

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 1 – Introduksjon til programmering

Analog introduksjon til programmering

Oppgave 1

De skal jobbe i par.

Begge bygger ein liten figur av dei utdelte legoklossane. Ta vare på figuren din heilt til slutten, men ikkje vis han til nokon andre.

Lag ei oppskrift slik at den andre du jobbar med klarer å bygge ein heilt lik figur. Du får ikkje forklare ekstra undervegs, alt må vere skrive ned.

Så byter de oppskrift og bygger den andre sin figur.

- Blei dei like?
- Kva måtte du endre i oppskrifta for at figurane skulle bli heilt like?
- Kunne du ha laga oppskrifta kortare eller enklare?



Oppgave 2

De skal jobbe i par.

Begge lagar ei oppskrift på korleis den andre må gå for å kome seg ut døra. Det er ikkje lov å forklare undervegs.

Så byter de oppskrift, og ein om gongen prøver ut den andre si oppskrift og ser kor du hamnar.

Deretter justerer de begge oppskriftene, men da kan de berre bruke kodeklossane på neste side.

Kor få klossar er mogleg å bruke for å få til oppgåva?



Feilsøking og forbedring

1. Skjedde det som skulle skje? / Fekk du til oppgåva?
2. Kva måtte du endre på i oppskrifta for å få til oppgåva?
3. Kunne du ha laga oppskrifta kortare eller enklare?

Kodeklossar – eit lite oversyn

Gå ett steg

Gå to steg

Gå tre steg

Desse klossane seier kor mange steg du skal gå bortover.

Sving 90° mot høgre

Sving 90° mot venstre

Snu 180°

Desse klossane seier kor mange grader du skal snu.

Stopp

Denne klossen seier at du skal stoppe opp og stå i ro.

Repeter 5 gonger

Repeter 10 gonger

Repeter 20 gonger

Dette er lykkjer. Dei seier kor mange gonger du skal repetere noko. Kor mange gonger du skal gå rundt gjennom lykkja.

Difor må du alltid kople han saman med ein annan kloss som seier akkurat kva du skal repetere i lykkja di.

Dersom du krasjar, så

Dersom du somnar, så

Dersom du snublar, så

Dette er dersom-klossar. Dei inneheld eit vilkår, altså DERSOM noko hender, SÅ skal du gjere noko.

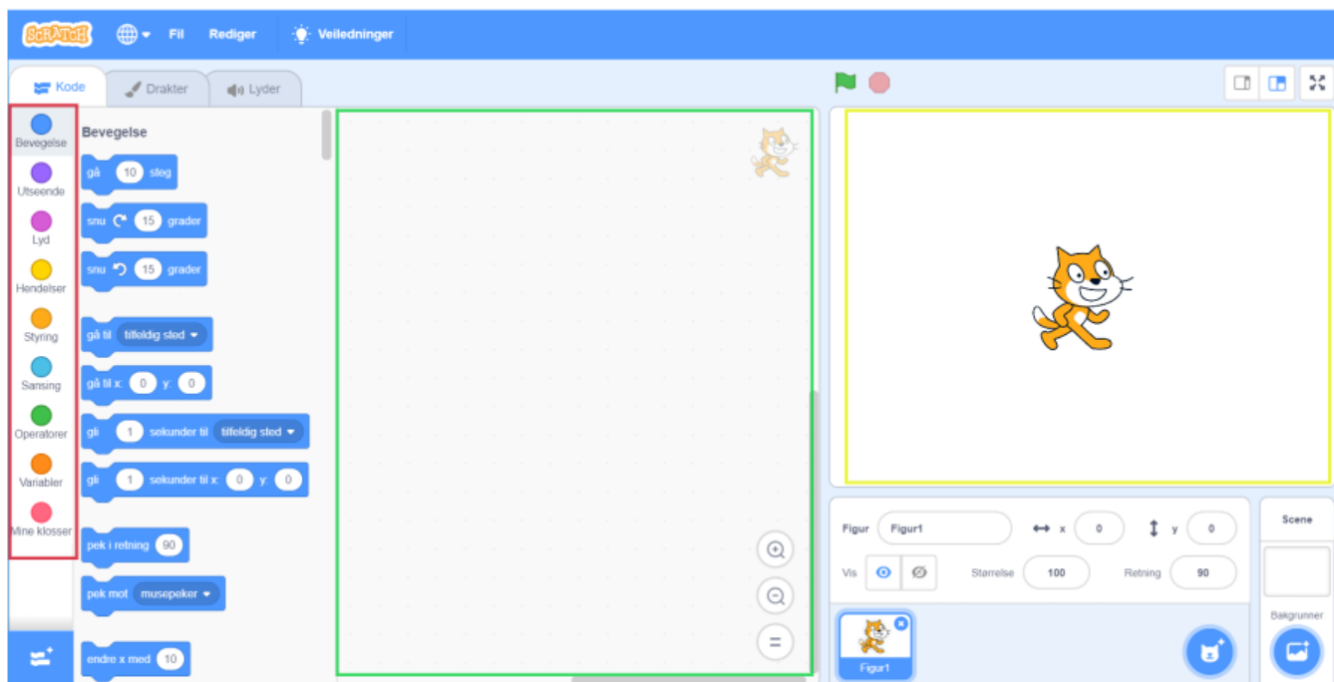
Difor må du alltid kople han saman med ein annan kloss som seier akkurat kva du skal gjere dersom du krasjar, somnar eller snublar.

Scratch

- ein introduksjon



Gå inn på nettadressa: <https://scratch.mit.edu>



I den raude ramma er det ulike kategoriar som gjer det enklare å finne dei klossane ein treng.

Rørsle: Her ligg klossane for kvar katten skal røre seg.

Styring: Her ligg klossane som seier kor mange gonger noko skal gjerast eller om det skal gjerast i det hele teke.

Operatorar: Her ligg ein del rekneoperasjonar, som addisjon, subtraksjon og samanlikningar.

I den grønne ramma er feltet der ein lagar programmet.

Du trekker klossen du vil nytte inn hit, da er han ein del av programmet ditt.

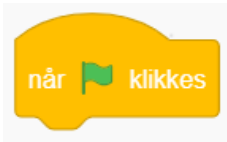
I den gule ramma ser du resultatet av programmet ditt.

Katten vil røre seg slik programmet gir han melding om.

Dette området fungerer som eit koordinatsystem, der katten startar i origo.

Intro til nokre klossar i Scratch

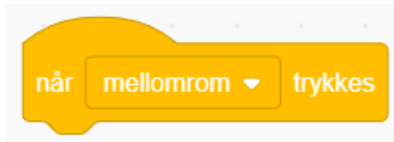
- Alle desse klossane finn du i kategorien som har lik farge som klossane



Når ein klikkar på det grønne flagget, skjer det som står i klossen kopla fast i denne.



Denne klossen kallast en viss-kloss. Og dersom det som står i det grå feltet er sant, vil det som står inni denne klossen hende.



Når ein klikkar på mellomrom-tasten, hender det som står i klossen som er festa i denne.



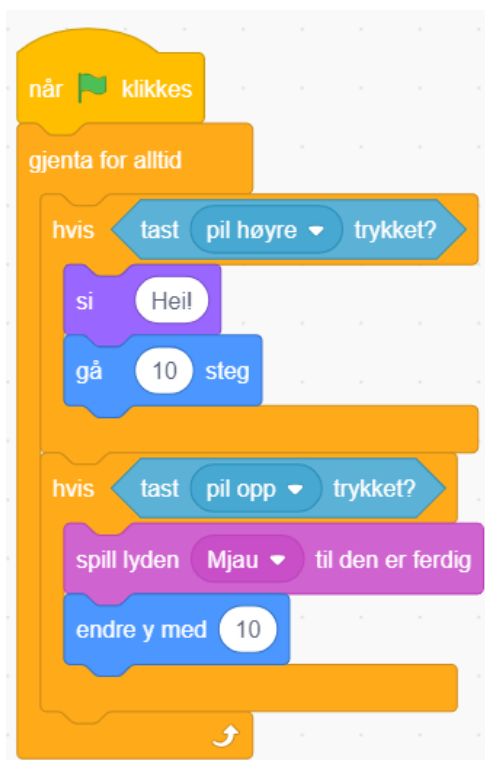
Døme på eit vilkår som kan settast inn i det grå feltet til dersom-klossen. Dersom vi set inn eit tal som er større enn 50, vil det som står inne i viss-klossen hende.



Denne klossen kallast ei lykkje. Det som står inne i denne klossen vil repeterast for alltid.



Denne klossen sier kva for nokre koordinatar katten skal gå til.



Oppgåve

1. Giss kva dette programmet gjer.
 - Fyll inn i det utdelte arket
2. Lag programmet i Scratch og sjå kva som hender.
3. Kva må du gjere for at katten skal kunne kome attende til origo?
 - HINT: Tenk på x- og y-koordinatar.

Ekstraoppgåve

1. Kan du få katten til å flytte seg i ei rektangelform utan å bruke nokon tastar?
2. Kva med ei trekantform?

Introduksjon til vilkår, viss-klossar og lykkjer i Scratch

Vilkår

Eit vilkår er ein påstand som kan vere sann eller usann. Vi samanliknar ofte to tal og ser kva for eit som er størst, eller om dei er like store.

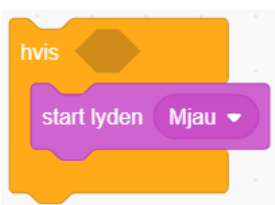
Døme på vilkår:

- Mamma er høgare enn pappa.
- Det er høgare temperatur om sommaren enn om vinteren.
- Læraren har lik skostorleik som eg har.

I Scratch er vilkåra grøne, sekskanta klossar.

Viss-blokker

- Kva: Ein viss-kloss inneheld eit vilkår. Når vilkåret er sant, vil det som står inni viss-klossen hende.
- Kvifor: Ein viss-kloss gjer at eit program kan gjere fleire ulike ting avhengig av om vilkåret er sant eller usant.



Diskuter

For kva for nokre av dei tre grøne vilkåra vil vi høyre mjauing med programmet til venstre?

Lykkjer

- Kva: Lykkjer gjer at ein del av koden i eit program blir teke opp att fleire gonger. Det finst fleire typer lykkjer. Nokre tek opp att det du skal gjere et visst tal gonger, medan andre lykkjer held fram for evig. Den tredje typen tek opp att fram til eit vilkår ikkje lenger er sant.



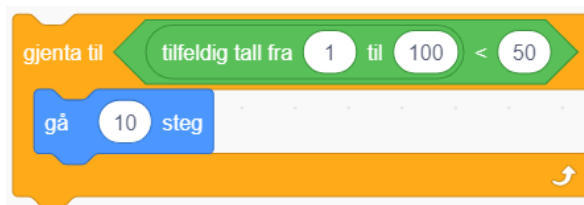
- Kvifor: Programmet blir lettare å få oversikt over, og det blir færre klossar enn om ein ikkje nyttar lykkjer.

Oppgåver

1. Lag to vilkår som er sanne og to som er usanne.
2. Er det alltid enkelt å vite om eit vilkår er sant eller usant?
3. Kva type vilkår kan ein datamaskin forstå?

Grubleoppgåve

Kva trur du hender i programmet under?
Vil det same hende kvar gong?



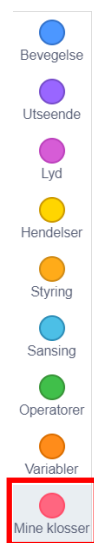
Introduksjon til funksjonar i programmering

- korleis lage egne klossar

Kva? Ein funksjon er ein kloss som du har laga sjølv. Altså eit slags eige program. Det som skil ein funksjon frå eit vanleg program, er at du først må definere ein funksjon. Det vil seie at du må lage han. Etterpå må du køyre funksjonen i eit program. Det som står inni ein funksjon, hender ikkje av seg sjølv før du nyttar funksjonen i eit program.

Kvifor? Programmet du lagar tek mindre plass, akkurat som når du bruker lykkjer. Funksjonen kan bli nytta fleire stader i programmet utan å måtte lage alle blokkene på ny.

Korleis?



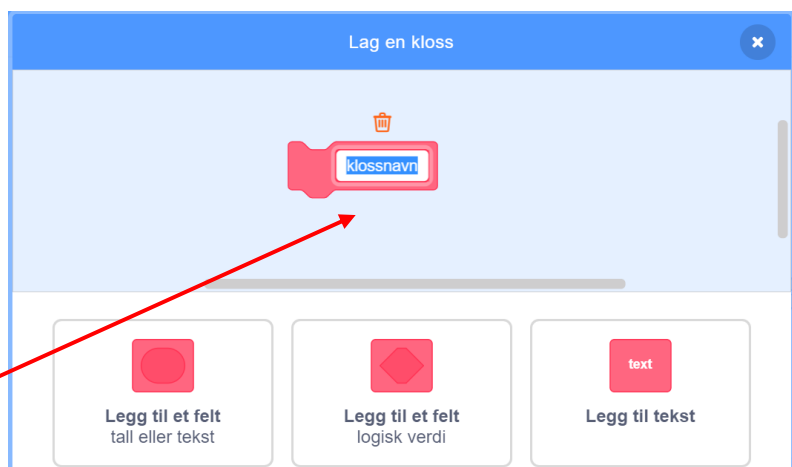
1. steg: Vel mine klosser i menyen på venstre sida.

2. steg: Trykk på lag en kloss-knappen som da dukker opp.

Mine klosser

Lag en kloss

3. steg: Nå må du skrive inn kva funksjonen din skal heite der det står klossnavn.



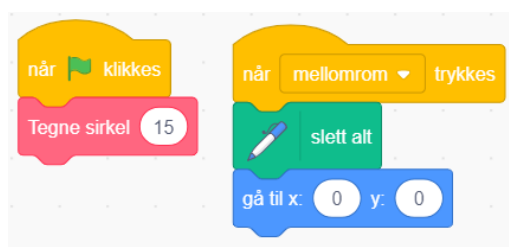
4. steg: Du kan velje å legge til om funksjonen din treng å få noko informasjon for å verke. Dette kallast ofte for inndata eller input. Det treng vi til dømes om funksjonen skal rekne ut ulike reknestykke.

Dersom du lagar ein funksjon som skal rekne om ein verdi frå meter til centimeter, må funksjonen din få vite kor mange meter du vil gjere om til centimeter. Da må du legge til eit tal når du opprettar klossen til funksjonen din.



Diskuter

1. Kva gjer funksjonen til venstre?
2. Kvifor er det ein funksjon, og ikkje eit eige program?
3. Kva gjer programmet under?
4. Kvifor er dette eit program?



Oppgave

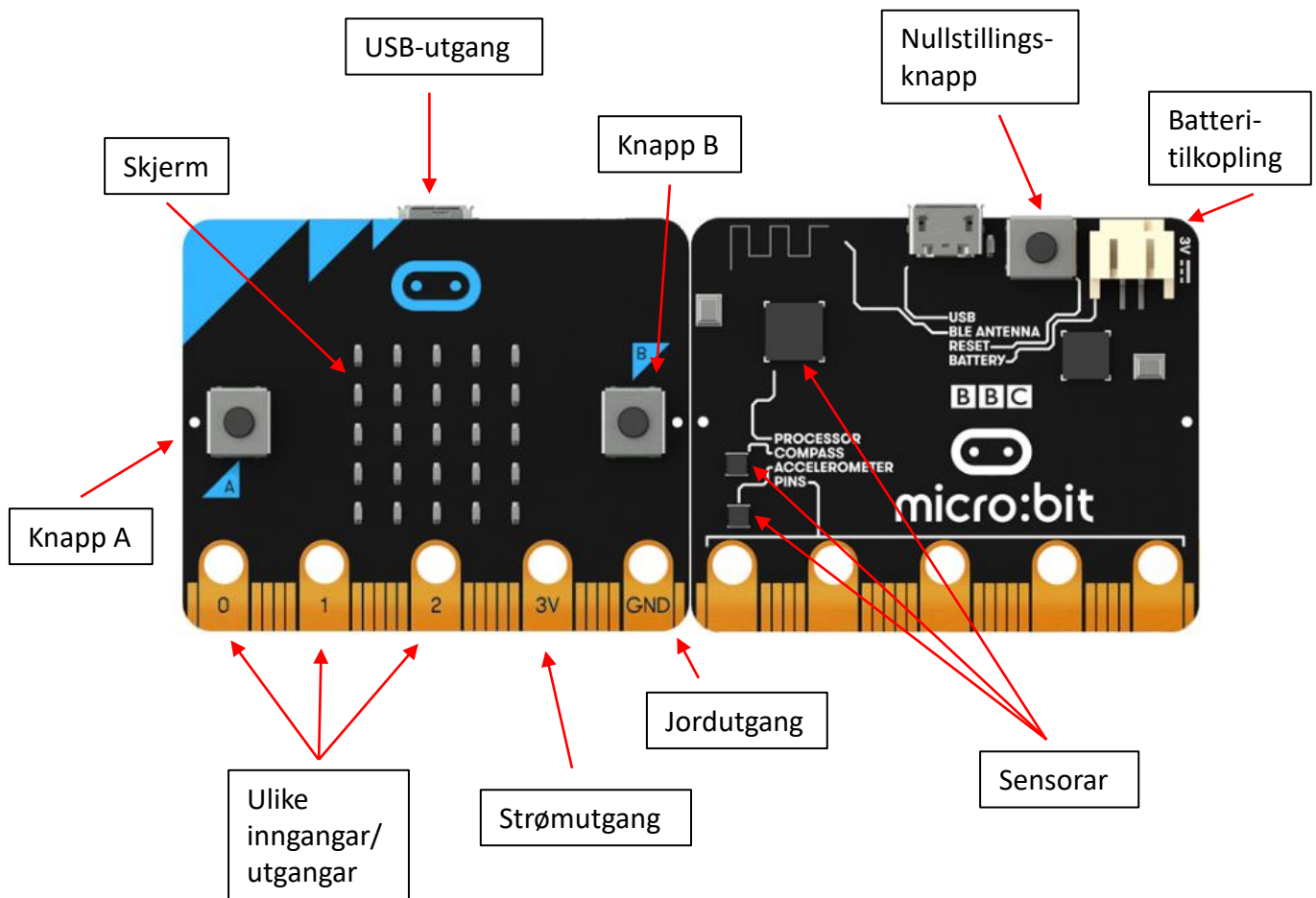
Lag ein funksjon for at katten flyttar seg i desse formene:

1. Kvadrat
2. Rektangel
3. Trekant

Kan du bruke funksjonane dine til å teikne ein samansett figur?

Introduksjon til micro:bit

Micro:bit er ein mikrokontrollar, det vil seie ein enkel datamaskin som vi kan programmere. Vi kan programmere med hjelp av ein datamaskin, eit nettbrett eller ein mobiltelefon.



Skjerm: Micro:biten har ein «skjerm» som er samansett av 25 LED-lamper som vi kan programmere til å lyse, blenke og vise ulike bilete, tal eller bokstavar.

Knappar: Han har to knappar på framsida, og ein nullstillingsknapp på baksida. Nullstillingsknappen gjer at programmet på micro:biten startar på ny.

Mini-usb: Her kan du kople micro:bit til ein USB-kabel dersom du bruker ein datamaskin med USB-utgang.

Batteritilkopling: Her kan du kople til batteri for at micro:biten skal få straum.

Sensorar: Micro:biten har nokre innebygde sensorar som kan måle akselerasjon, temperatur og kompassretningar.

Tilkoplingar: Det går an å kople på andre sensorar, motorar, LED-pærer eller høgtalarar på micro:biten med hjelp av alligatorleidningar.

Bluetooth (blåtann): Micro:biten har bluetooth-tilkopling (blåtann-tilkopling) slik at han kan få overført program frå til dømes iPad som ikkje har USB-utgang.

Introduksjon til bruk av micro:bit

- for datamaskinar

Gå inn på nettadressa: <https://makecode.microbit.org/#editor>

Da kjem du inn på denne nettsida, der du kan byrje å programmere micro:biten.

Simuleringsområde

I den gule ramma ser du resultatet av programmet ditt.

Klosseliste

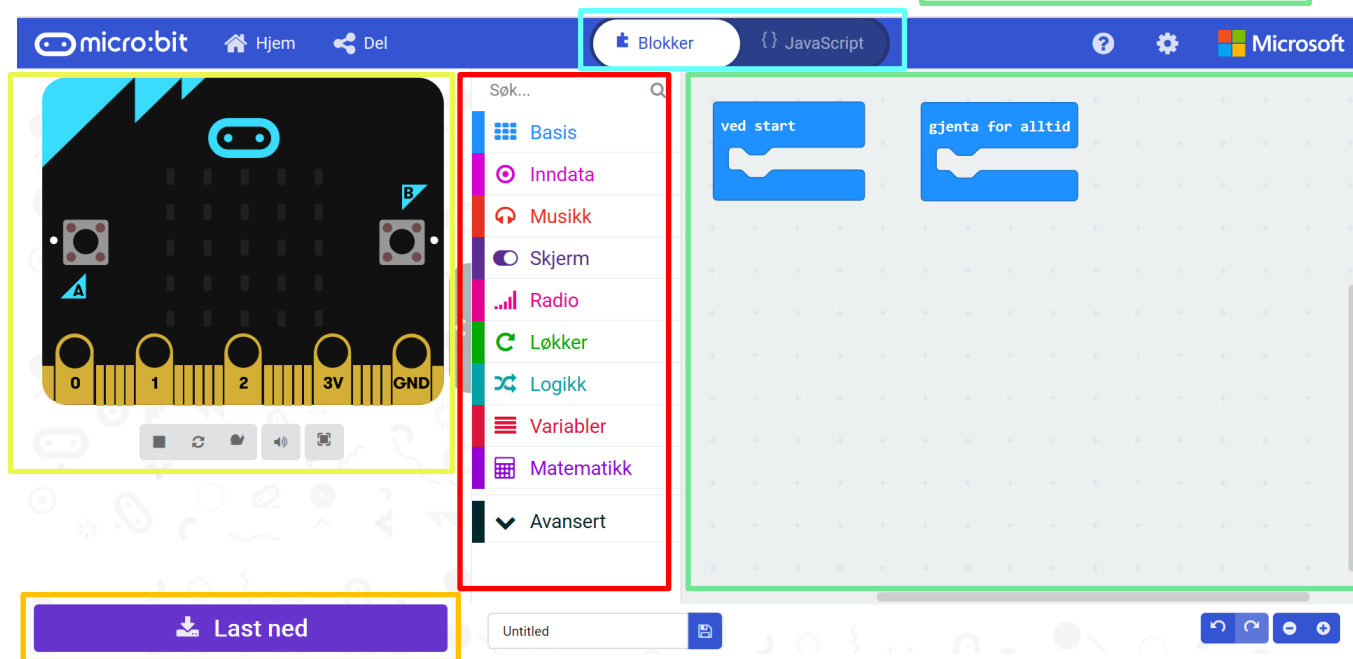
I den raude ramma er det ulike kategoriar som gjer det enklare å finne dei klossane ein treng.

Når du trykker på ein av kategoriane, vil klossane som høyrer til komme fram rett til høgre.

Programmeringsområde

I den grønne ramma er feltet der ein lagar programmet.

Du trekk klossen du vil bruke inn hit, da er han ein del av programmet ditt.



Nedlastingsområde

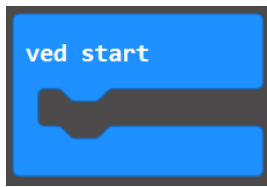
I den oransje ramma kan vi laste programmet over til micro:biten.

Programmeringsspråk

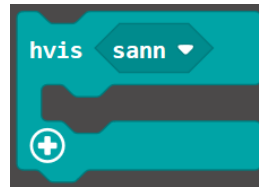
I den turkise ramma er feltet der du kan velje om du skal bruke klossar å programmere med, eller eit tekstbasert programmeringsspråk kalla Python eller JavaScript.

Intro til MakeCode for micro:bit

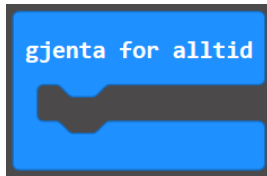
- Alle disse klossane finn du i kategorien som har lik farge som klossane



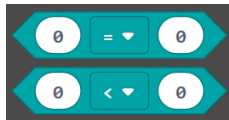
Straks programmet er overført til micro:biten hender det som står inni denne klossen.



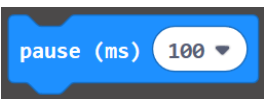
Denne klossen kallest ein viss-kloss. Og dersom det som står i det mørkegrøne feltet er sant, vil det som står inni denne klossen hende.



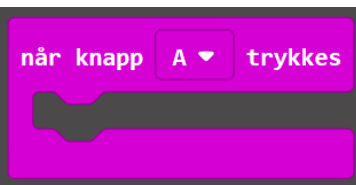
Denne klossen kallast ei lykkje. Det som står inne i denne klossen vil bli teke opp att for alltid.



Døme på vilkår som kan bli sett inn i det mørkegrøne feltet i viss-klossen. I det øvste vilkåret kan vi sette inn to tal som han sjekker om er like. I den nedste kan vi sette inn to tal der vilkåret sjekker om det første er mindre enn det andre.



Denne klossen gjer at programmet tek ein pause før det går vidare til neste kloss.



Denne klossen gjer det som står inni han når ein trykker på knapp A på micro:biten.

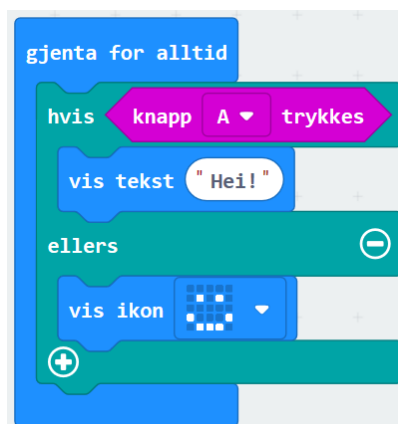
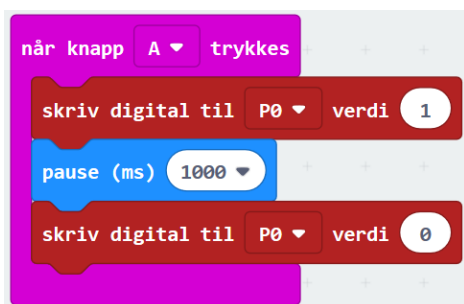


Når verdien er 1, vil denne klossen gjere at det vert sendt straum ut gjennom P0-utgangen. Når verdien er 0, blir det ikkje sendt straum ut gjennom utgangen.

Output til skjerm

Eit viktig poeng med å bruke micro:bit er at vi kan vise informasjon på han. Det blir gjerne kalla utdata eller output, og kan visast på skjermen til micro:biten. Skjermen er ikkje ein vanleg skjerm, men er samansett av 25 raude lysdiodar som kan vere av- eller påskrudde.

I micro:biten har vi ulike klossar som kan vise ting på skjermen. Vi kan vise tekstar, bilete eller tal. Tekstane kan vere litt vanskelege å lese, for dei ruller over skjermen og kjem som ei blanding av store og små bokstavar. Vi kjem til å vise flest tal i denne boka.



Oppgave

Giss kva dei to programma gjer.

Lag programma i makecode og før over til micro:biten – eitt om gongen.

Skjedde det du trudde?

Kan du gjere nokre morosame endringar?

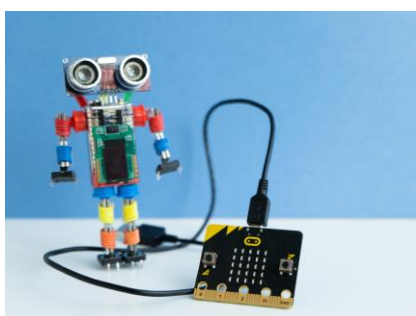
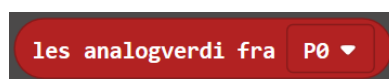
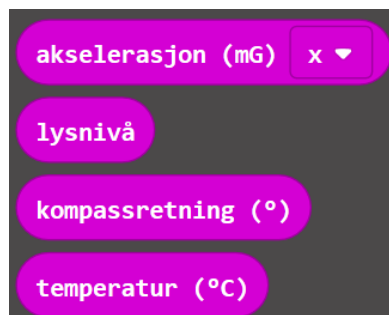
Inndata/input for micro:bit

Input frå sensorar

Mesteparten av poenget med å bruke micro:bit er at vi kan samle inn informasjon med han. Det kallast gjerne inndata eller input, og kjem frå ulike sensorar. Dei kan være innebygde i sjølve micro:biten, eller vi kan kople til andre sensorar.

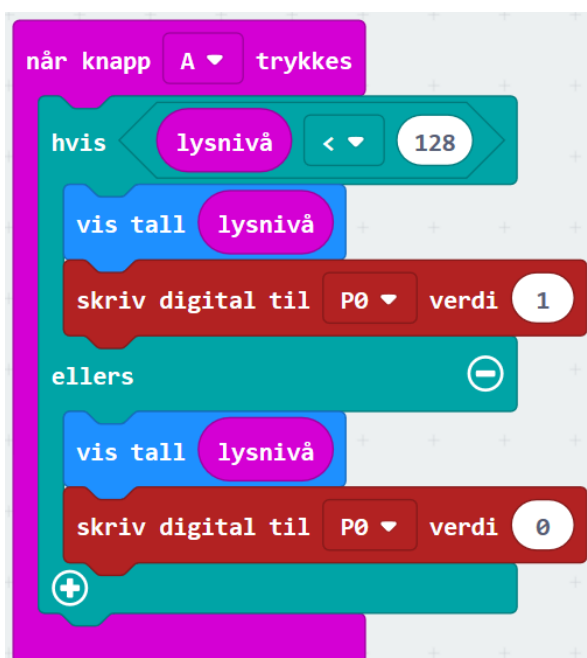
I micro:biten har vi innebygde sensorar som kan måle akselerasjonen, lysnivået, kompassretninga og temperaturen. Vi kjem oftast til å bruke temperaturen i denne boka.

Dersom vi ønsker å bruke ein sensor som ikkje er innebygd i micro:biten, må vi først kople han til ein inngang på micro:biten. I tillegg treng vi ein kloss som les av kva for ein verdi sensoren sender til micro:biten. Da nyttar vi denne klossen. Klossen les berre av kva verdien er, men gjer ingenting med han. Dei fleste sensorar vi kopler på, gir verdier mellom 0 og 1023, men det kan variere frå sensor til sensor. Det står meir om sensorar og korleis de verkar på neste side.



Snakk om

1. Kva for nokre sensorar veit du om?
2. Korleis kan ein micro:bit vise fram målingane frå sensorane?
3. Kva kan ein bruke ulike sensorar til?
4. Kan du tenke ut ei oppfinning som nyttar ein sensor for å løyse ei oppgåve.



Oppgåve

1. Giss kva dette programmet gjer.
2. Lag programmet i MakeCode, overfør det til micro:biten og sjå kva som hender.
3. Kva kan det tenkast skal vere kopla til micro:biten?
4. Kva ville du gjort med programmet for å forbetre det?

Sensorar

Ein sensor blir brukt til å måle ulike verdier, slik som temperatur, lysstyrke, fart, akselerasjon eller lydnivå. Sensoren vert kopla til ein micro:bit og ein kan måle kor stor spenning som krevst for å trykke elektrona gjennom han. Da er det slik at i ein temperatursensor vil spenninga variere i same takt som temperaturen, og vi kan bruke spenninga vi måler til å rekne ut nøyaktig temperatur.



Termistor
- måler temperatur



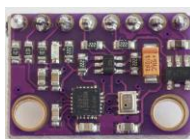
Lyssensor
- måler lysstyrke



Halleffekt-sensor
- måler magnetfelt



Ultralydsensor
- måler avstand



Akselerometer
- måler akselerasjon



Lydsensor
- Måler lydstyrke

Kva er motstand?

Når eit materiale har stor motstand, er det vanskeleg for elektrona å flytte seg gjennom materialet. Da treng ein høg spenning for å dytte elektrona gjennom. Motstand blir og kalla resistans, og eininga er ohm.

Elektriske leiingar har liten motstand slik at elektrona kan røre seg lett gjennom dei, medan isolatorar har veldig stor motstand slik at det er mest umogleg for elektron å flytte seg gjennom dei.

I sensorar varierer motstanden, og da vil og spenninga variere. Denne spenninga er enkel å måle med ein micro:bit. I ein lyssensor vil motstanden variere med lysstyrken, og vi får målt ulik spenning om det er mørkt eller lyst. For ein ultralydsensor vil spenninga variere ut frå kor stor avstand sensoren har til ein gjenstand, og tilsvarende for andre sensorar.

Feilsøking

- debugging



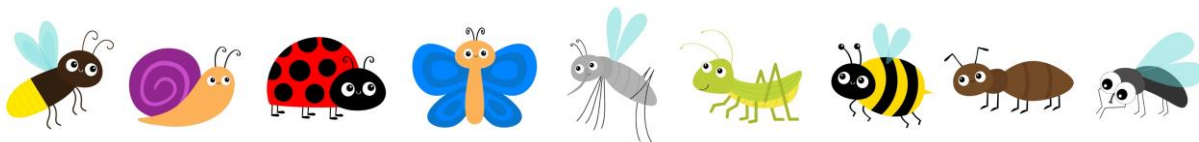
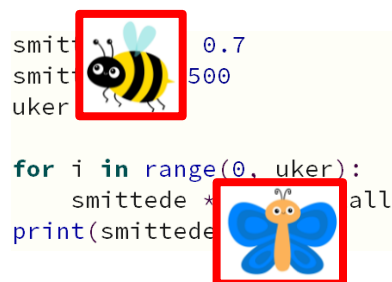
Finn meg!

Absolutt ingen som programmerar gjer alt rett på første forsøk. Ikkje ein gong dei aller beste programmerarane som har jobba med programmering i mange år! Når programmet ikkje oppfører seg som planlagt, treng vi å finne ut kva som er grunnen til det. Vi må finne feila i koden vår, og kalla bugs. Det kan vere lurt å lage seg ei eiga liste med vanlege feil for deg, og bruke denne lista for å finne bugs i nye program.

Døme på ei sjekkliste for feilsøking

- gjer gjerne det som er enklast først, ikkje slett heile programmet

1. Kjør programmet og sjå kva som hender
 - Kva er det som ikkje stemmer med det du ønsker at programmet skal gjere?
 - Skjer ting i den rekkefølga du hadde planlagt?
 - Får du nokon feilmeldingar (tekst-programmering)?
2. Gjer synleg det som skjer inni programmet og køyr det
 - Skriv variablane til skjerm
 - Legg inn pausar i koden for å rekke å sjå kva for nokre verdier ulike variablar har
3. Endre på programmet og køyr det
 - Har variablane rett namn med stor/liten bokstav alle stadene der du bruker same variabel?
 - Har lykkjene rett tal på repetisjonar?
 - Endre berre på ein ting om gongen, slik at det er enkelt å sjå kva ei endring fører til
 - Gir feilmeldingar nokon hint om i kva for nokre linjer feilen ligg (tekst-programmering)?
 - Gir feilmeldingar nokon hint om kva feilen er (tekst-programmering)?
 - Feil variabeltype
 - Glømt innrykk
 - Manglande kolon eller andre teikn
 - Søk på feilkodar på internett (tekst-programmering)



Søke etter svar

- Sjølv dei flinkaste programmerarane bruker mykje av tida på å søke opp feilkodar på internett. Det kan og vere lurt å søke på namnet til ulike funksjonar for å sjå kva for nokre inndata dei skal ha, og kva for nokre utdata dei gir.
- Står det noko i læreboka di? Les gjerne nærmare gjennom oppgåva.
- Spør nokon i klassen eller læraren.

Korleis verkar ein servo?

En servo er ein liten elektrisk motor som kan dreie fram og attende eller rundt og rundt. Han som berre kan dreie fram og attende blir kalla ein 180°-servo. Han som kan dreie rundt og rundt blir kalla ein 360°-servo (eller kontinuerlig servo).

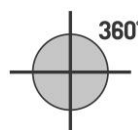
Vinklar



Servoane får namn frå vinklar. Ein vinkel kan vere mellom 0° og 360°. Ein vinkel på 90°, kallast ein rett vinkel og ser slik ut.



Ein vinkel på 180° er det doble av ein rett vinkel, og da får vi faktisk ei rett linje!



Dersom du tek fire rette vinklar, endar du opp med ein vinkel på 360°. Da er du komen heilt rundt! Dei servoane som kallast 360°-servoar går jo rundt og rundt.



Kva tyder kontinuerlig?

Noko som held fram utan opphald.

Ein 180°-servo kan berre dreie 180°, og ikkje rundt og rundt.

Det går typisk ikkje an å styre farten til denne typen servoar, berre vinkelen dei dreier til.

Programmering av servoar

Da kjem du til å trenge nokre av desse klossane.

180°-graders servo:

Denne set kva for ein vinkel servoen skal dreie til.

360°-graders servo:

Denne set farten til servoen.

Denne stoppar servoen.

Sett vinkel på servo P0 til 90°

servo P0 kjører kontinuerlig på 50 %

stans servo P0

Diskuter

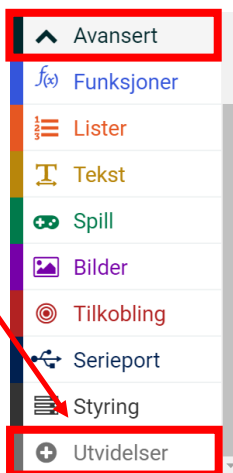
Kva må du gjere for å få 360°-servoen til å dreie i motsett retning?

Desse klossane finn du slik

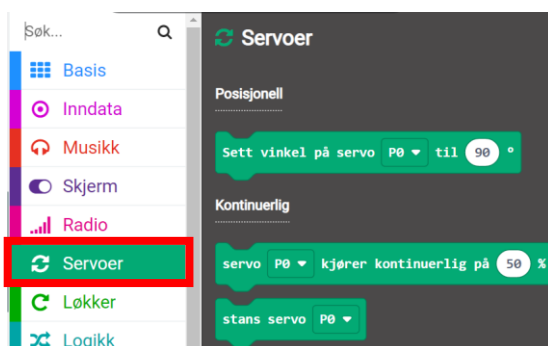
1. Vel avansert i menyen, han er svart.

2. Trykk der det står utvidelser, nedst.

3. Så vel du servo-utvidelsen.



4. Vel servo i menyen, så finn du klossane du treng.



Servoen treng litt meir straum enn det som kjem frå micro:biten. For å vere sikker på at han får nok straum til å virke slik han skal, må ein bruke eit kretskort som gjer at vi kan kople til batteri i tillegg til micro:biten.

De kan la vere å legge til servo-tillegget, da må de bruke ein annan kloss enn vist på førre side. Det er berre ein kloss uavhengig av kva type servo ein nyttar, men verdiane ein skriv inn, får ulikt resultat ut frå servotypen.

180°-servo: Verdien angir vinkelen servoen skal dreie til.

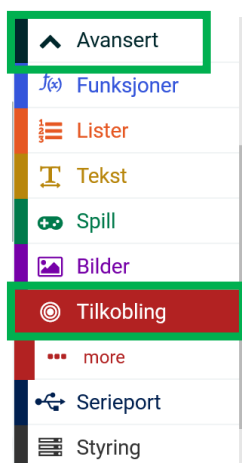
360°-servo: Verdien angir dreiefart som servoen dreier rundt og rundt med.

servo skriv til P0 ▼ verdi 180

Denne klossen finn du slik

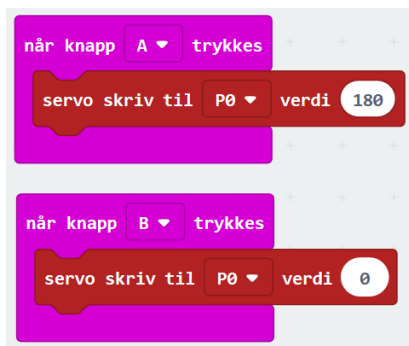
1. Vel avansert i menyen, han er svart.

2. Gå nedover i menyen til du finn tilkopling. Han er mørkeraud.



3. Så finn du klossen du treng nest nedst (og nedst).

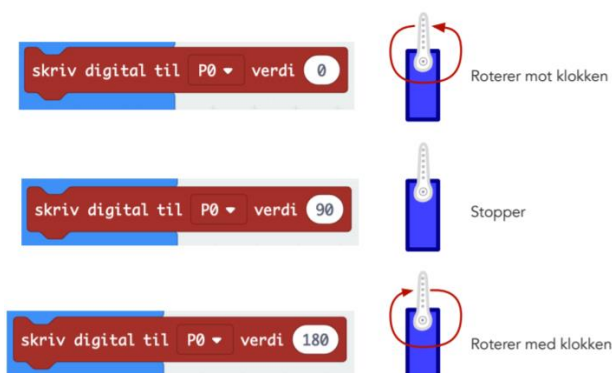
Programmering av 180°-servoar



Ein 180°-servo kan dreie 180° kvar veg, men ikkje dreie heilt rundt. Ein kan typisk ikkje styre farten til denne typen servoar, berre kva for ein vinkel dei dreier til.

Det øvste programmet gjer at servoen dreier 180° i den eine retninga. Det nedste programmet gjer at servoen dreier attende til utgangspunktet.

Programmering av 360°-servoar (kontinuerlege servoar)



Om vi skriv verdien 0 til servoen, vil han rotere mot klokka så fort han kan.

Når vi aukar verdien, vil han rotere saktare heilt til vi kjem til 90, der motoren står stille.

Når vi aukar verdien over 90 opp mot 180 grader, roterer han raskare og raskare med klokka.

Her er verdien altså ikkje vinkelen, men dreiefart.

Miniutfordringar

Det skeive tårnet i Pisa



Materiale: Papplate (30 cm x 30 cm), 18 papirark, tape og saks.

Bruk av tid: Planlegging 2 min, gjennomføring 10 min.

Krav:

Bruk dei utdelte materiala.

Bygg tårn med gjennomsnittshøgde på minst 160 cm på papplata.

Klossane må vere i kontakt med plata, og plata skal vere intakt.

Utskytingsmekanisme



Materiale: Alt som ligg klart.

Bruk av tid: Planlegging 2 min, gjennomføring 10 min.

Krav:

Lag ein utskytningsmekanisme som får svampen til å flyge minst 5 meter.

Utskytingsmekanismen må vere samansett av 5 ulike materialar.

Uidentifisert flygande objekt (UFO)



Materiale: Alt som ligg klart.

Tidsbruk: Planlegging 2 min,
gjennomføring 10 min.

Krav:

Lag eit flygande objekt som skal vere i lufta lengst mogleg.

Bruk dei utdelte materiala.

Det flygande objektet skal dra ein blyant.

Bordtennis-basketball



Materiale: Alt som ligg klart.

Tidsbruk: Planlegging 2 min,
gjennomføring 10 min.

Krav:

Ballen må følge ein bane og gå gjennom ei korg.

Ballen kan ikkje rørast etter at han er sett i gang.

Korga som ballen skal gå gjennom må lagast.

Banen må vere minst 2 meter og kan ikkje vere ei rett linje.

Bakkeløp



Materiale: Alt som ligg klart.

Tidsbruk: Planlegging 2 min,
gjennomføring 10 min.

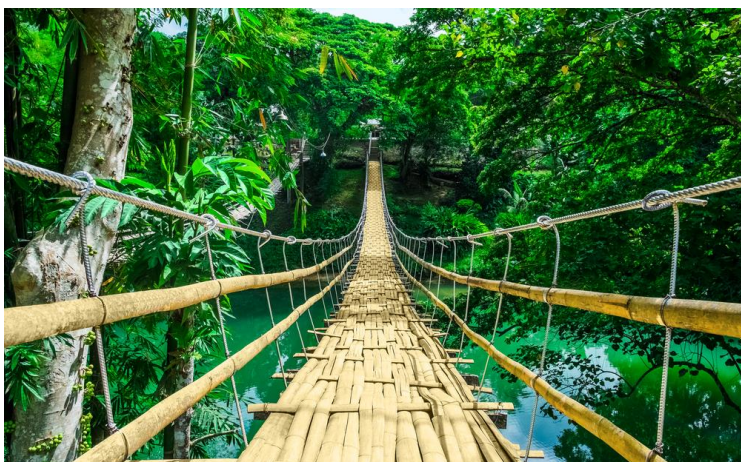
Krav:

Lage ein gjenstand som bruker nøyaktig 5 sekund på å trille ned papplatebakken.

Bruk dei utdelte materiala.

Det er ikkje lov å røre gjenstanden etter at han er sloppen.

Brubygging



Materiale: Alt som ligg klart.

Tidsbruk: Planlegging 2 min,
gjennomføring 10 min.

Krav:

Bruk dei utdelte materiala.

Bygg ei bru med eit spenn på minst 50 cm.

Brua skal kunne bere ein full melkekartong midt på bruspennet (1 kg).

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 2

Opplegg 1 - Innovasjonsmetoden

- Korleis løyse ei oppgåve?

1. Undersøke og finne informasjon

- Leit etter informasjon om temaet.
- Bruk ulike kjelder.
- Finn ut kva vanskelege omgrep tyder.



2. Planlegge og idémyldre

- Bruk den informasjonen du fann for å prøve å finne løysingar.
- Finn flest mogleg løysingar og vel den løysinga du trur er best.
- Planlegg gjennomføringa.



3. Utføre

- Teikne, bygg eller lag modellen.
- Lag første versjon av dataprogrammet (dersom det er ein del av oppgåva).



4. Teste

- Prøv ut det du laga i punkt 3.
- Gjer eventuelle målingar.



5. Evaluere

- Verkar det du laga slik som du hadde tenkt?
- Kva kan du gjere for at det skal verke betre?



6. Forbetre

- Gjer det du fann ut under punkt 6 som vil få det du laga til å verke betre.
- Gå gjerne attende til punkt 1 – 6 flere gonger.



7. Dokumentere

- Gjer ein ny test av det du har laga, verkar det nå?
- Ta eit bilete eller skriv ein liten tekst om det du har laga og korleis det verkar.

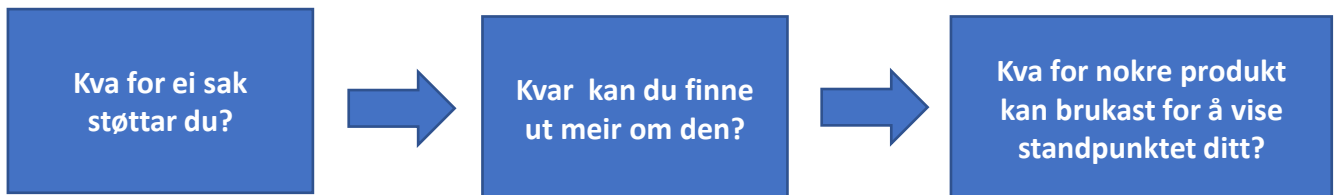


Demonstrer for standpunktet ditt



Bruk innovasjonsmetoden for å lage eit produkt som kan nyttast for å fremje synet ditt på ei viktig sak – det kan vere ei teikning, ein plakat eller noko heilt anna.

Fase 1: Undersøke og finne informasjon



Fase 2: Idémyldre og planlegge

Kva er det som gjer at du støttar saka du valde?

Korleis kan du lage eit produkt som overtyder andre?

Kva for nokre verkemiddel kan du bruke for å få fram standpunktet ditt?



Fase 3: Lag ei skisse av produktet.

Fase 4: Spør andre om dei ser kva for ei sak du demonstrerer for.

Fase 5: Kan du forbetre produktet på nokon måte?

Fase 6: Lag produktet.

Fase 7: Skriv ei forklaring på kvifor du valde å demonstrere for akkurat den saka med hjelp av det produktet du gjorde.

Opplegg 2 - Celler

Alle levande organismer er bygd opp av ei eller flere celler. Dei fleste cellene er så små at ein treng mikroskop for å sjå dei, men mennesket sine eggceller er store nok til å sjå utan hjelpemiddel. Det finst mange forskjellige typar celler, og i menneskekroppen er det om lag 220 ulike typar. Det kan vere alt frå muskelceller til nerveceller og kjønnsceller. I menneskekroppen har vi om lag 37 billionar celler (37 000 000 000 000).

Organeller (cella sine organ)

Cellemembran – er ei hinne rundt cella. Denne slepp inn næringsstoff, og slepp ut avfallsstoff og andre produkt.

Cytosol – ei tjuktflytande væske som fyller innsida av cella. Ho er i hovudsak laga av vatn, mineral og næringsstoff.

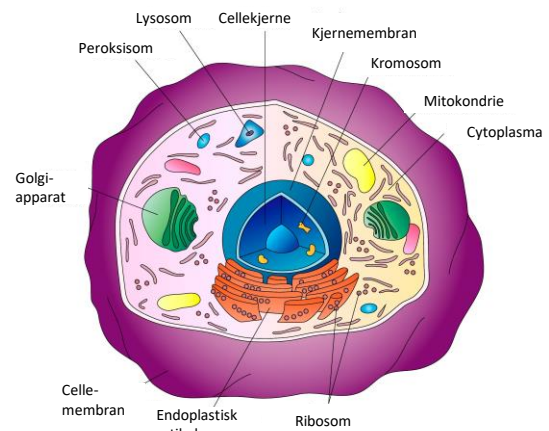
Cellekjerne – cella sin kommandosentral, som avgjer det som skjer i cella.

DNA – er arvestoffet vårt, og det ligg inni cellekjernene. Det fastset kva for nokre protein kroppen vår skal produsere.

Ribosom – er viktig i prosessen der cellene lagar protein.

Mitokondriar – cellene sine kraftverk, dei omformar den kjemiske energien frå maten vi et til energi kroppen kan bruke (celleanding).

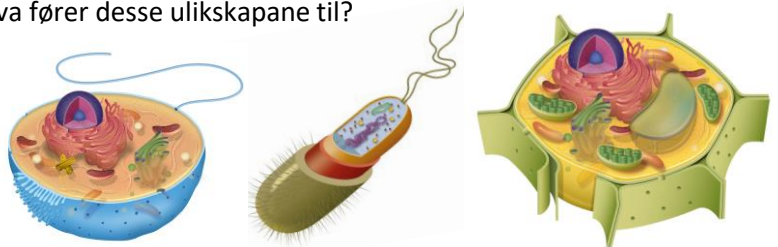
Lysosomer – cellene sin avfallsstasjon, dei bryt ned avfallsstoffa i cellene.



Snakk om

Kan de finne ut nokon skilnader på dyreceller, planteceller og eincella organismar? HINT: Cellevegg, kloroplaster, lysosomer og cellekjerne.

Kva fører desse ulikskapane til?



Du kjem til å trenge denne klossen. Når verdien er 1, vil denne klossen gjere at det blir sendt strøm gjennom LED-pæra slik at ho lyser.

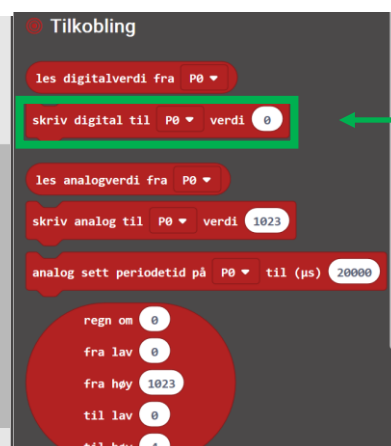
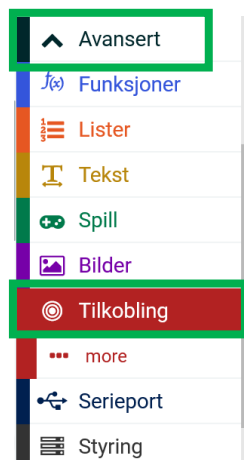
Verdi 1: Strøm på

Verdi 0: Strøm av

skriv digital til P0 ▼ verdi 0

Denne klossen finn du slik

- Vel avansert i menyen, han er svart.
- Gå nedover i menyen til du finn tilkobling. Han er mørkeraud.



- Så finn du klossen du treng nest øverst.

Lag ein modell av ei celle

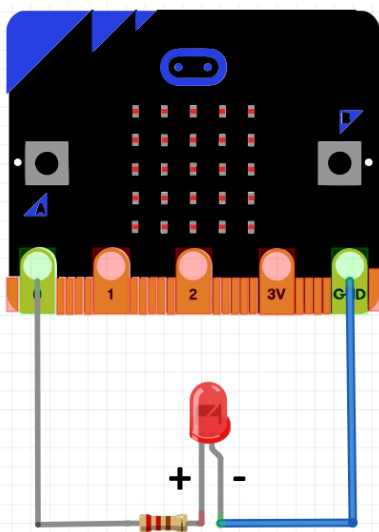
Oppgåve

Lag ein modell av ei celle som har minst éi lysande eller blenkande LED-pære.

Fase 1: Finn informasjon og inspirasjon til ein cellemodell. Kva for nokre delar er han samansett av? Korleis ser desse delane ut?

Fase 2: Ha ei idémyldring for deg sjølv. Korleis vil du at cellemodellen din skal sjå ut, og kva utstyr skal du bruke? Teikne gjerne ei skisse før du tek til med å lage modellen. Han skal og ha minst eitt lys, kva for nokre fargar skal desse ha, og kvar passar dei best?

Fase 3: Tid for å lage cellemodellen og programmere micro:biten. Sjå under for oppkopling og programmering.



Kople opp micro:bit som vist på figuren.

Det lange beinet til LED-pæra skal koplast til ein mostand og så til P0-utgangen. Dette blir den positive polen, og vi kan lage eit program som sender strøm gjennom pæra.

Det korte beinet til pæra skal koplast til GND-utgangen. Det står for ground som tyder jord, og det blir den negative polen.

Kva for nokre endringar må vi gjere for å få fleire LED-pærer til å lyse?

gjenta for alltid

skriv digital til P0 ▼ verdi 1

Døme på program

Dette programmet får LED-pæra til å lyse konstant. Kva må du gjere for å få henne til å blenke?

Fase 4: Test programmet ditt.

Fase 5: Verkar det slik det skal?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få ein best mogleg cellemodell. Gjer gjerne endringar i micro:bit-programmet ditt.

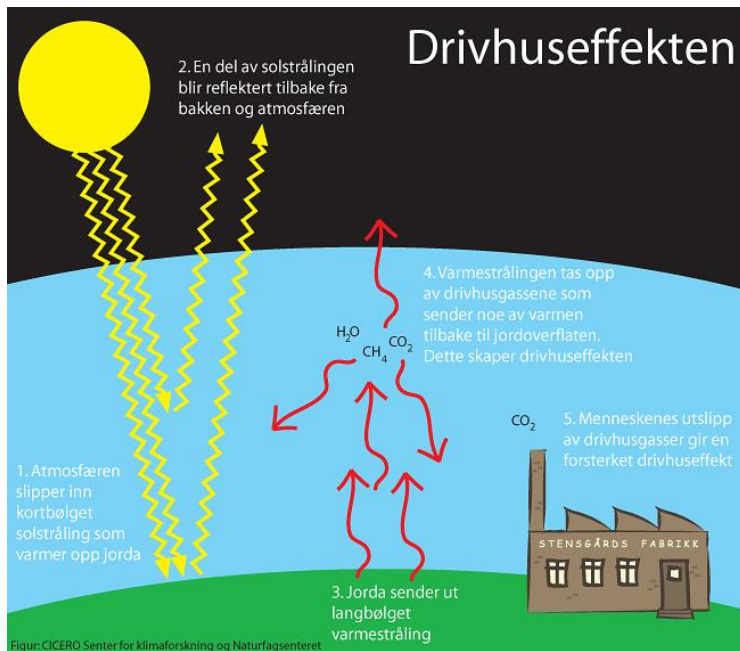
Fase 7: Ta gjerne en filmsnutt av den ferdige cellemodellen for å vise korleis han verkar.

Oppgåve

Berre planteceller inneheld kloroplaster som gjer at dei kan omforme energi frå sollyset til kjemisk energi i karbohydrat. Men alle celler kan omforme energien frå næringsstoff til den energien dei treng for å leve. Denne prosessen blir kalla celleanding.

1. Finn ut kva for nokre stoff som er med i reaksjonane for fotosyntese og celleanding.
2. Kva har desse stoffa med karbonkrinsløpet å gjere?
3. Kan dette påvirke drivhuseffekten på nokon måte?

Opplegg 3 - Drivhuseffekten



Drivhuseffekten gjer at gjennomsnittstemperaturen på jordkloden er omtrent 15°C i staden for -18°C , som han ville ha vore utan nokon drivhuseffekt.

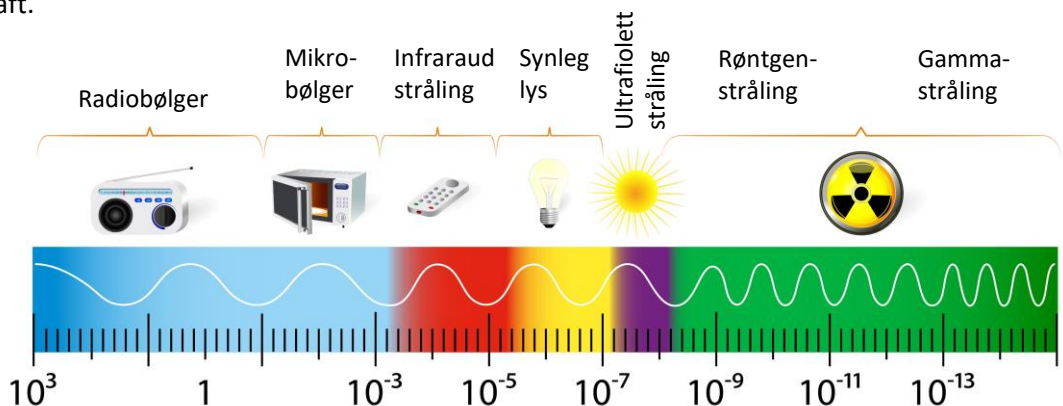
Det blir sendt ut stråling med mykje energi frå sola, og denne treff jorda sin atmosfære. Ein mindre del av denne strålinga forsvinn berre ut i verdsrommet, for atmosfæren og skyene verkar som ein spegel og reflekterer ein del av strålinga bort. Men mesteparten av denne strålinga blir absorbert (teken opp) av atmosfæren og bakken. Da har jordkloden motteke energi.

Jordkloden sender og ut si eiga stråling, varmestråling. Det er fordi

jordkloden har ein høgare temperatur enn verdsrommet. Verdsrommet har ein temperatur på om lag -270°C .

Varmestrålinga som jorda sender ut att, vert i stor grad teken opp av drivhusgassane i atmosfæren. Mengda av varmeenergi som blir teken opp i atmosfæren, blir større når det blir meir av drivhusgassane (klimagassane). Da vil gjennomsnittstemperaturen på jordkloden auke, slik at han blir høgare enn 15°C . Dersom gjennomsnittstemperaturen stig med over 2°C , vil det få store konsekvensar for livet på jorda. Store landområde vil bli liggande under vatn, det vil bli meir ekstremver; som kraftige stormar og uver, og langvarig tørke som kan føre til stor fare for skogbrannar. Dette vil verke på både menneske og dyre- og plantelivet på jordkloden.

Dei vanlegaste drivhusgassane er vassdamp (H_2O), karbondioksid (CO_2), metan (CH_4), lystgass (N_2O) og ozon (O_3). Av desse gassane er det i hovudsak CO_2 menneska slepp ut, og som dermed auker drivhuseffekten frå kor kraftig han var før den industrielle revolusjonen, da vi tok til med å bruke kol til dampkraft.



Meir om stråling og bølgelengd

Strålinga frå sola er kortbølga, det vil seie at bølgelengda er liten, og fargen vi kan sjå er gul/kvit. Strålinga med lita bølgelengd er den som inneheld mest energi. Denne strålinga blir absorbert av jorda, og sendt ut att som meir langbølga varmestråling. Varmestråling blir og kalt infraraud stråling og er usynlig for menneske, sidan ho har større bølgelengd enn 800 nm. Menneske kan berre sjå lys med bølgelengd mellom 400 nm og 800 nm, der lys med bølgelengd på 400 nm svarer til ein lillafarge, og lys med bølgelengd 800 nm svarer til raud farge.

Lag ein modell av drivhuseffekten

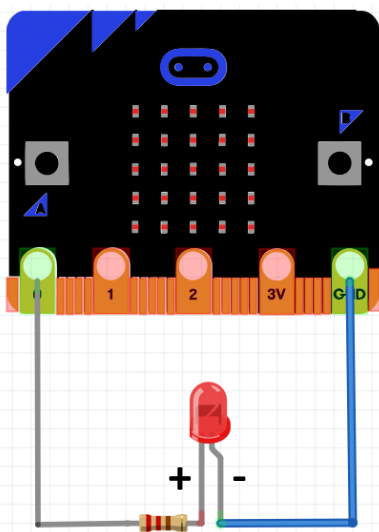
Oppgåve

Lag ein modell av drivhuseffekten som har minst éi lysande eller blenkande LED-pære.

Fase 1: Finn informasjon og inspirasjon til ein modell av drivhuseffekten. Kva for nokre delar er han samansett av? Korleis ser desse delane ut?

Fase 2: Ha ei idémyldring for deg sjølv. Korleis vil du at din modell skal sjå ut? Teikne gjerne ei skisse før du tek til med å lage modellen. Han skal og ha lys, Kva for nokre fargar skal desse ha, og kvar passar dei ulike fargane best?

Fase 3: Tid for å lage modellen, og programmere micro:biten. Sjå under for oppkopling og programmering.



Kople opp micro:bit som vist på figuren.

Det lange beinet til LED-pæra skal koplast til ein mostand og så til P0-utgangen. Dette blir den positive polen, og vi kan lage eit program som sender strøm gjennom pæra.

Det korte beinet til pæra skal koplast til GND-utgangen. Det står for ground som betyr jord, og det blir den negative polen.

Kva for nokre endringar må vi gjere for å få fleire LED-pærer til å lyse?

gjenta for alltid

skriv digital til P0 ▼ verdi 1

Døme på program

Dette programmet får LED-pæra til å lyse konstant. Kva må du gjere for å få henne til å blenke?

Fase 4: Test programmet ditt.

Fase 5: Verkar det slik det skal?

Fase 6: Hopp gjerne tilbake til tidlegare punkt og gjer endringar for å få ein best mogleg modell. Gjer gjerne endringar i micro:bit-programmet ditt.

Fase 7: Lag ein tilhøyrande tekst eller ei lydfil som forklarar korleis drivhuseffekten verkar og kva for nokre konsekvensar auka drivhuseffekt kan føre til.

Oppgåve

Dei globale klimaendringane skuldest i hovudsak menneska sine utslepp av CO₂.

1. Kor mykje har mengda CO₂ auka sidan den industrielle revolusjonen?
1. CO₂ er den den minst effektive klimagassen, men likevel han det blir fokusert på. Kvifor er det slik?
2. Konsekvensar av auka drivhuseffekt, er auka gjennomsnittstemperatur, som mellom anna fører til meir issmelting og meir ekstremvær. Kva for nokre konsekvensar kan dette igjen ha for menneske, dyr og planter?

Opplegg 4 - Energiformer og -overgangar

Kva er energi?

Energi er det som får ting til å skje. Utan energi kunne ikkje noko røre på seg. Energi kan aldri oppstå eller forsvinne, han kan berre gå over til andre energiformer. Vi seier at energien er teken vare på.

Stillingsenergi (potensiell energi)

Når vi løftar ein gjenstand høgare opp frå bakken, tilfører vi han energi. Jo høgare opp gjenstanden er, dess høgare energi har han. Denne energien kan enkelt bli omforma til andre typar energi med ein gong du slepp gjenstanden og han tek til å falle.

Rørsleenergi (kinetisk energi)

Ein gjenstand som rører på seg har meir energi jo større fart han har. Det er difor det blir så mykje større skade om en bil kolliderer når han har stor fart, sidan energien er mykje større enn om bilen hadde kjørt saktare.

Lyd

Lyden energi finnest i lydbølgene som rører seg til øyret ditt. Lyd er eigentleg trykkbølger i luft (eller andre materiale) som får luftmolekylane til å svinge. Sidan luftmolekylane rører på seg, får dei ein type rørsleenergi som vi kallar lyden energi.

Varme

Varme er ein form for energi, men en gjenstand kan ikkje innehalde varme, han kan berre overføre varme. Varme er energi som går frå ein gjenstand med høg temperatur til ein gjenstand med lågare temperatur.

Elektrisk energi

Elektrisitet er ladningar som rører på seg, som regel elektron. Da har desse ein type rørsleenergi som vi kallar elektrisk energi.

Strålingsenergi (for elektromagnetisk stråling)

Stråling er samansett av bølger av elektrisitet og av magnetisme, ein kan seie at han rører på seg. Da har dei en type rørsleenergi. Han kan forplante seg i lufttomt rom, og all energi vi får frå sola er av denne typen.

Kjemisk energi

Alle kjemiske sambindingar har lagra energi i bindingane sine. Denne energien kan bli frigjeven ved at vi bryt desse bindingane. For nokon stoff er det vanskeleg å bryte bindingene for å frigi denne energien, men for andre stoff er

det veldig enkelt. I fossile brensel er dette enkelt med å varme stoffet opp slik at vi får ein forbrenningsreaksjon (se opplegg 31). Det er faktisk forbrenningsreaksjonar som gjer at vi får frigitt den kjemiske energien frå maten vi et.



Snakk om

Du slepp ein stein frå 1 meter over bakken, og etterpå slepp du han frå 2 meter over bakken.

1. Kva skjer med farten til steinen akkurat i det han treff bakken i dei to tilfella?
2. Kvifor er det slik, trur du?

Kvifor kan vi ikkje høyre eksplosjonar som skjer i verdensrommet?

Eksempel på energikjeder

Sleppe ein ball frå ei høgd:

Stillingsenergi → rørsleenergi → lyden energi og varmeenergi

Bilkollisjon:

Rørsleenergi → lyden energi og varmeenergi

Energiproduksjon er eigentleg berre energiomforming

Ofte snakkar vi om å produsere energi, men det går ikkje an å produsere eller lage energi, han kan jo berre omformast! Når vi snakkar om energiproduksjon, meiner vi som regel å omforme ein energiform til elektrisk energi. Eksempel på energiomforming:

- Vasskraft som omformar stillingsenergien i store vassmengder til elektrisk energi
- Vindkraft som omformar vinden sin rørsleenergi til elektrisk energi
- Solceller som omformar strålingsenergien frå sola til elektrisk energi
- Fossil energi som omformar den kjemiske energien i olje, kol og gass til elektrisk energi

Kva er energikvalitet?

Han seier noko om kor enkelt det er å bruke energien til noko nyttig. I ei energikjede vil energikvaliteten alltid synke når energien blir omforma til andre typar. Han blir mindre nyttig. Til slutt endar all energi opp som svak varmeenergi. Og denne er veldig vanskeleg å bruke til noko nyttig. Elektrisk energi er veldig enkelt å bruke til noko nyttig. Kva med dei andre energitypane?

Lag ein kjedereaksjon

- programmer ein servo

Oppgåve

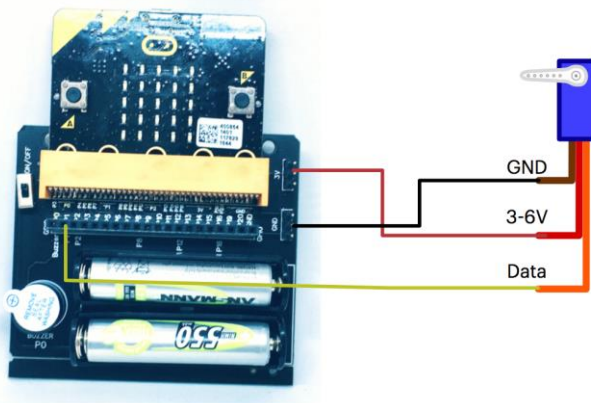
Lag ein kjedereaksjon som forflyttar ei kule minst to meter utan at nokon tek på henne undervegs. Det må nyttast minst tre delar og minst ein servo.

Fase 1:



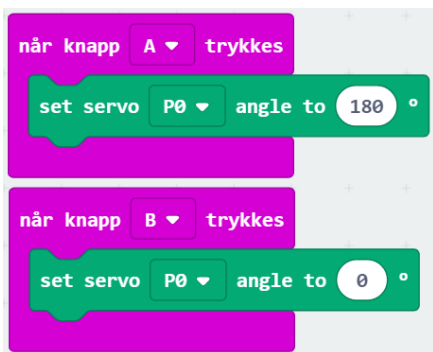
Fase 2: Kvar skal kjedereaksjonen få energien frå? Kva slags materiale og delar skal han vere laga av? Korleis kan du bruke minst ein servo som ein del av kjedereaksjonen din? Teikne ei skisse.

Fase 3: Bygg kjedereaksjonen og lag programmet til servoen.



Kople opp micro:bit og kretskort med servoen som vist på figuren.

Servoen har tre fargar på leidningane der det er viktig at den rette leidningen havnar på den rette plassen.



Døme på program

Det øvste programmet gjer at servoen dreier 180° i den eine retninga.

Det nedste programmet gjer at servoen dreier attende til utgangspunktet.

Kan de bruke noko av dette til dykkar kjedereaksjon?

Fase 4: Prøv ut kjedereaksjonen.

Fase 5: Kan de gjere han betre på nokon måte? Verkar han som planlagt? Må noko bli justert?

Fase 6: Lag den ferdige kjedereaksjonen.

Fase 7: Lag ein filmsnutt av kjedereaksjonen, og skriv eller lag eit lydklipp med forklaring på kvifor de valde å lage kjedereaksjonen akkurat slik de gjorde.

Oppgåve

Studer kula heile vegen gjennom løypa og forklar kva typar energi ho har, og kva typar energi ho mottek og mister undervegs i løypa.

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 3 – Modellering med Geogebra

Lineær modellering

I dette kapitlet skal vi lære litt om lineær regresjon. For å forstå kva det er, skal vi sjå på eit døme.

Når Line kjem inn i klasserommet ein morgon, er det veldig kaldt. Varmeanlegget har vore avslått heile natta, men heldigvis har vaktmeisteren skrudd det på att. Line har eit termometer liggande på pulten sin og ønsker å finne ut kor fort klasserommet blir varma opp att. Ho måler temperaturen kvar halvtime, åpner Geogebra og skriv resultata inn i reknearket. Resultata ser du nedanfor.

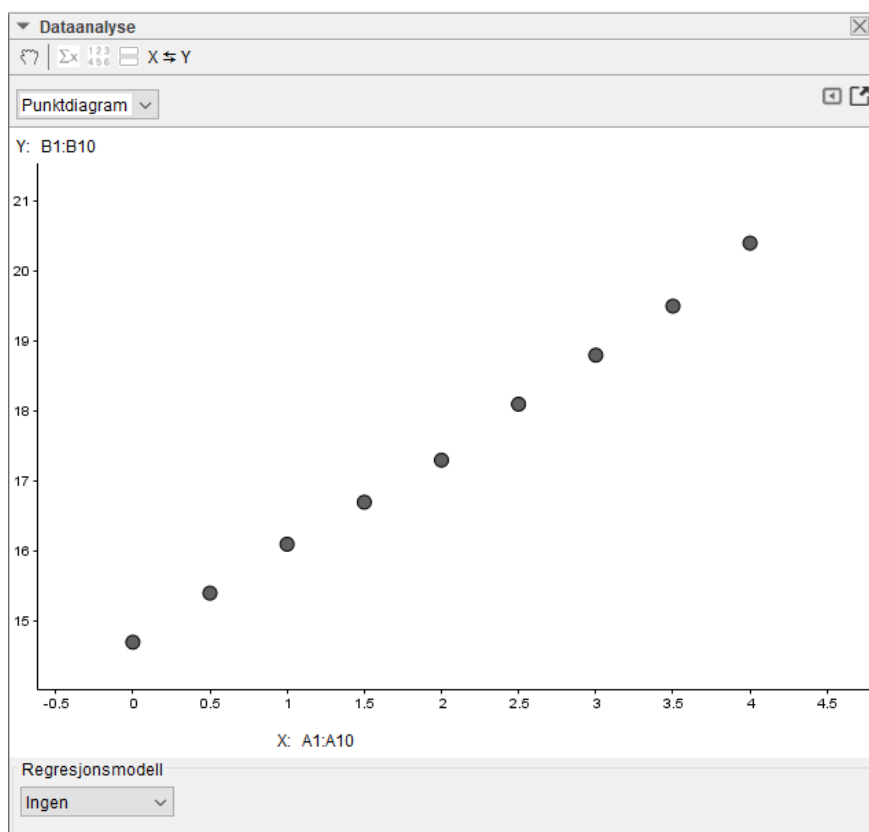
Timer	Grader
0	14.7
0.5	15.4
1	16.1
1.5	16.7
2	17.3
2.5	18.1
3	18.8
3.5	19.5
4	20.4

For å få betre oversikt, ønsker Line å lage ein graf av desse tala. Det er det som blir kalla regresjon og vi skal nå sjå korleis Line gjer det.



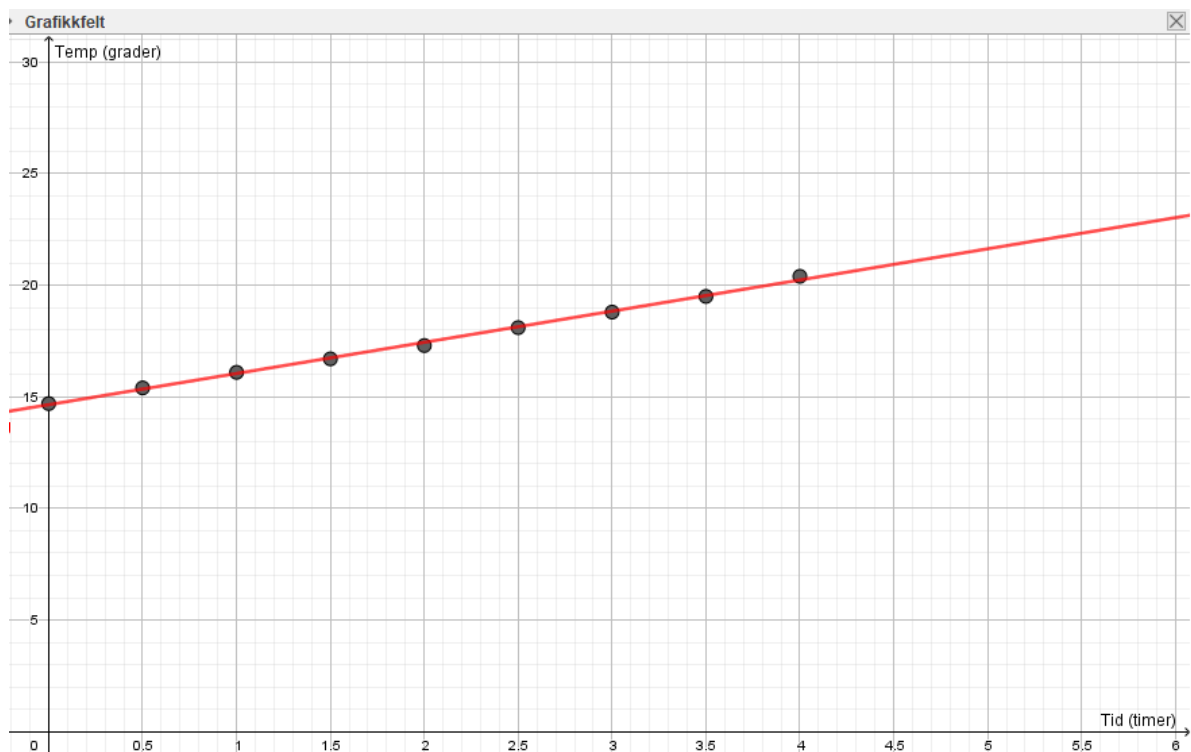
Først markerer ho tala, og så trykker ho på dette symbolet:

Deretter vel ho «regresjonsanalyse» og «analyser». Ho får da opp tala i tabellen som punkt. Vi hugser at det var 14,7 grader da hun starta å måle. Det svarer til punktet nedst til venstre. Etter ein halvtime var det 15,4 grader. Det svarer til det neste punktet, og slik held det fram.



Line ser at punkta ligg omtrent langs ei rett linje, og ho ønsker å få PC-en til å velje den linja som passar best med punkta. Det gjer ho ved å velje «regresjonsmodell» nedst i bildet. Hun vel «lineær» som tyder at grafen skal bli ei rett linje. Hvis punkta ikkje hadde lege langs ei linje, kunne ho valt noko anna, og det skal vi sjå på seinare.

Til slutt høgreklikkar ho i biletet og vel «kopier til grafikkfeltet» og dermed er grafen ferdig.



Funksjonsuttrykket til denne grafen er $y = 1,4x + 14,65$. Kva tyder disse tala?

Det talet som står framfor x-en, blir kalla eit stigningstal. Jo større dette talet er, jo brattare er grafen. Dersom talet er negativt, synk grafen. Her er det 1,4. Det tyder at det blir 1,4 grader varmare for kvar time som går.

Det andre talet blir kalla konstantledd. Her er det 14,65. Det tyder at grafen startar på 14,65 grader, altså at det ifølge grafen var så mange grader da forsøket starta.

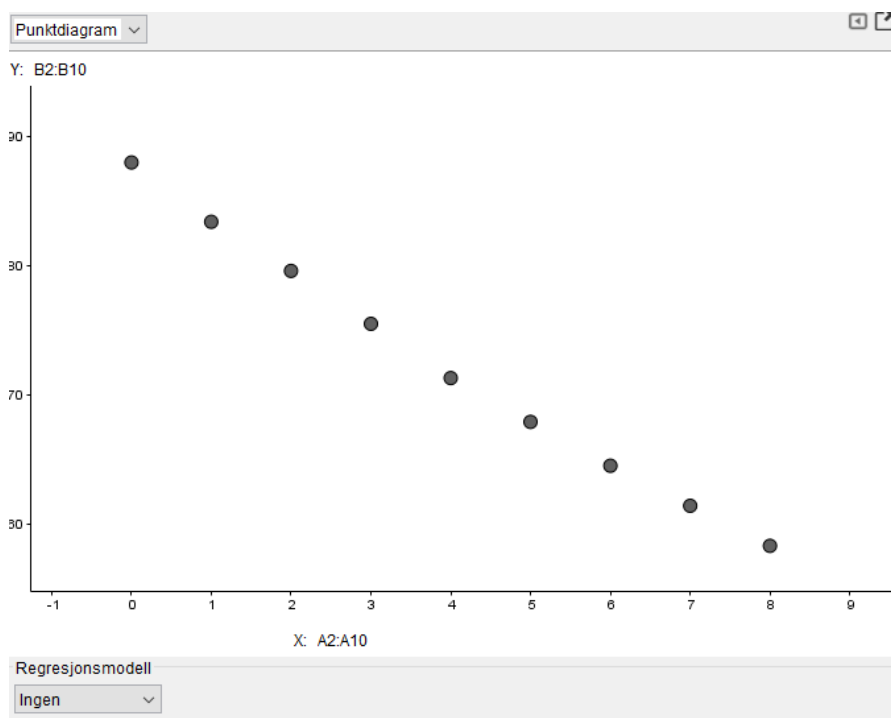
Og med det har Line funne svaret. Klasserommet blir varma opp med 1,4 grader kvar time.

Ikkje-lineær modellering

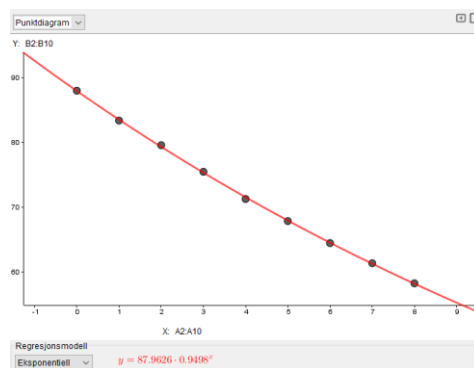
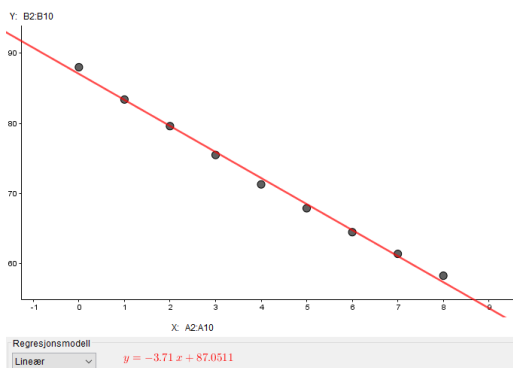
Vi skal nå sjå korleis vi kan bruke Geogebra til å lage grafar når punkta ikkje ligg på ei rett linje.

Markus liker å ha med kakao på tur, og han har fått ein ny termos som han håper skal halde godt på varmen. Han har lyst til å teste termosen ut heime, og fyller han med varmt vatn ein morgon. Så måler han temperaturen ein gong i timen. Han legg inn punkta på same måten som Line (sjå dømet ovanfor).

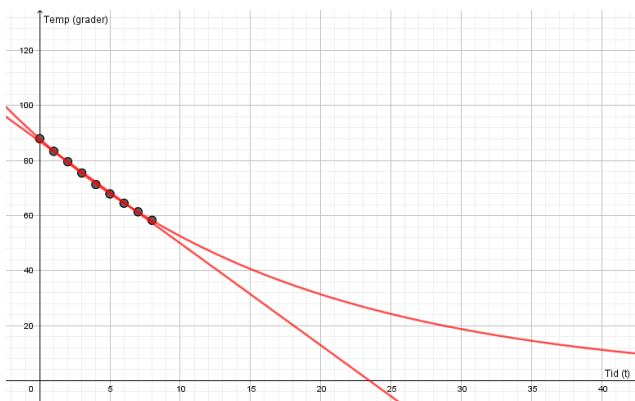
Timer	Temp
0	88
1	83.4
2	79.6
3	75.5
4	71.3
5	67.9
6	64.5
7	61.4
8	58.3



Vi ser at punkta til Markus kan hende ikkje ligg heilt langs ei rett linje slik punkta til Line gjorde. Markus prøver ut forskjellige regresjonsmodellar. Da får han følgande bilete.



Desse grafane ser kan hende litt like ut, men vi ser klart skilnad om vi zoomer litt ut.



Korleis kan vi finne ut av kva for ein av desse som passar best med tala til Markus? For å finne ut av det, må vi sjå på eit tal som blir kalla korrelasjonskoeffisient. Det finn vi ved å trykke på «vis statistikk»-knappen.



Da får vi tabellar som desse.
Det er det nest nedste talet
(R^2) vi er interesserte i.

For den rette linja får vi 0.996.
For den krumme linja får vi 0.9998.

Jo nærmare dette talet er 1, jo betre
passar grafen. Dette tyder at det er den
krumme grafen (den eksponentielle) som
passar best.

Nå har Markus fått en fin graf som viser
temperaturendringa i termosen over tid.

Lineær graf

Statistikk	
GjennomsnittX	4
GjennomsnittY	72.2111
Sx	2.7386
Sy	10.1807
r	-0.998
p	-1
Sxx	60
Syy	829.1689
Sxy	-222.6
R^2	0.996
SSE	3.3229

Krum graf (eksponentiell)

Statistikk	
GjennomsnittX	4
GjennomsnittY	72.2111
Sx	2.7386
Sy	10.1807
r	-0.998
p	-1
Sxx	60
Syy	829.1689
Sxy	-222.6
R^2	0.9998
SSE	0.2014

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 4

Variablar i Scratch og MakeCode

Ein variabel er som ei lita skuffe som kan innehalde ein verdi. For å kunne utføre reknestykke, treng vi skuffer til å ha tala våre i. Kvar gong du puttar eit nytt tal i skuffa, forsvinn det gamle, for skuffene er små og har berre plass til det nyaste talet.

Det finst fire typar variablar:

Heiltal – Tal utan desimalar

Desimaltal – Tal med desimalar

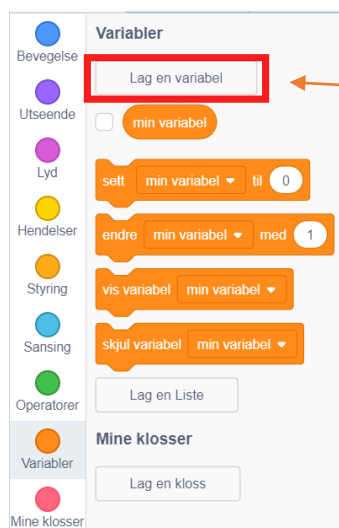
Tekst/strenger – Ord eller bokstavar

Boolske verdier – Sant eller usant



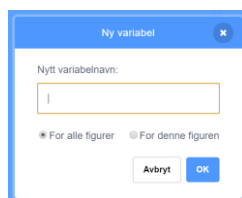
Variablar er nyttige når ein skal kunne lage eit program som kan rekne ut reknestykke med ulike tal, utan å endre heile programmet. I tillegg kan variablar bli brukte for å halde styr på talet på rundar i ei lykkje med ein tellevariabel. Sjå Pythonheftet for meir om variablar.

Variablar i Scratch



For å kunne lage variablar i Scratch, må du finne menyen for variablar – han er mørk oransje. Så må du trykke på den øvste knappen der det står Lag en variabel. På figuren her markert med raudt.

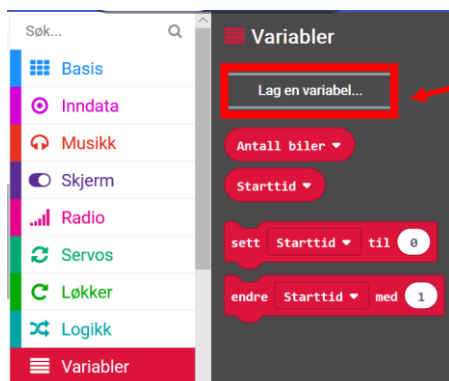
Da dukker dette vindauget opp, og du lagar eit namn på variabelen din.



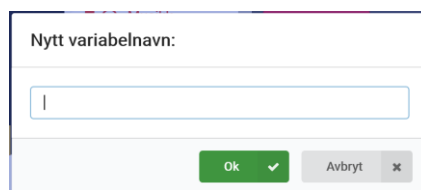
Dette gjentek du til du har mange nok variablar.

Variablar i MakeCode

For å kunne lage variablar i MakeCode, må du finne menyen for variablar – han er raud. Så må du trykke på den øvste knappen der det står Lag en variabel. På figuren markert med raudt.



Da dukkar dette vindauget opp, og du lagar eit namn på variabelen din.



Dette gjentek du til du har mange nok variablar.

Opplegg 5 - Massetettleik og periodesystemet

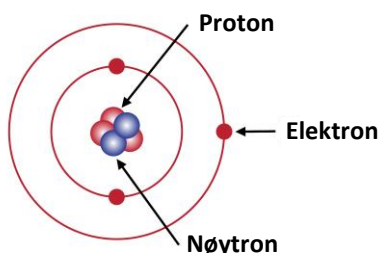
Kva er masse?

Masse er det vi måler når vi veg noko. Vi kallar det ofte tyngd, men tyngda er eigentleg den krafta som jordkloden dreg oss ned mot bakken med (tyngdekraft). Masse måler vi i kilogram.

Kva er massetettleik?

Han seier noko om kor stor tettleik eit stoff har. Om du held ein liten bit av eit stoff med stor massetettleik, vil det vere tungt å halde. Eininga for massetettleik er enten kg/m^3 eller g/cm^3 .

$$\text{massetettleik} = \frac{\text{masse}}{\text{volum}}$$



Alle atom er bygd opp av proton, nøytron og elektron. Atomkjernen består av proton og nøytron. Protona har ei positiv ladning, men nøytrona har inga ladning, dei er nøytrale. Elektrona går i bane rundt atomkjernen og dei har ei negativ ladning. I eit nøytralt atom er det alltid like mange elektron og proton.

Visste du at?

Ei teskei med nøytronstjerne veg omtrent 500 millionar tonn!

Atommassane til grunnstoffa er i hovudsak fastsette av talet på proton og nøytron i kjernen, sidan elektron har så liten masse til samanlikning. Atommassane aukar når atomnummeret aukar, for det seier kor mange proton atomet har i kjernen. For dei lettaste grunnstoffa har atomet omtrent like mange nøytron som proton, men for de tyngre grunnstoffa er det fleire nøytron enn proton.

Atommassane avgjer svært lite av massetettleiken til grunnstoffa. Det er ikkje slik at massetettleiken automatisk blir høgare for grunnstoffa med høgare atomnummer. Ein del av grunnen er at for dei fleste grunnstoffa er atoma bundne til kvarandre, slik at det blir molekyl av grunnstoffet. Måten desse er bundne saman på, er med på å fastsetje kor stor plass molekyla tek. Altså kor stort volum dei har.

Innanfor kvar kolonne (gruppe) i periodesystemet er det likevel slik at grunnstoffa med høgast atomnummer har størst massetettleik. Grunnen er at desse grunnstoffa har liknande kjemiske eigenskaper sidan dei har like mange valenselektron (elektron i ytste skal), som er det viktigaste for å avgjere bindingstype og kjemiske eigenskaper til stoffet.

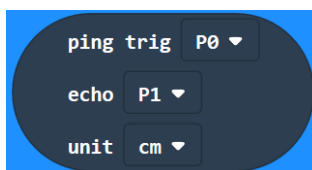
Diskuter

Kva for nokre ulike typar bindingar veit du om?

Ultralydsensor (sonar)

1. For å kunne nytte ein ultralydsensor (sonar), må du først trykke på Avansert i menyen. Så må du trykke på Utvidelser, nedst i menyen; markert med raudt på figuren her.

2. Trykk så på ruta der det står sonar.



Da får du tilgang til denne programmerings-klossen. Han viser avstanden frå sonaren målt i cm eller mikrosekund.

Ultralydsensoren måler ikkje avstand direkte. Han måler tida det tek for ultralyd å gå frå sensoren, bort til ein overflate, så attende. Sidan sensoren veit at lydfarten er ca 340 m/s i luft, kan han rekne tida om til avstand.

Dessverre oppgjer sensoren avstanden i heile cm, altså ikkje særleg nøyaktig. Kva kan vi gjere med det? Kan det vere meir nøyaktig å måle tida, og heller rekne om sjølv? (HINT: $s = v \cdot t$)

Lag ein volummålar

- finn massetettleiken til eit stoff

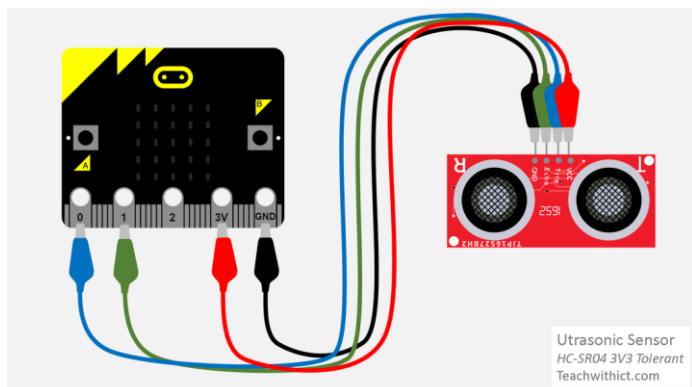
Oppgåve

Lag eit produkt som måler volumet til eit stoff med å bruke ein ultralydsensor (sonar) til å måle kor mykje vasshøgda i eit begerglas aukar når du legg i ein bit av stoffet. Bruk massen av biten og volumet for å rekne ut massetettleiken og finne ut av kva for eit stoff det er.

Fase 1: Kva for nokre delar er ein volummålar samansett av? Korleis ser disse delane ut? Kva må bli bygd saman av desse delane?

Fase 2: Ha ei idémyldring for deg sjølv. Korleis vil du at volummålar din skal sjå ut? Teikne gjerne ei skisse før du diskuterer med dei andre.

Fase 3: Tid for å lage volummålar og programmere micro:biten. Kople opp micro:bit som vist på figuren.



Naudsynt utstyr til volummålar

Kalibrering av volummålar (enkel regresjon)

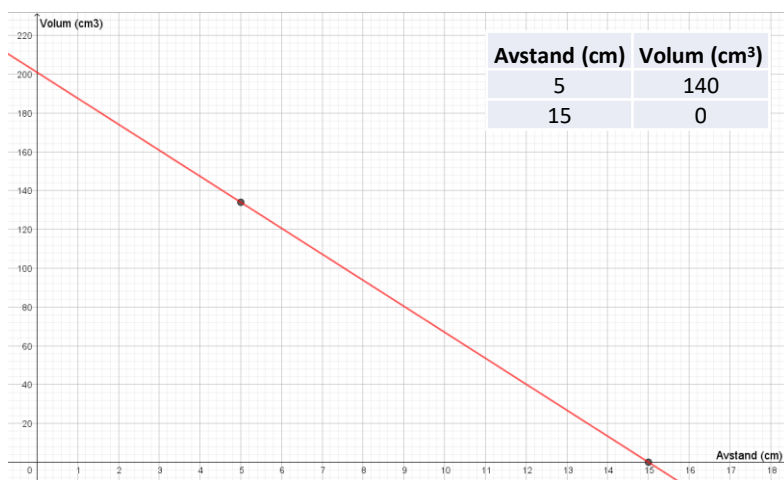
Vi skal finne massetettleiken til ulike stoff. Da må vi måle kor stor masse og volum dei har. Men det identifikatoren vår måler, er avstand, og ikkje volum. Heldigvis er det ein samanheng mellom dei!

Vi gjer først ei måling av avstanden til overflata på vatnet utan noko oppi det, og noterer avstanden (volumet er 0 sidan vi ikkje har noe oppi vatnet). Så legg vi oppi ein halvstor gjenstand og les av både avstanden til vatnet og volumet på skalaen på sida av begerglaset. Da har vi fått to punkt som vi kan teikne inn i et koordinatsystem.

Dette kan du enten gjere for hand eller digitalt. Når du har teikna de to punkta, trekk du ei rett linje mellom dei. Da har du fått ein graf du kan bruke til å lese av volumet for den avstanden som blir vist på micro:biten.

Dersom du vil lage ein meir nøyaktig graf, gjer du flere målingar og teiknar inn punkta i same koordinatsystem. Da vil det ikkje passe med ei rett linje som treff alle punkta dine, men du teiknar opp den rette linja som treff best mogleg med dine punkt.

Da har du gjort ein regresjon. Det vil seie at du har laga ein matematisk modell! Gratulerer! Det er det som er det største ønsket til mange forskarar innanfor realfag. Det finst enklare måtar å gjere det digitalt på, slik at du slepp å teikne inn alle punkta og linja for hand. Det står det meir om i modellerings-kapitlet.



Diskuter

Kvifor vil det ikkje vere mogleg å finne massetettleiken til salt på denne måten?

Gjer fase 4 – 7.

Opplegg 6 - Friksjon og omforming av energi

Friksjon hindrar overflater i å røre seg i høve til kvarandre. Han verkar slik at han alltid bremsar ei rørsle. Hadde det ikkje vore for friksjon og luftmotstand (som er ein slags friksjon), hadde alle rørsler fortsett utan å stoppe opp!

Tenk at du aker ned ein bakke med akebrett. Ved enden av bakken er det ei diger slette som er dekkja av eit islag med vatn oppå. Kva skjer med farten din på sletta? Kva om det i staden hadde vore snøslaps, eller kan hende gras? Skilnaden på is med vatn, snøslaps og gras er kor stor friksjonen blir. Jo glattare overflata er, desto mindre friksjon har vi, og akebrettet held fram veldig langt før det stoppar. Vi har uansett luftmotstanden som bremsar rørsla, så akebrettet held ikkje fram uendeleg lenge.

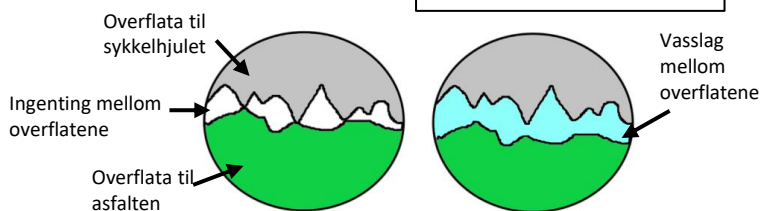
Det er kor ujamne overflatene er, som avgjer kor stor friksjonen er. Vi kan tilsetje smøremiddel mellom ujamne overflater, slik som i bilmotorar og andre stader der overflater skal røre seg i høve til kvarandre. Kva skjer om friksjonen er stor inni ein bilmotor?

I andre høve ønsker vi ikkje å redusere friksjonen. Kva med når du

syklar, veit du kva vassplaning er? Det er når det legg seg eit lag med vatn på asfalten, og det verkar på same måten som eit smøremiddel. Skummelt!

Diskuter

Veit du om ein stad utan luft, og dermed utan luftmotstand? Kva for nokre konsekvensar får det for rørsle?



To overflater når det ikkje er tilsett noko mellom dei – friksjonen er stor

Eit tynt lag vatn skil dei to overflatene frå kvarandre – friksjonen er liten

Omforming av energi

Friksjon gjer at farten minkar, og da blir rørsleesenergien og mindre. Kvar blir resten av energien av? Han kan jo ikkje berre forsvinne.

Energien går stort sett over frå rørsleenergi til varme- og lydenergi. Han kan og gå med til å deformere overflatene.

På biletet ser du ein ball som vert sleppt frå ei høgd, og for kvar gong han sprett, når han kortare opp. Det er friksjonen mellom ball og golv som omformar (stel) noko av rørsleenergien til ballen, slik at han ikkje når like høgt neste gong.

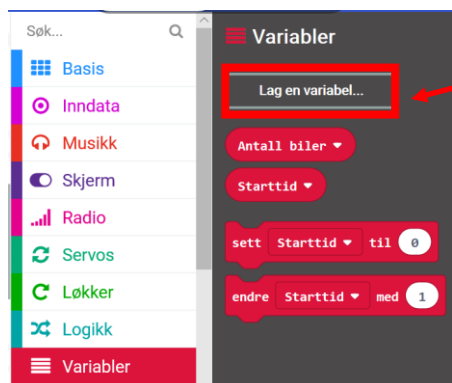


Variablar i MakeCode

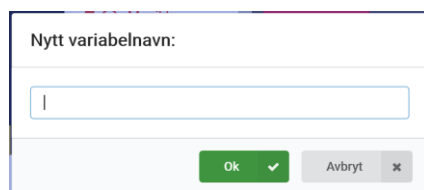
Ein variabel er som ei lita skuffe som kan innehalde ein verdi. For å kunne utføre reknestykke, treng vi skuffer til å ha tala våre i. Kvar gong du legg eit nytt tal i skuffa, forsvinn det gamle, for skuffene er små og har berre plass til det nyaste talet. Desse skuffene må vi først lage.



For å kunne lage variablar i MakeCode må du finne menyen for variablar – han er raud. Så må du trykke på den øvste knappen der det står Lag en variabel, markert med raudt på figuren.



Da dukker dette vindauget opp, og du lagar eit namn på variabelen din.



Dette tek du opp att til du har mange nok variablar.

Lag eit skråplan der vinkelen kan justerast - og ei stoppeklokke med micro:bit

Oppgåve

Lag eit justerbart skråplan som de skal nytte til å la ei kule trille ned med ulike vinklar. Mål tida det tek for kula å trille ein avmerkt avstand på skråplanet med micro:biten. Lag ein lineær modell for korleis trilletida avheng av vinkelen til skråplanet.

Fase 1: Kva er eit skråplan? Kva kan det bli bygd av? Korleis få det regulerbart?

Fase 2: Det er viktig at de er opne for alle slags idear og ikkje er kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

1. Tenk sjølv først og teikne gjerne skisser.
2. Forklar ideen din for dei andre på gruppa.
3. Heile gruppa diskuterer dei ulike ideane, og lagar ein felles hypotese for resultatet av målingene.

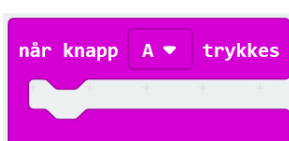
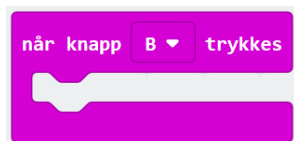
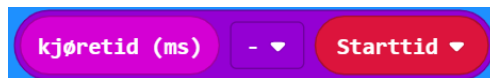
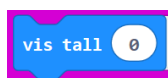
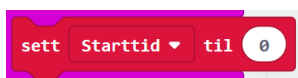
Fase 3: Tid for å lage skråplanet og programmere micro:biten. Under er det eit puslespel-hint til stoppeklokke-programmet.

Kjøretid-klossen

Denne klossen måler kor lang tid det har gått sidan programmet tok til å køyre. Han byrjar å telje med ein gong programmet er ført over til micro:biten. Han finst under kategorien inndata, der det står meir...

Han måler tida i millisekund. Kva må du gjere for å få tida i sekund?

kjøretid (ms)



Fase 4: Prøv ut skråplanet og stoppeklokka.

Fase 5: Verkar det som planlagt? Må noko justerast? Kan du gjere skråplanet eller stoppeklokka betre på nokon måte?

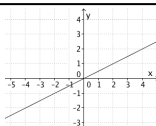
Fase 6: Lag det ferdige skråplanet og stoppeklokka.

Fase 7: Gjennomfør målingar av trilletid for minst 5 ulike vinklar. Skriv ned målingene og ta bilete.



Regresjonsoppgave – lineær modell

1. Plott alle dei målte verdiane i Geogebra.
2. Finn ein matematisk modell ved å gjere ein regresjon for dei målte dataa. Sjå modelleringskapittelet for framgangsmåte.
3. Korleis trur du modellen ville blitt om de hadde gjort fleire målingar?
 - Prøv!
 - Passa dei nye målingane inn i modellen dykkar?
 - Var dei nye målingane innimellom dei gamle verdiane, eller utanfor? Trur du det har noko å seie for resultatet?



Diskuter

Stemte hypotesen dykkar? Kvifor/kvifor ikkje?

Oppgåve

Modellen dykkar er ein lineær funksjon som kan skrivast på denne formen

$$f(x) = a \cdot x + b$$

Dersom de set dykkar funksjon lik ei anna gruppe sin funksjon har de laga ei likning! Løys denne likningen, og diskuter kva de har rekna ut.

Opplegg 7 - Solenergi

- Korleis kan vi varme vatn ved hjelp av solenergi?

Energiproduksjon fra sollys



Energien frå sollyset er fornybar energi, sidan vi ikkje kan bruke han opp. Sollyset som treff jordkloden tilsvarar ei mengd energi som er 15 000 gonger høgare enn energiforbruket til alle menneske på jorda. Dersom vi hadde klart å utnytte heile energien frå sola, hadde vi ikkje trengt annan energiproduksjon.

Det er hovudsakeleg to måtar å produsere energi frå sollys. Vi kan bruke solceller eller solfangarar. I ei solcelle blir det

danna elektrisitet, sidan sollyset kan slå laus elektron. Disse elektronene rører seg i ein krins på grunn av måten solcellene er bygde opp på, og vi får danna elektrisitet.

Det finst to typar solfangarar. Begge verkar slik at dei varmar opp vatn eller ei anna væske. Skilnaden på dei to typane er kor høg temperatur dei varmar væska opp til, og kva dei gjer med væska etter at ho er oppvarma.

1. **Solfangar til elektrisitetsproduksjon** – denne typen inneheld mange speglar som reflekterer sollyset slik at det treff ein liten behaldar med væske. Da vil temperaturen til denne væska auke kraftig og væska fordampar. Det vil seie at ho blir til gass. Da kan gassen drive ein turbin som omformar gassen sin rørsleenergi til elektrisitet. Dette er typisk større energikraftverk som produserer elektrisitet nok til mange tusen husstandar, slik som på biletet over.
2. **Solfangar til oppvarming av vatn** – denne typen er langt enklare, og noko privatpersonar kan ha heime. Dei bruker energi frå sollyset til å varme opp vatn til opp mot kokepunktet. Mange nyttar slike solfangarar på hytta eller campingtur, for å varme opp dusjvatn. Dei er laga av en svart pose med vatn oppi, sidan svart farge absorberer mest mogleg av strålinga frå sollyset, og ein plastslange med eit lite dusjhovud. Her får vatnet som regel ikkje veldig høg temperatur, men det kan vere lurt å sjekke litt forsiktig før ein dusjer heile seg.

Liknande solfangarar kan og brukast av menneske som bor på stader utan tilgang til elektrisitet for å koke vatnet sitt, slik at bakteriar og parasittar dør. Da er det mindre fare for å bli sjuke. I tillegg kan dei få laga varm mat, sjølv om dei ikkje har tilgang på ved. Desse solfangarane er gjerne laga av papp eller tre, noko reflekterande, noko absorberande, noko som isolerer, og ein behaldar til vatnet.



Materiala som ein solfangar er laga av har alle ein funksjon.



Papp - byggemateriale



Plastfolie - isolerer



Aluminiumsfolie - reflekterer sollys



Svart maling – absorberer sollys

Snakk om

All energi på jordkloden stammar til sist frå solenergi, både fornybare energikjelder og ikkje-fornybare energikjelder. Til dømes vindkraft, vasskraft, olje, gass og kol, endatil energien i maten vi et!

1. Korleis kan det stemme? Forklar korleis det kan vere slik.
2. Kan du komme på nokon unnatak? Er det nokon energikjelder som ikkje har sitt opphav i solenergi?



Lage ein effektiv solfangar

Oppgåve

Lag ein solfangar. Mål kor mykje temperaturen stig for kvart minutt solfangaren står i sola. Lag ein lineær modell for korleis temperaturen avheng av tida.

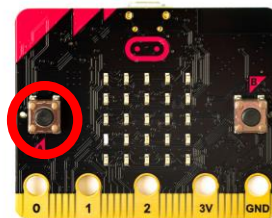
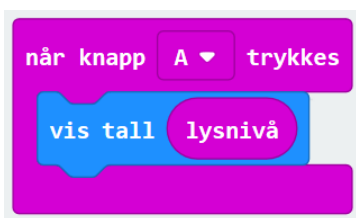
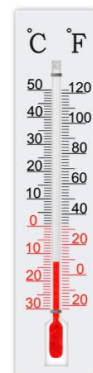
Fase 1: Undersøk litt meir om solfangarar om de ønsker det.

Fase 2: Det er viktig at de er opne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

- Tenk sjølv først, og teikne gjerne skisser.
- Forklar ideen din for dei andre på gruppa.
- Heile gruppa diskuterer dei ulike ideane, og lagar ein felles hypotese for bygging.

Fase 3: Gjennomfør planen dykkar for å lage solfangaren og lag programmet for å måle temperaturen med micro:bit.

Lag eit program som måler temperaturen når du trykker på den eine knappen. Under ser du eit programdøme som måler lysstyrken når ein trykker på knapp A, og skriv verdien til LED-lysa på framsida av micro:biten (skjermen). Kva må du endre på?



Knapp A finn du her på micro:biten

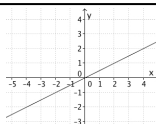
Fase 4: Test kor mykje temperaturen aukar.

Fase 5: Sammenlikne resultata med andre i klassen. Fekk nokre andre større temperaturauke? Kvifor gjorde dei det, trur du? Kan gruppa di gjere noko av det same?

Fase 6: Gå attende til dei andre fasane for å gjere dei planlagte forbetringane.

Fase 7: Gjennomfør dei siste målingane, desse skal de bruke til å plote ein graf med.

Regresjonsoppgåve – lineær modell



1. Plott alle dei målte verdiane i Geogebra.
2. Finn ein matematisk modell ved å føreta ein regresjon for dei målte dataa. Sjå modelleringskapittelet for framgangsmåte.
3. Korleis ser funksjonen ut?
4. Passar han bra med datapunkta dykkar?
5. Trur de modellen vil passe bra dersom de let solfangaren stå ute i sola ein heil dag?

Ekstraoppgåver

1. Kva kan vere moglege feilkjelder for måling av temperaturen?
2. Forklar kor viktig det er å teste-evaluere-forbetre.
3. Kan de finne ut av kva faseovergangar er?
4. Kople resultata dykkar mot faseovergangar, kan hende går ein del av energien til at vatnet fordampar. Korleis kan ein unngå dette?

Oppgåve

Modellen dykkar er ein lineær funksjon som kan skrivast på denne formen

$$f(x) = a \cdot x + b$$

Dersom de set funksjonen dykkar lik ei anna gruppe sin funksjon, har de laga ei likning! Løys denne likninga, og diskuter kva de har rekna ut.

Opplegg 8 - Formlar, likningar og fart

Kva er fart?

Farten fortel oss kor raskt vi flyttar oss frå ein stad til ein annan. Når en gjenstand flyttar seg, er gjennomsnittsfarten lik strekninga dividert med tida det tok. Dette kan vi skrive med formelen $v = \frac{s}{t}$ der v er farten, s er strekninga og t er tida.



Likningar

Ei likning er samansett av ein ukjend variabel (ofte bruker vi x), eit likskapsteikn og minst eitt tal. Det som er hovudpoenget med likningar, er at dei inneheld eit likskapsteikn som fortel oss at det på venstre side er likt som det på høgre side av likskapsteiknet.

Dette er like stort $\rightarrow 3x - 2 = x + 4 \leftarrow$ som dette

Sidan vi veit at det som står på begge sidene av likskapsteiknet er like stort, må vi alltid gjere det same på begge sider av likskapsteiknet. Dersom vi legg til eit tal, MÅ vi gjere det på begge sidene. Det same gjeld om vi trekk frå eit tal, multipliserer eller dividerer. Når vi har to ting som er like, må vi alltid gjere det same med begge for at dei framleis skal vere like. Er ikkje det logisk?

Vi prøver å løyse dømet vårt. Det finst fleire moglege rekkefølger for kva vi gjer, og de kan velge heilt fritt blant dei, så lenge de ALLTID GJER DET SAME PÅ BEGGE SIDER.

Samle alle ledd med x på den venstre sida, da må vi trekke frå alle x -ane som er på høgre sida:

$$3x - 2 - x = x + 4 - x$$

Så trekk vi saman uttrykka på kvar side og får:

$$2x - 2 = 4$$

Det neste steget er å samle alle tala på høgre sida, da må vi finne ut kva vi må legge til eller tekke frå for å kvitte oss med tala på venstre sida:

$$2x - 2 + 2 = 4 + 2$$

Så trek vi saman uttrykka på kvar side og får:

$$2x = 6$$

Da er vi nesten i mål, men vi må berre kvitte oss med talet framfor x . Det står jo eigentleg eit gangeteikn mellom talet og x , så dersom vi deler på same talet, kan vi stryke talet. Dette fordi det blir som å multiplisere med ein brøk som er lik 1, slik som $\frac{2}{2} = \frac{3}{3} = \frac{6}{6} = \frac{1000}{1000} = 1$.

Da får vi:

$$\frac{2x}{2} = \frac{6}{2}$$

Når vi til slutt forkortar, står vi att med:

$$x = 3$$

Og vi har funne svaret vårt! Så lenge vi gjer det same på begge sidene av likskapsteiknet, treng vi eigentleg ikkje hugse noko meir for å løyse likningar.

Formelrekning

Ein formel er eigentleg veldig lik ei likning, og vi kan nytte akkurat dei same reglane for å omforme han. For $v = \frac{s}{t}$ som vi starta med, så kan vi omforme han til $s = v \cdot t$. Klarer du sjå kva ein må gjere?

Lag det raskaste køyretøyet

- med kontinuerlig servo

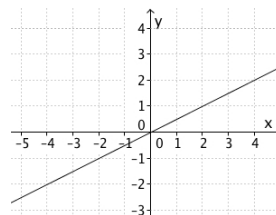
Oppgåve

Lag eit køyretøy som bruker ein kontinuerlig servo. Mål kor lang tid det brukar på minst fem ulike strekningar. Lag ein lineær modell for korleis tida avheng av strekninga.

Fase 1: Undersøk gjerne litt for å få inspirasjon til køyretøyet dykkar.

Fase 2: Det er viktig at de er opne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

- Tenk sjølv først, og teikne gjerne skisser.
- Forklar ideen din for dei andre på gruppa.
- Heile gruppa diskuterer dei ulike ideane, og lagar ein felles hypotese for bygging.



Fase 3: Gjennomfør planen dykkar for å lage køyretøyet, og lag programmet for å styre servoen.

Ekstraoppgåve

Lag ei stoppeklokke for micro:bit for å måle tida på strekningane.



Fase 4: Test kor raskt køyretøyet er.

Fase 5: Samenlikne resultatane med andre i klassen. Fekk nokon andre større fart? Kvifor trur du deira køyretøy fekk større fart? Kan de gjere noko av det same?

Fase 6: Gå attende til dei andre fasane for å gjere dei planlagde forbetringane.

Fase 7: Gjennomfør dei siste målingane for strekning og tid, disse skal de bruke til å plote ein graf med.



Regresjonsoppgåve – lineær modell

1. Plott alle dei målte verdiane i Geogebra. Tida tilsvarende x-verdiane og strekninga tilsvarende y-verdiane.
2. Finn ein matematisk modell med å gjere ein lineær regresjon for dei målte data. Sjå modellerings-kapitlet for framgangsmåte.
3. Passar modellen bra med datapunkta dykkar?
4. For denne lineære modellen vil stigningstallet vere gjennomsnittsfarten til køyretøyet. Det køyretøyet med størst stigningstal, har størst fart. Korleis kan de sjå på grafen kva for eit køyretøy som har størst fart?
5. Tror de modellen vil passe bra dersom de lar køyretøyet køyre i 12 timar?

Oppgåve

Modellen dykkar er ein lineær funksjon som kan skrivast på denne formen

$$f(x) = a \cdot x + b$$

Dersom de set funksjonen dykkar lik ei anna gruppe sin funksjon, har de laga ei likning! Løys denne likninga, og diskuter kva de har rekna ut.

Ekstraoppgåve

Når de har funksjonen for samanhengen mellom tida og strekninga til køyretøyet, finn ut hva strekninga måtte vore for at køyretøyet skulle kjørt i 5 timar.

Tror de at svaret er realistisk?

Kan de laga ei anna likning basert på tida og strekninga til køyretøyet dykkar?

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 5

Opplegg 9 - Rytme

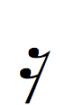
Rytme er sett saman av fleire andre omgrep, slik som betoning, puls og taktart. Korleis ein opplever tempoet varierer og med kva for nokre slag som blir markerte med trommer eller eit anna rytmeinstrument.

Det finst mange ulike taktartar, og noko musikk har til og med fleire taktartar i same stykket! Taktartar skrivest som brøkar, og i vesten nyttar vi i hovudsak disse taktartane: $\frac{2}{4}$, $\frac{3}{4}$, $\frac{4}{4}$ og $\frac{6}{8}$, men dette avheng av kultur.

Marsjane til skulekorpset på 17.mai går typisk i $\frac{2}{4}$ -takt eller $\frac{4}{4}$ -takt, for at ein skal kunne marsjere til dei. Vals går i $\frac{3}{4}$ -takt, og det er ganske håplaust å danse vals til musikk med annan taktart.

Notar og varigheit

Notane har ulik utsjånad ut frå kor lenge ein skal halde dei når ein spelar eller syng. Under visest ein oversikt over varigheit til notar og pausar.

	Helnote		Helpause
	Halvnote		Halvpause
	4.delsnote		4.delspause
	8.delsnote		8.delspause
	16.delsnote		16.delspause



Takt

Vi nyttar taktar for å dele opp musikken, og ut frå kva for ein taktart som er nytta, vil musikken ha mange ulike stilar. Taktarten er ein brøk, og vi treng faktisk brøkrekning for å skrive kva for nokre notar som kan vere med i ein takt med en bunden taktart. For $\frac{4}{4}$ -takt, vil vi ha enten ein heilnote, to halvnotar, fire 4.delsnotar, eller andre kombinasjoner som til saman blir $\frac{4}{4}$ (eller ein heil). Sjå døme på lovlege/ulovlege kombinasjoner under.

Ulovleg $\neq \frac{3}{4}$

$$\frac{3}{4} \text{ (Ulovleg)} \quad \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{4} = \frac{7}{8} < \frac{3}{4}$$

$$\frac{3}{4} \text{ (Ulovleg)} \quad \frac{1}{8} + \frac{1}{2} = \frac{5}{8} < \frac{3}{4}$$

$$\frac{3}{4} \text{ (Ulovleg)} \quad \frac{1}{16} + \frac{1}{16} + \frac{1}{16} + \frac{1}{2} + \frac{1}{8} = \frac{13}{16} > \frac{3}{4}$$

Betoning

Det at ein legg meir trykk på nokre notar/tonar enn andre og framhever disse. Dei betonte tonane verkar dermed tyngre enn dei andre.

Kva er puls?



Pulsen fortel oss kor raskt musikken går, eller kva for ein fart han har. Det kallast og for tempo. Vi bruker gjerne eit metronomtal for å seie noko om tempoet, og det målest i BPM. BPM står for beats per minute, altså taktslag i minuttet.

Det går an å bruke ein metronom for å halde tempo. Før var disse analoge og hadde eit slags urverk som gjorde dei i stand til å lage eit visst tal klikkelyder i minuttet. No er det vanlegare med digitale metronomar, der det er elektronikk som gjer at vi får svake tikkelydar.

Det går og an å ha ein metronom som likner på eit armbandsur. Han kan gje ein liten vibrasjon i staden for lyd ved kvar puls i musikken.

Lovleg $= \frac{3}{4}$

$$\frac{3}{4} \text{ (Lovleg)} \quad \frac{1}{4} + \frac{1}{4} + \frac{1}{4} = \frac{3}{4}$$

$$\frac{3}{4} \text{ (Lovleg)} \quad \frac{1}{2} + \frac{1}{8} + \frac{1}{8} = \frac{6}{8} = \frac{3}{4}$$

$$\frac{3}{4} \text{ (Lovleg)} \quad \left(\frac{1}{4} + \frac{1}{8}\right) + \frac{1}{8} + \frac{1}{4} = \frac{6}{8} = \frac{3}{4}$$

Lag eit programmerbart instrument

Oppgåve

Lag eit rytmeinstrument som du må bruke minst ein servo for å få lyd i.

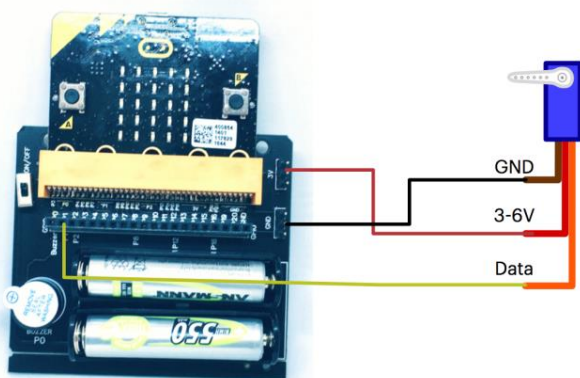


Fase 1: Veit du kva eit rytmeinstrument er? Kan du finne døme på ulike typar rytmeinstrument? Kva er skilnaden på eit rytmeinstrument og eit strengeinstrument eller eit blåseinstrument?

Fase 2: Ha ei idémyldring for deg sjølv. Korleis vil du at instrumentet ditt skal sjå ut? Kva for nokre materiale vil du nytte? Korleis kan du få akkurat den typen lyd du har mest lyst å lage?

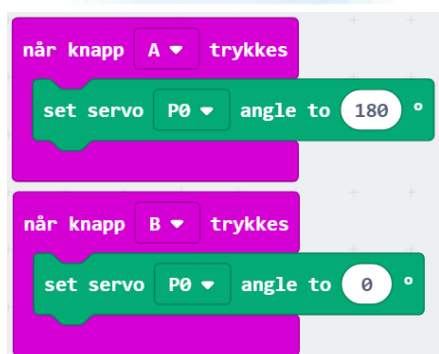
Det kan vere lurt å teikne ei skisse av instrumentet ditt.

Fase 3: Tid for å lage instrumentet. Bruk gjerne skissa frå fase 2. Så må du lage eit program som får rytmeinstrumentet til å lage ein rytmisk lyd. Kva slags type servo er den beste å bruke til ditt instrument?



Kople opp micro:bit og kretskort med servoen som vist på figuren.

Servoen har tre fargar på leidningane der det er viktig at den rette leidningen havnar på den rette staden.



Døme på program

Det øvste programmet gjer at servoen dreier 180° i den eine retninga.

Det nedste programmet gjer at servoen dreier attende til utgangspunktet.

Kan du bruke noko av dette til instrumentet ditt?

Fase 4: Nå er det på tide å kombinere instrumentet du har laga med programmet du har laga.

Fase 5: Verkar instrumentet ditt slik du hadde tenkt? Blei lyden fin? Har nokon andre i klassen kulare lyd? Kvifor er den lyden kulare? Eller vil du endre rytmen?

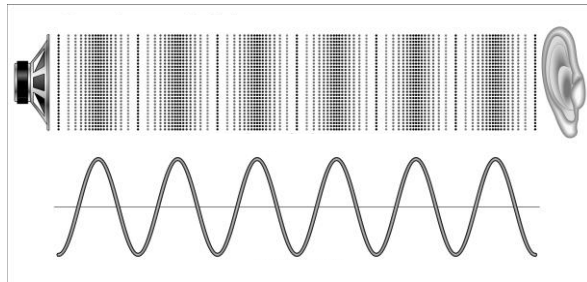
Fase 6: Kan du gå attende til ein av de andre fasane for å lage eit enda betre instrument? Du kan endre både programmet og sjølve instrumentet ditt.

Fase 7: Til slutt kan du forklare korleis du har gjort for å lage instrumentet ditt, og kvifor du valde akkurat din måte å lage rytme på. Skriv, teikne eller lag ei lydfil med forklaringa.

Ekstraoppgåver

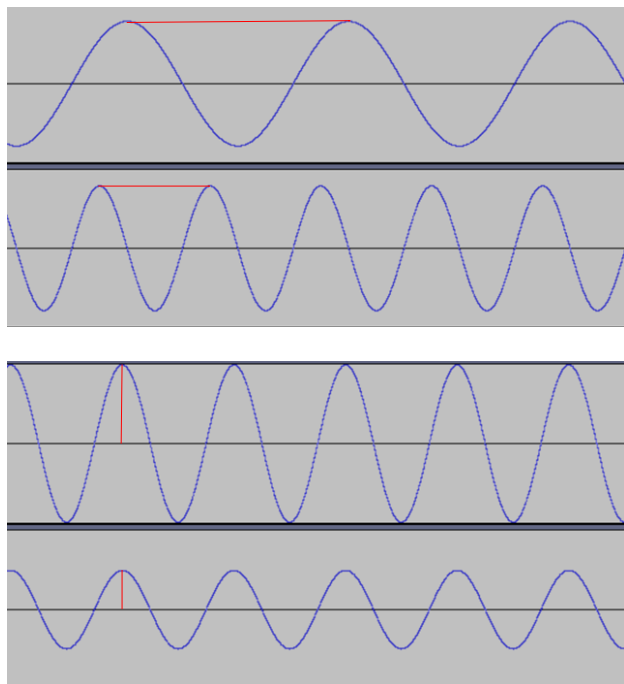
1. Kan du kombinere fleire ulike måtar å lage lyd på, og lage eit orkester?
2. Gå saman med nokre andre og lag musikk som spelar i takt.

Opplegg 10 - Lyd og lydbølger



Kva er lyd?

Lyd er når luftmolekyla vert sett i rørsle, og denne rørsla forplantar seg vidare. Da vil luftmolekyla røre seg på ein slik måte at det blir danna område med større trykk og område med mindre trykk. Disse områda forflyttar seg, og fører til at lyden forplantar seg vidare. Etter kvart kan dei svingande molekyla sette luftmolekyla inni øyrene våre i rørsle, og da vil trommehinna og byrje å svinge på samme måten som skinnen på ei tromme. Denne svinginga forplantar seg til dei små beina i det indre øyret, og det blir sendt signal til hjernen om desse svingingane. Hjernen set om desse signala til lyd – enten det er fin musikk eller berre støy.



Bølgelengd

Avstanden mellom to bølgetoppar. På figuren har den øvste bølga dobbelt så stor bølgelengd som den nedste. Bølgelengd vert målt i meter.

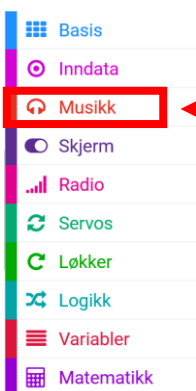
Frekvens

Talet på svingingar per sekund. Når frekvensen aukar, blir tonen lysare. Frekvens vert målt i hertz (Hz).

Amplitude

Avstanden frå likevektsstillinga (nullpunktet) til toppunktet slik det er merka på figuren. Den øvste bølga har dobbelt så stor amplitude som den nedste bølga. Volumet på lyden aukar når amplituden aukar. Amplituden vert målt i ulike einingar alt etter kva slags bølger vi ser på. For lydbølger blir eininga den same som for lufttrykk (Pascal eller Pa).

Musikk i MakeCode



For å kunne bruke musikklossane, må du først trykke på musikk i menyen.

Da får du tilgang til ei rekke klossar som du kan bruke for å komponere din eigen melodi og spele han av.

Denne klossen angjer tempoet (pulsen) på melodien din.

sett tempo til (bpm) 120

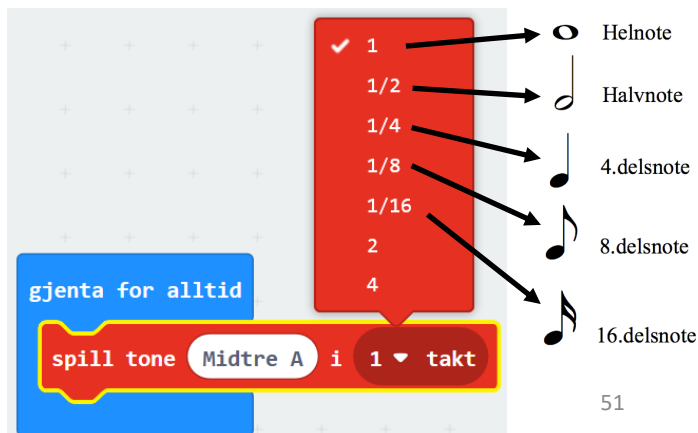
pause (ms) 1 takt

Denne klossen lagar ein pause i melodien din.

Denne klossen spelar av ein tone i så lang tid som du vel.

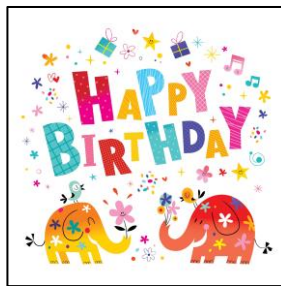
spill tone Midtre C i 1 takt

Vala som kjem opp for kor lenge tonen skal spele, svarer til dei ulike notetypane som vist under.



Lag eit musikkort

- som speler av ein eigenkomponert melodi



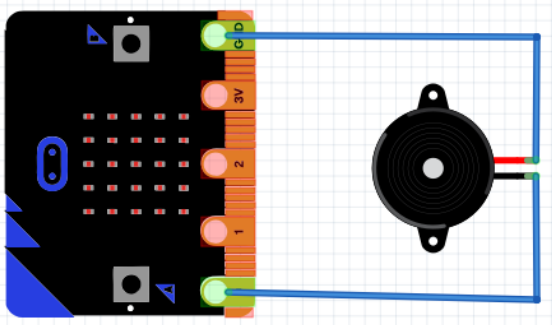
Oppgåve

Lag eit kort som speler ein eigenkomponert melodi når det blir opna.

Fase 1: Finn informasjon og inspirasjon til eit musikkort. Kva slags kort skal det vere? Morsdags- eller farsdagskort? Julekort? Gratulerer-med-dagen-til-gullfisken-din-kort?

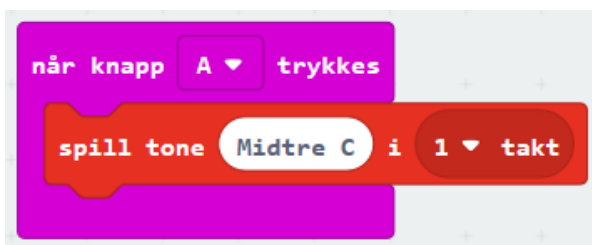
Fase 2: Ha ei idèmyldring for deg sjølv. Korleis vil du at kortet skal sjå ut? Teikne gjerne ei skisse før du tek til å lage det. Kva for nokre materiale skal du bruke? Kortet skal spele av ein melodi, korleis skal melodien din vere?

Fase 3: Tid for å lage kortet og programmere micro:biten slik at han speler ein melodi når du trykker på knappen.



Kople opp micro:bit som vist på figuren.

Korleis må vi koble om vi vil få tilkopla LED-pærer til å lyse?



Dette programmet får høgtalaren til å spele C i én takt.

Korleis kan du få han til å spele ein heil melodi?



Fase 4: Test programmet ditt.

Fase 5: Verkar det slik det skal? Sjekk gjerne om det er nokon andre i klassen som har spennande idear dei har brukt til kortet sitt. Kan du bruke noko av dette til kortet ditt?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få eit best mogleg kort. Gjer gjerne endringar i micro:bit-programmet ditt.

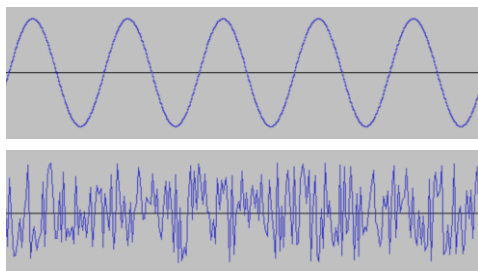
Fase 7: Forklar for ein annan i klassen korleis du gjekk fram for å lage kortet ditt.

Ekstraoppgåve

Kunne du fått kortet ditt til å spele ein kjend melodi?

1. Søk gjerne opp noteark med melodien på internett.
2. Lag programmet til micro:biten.
3. Sjekk om dei andre i klassen klarer å gjenkjenne melodien.ⁱⁱ⁵²

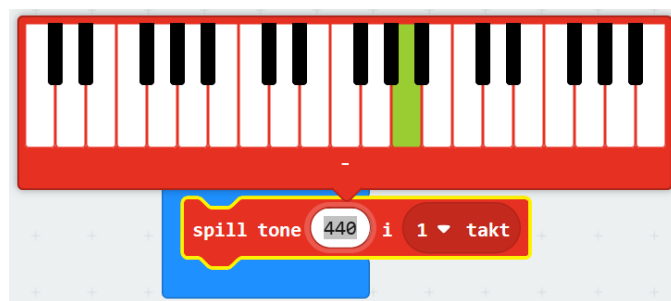
Opplegg 11 - Tonar og frekvens



Skilnaden på ein rein tone og støy eller annan lyd, er at tonen har fast bølgelengd og frekvens. Da vil lufttrykket variere på same måten heile tida. Da kan vi få ein graf som viser ei fin bølge når vi plottar lufttrykk som ein funksjon av tida.

For støy vil lufttrykket variere på ein kaotisk måte, og vi får ingen fin bølgegraf. Dersom vi plottar lufttrykk som ein funksjon av tida, vil han likne den nedste grafen til venstre.

I MakeCode kan du velge toner ut frå namnet på tonane, frekvensen deira, eller frå tangentene på eit slags piano. Du kan velge mellom tre ulike oktavar (frekvensintervaller) der dei har kalla de mørkaste tonane for låg, og de lysaste tonane for høg. I musikkteorien blir gjerne desse mørke variantene kalla for liten oktav, dei midtre tonane for einstrokne og dei høge tonane for tostrokne.

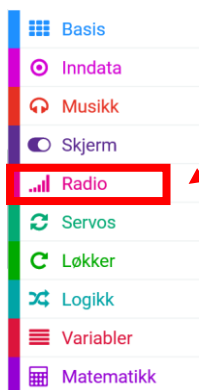


Sammenheng mellom tone og frekvens til lydbølgen					
Tone - lav	Frekvens (Hz)	Tone - midtre	Frekvens (Hz)	Tone - høy	Frekvens (Hz)
C	131	C	262	C	523
D	147	D	294	D	587
E	165	E	330	E	659
F	175	F	349	F	698
G	196	G	392	G	784
A	220	A	440	A	880
H	247	H	494	H	988

Snakk om

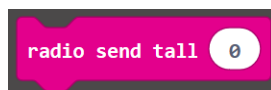
Kan de sjå nokre mønster for frekvensen til dei ulike tonane i tabellen?

Radio i MakeCode

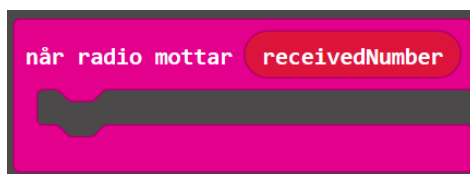


For å kunne nytte radioklossane, må du først trykke på radio i menyen.

Da får du tilgang til ei rekke klossar som du kan nytte for å sende informasjon frå ein micro:bit til ein annan micro:bit.



Denne klossen sender eit fastsett tal til ein annan micro:bit som er i same radiogruppe som den micro:biten som sender talet. Det går an å sende ein variabel som ein set inn i denne blokka. Det finst andre klossar for å sende tekstar og.



Denne klossen mottok eit tal frå ein annan micro:bit som er i same radiogruppe som han sjølv.



Denne klossen avgjer kva for ein «kanal» radioen bruker. Micro:bitane i same radiogruppe bruker same frekvens for å kommunisere.

Dette er ein type viss-kloss, og det som står inni klossen blir gjort når han mottok eit tal.

Radiogruppa må vere lik på begge micro:bitane for at dei skal kunne «snakke saman». Pass på at det ikkje er fleire grupper i klassen som bruker same kanal, for da vil alle micro:bitene «snakke i munnen på kvarandre», og det kan bli litt kaotisk. Verdien for radiogruppa kan vere mellom 0 og 255.

Lag ei ringeklokke

- med eit radiostyrt micro:bit-program



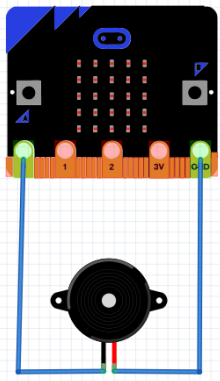
Oppgåve

Lag ei ringeklokke som speler en eigenkomponert melodi minst 5 meter borte når nokon trykker på henne.

Fase 1: Finn informasjon og inspirasjon til ei ringeklokke. Korleis kan ringeklokker sjå ut?

Fase 2: Ha ei idèmyldring for deg sjølv. Korleis vil du at ringeklokka di skal sjå ut? Teikne gjerne ei skisse før du tek til å lage ringeklokka. Kva for nokre materiale skal du bruke? Ringeklokka skal spele av ein melodi, kva for ein melodi vil du ha?

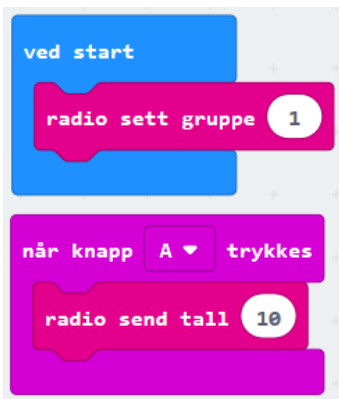
Fase 3: Tid for å lage kortet og programmere micro:biten slik at LED-pærene lyser.



Kople opp micro:bit som vist på figuren.

Korleis må vi kople om vi vil bruke fleire høgtalarar samstundes?

Kva for nokre endringar må vi gjere dersom vi ønsker å også få LED-pærer til å lyse?



Desse programma får høgtalaren som er kopla til den eine micro:biten til å spele C i éin takt når ein trykker på knapp A på den andre micro:biten.

Korleis kan du få han til å spele ein heil melodi?

Fase 4: Test programmet ditt.

Fase 5: Verkar det slik det skal?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få ei best mogleg ringeklokke. Gjer gjerne endringar i micro:bit-programmet ditt.

Fase 7: Forklar for ein annan i klassen korleis du gjekk fram for å lage ringeklokka di.

Kopling B

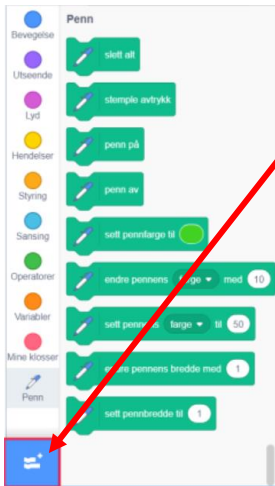
Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 6

Opplegg 12 – Teikne geometriske former med programmering



Det går an å nytte programmering for å teikne geometriske figurar. I Scratch treng vi å bruke nokon teiknefunksjonar for å få til dette. Dei må importerast frå ein tilleggskategori, og framgangsmåten blir synt under.



1. For å finne tillegg i Scratch, må du trykke på den blå knappen nedst i det venstre hjørnet.

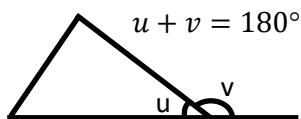
2. Da kjem du inn i ein meny der ein kan velje mellom mange ulike tillegg. Trykk så på ruta der det står penn.



3. Da blir det lagt til ein meny med ulike framgangsmåtar for å teikne med Scratch.

Teikning av geometriske former

Det er sær s éin ting ein må tenke på når ein skal teikne geometriske former med Scratch, og det er vinklar. Alle vinklane i Scratch er dei ytre vinklane, det vil seie supplementvinklane til dei indre vinklene. To vinklar som til saman danner 180° er supplementvinklar til kvarandre. I trekanten under er u og v supplementvinklar til kvarandre, og i Scratch er det vinkel v du skal bruke.



Diskuter

1. Kva for ein geometrisk figur blir teikna med de to programma til høgre?
2. Kva er poenget med den siste delen av koden heilt til høgre – må han alltid vere med?
3. Kva tyder tala i dei to programma? Blir det same figurane dersom nokre av tala blir endra?
4. Korleis ville du teikna ein sirkel med Scratch?



Teikning av hus til fargeleggingsbok

- Finn arealet av samansette figurar

Oppgåve

Lag eit program som teikner eit hus til ei fargeleggingsbok. Bruk minst tre ulike geometriske figurar i teikninga di. Ta skjermbilete av huset ditt og skriv det ut på papir slik at det blir ei side i ei matematisk fargeleggingsbok.

Fase 1: Kva for nokre geometriske former finnest? Sjekk at du veit om minst fem ulike, og kva for nokre kjenneteikn dei har. Kan du finne formlane for arealet for formene?

Fase 2: Det er viktig at de er åpne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

Når du har klar ein kladd over korleis huseikninga di skal vere, tek planlegginga til.

1. Del oppgåva opp i fleire små delar
 - Kva for nokre geometriske figurar er teikninga di samansett av?
 - Veit du korleis du må programmere for å teikne dei ulike delane?
 - Dersom du ikkje veit, kan du finne det ut?
 - Dersom du ikkje kan finne det ut, skal du endre teikninga di litt?
 - Korleis skal du kombinere dei ulike delane?
2. Planlegg gjennomføringa
 - Berekne kor lang tid kvar del tek.
 - Har du tid til å teikne alle delane?
 - Dersom ikkje, kva skal du kutte ut?
 - Kva for nokre former *må* vere med?

Fase 3: Lag første versjon av programmet ditt.

Fase 4: Test programmet ditt, og utforsk kva som skjer om du varierer kva for nokre tal du nyttar i dei ulike versjonane av programmet ditt.

Fase 5: Verkar det slik det skal? Blei teikninga av huset slik du hadde planlagt?

Fase 6: Hopp gjerne attende til tidlegare punkt, og gjer endringar for å få ei best mogleg huseikning. Gjer gjerne endringar i programmet ditt.

Fase 7: Skriv ut biletet ditt slik at du kan bruke det i ei matematisk fargeleggingsbok.



Oppgåve

1. Finn arealet av golvflata i huset ditt.
2. Finn overflatearealet av huset ditt. Bruk formlane for areal av dei ulike geometriske figurane.
3. Finn volumet av huset ditt. Må du gjere nokon gissingar?

Diskuter

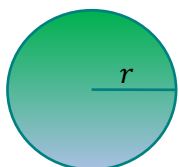
1. For at huset skal ha dobbelt så stort golvareal, kva blir dei nye måla?
2. For at huset skal få dobbelt så stort volum, kva blir dei nye måla?
3. Kva blir svara på spørsmål ein og to, dersom husa må vere formlike?

Opplegg 13 – Sirklar, kuler og naturressursar

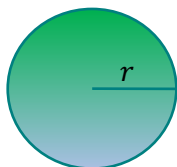


Under blir det vist figurar av to sirklar med same radius, og to kuler med same radius. Ein sirkel er to-dimensjonal, medan ei kule er tre-dimensjonal. Det tyder at vi ikkje kan rekne ut volumet av ein sirkel, for han er heilt flat, han har inga høgd.

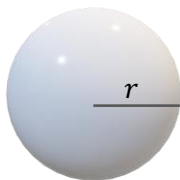
$$O = 2\pi r$$



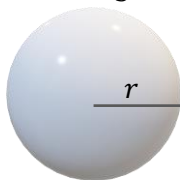
$$A = \pi \cdot r^2$$



$$A = 4\pi r^2$$



$$V = \frac{4\pi r^3}{3}$$



Diskusjonsoppgåve

1. Kan vi samanlikne arealet av ein sirkel med overflatearealet til ei kule? Korleis er denne samanhengen?
2. Dersom du har en sirkel med radius r . Kva blir formlane for omkrinsen og arealet? Kva skjer med formlane dersom radiusen doblast, altså at han er lik $2r$?
3. Kva skjer ved ei dobling av radius for volumet av ei kule? HINT – sjå på formel for volum.
4. Samanlikne kor mykje omkrins og areal av sirkel og volum av ei kule aukar når radiusen blir dobbelt så stor. Kan de seie noko om kvifor dei aukar på akkurat den måten?

Naturressursar

Jordkloden har mange naturressursar som menneske har bruk for på ulike måtar. Det kan til dømes vere metallressursar, drikkevatt eller dyrkande areal. Eller oljeressursar som kan nyttast til å lage plast eller bensin. Problemet er om menneske nyttar for mykje av desse ressursane, slik at dei blir oppbrukte, eller at bruken av ressursane skapar problem for naturen og menneske. Dette kan mellom anna føre til at dyre- og planteartar blir truga og døyr ut, da får vi eit tap av biologisk mangfald.

Diskuter

1. Kva er mest miljøvenleg av eit bomullsnett, eit plastbasert handlenett eller ein plastpose?
2. Kva meiner du mest miljøvenleg? Sjå på nokon av framlegga i lista og argumenter for kva for nokre perspektiv du meiner er viktigast. Tenk gjerne ut fleire perspektiv.
 - CO₂-utslepp
 - Dyrkinga/produksjonen påverkar dyre- og planteartar
 - Utslepp i naturen (forureining)
 - Plast i havet
 - Bruk av dyrkingsareal (fortrenger dyrking av mat)
 - Vassforbruk
 - Moglegheit for gjenbruk
3. Påverkar val av perspektiva i spørsmål 2 svaret på spørsmål 1? Kva bør vere med i vurderinga? Fint det noko fasitsvar?



Teikning av sirklar til handlenett

- Måling av diameter og omkrins

Oppgåve

Teikn minst 5 ulike sirklar med hjelp av programmering, skriv dei ut, og mål diameter og omkrins. Bruk sirklane som mønster til dekorasjon på eit handlenett. Finn ein matematisk modell for samanhengen mellom diameter og omkrins til sirklane.

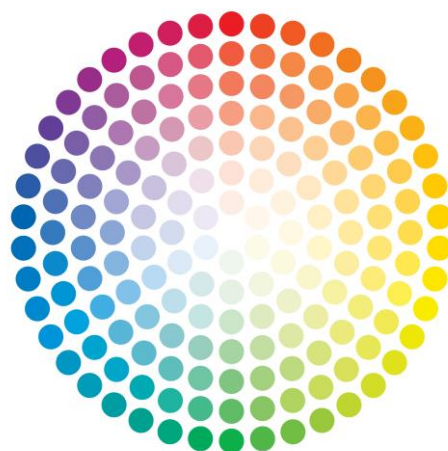
Fase 1: Er det nokon idear frå oppgåva du er usikker på, prøv å finne ei forklaring.

Fase 2: I denne oppgåva er det logisk å ta til med fase 2 i modellen, sidan vi ikkje treng å finne ut så mykje om sirklar på førehand.

Fase 3: Tid for å lage sirklane, skrive dei ut, måle dei og bruke som mønster til stoffsirklar til handlenettet.

Del oppgåva opp i fleire små delar

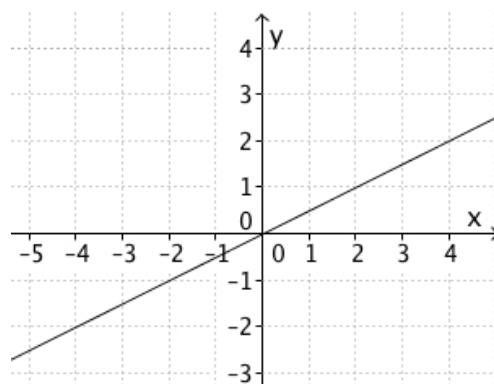
- Korleis skal du lage minst 5 ulike sirklar?
- Korleis bruke programmering?
- Har du nokre liknande program frå før?
- Korleis skal du måle diameter og omkrins?
- Treng du noko utstyr for å gjennomføre målingane?



Gjer fase 4 – 7.

Oppgåver

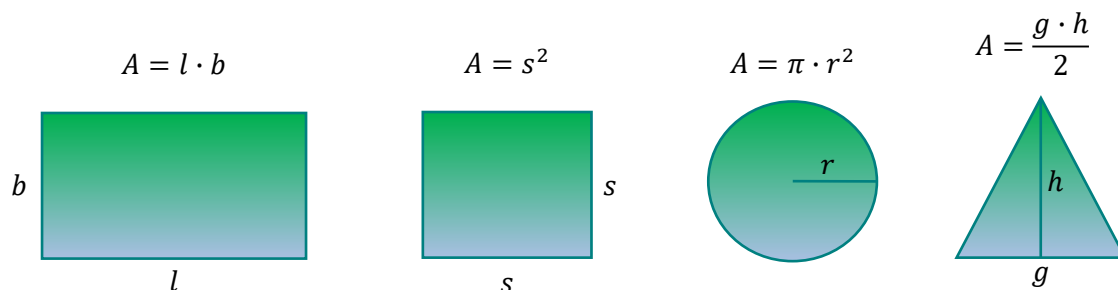
1. Mål diameter og omkrins for dei ulike sirklane du teikna med hjelp av programmering.
2. Plott alle dei målte verdiane i Geogebra.
3. Finn ein matematisk modell, ved å gjere ein regresjon for dei målte dataa.
4. Kva er hovudskilnaden mellom andre modellar de har laga tidlegare og denne modellen?



Opplegg 14 - Volum for ulike geometriske former

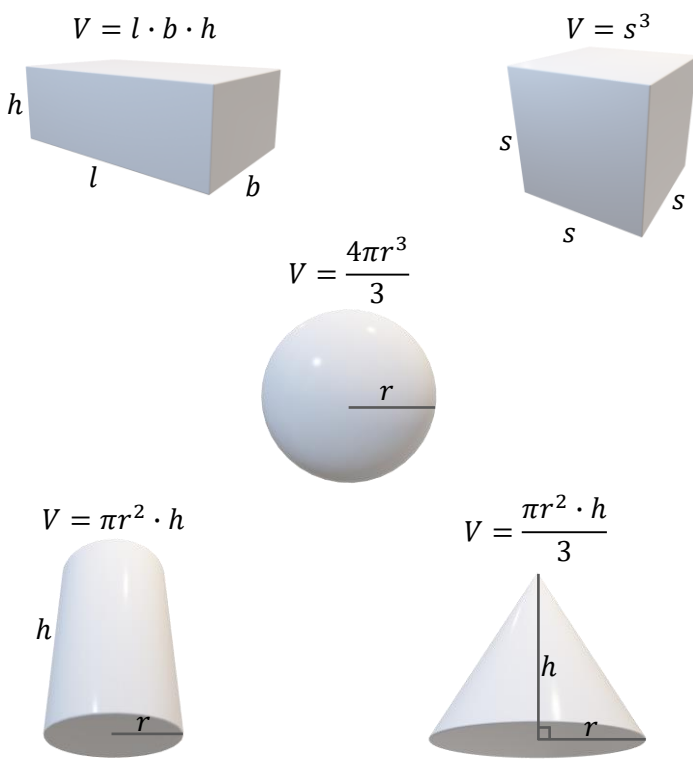
Lengd fortel oss kor stor avstanden er mellom to punkt. Lengda er eindimensjonal, og målest i til dømes einingane cm eller meter.

Areal gir oss informasjon om kor stor ei flate er, slik at vi kan samanlikne storleiken på grunnflata til ulike figurar, til dømes om ein skal kjøpe eit hus. Areal er todimensjonalt, det vil seie at det er samansett av to lengder. Men som regel kallar vi det lengd og breidd, eller grunnlinje og høgd. Desse to avstandane må multipliserast for å få eit areal, dermed blir eininga for areal alltid ei lengd opphøgd i andre, for eksempel cm^2 eller m^2 .



På same måten gir volum informasjon om kor stor plass noko tek, eller kor stor plass det er inni noko. Volum er ein 3-dimensjonal storleik, og vi får difor ikkje teikna han skikkeleg på ei 2-dimensjonal flate slik som i ei bok.

For å rekne ut volum må vi alltid multiplisere tre lengder med kvarandre. Sjekk gjerne med formlane under! Da blir eininga for volum alltid ei lengd opphøgd i tredje, slik som cm^3 eller m^3 . Alle fysiske gjenstandar har tre dimensjonar, men nokon eller alle lengdene kan vere veldig korte slik at volumet blir veldig lite. Dette gjeld til dømes eit hårstrå eller eit ark silkepapir eller bladgull.



Diskuter

1. Har nokon av 3D-figurane noko til felles?
2. Korleis vil de dele figurane inn i grupper? Er det nokon figurar som passar i fleire grupper?
3. Kan de sjå noko felles for volumformlane for dei ulike gruppene?

Utforsk volum med ein volummålar

Oppgåve

Lag ei geometrisk form i minst 5 forskjellige storleikar. Dei skal nyttast i ein målesylinder, for å måle volumet på dei. Så skal de bruke målinga for volum til å finne formelen for volum med hjelp av regresjon i geogebra. Til slutt skal de lage eit program som de kan bruke til å utforske kva som skjer med volumet til figuren dersom du doblar sidelengda, radius eller høgd.

Fase 1: Finn kva for nokre former det kan vere lurt å lage? HINT: Det er ei stor føremon med berre éin variabel. Spør gjerne lærar om tips.

Fase 2: Ha ei idémyldring for deg sjølv. Teikne gjerne ei skisse før du diskuterer med dei andre.

Fase 3: Tid for å lage figurane, og måle volum og den eine variabelen (sidelengd, radius eller høgd). Regresjonen og programmeringen må vi gjere til slutt, for vi må ha alle målingane klare først.

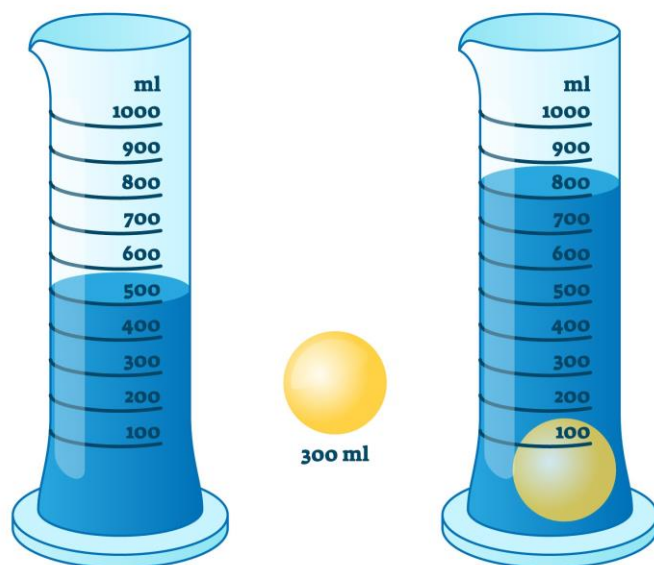
Gjer fase 4 – 7.

Regresjon – vise volumformlar

1. Mål volum og sidelengd, radius eller høgd.
2. Plott alle dei målte verdiane i Geogebra.
3. Finn ein matematisk modell med å gjere ein regresjon for dei målte data.
 - Kva form trur du grafen vil ha?
4. Kva er hovudskilnaden mellom andre modellar de har laga og denne modellen?

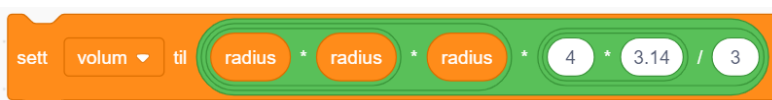
Programmering

- Bruk volumformelen de kom fram til i regresjonen, og sjekk om han stemmer med biletet på førre side.
- Lag eit program der de skriv inn kor stor variabelen skal vere (sidelengd, radius eller høgd) og som reknar ut kor stort volumet blir i det høvet.
- Kva meir kan du utforske med å forandre litt på programmet?



Programmeringstips

For å rekne ut volumet til ei kule vil de trenge ein kloss som er samansett på denne måten.



Opplegg 15 - Matematiske mønster

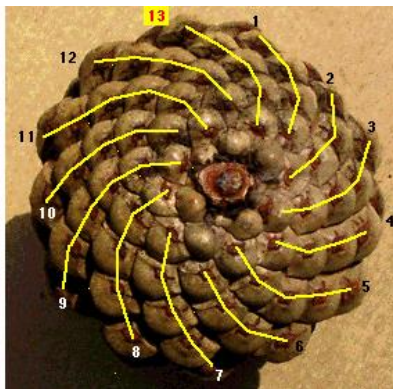
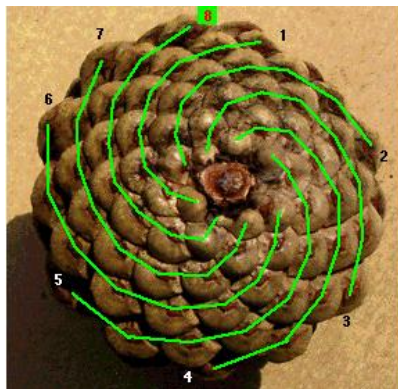
Det finst mange matematiske mønster i naturen. Nokon er baserte på ei talrekke som vert kalla Fibonacci-tala. Denne rekka er laga av tal der kvart tal er lik summen av dei to føregåande tala i rekka.

Fibonacci rekka: 0 – 1 – 1 – 2 – 3 – 5 – 8 – 13 – 21 – 34



Fibonacci-talla i naturen

Dersom ein ser på spiralane i ein konge eller ananas, kan ein sjå at dei liknar spiralane i solsikkefrøa eller romanesco-kålen. Talet på spiralar i kvar retning er to påfølgande Fibonacci-tal. Med å sjå på frøkapselstrukturen til ein mogen hokonge, kan ein telje seg fram til 8 spiralar i den eine retninga (merka med grønt) og 13 spiralar i den andre retninga (merka med gult).



Illustrasjon: John R. Simmons, University of Georgia

Diskuter

1. Kva trur du dei tre neste talla i rekka blir?
2. Lag minst to eigne talrekker, og la ein annan i klassen finne det neste talet.
3. Kan du finne ut kva trekanttalet er?

Geometriske mønster

Vi kan og lage mønster med hjelp av geometriske figurar, og det passar veldig fint å gjere med hjelp av programmering!

Ofte kan vi få mønster med mange symmetrilinjer med å teikne ein geometrisk figur, og så forskyve han med til dømes 20 grader, og ta dette oppatt heilt til figuren har blitt dreid 360 grader. Da vil vi få ein figur som kan likne litt på en blome.

Oppgåver

1. Kva er poenget med koden til høgre i Scratch-programmet på denne sida?
2. Kor mange symmetrilinjer har «blomen» teikna av dette Scratch-programmet?
3. Kva er det i programmet som avgjer kor mange symmetrilinjer vi har?
4. Korleis ville ein trekantblome sjå ut? Lag programmet og sjekk.

Dette programmet tegner denne «blomen».

Lag eit putetrekk

- med eit matematisk mønster



Oppgåve

Lag eit program som teiknar eit matematisk mønster. Dette mønsteret skal overførast til eit putetrekk med bruk av brodering/symaskin, varmpressa vinyl, pålimt tråd, stoffmåling eller andre måtar.

Fase 1: Undersøk ulike matematiske mønster. Det kan vere geometriske mønster, talmønster eller andre typar mønster. Kva blir ulike typar matematiske mønster kalla? Kva har dei til felles?

Fase 2: Det er viktig at de er opne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

Tips: Kva for eit motiv kan få folk til å meine at matematiske mønster er spennande? Kan du gjere det enklare slik at du klarer å teikne det med Scratch?

- Tenk sjølv først og teikne gjerne fleire skisser.
- Forklar ideen din for dei andre på gruppa.
- Heile gruppa diskuterer dei ulike ideane, og alle avgjer kva for eit matematisk mønster dei vil nytte.

Tenk gjennom korleis du skal få mønsteret ført over til eit putetrekk. Vil du helst brodere mønsteret for hand, sy på symaskin, lime det på eller noko heilt anna? Diskuter gjerne med dei andre kva dei har tenkt.

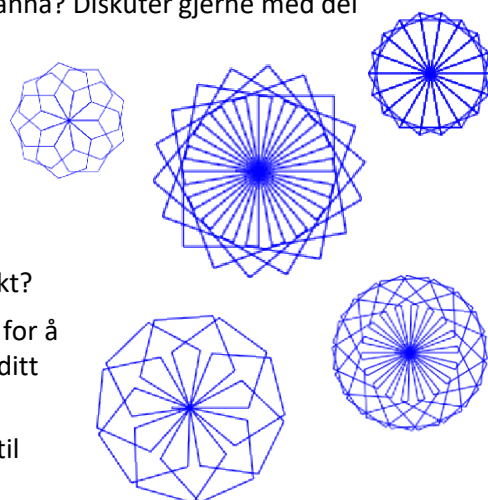
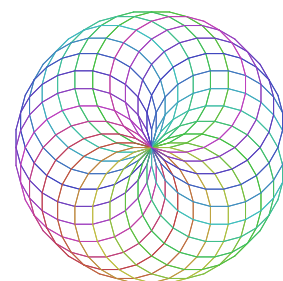
Fase 3: Lag første versjon av programmet ditt.

Fase 4: Test programmet ditt, og utforsk kva som hender om du varierer kva for nokre tal du bruker i dei ulike versjonane av programmet ditt.

Fase 5: Verkar det slik det skal? Blei mønsteret slik du hadde tenkt?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få eit best mogleg mønster. Gjer gjerne endringar i programmet ditt og/eller planen din for å føre mønsteret over til putetrekket.

Fase 7: Skriv ut biletet ditt slik at du kan bruke det som mønster til putetrekket ditt.



Oppgåver

1. Samanlikne mønsteret ditt med mønstra til dei andre i klassen. Har nokon av dykk formlike eller kongruente mønster?
2. Kan du programmere nokre formlike versjonar av ditt eige mønster?
3. Kva er likt, og kva er ulikt i programma for dei ymse formlike mønstra?

Ekstraoppgåve

1. Kva blir arealet av figuren din? Kan du justere storleiken slik at figuren din dekker eit område på 200 cm^2 ?
2. Må du gjere nokon gissingar for å klare det? Kva for nokre?

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 7

Forsøk, simuleringar og sannsyn.

Korleis finn vi sannsynet for at noko spesifikt skal hende? Dersom vi vil finne sannsynet for å få ein seksar på eit terningkast, kan vi kaste ein terning fleire gonger og telle opp kor mange gonger vi får seksar. Dersom du kastar terningen 50 gonger, og du får seksar 11 av gongene, kan vi finne sannsynet for å få seksar med å dele talet seksarar på så mange gonger du kasta terningen.

$$P(\text{seksar}) = \frac{11}{50} \approx 0,22$$

Da har du gjort eit forsøk, og funne ut at sannsynet er 0,22 eller 22 % for å få seksar. Men dette høyrer vel litt rart? Burde det ikkje vere like stor sjanse for å få seksar som einar eller firar? Dersom det er like stor sjanse for å få dei ulike verdiane på terningen, burde kvar av dei seks sjansane ha sannsyn på $\frac{1}{6} \approx 0,17 = 17\%$.

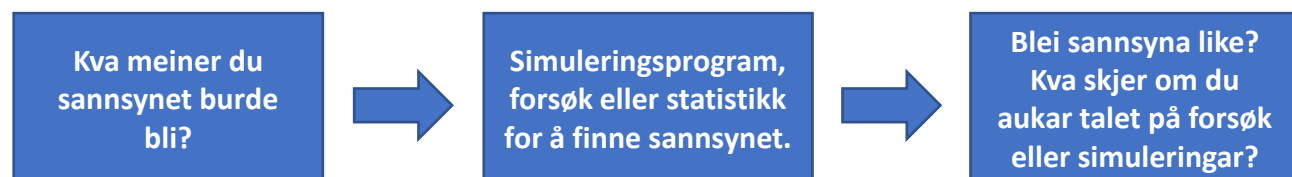
$$0,17 \neq 0,22$$

Kvifor blir det ikkje likt når vi gjer forsøket i røynda? Er terningen vår ein juksesterning? Nei, det er slik at dersom vi gjer eit forsøk i røynda, kan vi få litt rare svar om vi ikkje gjer forsøket så mange gonger. Dette kjem av at terningen ikkje veit kva han landa på siste gong, og det er difor heilt tilfeldig kva han hamnar på neste gong. Vi seier at dei to kasta er uavhengige av kvarandre.

Dersom du kastar terningen 100 gonger i staden, vil sannsynet for å få seksar bli nærmare 0,17. Men du må kan hende kaste nesten 10000 gonger før det blir nøyaktig nok. Det blir nok litt kjedelig!

Da kan ein bruke programmering i staden! Når ein lagar eit program som «lèt som» om det gjer eit forsøk, så har vi laga ei simulering. Denne simuleringa (programmet vi laga) kan vi enkelt bruke for å simulere terningkast, og sjekke at det simulerte sannsynet blir likare og likare den verdien det burde ha, jo fleire simuleringar vi gjer.

Det finst altså hovudsakleg tre måtar for å finne kva eit sannsyn er. Vi kan tenke logisk, og så gjere eit forsøk (eller finne statistikk) eller ei simulering for å sjekke om det stemmer. Ofte er det upraktisk med forsøk, det tek gjerne for lang tid, da kan ein nytte simulering i staden.



Utfall-

det som kan skje når vi gjer eit statistisk forsøk. Det å kaste ein einar på terningen er eit utfall, og det å kaste en firar på terningen er eit anna utfall. Sannsynet for alle moglege utfall vil alltid til saman bli 1.

Korleis rekne ut sannsyn

Vi skal kaste ein terning og finne sannsynet for å få einar eller toar. Først må vi telle opp kor mange utfall som er gunstige for resultatet vårt, altså kor mange utfall som gir oss einar eller toar. Det blir 2 utfall (einar er eitt utfall og toar er eitt til utfall, to til saman). Så må vi finne kor mange moglege utfall vi har, altså kor mange ulike sider vi kan treffe på når vi kastar ein terning. Det er seks moglege utfall, sidan ein terning har seks sider. For å finne sannsynet må vi dividere gunstige utfall på moglege utfall:

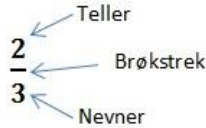
$$P(\text{einar eller toar}) = \frac{\text{gunstige utfall}}{\text{mulige utfall}} = \frac{2}{6} = \frac{1}{3}$$

Opplegg 16 - Sannsyn og representasjonar

Sannsyn kan skrivast som brøk, desimaltal eller prosent, og slik er samanhengen mellom dei:

Brøk

Ein brøk består av ein teljar, ein brøkestrek og ein nemnar. Brøkestreken kan vi sjå på som eit deleteikn.



Desimaltal

Sidan brøkestreken kan sjåast på som eit deleteikn, kan vi enkelt rekne ut desimaltalet frå brøken. Vi deler teljaren på nemnaren og får då desimaltalet: $\frac{2}{3} = 2 : 3 = 0,67$

Prosent – tyder hundredel

Sidan desimaltalet viser kor mange einarar vi har, kan vi multiplisere med 100. Da får vi prosent, altså kor mange hundredelar vi har: $0,67 \cdot 100 \% = 67 \%$

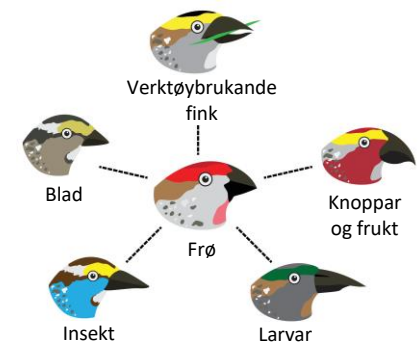
Evolusjonsteorien

I 1836 kom Charles Darwin heim frå ei segling jorda rundt med skipet Beagle. På turen hadde dei vitja Galapagosøyane, ei øygruppe som høyrer til Ecuador i Sør-Amerika. Darwin observerte dyrelivet på øyane og la merke til at finkane på ulike øyar hadde ulike nebb. Dette kom til å bli ein viktig observasjon i arbeidet hans med evolusjonsteorien.

Grunnen til at finkane hadde ulike nebb, var at dei ulike øyane hadde tilgang til ulik mat for finkane. Finkane hadde litt avvikande nebb når dei blei klekka. Dei finkane som hadde nebb som gjorde det enkelt å ete maten det var størst tilgang til, ville og overleve lengst når det blei konkurranse om maten.

Dei finkane som overlevde lengst, var og dei som fikk flest etterkomarar. Over tid ville denne gradvise endringa føre til at det ville leve ulike typar finkar på ulike øyar. Sidan øyane hadde tilgang på ulike typar mat, ville det berre vere den finketypen med nebb tilpassa akkurat den tilgjengelege maten, som levde på kvar øy.

Det at den arten eller varianten av ein art som er best tilpassa omgivnadane sine (skaffe seg mat, ly, få mange avkom o.l.), vil bli dominerande, er ein viktig del av evolusjonsteorien. Ein art vil på det viset utvikle seg slik at han er tilpassa omgivnadane sine. Sidan dette kan gjerast på utruleg mange ulike måtar, har vi stor variasjon mellom dei ulike artane.



Snakk om

Kva har klimaendringar med evolusjonsteori å gjere?

Korleis kan ulike artar tilpasse seg klimaendringane?

Artskjenneteikn	Talet på daude per 1000 dyr/fuglar første leveår
Venger	150
Nattaktiv	100
Gode graveklør	80
Tjukk pels	75
Skarpe tenner	50
Lange øyre	250

Dette er fiktive tal, det finnest ikkje nokon ekte oversikt.

Sannsyn frå statistikk

I tabellen er det oversikt over dødstalet per 100 dyr eller fuglar for deira første leveår. Dermed kan vi bruke dei verdiane for å finne ut sannsynet for at dyret eller fuglen dør før eittårsdagen.

Vi vel å sjå på fuglar med venger, og der dør 150 fuglar av 1000.

Sannsynet for at fuglane med venger vil døy i det første leveåret sitt, blir da:

$$P(død) = \frac{150}{1000} = \frac{15}{100} = \frac{3}{20}$$

Lag ein klimatilpassa art

- lag eit program som reknar om brøk til desimaltal og prosent

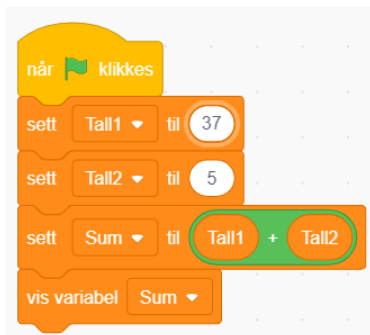
Oppgåve

Lag ein art som er betre tilpassa klimaendringar. Du skal rekne ut sannsynet for at han dør i det første leveåret med å bruke tabellen med artskjenneteikn på førre sida, oppgitt som ein brøk. Lag eit program som reknar om ein brøk til desimaltal og prosent, og skriv sannsynet som både desimaltal og prosent.

Fase 1: Veit du kva moglege konsekvensar av klimaendringar kan vere? Dersom ikkje, finn det ut. Finn informasjon om klimaendringar, og korleis det verkar på ulike stadar på jordkloden. Bruk dette som inspirasjon til ein art som er betre tilpassa klimaendringane.

Fase 2: Ha ei idémyldring for deg sjølv. Korleis vil du at arten din skal sjå ut? Teikne gjerne ei skisse før du diskuterer med dei andre. Deretter må gruppa avgjere korleis arten skal sjå ut. Hugs å ha med noko der de kan skrive opp sannsynet som de skal rekne ut.

Fase 3: Tid for å lage eit program som gjer om frå brøk til desimaltal og prosent, og å lage ein modell av den nye arten.



Kva trur du desse programma gjer?

Kva er skilnaden på dei?

Korleis kan de bruke noko av dette i programmet dykkar?

Lag omrekningsprogrammet.

Del gjerne oppgåva opp i fleire små delar

- Veit du korleis du må programmere for å rekne ut frå brøk til desimaltal?
- Veit du korleis du må programmere for å rekne ut til prosent?
- Dersom du ikkje veit, kan du finne det ut?
- Korleis skal du kombinere dei ulike delane?



Fase 4: Test programmet ditt. Be andre i klassen om tilbakemeldingar på modellen. Kan dei sjå kva det skal vere?

Fase 5: Verkar alt slik det skal? Er svaret frå programmet logisk?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få ein best mogleg klimatilpassa art. Gjer gjerne endringar i programmet ditt. Kan det gjerast meir effektivt eller ryddigare?

Fase 7: Skriv ned utrekninga for sannsynet, og vis det som både brøk, desimaltal og prosent. Forklar korleis gruppa di tenkte for ein person frå ei anna gruppe.

Oppgåve

Kva for nokre konsekvensar av klimaendringane er arten dykkar særst godt tilpassa? Forklar kvifor.

Fører denne tilpassinga til større overleving over alt trur du? Kvifor/kvifor ikkje?

Opplegg 17 - Uniformt sannsyn

Uniformt sannsyn

Alle utvegar har like stort sannsyn for å hende.

Døme på uniformt sannsyn kan vere å trekke eit kort frå ein kortstokk, eller å kaste mynt eller krone. Det er ingen grunn til at det skal vere meir sannsynleg å trekke akkurat hjerter tre enn kløver fem. Det er heller ingen grunn til at det skulle vere meir sannsynleg at ein mynt landar med krona opp, enn myntsida opp (den med hovudet). Vi ser bort fra det utruleg vesle sannsynet for at mynten landar på kanten.



Diskuter

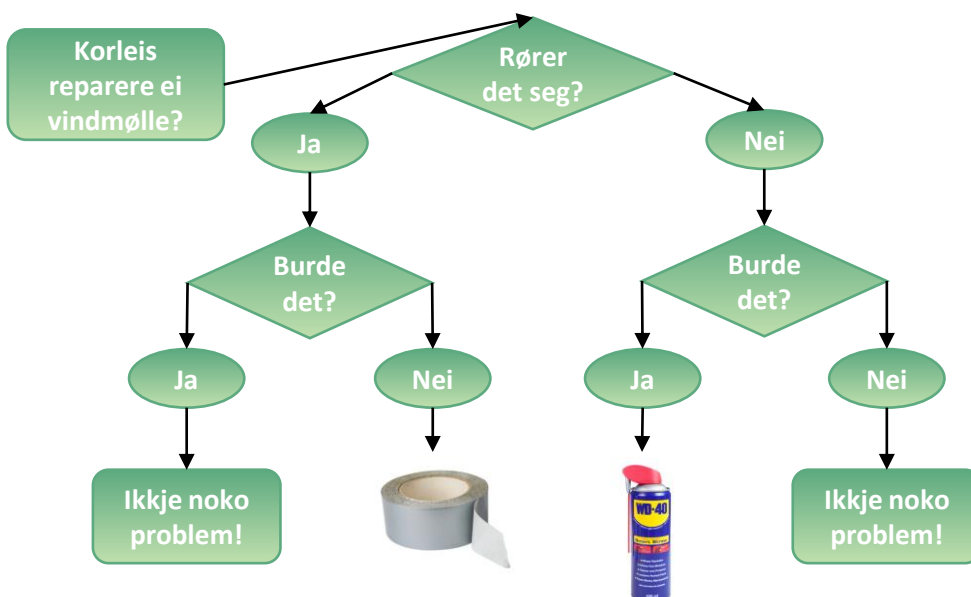
1. Kan du finne nokon andre eksempel på uniformt sannsyn?
2. Kan du finne nokre døme som ikkje er uniforme?
3. Korleis kan vi sjekke om døma våre er uniforme eller ikkje?

Pseudokode og flytskjema

I byrjinga av boka er det to oppgåver der de skal programmere kvarandre, utan å bruke eit programmeringsspråk. Da endte de opp med å skrive ei oppskrift, ein algoritme, med vanlege ord. Dette kan kallast pseudokode. Altså det å skrive eit program med det vanlege språket vårt, gjerne med stikkord, men ikkje med eit programmeringsspråk.

Når ein skal lage eit litt større program, kan det vere lurt å bruke pseudokode først. Da skriv du med eigne ord kva programmet skal gjere, og etterpå set du dette om til enten Scratch- eller Pythonkode.

Eit alternativ er å lage eit flytskjema først. Et flytskjema er eit slags oversyn over programmet. Ein nyttar ulike små figurar som tyder ulike ting. Ein form for spørsmål, ein form for svar, og ein form for byrjing og slutt. Du kan bruke både flytskjema og pseudokode når du skal lage eit program, men som regel er det greitt å velje det som passar best for deg.



Byrjing eller slutt

Svar

Spørsmål

Oppgåver

1. Kva viser dette flytskjemaet?
2. Lag eit flytskjema for korleis rydde rommet ditt (søk på «funny flowcharts» for inspirasjon).
3. Lag ein pseudokode for å rydde rommet ditt.



Lag ein terning eller eit lukkehjul

- og eit program som simulerer sannsynet



© Inter IKEA Systems B.V. 2020

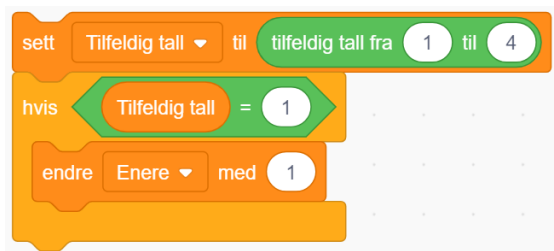
Oppgåve

Lag ein eigendesigna terning som ikkje har seks sider, eller eit lukkehjul som seinare skal vere ein del av eit spel. Du skal lage eit program som du kan bruke til å simulere terningkast eller lukkehjul-snurr og finne sannsynet for dei ulike utfalla.

Fase 1: Finn informasjon og inspirasjon til ein terning eller et lukkehjul. Kva for ei form kan ein terning ha? Kva for nokre delar er eit lukkehjul samansett av? Korleis ser desse delane ut?

Fase 2: Ha ei idemyldring for deg sjølv. Teikne gjerne ei skisse før du diskuterer med dei andre. Deretter må gruppa avgjere korleis terningen/lukkehjulet skal sjå ut.

Fase 3: Tid for å lage simuleringsprogrammet, og å lage terningen eller lukkehjulet. Sjå på programmeringsdøma for eit firedelt lukkehjul under:



Kva trur du desse programma gjer?
Korleis kan du bruke det i ditt
simuleringsprogram?

Lag eit simuleringsprogram – det finnest ark med HINT (spør lærar)

Del oppgåva opp i fleire små delar

- Veit du korleis du må programmere for å få eit tilfeldig terningkast eller lukkehjul-spinn?
- Veit du korleis du må programmere for å telje opp kor mange det blir av kvart utfall?
- Dersom du ikkje veit, kan du finne det ut?
- Kan du bruke noko frå programmeringsdøma over?
- Korleis skal du kombinere dei ulike delane?

Kan du lage eit
flytskjema her?



Fase 4: Test terningen eller lukkehjulet og programmet ditt.

Fase 5: Verkar alt slik det skal?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få terningen eller lukkehjulet best mogleg. Gjer gjerne endringar i modelleringsprogrammet ditt. Kan det gjerast meir effektivt eller ryddigare?

Fase 7: Lag eit ark med bilete frå heile prosessen saman med resultatet for sannsynet til dei ulike utfalla.

Opplegg 18 - Antall moglege kombinasjonar



Regel for sannsyn

Vi har alltid denne regelen for berekning av sannsyn:

$$P(\text{utfall}) = \frac{\text{gunstige utfall}}{\text{moglege utfall}}$$

Hending

Sett saman av eitt eller fleire utfall.

Gunstige utfall

Utfalla som er med i hendinga du skal finne sannsynet til.

Moglege utfall

Alle utfalla som er moglege å få.

For å bruke denne regelen må vi kunne finne ut kor mange gunstige utfall vi har, og kor mange moglege utfall vi har. Ein måte å gjere dette på, dersom ein ikkje har altfor mange utfall, er å teikne opp alle utfalla i ein tabell og telje opp kva for nokre som er gunstige.

Vi skal finne ut kor stort sannsyn det er for at summen av auge på terningane blir 9 når vi kastar to ulike terningar. Ein av terningane er vanleg og har seks sider, medan den andre har åtte sider. Vi lagar ein tabell (som ein krysstabell for gener, men med fleire utfall):

+	1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8	9
2	3	4	5	6	7	8	9	10
3	4	5	6	7	8	9	10	11
4	5	6	7	8	9	10	11	12
5	6	7	8	9	10	11	12	13
6	7	8	9	10	11	12	13	14

For å finne kor mange gunstige utfall vi har, må vi telje alle utfalla der vi får at summen blir 9. Det blir 6 utfall.

For å finne kor mange moglege utfall vi har, må vi telje alle utfalla i tabellen vår. Eller må vi det? Dersom vi skal finne ut kor mange ruter det er i tabellen vår, kan vi vel berre multiplisere antall utfall for kvar av terningane?

Den eine terningen har 6 sider, og dermed 6 moglege utfall. Den andre terningen har 8 sider, og dermed 8 moglege utfall. For kvar av dei 6 sidene på den eine terningen, kan vi kombinere henne med kvar av de 8 sidene på den andre terningen. Altså blir det 8 moglege summer for kvart utfall på den 6-sida terningen. Dermed kan vi multiplisere 8 med 6 for å finne alle dei moglege utfalla: $6 \cdot 8 = 48$. Da kan vi rekne ut sannsynet for at summen på terningane blir 9:

$$P(\text{summen er 9}) = \frac{\text{gunstige utfall}}{\text{moglege utfall}} = \frac{6}{48} = \frac{1}{8}$$

Regel for talet på moglege utfall

For å berekne talet på moglege utfall for ei hending, kan vi multiplisere talet på moglege hendingar for kvart av utfalla hendinga er samansett av.



Oppgåver

1. Finn ut kor stort sannsyn det er for å trekke eit biletekort (inkludert ess) frå ein vanleg kortstokk. Treng du teikne ein tabell?
2. Finn ut kor stort sannsyn det er for at produktet av augene på to vanlege terningar blir 14. Treng du teikne ein tabell?
3. Kva gjorde at du trong teikne tabell på ei av oppgåvene, men ikkje den andre? Kan du lage ein regel for kva tid det er lurt å tegne ein tabell?

Lag ei magisk krystallkule

- og tilpass programmet som simulerer sannsynet

Oppgåve

Bruk tilgjengeleg materiale til å lage ei «krystallkule» med plass til micro:biten inni. Det skal nyttast eit program som tilfeldig vel mellom minst 8 spådommar. Du skal og lage eit program som du kan bruke til å simulere trekning av spådommar og finne sannsynet for dei ulike utfalla.

Fase 1: Finn inspirasjon til spådommane og korleis «krystallkula» skal sjå ut. Kva for slags materiale skal du bruke? Kor mange spådommar skal du lage?

Fase 2: Ha ei idemyldring for deg sjølv. Korleis skal micro:bit-programmet ditt vise kva for ein spådom som blir tilfeldig trekt? Teikne gjerne ei skisse før du diskuterer med dei andre. Deretter må gruppa avgjere korleis krystallkula skal sjå ut.

Fase 3: Tid for å lage «krystallkula» og spådommane, i tillegg til programma:

- Lage eit program til micro:biten som vel tilfeldig mellom spådommane dine, og viser resultatet.
- Endre det tidlegare simuleringsprogrammet ditt (frå opplegg 17) for å bruke det til å simulere trekning av spådommar, og utrekning av sannsyn for dei ulike utfalla.



Kva trur du dette programmet gjer?

Korleis kan du bruke noko av det i ditt program på micro:biten?



Fase 4: Test krystallkula og programma dine.

Fase 5: Verkar alt slik det skal?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få ei best mogleg krystallkule og micro:bit-program. Gjer gjerne endringar i simuleringsprogrammet ditt. Kan det bli gjort meir effektivt eller ryddigare?

Fase 7: Dokumenter heile prosessen saman med resultatet for sannsynet til dei ulike utfalla på ein måte du vel sjølv.

Opplegg 19 - Sannsyn og valtre



Regel for multiplikasjon av sannsyn

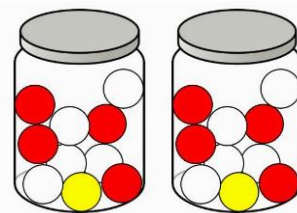
Dersom vi har ei hending som er samansett av utfall som er uavhengige av kvarandre, kan vi finne sannsynet for kvart av utfalla, og multiplisere dei saman for å finne sannsynet til den samansette hendinga:

$$P(\text{hendelse}) = P(\text{utfall 1}) \cdot P(\text{utfall 2})$$

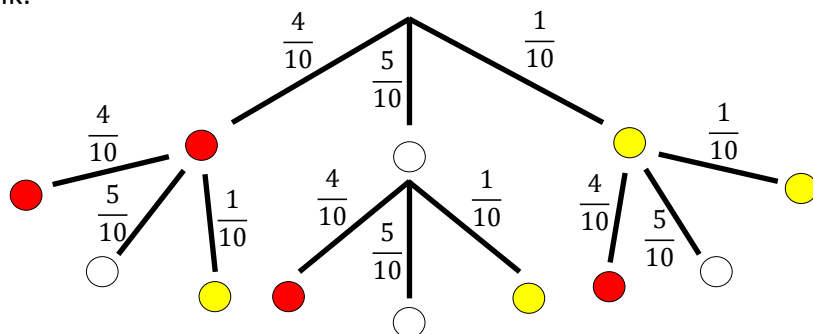
Uavhengige utfall

Eit utfall påverkar ikkje eit anna utfall. Døme er å trille ein terning fleire gonger eller å slå mynt eller krone. Det gjeld og når ein trekker fleire kort frå ein kortstokk, dersom ein legg korta attende mellom kvar gong ein trekker.

Et valtre blir nytta når ein skal finne sannsynet til ei samansett hending som består av fleire tilfeldige delforsøk. Vi kan sjå på to glas med 10 farga klinkekuler i kvart. 4 er raude, 5 er kvite og ei er gul i kvart av glasa. Vi skal finne sannsynet for å trekke ei raud kule frå kvart av glasa.



Først kan vi starte med å få oversyn over kva for nokre moglege utfall vi har, og da kan vi bruke eit valtre. Her har vi to uavhengige delforsøk, sidan det vi trekker opp av det ene glaset ikkje påverkar kva vi trekker opp frå det andre glaset. Et valtre byrjar med ei grein for kvart mogleg utfall for det eine delforsøket. Her har vi tre moglege utfall; raud, kvit eller gul klinkekule. For kvar grein deler vi opp i dei moglege utfalla i delforsøk 2, som og er raud, kvit eller gul. Da blir valtreet slik:



Til slutt må vi sette på kor stort sannsyn det er for kvar grein, slik at det blir enkelt å rekne ut sannsynet for å trekke ei raud frå kvart glas. Ser du korleis sannsynet er rekna ut? HINT: gunstige utfall / moglege utfall

For å finne ut sannsynet, finn vi ut av kor mange samansette greiner som gir eit gunstig utfall, altså raud kule frå begge glas. Vi ser av figuren at det berre er éi samansett grein som gir to raude kuler. Vi veit at dei to delforsøka er uavhengige av kvarandre, og dermed kan vi berre multiplisere sannsynet for å få raud frå den eine krukka med sannsynet for å få raud frå den andre krukka. Vi får da:

$$P(2 \text{ raude kuler}) = \frac{4}{10} \cdot \frac{4}{10} = \frac{16}{100} = \frac{4}{25}$$

TIPS:

Det kan være lurt å forkorte brøkane før ein set dei inn i valtreet, slik at det ikkje blir så store tal i reknestykka. Da blir det enklare å forkorte etter utrekning og.

Diskuter

Korleis vil valtreet bli dersom du berre har éi krukke du skal trekke to raude kuler frå?

1. Kva blir skilnaden frå dømet over?
2. Korleis vil sannsynet og valtreet da bli?

Lag eit spel

- og tilpass programmet som simulerer sannsynet

Oppgåve

Bruk terningen eller lukkehjulet frå opplegg 17 for å lage eit spel. Du skal og lage eit program som du kan bruke til å simulere at du triller to terningar eller snurrer to gonger på lukkehjulet og reknar ut sannsynet for dei ulike moglege utfalla.



Fase 1: Finn informasjon om ulike typar spel som finst, og bruk han som inspirasjon til ditt eige spel. Kva for nokre materiale skal du bruke? Hugs at det må inkludere minst éin terning eller eitt lukkehjul.

Fase 2: Ha ei idèmyldring for deg sjølv. Teikne gjerne ei skisse (eller mange skisser) før du diskuterer med dei andre. Deretter må gruppa avgjere korleis spelet skal sjå ut.

Fase 3: Tid for å lage spelet, i tillegg til simuleringsprogrammet:

- Endre det tidlegare simuleringsprogrammet (frå opplegg 17 og 18) til å kunne brukast for å simulere trilling av to terningar, eller å snurre to gonger på lukkehjulet, og rekne ut sannsynet for dei ulike utfalla.

HINT

Det kan vere lurt å teikne eit valtre for alle utfalla dine før du tek til å jobbe med simuleringsprogrammet.

Kan du lage eit flytskjema her?



Fase 4: Test programmet ditt. Ta ein liten prøverunde med spelet.

Fase 5: Verkar alt slik det skal? Var spelet gøy?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få eit best mogleg spel. Gjer gjerne endringar i simuleringsprogrammet ditt. Kan det gjerast meir effektivt, enklare eller ryddigare?

Fase 7: Dokumenter heile prosessen saman med resultatet for sannsynet til dei ulike utfalla på ein måte du vel sjølv.

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 8

Opplegg 20 - Statistikk og sentralmål



På Bore skule er det 23 lærarar, og desse har alle notert kva for ein skostorleik dei nyttar. Resultatet blei: 39, 40, 37, 37, 39, 38, 39, 36, 43, 38, 37, 38, 39, 42, 43, 38, 38, 38, 44, 40, 37, 39 og 43.

Dette er ganske uoversiktleg, difor finst det fleire måtar å få betre oversyn over innsamla data på. Vi kan til dømes sjå på ulike sentralmål. Dei seier noko om kvar midten av målingane er.

Sentralmål

Gjennomsnitt – er summen av alle målingane dividert med talet på målingar.

I dømet vårt blir det:

$$\frac{39 + 40 + 37 + 37 + 39 + 38 + 39 + 36 + 43 + 38 + 37 + 38 + 39 + 42 + 43 + 38 + 38 + 38 + 44 + 40 + 37 + 39 + 43}{23} \approx 39$$

Typetal – er den målinga det er flest av.

I dømet vårt blir det 38.

Median – er det talet som er i midten av målingane når dei er stilt opp i stigande rekkefølge.

I vårt døme blir det

36, 37, 37, 37, 37, 38, 38, 38, 38, 38, 38, **39**, 39, 39, 39, 40, 40, 42, 43, 43, 43, 44

Snakk om

Vi har samla inn informasjon om kor mykje 100 vaksne får i årsløn. Tre av dei er direktørar og tener over 5 millionar kroner. Resten av dei tener mellom 200 000 og 700 000 kroner. Kva for eit tal trur de er størst, gjennomsnittsløna eller medianløna? Kvifor?

Lag ein gjennomsnittskalkulator

- og ein del av ei statistikkutstilling

Oppgåve

De skal lage eit program som reknar ut eit gjennomsnitt. De skal bruke data samla inn frå solfangaren frå opplegg 7 i programmet. De skal måle solfangar-temperaturen etter kvart minutt i 10 minutt. Programmet skal rekne ut gjennomsnittet av målingane dykkar, og gjennomsnittet av temperaturstigningen etter 10 minutt for alle gruppene i klassen.

Lag ein del av ei statistikkutstilling ut frå første del av denne oppgåva. Det kan mellom anna vere ein plakat, ein tredimensjonal modell, ei utgreiing om korleis de gjorde det, bilete og tekst. Tenk gjennom korleis de skal utforme utstillinga for å få fram kva de gjorde på ein god og effektiv måte.



Fase 1: Undersøk gjerne kva måtar ein kan lage ei utstilling på.

Fase 2: Kva for nokre fargar vil de nytte i utstillinga? Skal de bruke nokre spesielle materiale? Vil de berre bruke papir i fleire fargar, eller materiale som silkepapir, stoffbitar, garn, papp? Skal de bygge ein modell av noko? Lag ei skisse over korleis de ønsker at utstillinga skal bli til slutt.

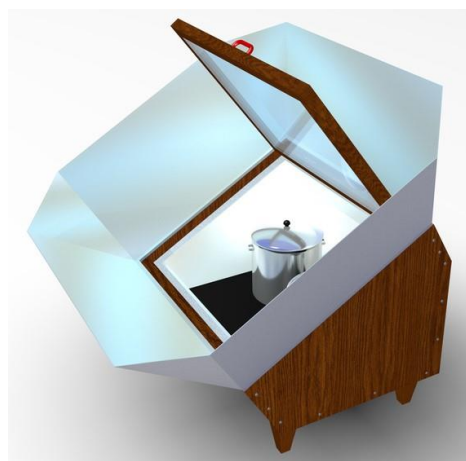
Fase 3: Lag første versjon av programmet for å rekne ut gjennomsnitt. Lag det som trengs til utstillinga.

Fase 4: Test programmet dykkar, får de eit svar som verkar fornuftig? Spør andre i klassen om utstillinga dykkar er tydeleg og god.

Fase 5: Samanlikne gjerne med dei andre i klassen, er det nokon som har ei betre utstilling? Kvifor meiner de at deira er betre? Kan de bruke noko av dei samme grepa i utstillinga dykkar ?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få ei best mogleg utstilling. Gjer gjerne endringar i programmet dykkar om det trengs.

Fase 7: Pass på å ta vare på bilete og notat de har gjort undervegs, slik at de kan vise kva de har tenkt. I denne oppgåva går dokumenteringa ut på å lage sjølve utstillinga.



Opplegg 21 - Frekvenstabell og sektordiagram



Nå skal vi sjå på ulike måtar å vise fram data, med hjelp av frekvenstabell og sektordiagram.

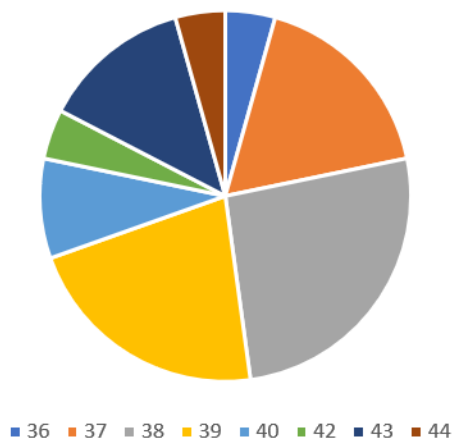
Frekvenstabell – Passar for å vise fram eit datasett med eit avgrensa tal på verdiar. Vi tel opp kor mange vi har av kvar måling og ordner det i ein tabell. Kor mange vi har av kvar måling blir kalla frekvensen til målinga. I sko-dømet vårt er frekvensen til skostorleik 39 lik 5. Frekvenstabellen for sko-dømet blir slik.

Måling	Frekvens
36	1
37	4
38	6
39	5
40	2
42	1
43	3
44	1

Sektordiagram – Passar for å vise fram eit datasett med et avgrensa tal på verdiar eller kategoriar. Det passar i sko-dømet vårt, der alle verdiane er frå 36 til 44. Det viser kor stor andel kvar måleverdi svarar til.

Vi nyttar ein frekvenstabell som utgangspunkt for å lage eit sektordiagram. Ein reknar ut kor stor brøkdelen kvar måling svarer til – kor stor del av lærarane som har kvar skostorleik. Denne brøkdelen kan vi multiplisere med vinkelsummen for ein sirkel, for da finn vi ut kor mange grader av sirkelen kvar skostorleik svarer til. Da kan vi til slutt teikne sektordiagrammet, og merke av storleiken på alle kakestykkene vi rekna ut storleiken på, med hjelp av ei gradskive.

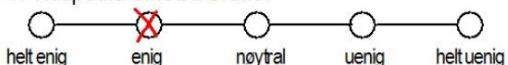
Måling	Frekvens	Brøkdelen/andel	Vinkel
36	1	$\frac{1}{23} \approx 0,043$	$0,043 \cdot 360^\circ \approx 15^\circ$
37	4	$\frac{4}{23} \approx 0,174$	$0,174 \cdot 360^\circ \approx 63^\circ$
38	6	$\frac{6}{23} \approx 0,261$	$0,261 \cdot 360^\circ \approx 94^\circ$
39	5	$\frac{5}{23} \approx 0,217$	$0,217 \cdot 360^\circ \approx 78^\circ$
40	2	$\frac{2}{23} \approx 0,087$	$0,087 \cdot 360^\circ \approx 31^\circ$
42	1	$\frac{1}{23} \approx 0,043$	$0,043 \cdot 360^\circ \approx 15^\circ$
43	3	$\frac{3}{23} \approx 0,130$	$0,130 \cdot 360^\circ \approx 47^\circ$
44	1	$\frac{1}{23} \approx 0,043$	$0,043 \cdot 360^\circ \approx 15^\circ$
SUM	23	1	360



Spørjeundersøkingar – Er ein metode for å samle inn data som vi kan handsame statistisk. Når ein utviklar ei spørjeundersøking, er det fleire ting ein bør tenke på.

- **Ja-seing:** Menneske har ein tendens til å vere samd i utsegner dei får presentert. Det er difor best å spørje om same ting minst to gonger, men med motsett vinkling.
 - Eg liker å lese bøker. / Eg misliker å lese bøker.
 - Eg synes det er spennande å finne ut av ting eg ikkje veit. / Eg er ikkje spesielt nyfiken.
- **Skala:** Ofte nyttar ein det som kallest ein likert-skala eller graderingsskala. Da har ein gjerne svaralternativ som ueinig, litt ueinig, nøytral, litt einig og einig, i tillegg til veit ikkje.

1. Wikipedia er lett å bruke.



Diskusjonsoppgåve: Kvifor trur de at det er vanleg med likert-skala? Kva meiner de om dømet? Er det et godt eller dårleg døme?

Lag ei spørjeundersøking

Oppgave

Design ei spørjeundersøking som samlar inn data, og samle resultata i ein frekvenstabell og eit sektordiagram, for minst eitt spørsmål. Lag eit program som bereknar vinklar til sektordiagrammet ut frå tala i frekvenstabellen. Bruk resultata som ein del av statistikkutstillinga dykkar, med ei utgreiing om undersøkinga og resultatet.

Fase 1: Undersøke og finne informasjon

Undersøk gjerne kva for nokre ulike typar data som kan vere fornuftig å samle inn? Er det noko spesielt de ønsker å finne ut av? Kva slags spørsmål kan vere bra å bruke?

Framlegg til datainnsamling – ulike typar spørsmål

Kor mange ein har av noko (talet på):

- Søsken, skjermar i heimen, bøker i heimen o.l.

Kor tilfreds/ikkje tilfreds ein er med noko

- Fritidsaktivitetar, leksene, klimapolitikk o.l.

Kor villig ein er til å endre på noko

Talet på flyreiser, kaste mindre mat, senke forbruket sitt o.l.



Fase 2: Idémyldring og planlegging

Ha ei idémyldring for deg sjølv. Kva for nokre data ønsker du å samle inn? Kvifor akkurat desse? Sjå etikkramma før du tek ei avgjerd. →

Teikne gjerne ei skisse over statistikkutstillinga før du diskuterer med dei andre. Deretter må gruppa samla avgjere kva de skal spørje om og korleis utstillinga dykkar skal sjå ut.

Planlegg gjennomføringa

- Anslå kor lang tid kvar del tar.
- Har de tid til å spørje alle spørsmåla?
- Dersom ikkje, kva skal de kutte ut?
- Korleis skal de dele inn svaralternativa?

Følg resten av fasane for å gjere oppgåva ferdig.

Etikk

Pass på at du ikkje spør om noko som kjennest negativt eller sensitivt for nokre personar.

Det må og vere helt frivillig å svare på spørjeundersøkinga.

Kan hende kan undersøkinga vere anonym?

Kvifor er dette så viktig? Kva har det med forskning å gjere?

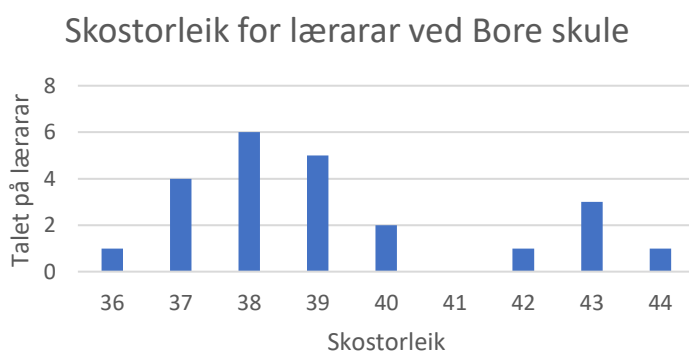


Opplegg 22 - Statistikk og diagram

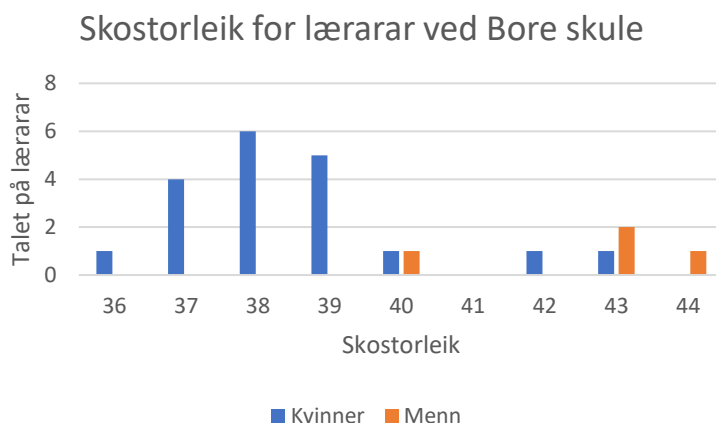


Nå skal vi sjå på ulike måtar å vise fram data med hjelp av diagram, og det finst fleire ulike typar diagram. Nokre av dei mest vanlege er:

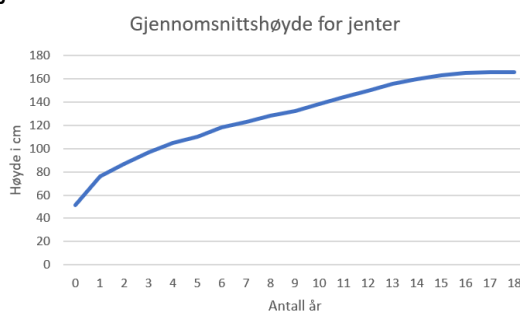
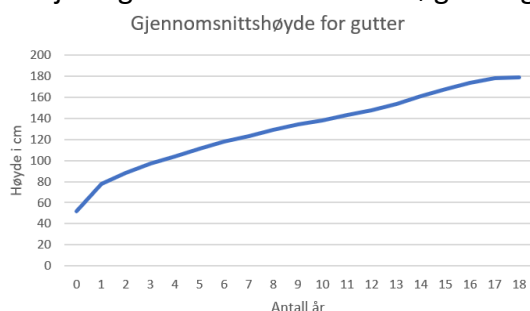
Stolpediagram (søylediagram) – Passar for å vise fram eit datasett med eit avgrensa tal på verdier eller kategoriar. Det passar i sko-dømet, der alle verdiane er frå 36 til 44.



Gruppert stolpediagram – Passar for å samanlikne to ulike grupper som du har same typen data for. Dersom vi visste kva for nokre skostorleikar som hørde til dei kvinnelege og dei mannlege lærarane, kunne vi laga eit gruppert stolpediagram som dette:



Linjediagram (kurvediagram) – Passar godt for å vise korleis noko endrer seg over tid. Det veit vi ikkje noko om i skostorleik-dømet vårt, så vi må finne eit anna døme. I desse to linjediagramma ser vi korleis høgda til gutar og jenter aukar når dei blir eldre.



Lag ei statistisk framstilling

- som skal nyttast i ei aktuell sak, og vidare som ein del av statistikkutstillinga dykkar

Oppgåve

Bruk micro:biten til å samle inn data, samle resultata i ein frekvenstabell og eit diagram, og lag eit aktualitetsoppslag ut frå målingane dykkar. Dette aktualitetsoppslaget skal vere ein del av utstillinga dykkar.

Fase 1: Undersøke og hente inn informasjon

Undersøk gjerne kva for nokre ulike typar data som kan vere fornuftig å samle inn? Sidan de skal bruke micro:bit til datainnsamling, bør de finne ut kva han kan brukast til å måle.

Fase 2: Idémyldre og planlegge

Ha ei idémyldring for deg sjølv. Kva for nokre data ønsker du å samle inn? Kvifor akkurat desse? Teikne gjerne ei skisse over aktualitetssaka før du diskuterer med dei andre. Deretter må gruppa samla avgjere kva de skal måle og korleis aktualitetssaka og utstillinga dykkar skal sjå ut.

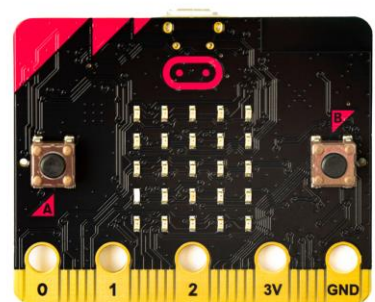
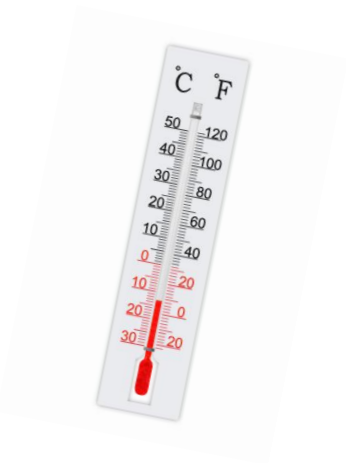
Fase 3: Lag første versjon av programmet for å gjere målingane dykkar. Lag førsteutkast av aktualitetssaka og det som de elles treng til utstillinga.,

Fase 4: Test programmet, får de målingar som verkar fornuftige? Spør andre i klassen om aktualitetssaka dykkar er tydeleg og god.

Fase 5: Samanlikne gjerne med dei andre i klassen, er det nokon som har ei betre sak? Kvifor meiner de at ho er bedre? Kan de bruke nokre av dei same grepa i saka dykkar?

Fase 6: Hopp gjerne attende til tidlegare punkt, og gjer endringar for å få ei best mogleg aktualitetssak og utstilling. Gjer gjerne endringar i programmet dykkar om det trengst.

Fase 7: Pass på å ta vare på bilete og notat de har gjort undervegs, slik at de kan vise kva de har tenkt. I denne oppgåva går dokumenteringa ut på å lage sjølvne aktualitetssaka og utstillinga.



Opplegg 23 - Statistikk og «fake news»

Det finst mange døme på dårlege eller villeiande framstillingar av data. Dette kan vere gjort med vilje for å fremje eit særst synspunkt, eller berre komme av dårleg utforma tabellar og diagram. Vi skal sjå på nokre måtar å lage dårlege og feil framstillingar.

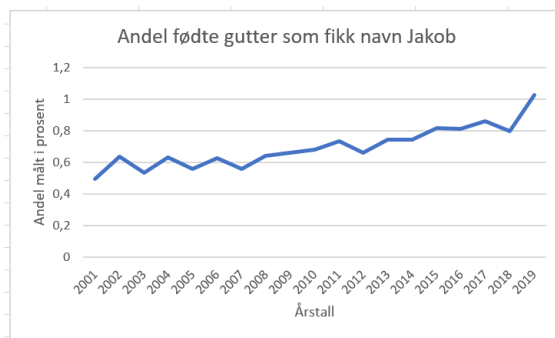
Kutte aksar – fører til overdrivne skilnader om ein kuttar på loddrett akse og feil inntrykk av utviklinga om ein kuttar på vassrett akse.

Kutte ut einskilde målingar – kan få statistikken til å vise noko heilt anna enn han faktisk gjer.

Avslutte med pil – ser ut som om utviklinga held fram i same retning, sjølv om vi ikkje kan vite noko om dette.

Utelate relevant informasjon – gjer det vanskelegare å danne seg eit fullstendig bilete av grunnar til utvikling.

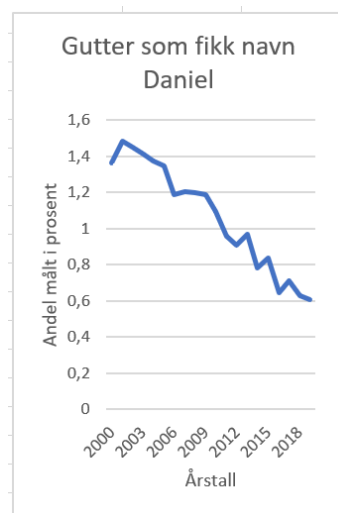
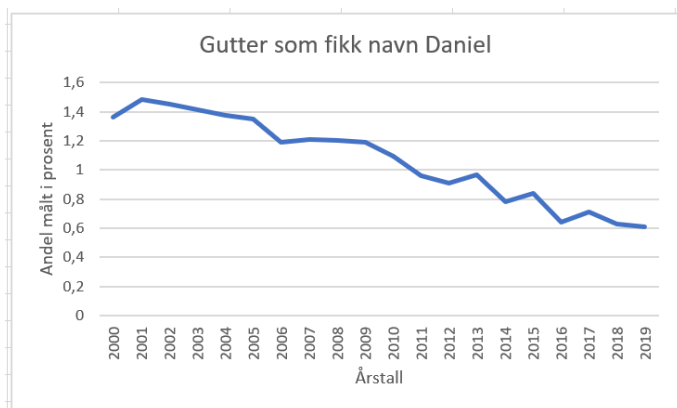
Ujamn avstand i aksane – kan føre til overdrivne eller underdrivne skilnader. Da kan ein få data til å gje inntrykk av akkurat kva ein sjølv ønsker.



Data henta frå: <https://www.ssb.no/statbank/table/10467/>

I diagrammet til venstre ser det ut til at andelen guttebabyar som blir kalla Jakob auker dramatisk og er på veg vidare opp, sidan vi avslutter med ein pil oppover. Den loddrette aksa er kutta, og det gjer at det ser ut til at endringane er veldig store. Dette diagrammet gjev inntrykk av at nesten alle guttebabyar vil få namn Jakob om nokre få år, og det er ikkje særst truleg. Dersom vi ser på diagrammet til høgre, ser det ikkje like dramatisk ut. Her har vi med nokre fleire målingar som viser at utviklinga ikkje er like dramatisk. Dersom du vil sjå korleis utviklinga er for ditt namn, kan du gå inn hit: <https://www.ssb.no/statbank/table/10467/>

Proporsjonar – dersom ein byter om på formatet, kan diagram gje inntrykk av større endringar enn det er grunn for. Standard er eit liggande format. Sjå døme under:



Lag ei villeiande statistisk framstilling

- som skal brukast i ei falsk aktualitetssak, og vidare som ein del av statistikkutstillinga dykkar

Oppgåve

Bruk micro:biten til å samle inn data, og samle resultata i eit diagram, lag eit falskt aktualitetsoppslag ut frå målingene dykkar. Dette falske aktualitetsoppslaget skal vere ein del av utstillinga dykkar.

Fase 1: Undersøke og innhente informasjon

Undersøk gjerne kva for nokre ulike typar data som kan vere fornuftig å samle inn? Sidan de skal nytte micro:bit til datainnsamling, bør de finne ut kva han kan brukast til å måle. Hugs at de skal lage ei falsk aktualitetssak til slutt, så det kan vere lurt å tenke på kva vinkling de ønsker å ha i saka.

Fase 2: Idémyldre og planlegge

Ha ei idémyldring for deg sjølv. Kva for nokre data ønsker du å samle inn? Kvifor akkurat desse? Teikne gjerne ei skisse over saka før du diskuterer med dei andre. Deretter må gruppa samla avgjere kva de skal måle og korleis aktualitetssaka og utstillinga dykkar skal sjå ut.

Fase 3: Lag første versjon av programmet for å gjere målinga dykkar. Lag førsteutkast til aktualitetssaka, og det som de elles treng til utstillinga.

Fase 4: Test programmet, får de målingar som verkar fornuftige? Spør andre i klassen om saka dykkar er tydeleg og god.

Fase 5: Samanlikne gjerne med dei andre i klassen, er det nokon som har ei betre aktualitetssak? Kvifor meiner de at deira er betre? Kan de bruke nokre av dei same grepa i saka dykkar?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få ei best mogleg aktualitetssak og utstilling. Gjer gjerne endringar i programmet dykkar om det trengs.

Fase 7: Pass på å ta vare på bilete og notat de har gjort undervegs, slik at de kan vise kva de har tenkt. I denne oppgåva går dokumenteringa ut på å lage sjølve aktualitetssaka og utstillinga.



Opplegg 24 - Statistikk og spreingsmål



På Bore skule er det 23 lærarar, og desse har alle notert kva for ein skostorleik dei bruker. Resultatet ble: 39, 40, 37, 37, 39, 38, 39, 36, 43, 38, 37, 38, 39, 42, 43, 38, 38, 38, 44, 40, 37, 39 og 43.

Dette er ganske uoversiktleg, difor finnest det fleire måtar å få betre oversikt over innsamla data på. Vi kan til dømes sjå på ulike spreingsmål. Dei seier noko om kor mykje målingane varierer.

Spreingsmål

Variasjonsbreidd – er skilnaden mellom høgaste og lågaste måling. I dømet vårt blir det:

$$44 - 36 = 8$$

Varsians – han seier noko om kor langt frå gjennomsnittet alle målingane er. Først må vi rekne ut gjennomsnittet, som i opplegg 20, der vi fekk at gjennomsnittet var 39.

Deretter må vi ta differansen mellom målinga og gjennomsnittet, og opphøge det i andre. Dette multipliserer vi med frekvensen til målinga, for han seier kor mange av målinga vi har. Til slutt summerer vi svara vi får for alle målingane våre og dividerer med talet på målingar. Det er lurt å bruke ein frekvenstabell for å gjere dette, elles er det fort gjort å gå i surr. I dømet vårt blir det slik:

Måling	Frekvens	$(\text{Måling} - \text{Gjennomsnitt})^2 \cdot \text{Frekvens}$
36	1	$(36 - 39)^2 \cdot 1 \approx 9$
37	4	$(37 - 39)^2 \cdot 4 \approx 16$
38	6	$(38 - 39)^2 \cdot 6 \approx 6$
39	5	$(39 - 39)^2 \cdot 5 \approx 0$
40	2	$(40 - 39)^2 \cdot 2 \approx 2$
41	0	$(41 - 39)^2 \cdot 0 \approx 0$
42	1	$(42 - 39)^2 \cdot 1 \approx 9$
43	3	$(43 - 39)^2 \cdot 3 \approx 48$
44	1	$(44 - 39)^2 \cdot 1 \approx 25$
SUM	23	115

Der vi til slutt må ta summen og dele på talet på målingar: $\frac{115}{23} \approx 5$

Standardavvik – er kvadratrota av variansen. I dømet vårt blir det:

$$\sqrt{5} \approx 2,2$$

Lag ein standardavvikskalkulator

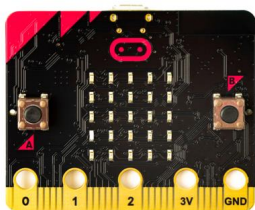
- og ein del av statistikkutstillinga dykkar

Oppgåve

Design ei statistisk undersøking som samlar inn data i naturfag, samle resultata i ein frekvenstabell og lag eit program som reknar ut standardavviket.

Lag ein del av utstillinga med ei utgreiing om undersøkinga og resultatet. Det kan vere ein plakat, ein tredimensjonal modell, ei skildring eller noko heilt anna. Tenk gjennom korleis de skal utforme utstillinga for å få fram det som de gjorde på ein god og effektiv måte.

Fase 1: Undersøk gjerne kva ulike typar data som kan vere fornuftig å samle inn. Vil de bruke micro:bit til dette, eller ei spørjeundersøking? Korleis skal denne delen av utstillinga sjå ut? Leit etter inspirasjon frå ulike typar kjelder.



Fase 2: Kva for nokre fargar vil de bruke i utstillinga? Skal de bruke noko spesielt materiale? Vil de berre bruke papir i fleire fargar, eller materiale som silkepapir, stoffbitar, garn, papp? Skal de bygge ein modell av noko? Lag ei skisse over korleis de ønsker at utstillinga skal bli til slutt.

Fase 3: Lag første versjon av programmet for å berekne standardavvik. Lag eventuelt eit program for å bruke micro:biten til å gjere målingar. Lag det som trengs til utstillinga.

Fase 4: Test programmet, får de eit svar som verkar fornuftig?



Fase 5: Samanlikne gjerne med dei andre i klassen, er det nokon som har ei betre utstilling? Kvifor meiner de at deira er betre? Kan de bruke noko av dei same grepa i utstillinga dykkar?

Fase 6: Hopp gjerne attende til tidlegare punkt og gjer endringar for å få ei best mogleg utstilling. Gjer gjerne endringar i programmet dykkar om det trengs.

Fase 7: Pass på å ta vare på bilete og notat de har gjort undervegs, slik at de kan vise kva de har tenkt. I denne oppgåva går dokumenteringa ut på å lage sjølve utstillinga.

Opplegg 25 - Statistikk i naturfag og samfunnsfag

- kva for nokre typar data finst, og korleis bør dei representast?

Bruk av statistikk i naturfag og samfunnsfag følger gjerne ei inndeling som dette. Det tyder ikkje at grensene for dei ulike typane data er fullstendig klare, eller at nokre av framstillingane ikkje kan brukast i begge fagområda.

Naturfag

1) Data som skal «bli éin verdi» og fordeler seg rundt denne.

Dette er typiske målingar vi gjer i naturfag, der vi held alle andre variablar mest mogleg fast. Da vil vi få ulike verdiar på grunn av feilkjelder og måleuvisse. Feilkjeldene får vi ikkje gjort så mykje med, men vi kan diskutere kva vi trur påverkar målingane våre. Måleuvisse kan vi finne ut av storleiken på, ved å rekne ut gjennomsnitt og standardavvik for målingane våre.

Døme: Måling av lengda på ein gjenstand, måling av farten til en bil, eller måling av vindstyrke.

2) Data som utviklar seg over tid.

Mange av målingane vi gjer i naturfag går ut på å måle ein variabel som endrar seg over tid. Desse målingane er det som regel nyttig å vise med eit linjediagram.

Døme: Temperaturen i løpet av eit døgn eller blodsukkernivået i løpet av ei veke.



Samfunnsfag

1) Data som spreier seg på ulike verdiar, med eit avgrensa tal på verdiar eller kategoriar.

Dette er typiske målingar vi gjer i samfunnsfag gjennom spørjeundersøkingar. Der er det gjerne lagt inn eit avgrensa tal svaralternativ med ein likert-skala. Denne typen data passar veldig godt å vise fram i ein frekvenstabell eller eit stolpediagram. Eit sektordiagram er fint å bruke i dei situasjonane at fordelinga av de ulike verdiane er viktig i høve til kvarandre.

Døme: Karakterfordelinga på ein prøve eller partitilhøyringa for folk.

2) Data som har så mange ulike verdiar at vi må dele dei inn i intervallar.

Mykje av dei dataa vi samlar inn i samfunnsfag, har så mange ulike verdiar at vi ikkje kan ta med alle enkeltverdiane. Da lagar vi grupper som inneheld fleire av verdiane, vi lagar intervall. Denne typen data kan presenterast med tabellar eller histogram.

Eksempler: Høgdefordeling for gutar på 10 år eller lengda på skulevegen for ein klasse.

3) Data som utviklar seg over tid.

Mykje av dataa vi samlar i samfunnsfag går ut på å måle éin variabel som endrar seg over tid. Desse målingane er det som regel nyttig å vise med eit linjediagram, akkurat som i naturfag. I samfunnsfag blir desse dataa gjerne samla inn i spørjeundersøkingar, og ikkje målte direkte med måleinstrument og sensorar.

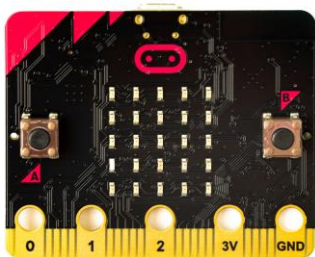
Døme: Talet på personar som stemmer på eit visst parti i løpet av 10 år eller korleis folketalet endrar seg over tid.

Lag ei statistisk fargeleggingsbok

- Bruk micro:bit til å samle inn data

Oppgåde

Design ei datainnsamling i naturfag (forsøk) der de bruker micro:bit for å samle inn data, og vis fram resultatata på ein ryddig måte der de sjølv vel kva slags tabellar og diagram de vil bruke. Lag ei side til ei fargeleggingsbok med ei enkel forklaring av undersøkinga, og resultatet. Denne sida skal seinare og nyttast i statistikkutstillinga.



Fase 1: Undersøke og finne informasjon

Undersøk gjerne kva for ulike typar data som kan vere fornuftig å samle inn? Er det verdier som endrar seg over tid? Ein verdi som skal vere konstant, som de måler fleire gonger? Eller er det verdier som bør delast inn i intervallar? Tenk gjennom korleis de best kan vise fram dataa de samlar inn – kva slags framstilling som blir tydelegast.

Sidan de skal bruke micro:bit til dette, må de finne ut kva han kan brukast til å måle. Hugs at det går an å kople til andre sensorar og elektronikk.

Fase 2: Idémyldre og planlegge

Ha ei idémyldring for deg sjølv. Kva for nokre data ønsker du å samle inn? Kvifor akkurat desse? Teikne gjerne ei skisse over sida til fargeleggingsboka før du diskuterer med dei andre. Deretter må gruppa samla avgjere kva de skal måle og korleis sida dykkar i fargeleggingsboka skal sjå ut.

Planlegg gjennomføringa

- Berekne kor lang tid kvar del tek.
- Har du tid til å gjere alle målingane?
- Dersom ikkje, kva skal du kutte ut?
- Korleis skal du dele inn målingane?

Følg resten av fasane for å gjere oppgåva.

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 9

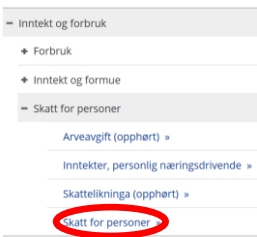
Opplegg 26 - Last ned data frå statistisk sentralbyrå

Mange forskarar bruker data frå statistisk sentralbyrå (SSB) til å forske på mange ulike tema. Det kan du og gjere! Men da treng du å vite korleis du kan laste ned data frå SSB, og korleis du kan bruke dei vidare. Det får du eit oversyn over her (eller en algoritme, om du vil):

1. Gå inn på www.ssb.no og trykk på statistikkbanken.

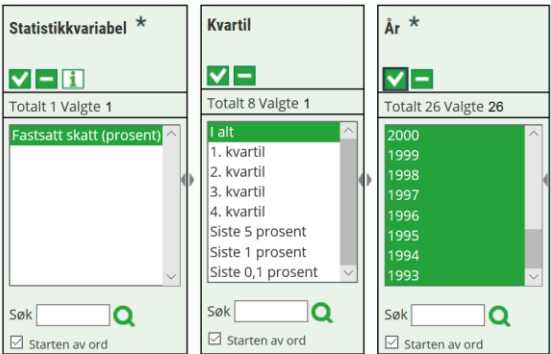


2. Vel den kategorien du vil finne data for, og så kva for ein tabell du vil sjå på.



08564	Oversiktstall fra skatteoppgjøret for alle personer. Antall, mill. kr og gjennomsnitt (F)	2006 - 2018
06593	Skatt i prosent av bruttoinntekt for ulike bruttoinntektsgrupper (prosent)	1993 - 2018
05802	Formuesskatt (før nedsettelse), etter alder og kjønn. Antall personer og gjennomsnitt, alle personer	1997 - 2018

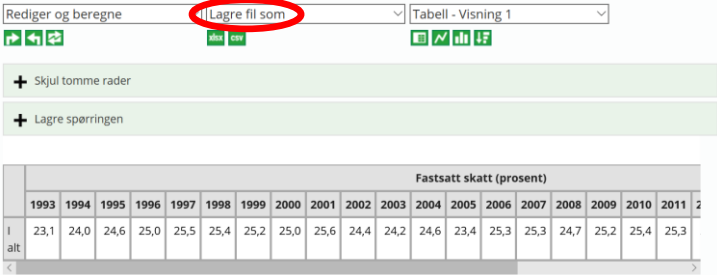
3. Vel dei variablane du vil ha med i tabellen din, og trykk på fortsett-knappen.



Antall valgte celler 26 (maksimalt tillatt er 300 000)
Presentasjon på skjermen er begrenset til 1 000 rader og 30 kolonner



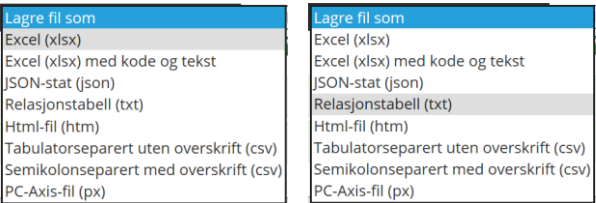
4. Da visest tabellen, og du kan laste han ned. Trykk på «Lagre fil som», og vel filtype.



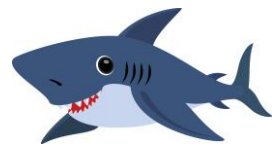
5. Dersom du skal bruke data i

- Geogebra eller Excel → bruk .xlsx-filer
- Python → bruk .txt-filer

6. Lagre filen i same mappe som du skal lagre Geogebra, Excel- eller Pythonfila di.



Lag figurar som illustrerer dataa du bruker



- Bruk statistikk frå SSB

Oppgåve

Finn eit datasett hos statistisk sentralbyrå (sjå oppskrift på førre side), og plott ein graf av dataa. Lag ein lineær matematisk modell for datasettet ditt og vurder om modellen er god. Til slutt skal de lage figurar som illustrerer kva for nokre data de har brukt, som skal nyttast i ei utstilling saman med grafen og modellen.

Fase 1: Tenk gjerne kva for nokre data som kan vere fornuftig å bruke. Sidan de skal bruke SSB som kjelde for data, må det vere eit datasett som finnest der.

Husk at de skal lage figurar som illustrerer datasettet til slutt, så det kan vere lurt å tenke på kva type figurar som kan passe til datasetta de ser på.

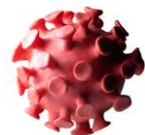
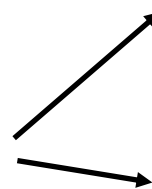
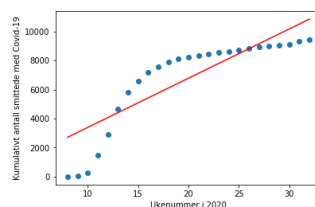
Kan hende kan ein smeltande snømann illustrere utslepp av CO₂ eller stigande gjennomsnittstemperaturar?



Fase 2: Ha ei idémyldring for deg sjølv. Kva for nokre data ønsker du å samle inn? Kvifor akkurat desse? Teikne gjerne ei skisse over figurane før du diskuterer med dei andre. Deretter må gruppa samla avgjere kva for nokre data de skal bruke, og korleis dykkar figurar og utstilling skal sjå ut.

Fase 3: Lag første versjon av grafen og den matematiske modellen. Lag figuren/figurane de treng til utstillinga.

Fase 4: Test grafen og modellen, får de ein modell som verkar fornuftig? Spør andre i klassen om utstillinga dykkar er tydeleg og god.



Fase 5: Samanlikne gjerne med dei andre i klassen, er det nokon som har ei betre utstilling? Kvifor meiner de at ho er betre? Kan de bruke nokre av dei same grepa i utstillinga dykkar?

Fase 6: Hopp gjerne attende til tidlegare punkt, og gjer endringar for å få ei best mogleg utstilling.

Fase 7: Pass på å ta vare på bilete og notat de har gjort undervegs, slik at de kan vise kva de har tenkt. I denne oppgåva går dokumenteringa ut på å lage sjølve utstillinga.

Opplegg 27 - Personleg økonomi



Personleg økonomi handlar om dei pengane folk tener, låner og bruker på ulike varer og tenester, og større ting som bolig og bil. Her kjem ei lita forklaring på nokre viktige omgrep.

Innskot – Pengar vi har ståande på konto i banken.

Lån – Vi kan låne pengar frå bankar eller kredittkortselskap, og da må vi betale ein viss prosentdel av lånesummen (det vi har lånt), i tillegg til sjølve lånesummen, attende. Det finst fleire typar lån, mellom anna serielån og annuitetslån.

Renter – Ein viss prosentdel av innskota eller låna våre. Dersom det er renter på innskot, er det pengar vi får frå banken. Men dersom det er renter på lån, er det pengar vi må betale til banken.

Forbruk – Det vi kjøper. Eit høgt forbruk tyder at vi bruker mykje pengar, medan eit lite forbruk tyder at vi bruker lite pengar.

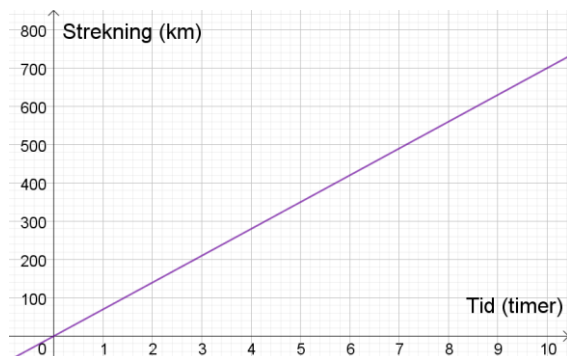
Løn – Dei pengane vi tener når vi arbeider.

Skatt – Når vi får løn, må vi betale ein viss prosentdel av pengane til staten, dette vert kalla skatt.

Budsjett – Eit oversyn over kor mykje pengar vi trur vi kjem til å bruke på ulike ting.

Rekneskap – Eit oversyn over kor mykje pengar vi faktisk har brukt på ulike ting.

Stigningstall til lineære funksjonar



Denne lineære grafen viser strekninga ein bil har kjørt, som funksjon av tida han bruker. Vi skal finne funksjonsuttrykket til grafen.

Alle lineære funksjonar kan skrivast på formen

$$f(x) = a \cdot x + b$$

der a er stigningstalet og b er konstantleddet.

Stigningstalet fortel kor bratt grafen er, og konstantleddet fortel kor grafen skjærer y-aksen.

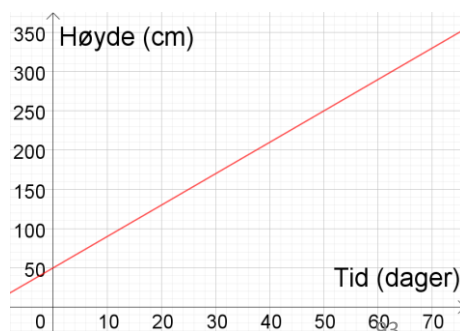
For grafen vår blir konstantleddet $b = 0$ sidan grafen krysser y-aksen i origo, altså når $f(x) = 0$. Når vi skal finne stigningstalet a , må vi finne kor mykje grafen går oppover for kvar eining på x-aksen. Da vel vi to punkt på grafen vår, og reknar ut kor stor endringa er på y-aksen mellom dei to punkta. Så reknar vi ut kor stor endringa er på x-aksen mellom de same to punkta. Etterpå deler vi endringa på y-aksen på endringa på x-aksen, og da finn vi stigningstalet.

$$a = \frac{\text{endring i } y}{\text{endring i } x} = \frac{700 \text{ km} - 0 \text{ km}}{10 \text{ t} - 0 \text{ t}} = \frac{700 \text{ km}}{10 \text{ t}} = 70 \text{ km/t}$$

Stigningstalet til ein lineær funksjon seier oss kor mykje y aukar for kvar auki i x . I ein lineær graf med strekning og tid, slik som vi har, vil stigningstalet alltid gje oss farten, altså kor mykje strekninga i kilometer endrar seg per time. Funksjonsuttrykket til grafen vår blir med det $f(x) = 70 \cdot x$.

Diskusjonsoppgåver

- Grafen til høgre viser høgda til ei solsikke som funksjon av talet på dagar etter ho blei sett ut.
 - Kva er funksjonsuttrykket for denne grafen?
 - Kva tyder stigningstalet for denne grafen?
 - For kva for nokre x -verdiar trur du denne grafen stemmer med målingane?
- Kva tyder det at ein graf har negativt stigningstal?
- Ein lineær graf viser talet på firarar som ein funksjon av talet på terningkast.
 - Kva blir stigningstalet (for store x -verdiar)?
 - Kva tyder stigningstalet for denne grafen?



Forsk på personleg økonomi

- Bruk statistikk frå SSB

Oppgåve

Finn data innan personleg økonomi som varierer over tid (<https://www.ssb.no/statbank/>), og utfør ein lineær regresjon (gjærne med Python) for å finne ein matematisk modell. Lag eit produkt som viser kva for nokre data du har brukt, og kva du har gjort med dei. Hugs grunngjeving for dei metodane du har nytta, og forklar kva du har funne ut.



Fase 1: Undersøk kva for nokre data som kan vere fornuftig å bruke. Kva for nokre data har med personleg økonomi å gjere? TIPS: Lån, renter, forbruk, arbeidsløysse og løn.

Tenk gjennom korleis de best kan vise fram dataa de samlar inn – kva slags framstilling som blir tydelegast. Vil de lage ein plakat, ein film, ei utstilling, eller noko anna?

Fase 2: Ha ei idémyldring for deg sjølv. Kva for nokre data ønsker du å bruke? Kvifor akkurat desse? Teikne gjerne ei skisse over produktet før du diskuterer med dei andre. Deretter må gruppa samla avgjere kva for nokre data de skal nytte og kva for eit produkt de skal lage.

Planlegg gjennomføringa, og berekne kor lang tid kvar del tar.

Fase 3: Lag første versjon av grafen for den matematiske modellen. Lag det som de treng til produktet dykkar.

Fase 4: Test produktet med å spørje dei andre i klassen om produktet dykkar er informativt og tydeleg.

Fase 5: Samanlikne gjerne med dei andre i klassen, er det nokon som har brukt interessante data eller gjort noko spennande med produktet sitt? Kan de bruke nokre av dei same grepa til dykkar produkt?

Fase 6: Hopp gjerne attende til tidlegare punkt, og gjer endringar for å få eit best mogleg produkt. Gjer gjerne endringar i grafen dykkar om det trengst.

Fase 7: Pass på å ta vare på bilete og notat de har gjort undervegs, slik at de kan vise kva de har tenkt. I denne oppgåva går dokumenteringa ut på å lage sjølve produktet.

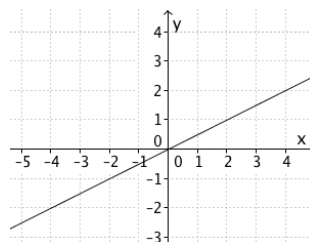


Oppgåve

Kva er stigningstalet for den lineære funksjonen din?

Forklar kva stigningstalet ditt tyder.

Forklar kva resultatet hadde blitt om stigningstalet ditt blei mindre.



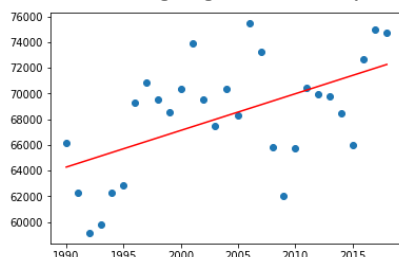
Opplegg 28 - Statistikk og korrelasjon

Korrelasjon
tyder ikkje
nødvendigvis at
vi har kausalitet!

Dersom vi har to datasett (for to variabler), kan vi finne ut kor godt dei korrelerer med kvarandre. Altså om dei varierer på same måten. Det vil seie at dersom den eine variabelen aukar, så aukar den andre variabelen tilsvarande, eller kan hende minkar han tilsvarande. Vi kan seie at korrelasjon er det same som samvariasjon.

For å sjekke kor godt to variablar korrelerer utfører vi ein lineær regresjon. Da kan vi få berekna korrelasjonen, det er nemlig eit mål på kor store avstandene er mellom regresjonsgrafen og kvart av punkta i grafen. Vi reknar ikkje dette for hand, men geogebra eller Python kan gjere det for oss.

Litt forenkla vil geogebra eller Python i ein lineær regresjon lage ei rett linje som gjer at dei samla avstandene mellom alle punkta og sjølve regresjonslinja blir minst mogleg. Altså at summen av avstandane mellom dei blå punkta og den røde lineære grafen blir minst mogleg.

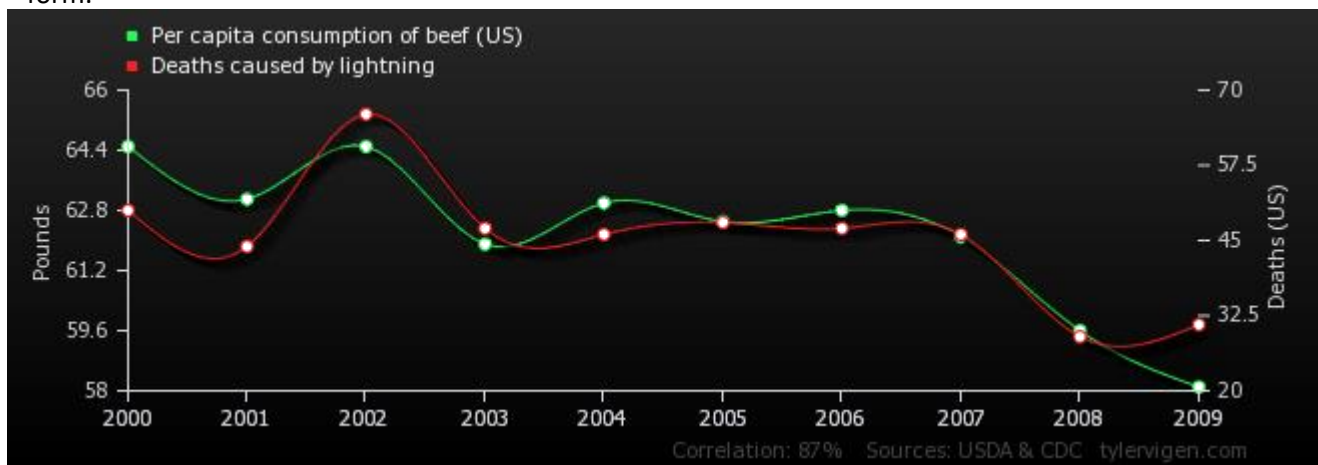


Kva er kausalitet?

Det tyder at det er ein årsak-verknad- samanheng mellom variablar. Dersom ein endrar på éin variabel, vil den andre variabelen og endre seg. Eit døme på en kausal samanheng kan vere at talet på timar du søv om natta påverkar kor trøtt du er neste dag. Eit anna ord for kausalitet er årsakssamanheng.

Ein kan samanlikne to variablar som varierer over tid, og sjekke om dei varierer på same måten. Til dømes kan ein velje to variablar som det er gjort målinger av årleg i perioden 1990 til 2020. For å sjekke kor godt dei to variablane korrelerer, vel ein den éine av variablane som x-verdi og den andre som y-verdi. Det er viktig å passe på at x-verdien og y-verdien er for same årstalet. Da kan ein utføre ein lineær regresjon på x- og y-verdiane og finne korrelasjonen.

Det er ikkje dette som er gjort med dei to grafane under. Her er begge variablane teikna inn som funksjon av årstal, slik at vi har to separate funksjonar. Men vi kan sjå at begge grafane har liknande form.



Korrelasjonskoeffisient: 0,87

Kilde: www.tyllervigen.com

Korrelasjonskoeffisienten – kan variere mellom -1 og 1.

Dersom han er 0, eller nær 0, tyder det at vi har ingen eller liten korrelasjon.

Dersom han er nær -1, minkar den eine variabelen når den andre aukar.

Dersom han er nær 1, vil begge variablane auke/minke på same måten.

Det er tilsvarande for R^2 , men han kan ikkje vere negativ.

Nokre moglege årsaker til korrelasjon:

- Ein variabel kan direkte forårsake eller avhenge av verdien til ein annan variabel.
- Ein variabel kan vere laust knytt til ein annan variabel.
- To variablar kan begge avhenge av ein ukjent tredje variabel.
- Det kan vere tilfeldig – størst sjanse for at dette skjer ved få målingar.

Lag ei villeiande aktualitetssak

- med bruk av korrelasjon

Oppgåve

Finn to datasett hos statistisk sentralbyrå (<https://www.ssb.no/statbank/>), og finn korrelasjonen med bruk av Geogebra eller Python. Lag eit aktualitetsoppslag som speulerer i kausalitet (at den eine variabelen er årsak til at den andre blir endra).

Fase 1: Undersøk kva for nokre data som kan vere fornuftig å bruke. Kva for nokre data kan tenkest å variere på same måten over tid? Hugs at de skal lage ei falsk aktualitetssak til slutt, så det kan vere lurt å tenke på kva vinkling de ønsker å ha i saka.

Fase 2: Ha ei idémyldring for deg sjølv. Kva for nokre data ønsker du å bruke? Kvifor akkurat desse? Teikne gjerne ei skisse over saka før du diskuterer med dei andre. Deretter må gruppa samla avgjere kva for nokre data de skal bruke og korleis aktualitetssaka dykkar skal sjå ut.

Fase 3: Lag første versjon av grafen for den matematiske modellen. Kva blir korrelasjonen mellom variablane? Lag aktualitetssaka basert på korrelasjonen.

Fase 4: Får de en korrelasjon som verkar fornuftig? Spør andre i klassen om aktualitetssaka dykkar er tydeleg og god.

Nokre døme på villeiande aktualitetssaker og konspirasjonsteoriar - basert på korrelasjonar

Ny studie:

Mye egg koblet til hjertesykdom og død

Kanskje er ikke egg så ufarlig likevel?

[Ny studie: Mye egg koblet til hjertesykdom og død \(forskning.no\)](#)

Folk som drikker brus – både sukkerfri og vanlig – lever kortere

Brus hører ikke til i hverdagen, sier forsker.

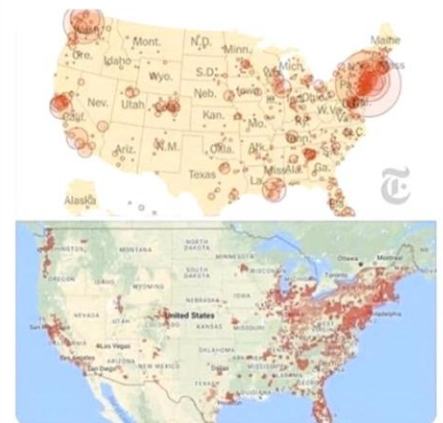
[Folk som drikker brus – både sukkerfri og vanlig – lever kortere \(forskning.no\)](#)

Håndskrift hjelper hukommelsen

Barn som lærer seg å skrive for hånd, har lettere for å huske og utvikle egne ideer enn barn som lærer seg å skrive via et tastatur, viser ny forskning.

[Håndskrift hjelper hukommelsen \(aftenposten.no\)](#)

One is a 5G Map, One is the CoronaVirus Map in the USA
Always a bigger picture 🕵️



Theodore R. Smith på Twitter: «Correlation != Causation. Tho this probably will correlate with population density which will correlate with mass infections, and 5G could be ruining people's immune systems. <https://t.co/r5as5Wfm2s> / Twitter

Fase 5: Samanlikne gjerne med dei andre i klassen, er det nokon som har ei betre aktualitetssak? Kvifor meiner de at deira er betre? Kan de bruke noko av dei same grepa i dykkar sak?

Fase 6: Hopp gjerne attende til tidlegare punkt, og gjer endringar for å få ei best mogleg aktualitetssak. Gjer gjerne endringar om det trengst.

Fase 7: Pass på å ta vare på bilete og notat de har gjort undervegs, slik at de kan vise kva de har tenkt. I denne oppgåva går dokumenteringa ut på å lage sjølve aktualitetssaka.

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 10

Opplegg 29 - Varme, isolasjon og energieffektivitet

Varme er energi som blir overført fordi gjenstandar har ulik temperatur. Ingen gjenstandar kan innehalde varme, det er kun noko som kan bli overført frå eit system til eit anna. Eit slikt system med høg temperatur kan til dømes vere deg sjølv, og eit system med lågare temperatur kan vere lufta rundt deg. Da vil det bli overført varme frå deg til lufta rundt deg. Som oftast ønsker vi ikkje å overføre så mykje varme, da minkar temperaturen vår, og vi frys. Da kan vi sørge for å bruke ei form for

isolasjon. Den vanlegste isolasjonen for oss menneske, er klede vi bruker. Dei hindrar at temperaturen vår søkk for mykje.



Grunnen til at klede verkar isolerende, er at dei inneheld mykje stillestående luft. Luft leier varme veldig dårleg, og verkar med det som ein varmeisolator. Men om lufta rører på seg, isolerer ho ikkje særst godt. Tenk berre på kor mykje kaldare det kjennest ute når det blæs kraftig, samanlikna med når det ikkje er vind. Dei fleste metall leier varme veldig godt. Tenk berre på grytene de har heime, dei er typisk laga av metall sidan vi ønsker at maten skal bli oppvarma raskast mogleg.

Kor gode varmeisolatorar materiale er, heng saman med varmeleiingsevna deira. Varmeleiingsevna seier noko om kor godt eit materiale leier varme. Dersom eit materiale har god varmeleiingsevne, er det ein dårleg varmeisolator.

I ein termos er heile poenget å sørge for at minst mogleg varme slepp ut til omgivnadene. Til vanleg er termosar i hovudsak samansette av glas, plast og stål.

Snakk om

Har du høyrte om leiar og isolator i andre samanhengar enn varmeleiing?



Papp



Plastfolie



Aluminiumsfolie



Glas



Garn



Plast



Kakelys

Snakk om

Kva for ein funksjon kan desse ulike materiala ha?

Er det nokon som isolerer betre enn andre?

Er det nokon andre grunnar til å bruke desse materiala i eit energieffektivt hus enn varmeleiingsevne?



Foto: Takako Picture Lab / Shutterstock.com

Eit passivhus er eit hus som bruker lite energi til oppvarming samanlikna med vanlege hus. Det er med andre ord veldig energieffektivt. For å kallast et passivhus må det tilfredsstille fire krav:

- 1) Lite varmetap.
- 2) Lite behov for oppvarming.
- 3) Minst halvparten av varmtvatnet skal varmast opp utan bruk av strøm eller fossile brensel.
- 4) Ventilasjonsanlegget og materiala må vere energieffektive.

Energiproduksjon

Sidan energi ikkje kan oppstå eller forsvinne, berre omdannast til andre typer energi, er all elektrisitetsproduksjon eigentleg berre ei omforming. Med vannkraft blir stillingsenergien til vatnet omgjort til elektrisitet, med vindkraft blir rørsleenergien til lufta (vinden) omgjort til elektrisitet. For fossile brensel er det kjemisk energi som blir omgjort til elektrisitet eller varme, men da blir store mengder CO₂ frigjeve. Dette fører til globale klimaendringar. Dersom vi klarer å bruke mindre energi, mellom anna ved å ha mest mogleg energieffektive hus, vil vi kunne redusere CO₂-utsleppa våre.

Lag eit energieffektivt hus

- mål temperaturen med radiostyrt micro:bit

Oppgåve

Lag eit mest mogleg energieffektivt hus. Mål kor mykje temperaturen endrar seg på 30 minutt med å bruke ein micro:bit inni huset, denne sender temperaturen til ein annan micro:bit.

Fase 1: Korleis verkar eit energieffektivt hus? De kan undersøke litt meir om teorien bak passivhus.

Fase 2: Det er viktig at de er åpne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

1. Tenk sjølv først og teikne skisser frå ulike vinklar.
2. Forklar ideen din for dei andre på gruppa. Bruk gjerne skissene i forklaringa.
3. Heile gruppa diskuterer dei ulike ideane og lagar ein felles hypotese for bygging.

Fase 3: Gjennomfør planen dykkar for å lage huset, og lag programmet for å måle temperaturen. Sjå tips under.



Lag ein hypotese

Lag ein hypotese som skildrar korleis det mest energieffektive huset ser ut.

Micro:bit nummer 1

Lag eit program som sender temperaturen med å bruke følgande kommandoar:

```
radio.send_bytes(x)    radio.on()
radio.config(channel=40)
temperatur = temperature() while True:
    sleep(500) from microbit import *
import radio x = str(temperatur)
```

Micro:bit nummer 2

Lag eit program som mottok og skriv ut temperaturen med å bruke følgande kommandoar:

```
x = radio.receive_bytes() sleep(500)
radio.on() import radio print(x)
while True: radio.config(channel=40)
from microbit import *
```

Fase 4: Test kor mykje temperaturen endrar seg.

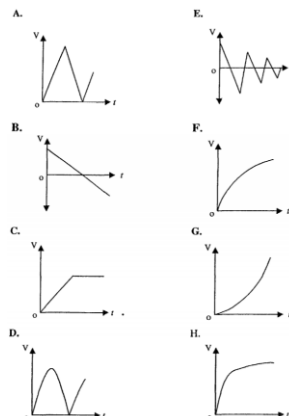
Fase 5: Samanlikne resultata med andre i klassen. Fekk nokon andre mindre temperaturnedgang?

Fase 6: Gå tilbake til dei andre fasane for å gjere planlagte forbetringar.

Fase 7: Gjennomfør dei siste målingane og dokumenter prosessen på ein valfri måte.

Modelleringsoppgåver

1. Mål temperaturen kvart minutt i 30 minutt.
2. Plott alle dei målte verdiane i Geogebra eller med Python
3. Finn ein matematisk modell med å føreta en regresjon for dei målte dataa.
4. Korleis kan de avgjere kva for ein modell som passar best?
5. Passar modellen bra med målingane dykkar?
6. Kva kan det gyldige området til modellen vere? Grunngi svaret.



Opplegg 30 - Populasjonar og populasjonsvekst

Kva er ein populasjon?

Det er alle individa av ein viss art som lever innanfor eit avgrensa område eller økosystem. Det blir og kalla ein bestand.

Kva er eit økosystem?

Det er samansett av alle populasjonane innanfor eit avgrensa område, samt dei abiotiske faktorane i området. Nokre døme kan vere ei myr, ein skog eller eit korallrev.

Ekspontialfunksjonar og prosentvis endring

Ein eksponentialfunksjon kan alltid skrivast på denne forma $f(x) = a \cdot b^x$ der a er startverdien og b er vekstfaktoren som alltid er eit positivt tal.

Vi har ein populasjon av rådyr og vi veit at han aukar med 20 % årleg. Kva blir vekstfaktoren til denne populasjonen?

Vi tek til med å rekne om 20 % til desimaltal.

$$20 \% = \frac{20}{100} = 0,20$$

Sidan heile den opprinnelege populasjonen alltid tilsvarer 100 %, må vi legge til dette

$$100 \% + 20 \% = 1,00 + 0,20 = 1,20$$

I dette høvet blir vekstfaktoren til rådyrpopulasjonen 1,20.

Diskuter

Kva blir vekstfaktoren om populasjonen minkar med 20 % årleg?

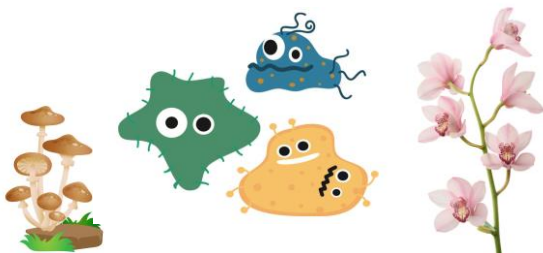


Startverdien er talet på rådyr i den opprinnelege populasjonen. Dersom vi har 250 i ein rådyrpopulasjon med ein årleg vekst på 20 %, vil funksjonen vår sjå slik ut:

$$f(x) = 250 \cdot 1,20^x$$

Kva er ein biotisk faktor?

Det er alle dei levande organismane i eit økosystem, slik som alle planter, dyr, bakteriar og sopp.



Kva er en abiotisk faktor?

Det er alle dei ikkje-levande omgivnadene som påverkar organismane som lever der, slik som jordsmonn, pH-verdi, klima og temperatur.



Lag ei øy!

- og programmer populasjonsveksten

Oppgave

Lag ein modell av ei lita øy, med eit oversyn over dei ulike biotiske og abiotiske faktorane. Tenk deg at det vert sett ut eit kaninpar på øya. Lag eit program som reknar ut kor mange kaninar det blir på øya kvart år etterpå, for de 10 første åra. Lag ein graf som viser populasjonsveksten for kaninane.

Fase 1: Kva for materiale skal de bruke for å lage øya? Kva for nokre abiotiske/biotiske faktorar skal de illustrere?

Fase 2: Det er viktig at de er åpne for alle slags idear og ikkje er for kritiske, da kan morosame framlegg bli kutta ut for tidleg.

1. Tenk sjølv først og teikne gjerne skisser.
2. Forklar ideen din for dei andre på gruppa.
3. Heile gruppa diskuterer dei ulike ideane, og lagar ein felles plan for øymodellen.



Fase 3: Lag ein modell av øya dykkar, med biotiske og abiotiske faktorar. Lag programmet som reknar ut populasjonsveksten. Spør gjerne læraren om ark med tips til programmeringa.

Puslespel-programmering

```
print(populasjon)

populasjon = populasjon * vekstfaktor

for i in range(0,10):
    vekstfaktor = 1 + økning/100
```

Fase 4: Korleis skal de teste modellen av øya dykkar? Kan hende kan det vere fornuftig å samanlikne resultata med andre i klassen.

Fase 5: Er det noko de vil endre?

Fase 6: Gå attende til dei andre fasane for å gjere planlagte forbetringar på modellen dykkar.

Fase 7: Dokumentér det som de har gjort med ein liten film og grunngje vala dykkar. Vis fram resultatet for resten av klassen med ei lita utstilling.

Refleksjonsoppgåver

1. Kva ville skjedd med dei ulike artane på øya om det flytta menneske dit?
2. Menneska ville trenge å dyrke mat, kva konsekvensar ville dette få for det biologiske mangfaldet på øya?
3. Kva er viktigast av at menneske kan bruke naturressursane, eller å ta vare på det biologiske mangfaldet?
4. Kva om menneska ville drive gruver eller lage eit kraftverk? Kva for nokre konsekvensar ville det fått?

Diskuter

1. Kva for ein type matematisk modell har vi brukt for populasjonen på øya?
2. Er denne realistisk?
3. Kvifor/kvifor ikkje?
4. Klarer du å finne ein modell som er meir realistisk?

Opplegg 31 - Vaksinar og immunforsvar



Korleis veit kroppen at nokre bakteriar og virus er farlege, men let yoghurtbakteriane vere? Kva er det som gjer at kroppen aktiverer immunforsvaret til den store sjukdomsjakta? Alle dei farlege virusa og bakteriane har nokre kjenneteikn som gjer det mogleg for immunforsvaret å oppdage dei.

Immunforsvaret har eit slags minne som gjer at når det blir aktivert, hugsar det kjenneteikna i lang tid, slik at det kan angripe sjukdomen neste gong vi blir smitta. Vaksinar inneheld viruset, eller bakterien vi blir vaksinerte mot, men dei er enten døde eller svekka slik at dei ikkje kan gi sjukdom. Når dei reduserte organismane kjem inn i kroppen, vil dei aktivere immunforsvaret, slik at immunforsvaret hugsar kjenneteikna til akkurat den organismen. Da kan kroppen sitt immunforsvar effektivt kjempe mot sjukdomen dersom ein blir smitta seinare.



Kva er smittetal?

Det fortel kor mange som gjennomsnittleg blir smitta av éin person medan han er sjuk. Smittetal er det samme som vekstfaktor i ein eksponential matematisk modell.

Vaksinemotstand mot meslingvaksinen (og andre)

Det finst personar som er motstandarar av vaksinar, og argumenterer for at andre heller ikkje skal vaksinast. Dette er hovudgrunnen til at sjukdomar som meslingar har fått utbrot der sjukdomen tidlegare var utrydda.

Det finst gjerne tre hovedargument dei bruker:

- Meslingvaksinen forårsakar autisme.
- Meslingar er bra for immunforsvaret.
- Vaksinane inneheld farlege stoff som aluminium, kvikksølv og formaldehyd.

Ein studie som ble gjort av Andrew Wakefield, og publisert i 1998, hevdar at det var ein samanheng mellom meslingvaksinen og autisme. Studien blei tilbakevist og trekt i 2004. Studien var bygd på forfalska data, og Wakefield blei fradømt retten til å praktisere som lege.

Meslingar er en sjukdom som kan få dødeleg utfall. Det er nokon som får milde biverknader av vaksinen, men dette er mykje mindre alvorlege biverknader enn dei som får meslingar. Det finst mange studiar som viser at immunforsvaret verkar dårlegare i lang tid etter sjukdomen. I slutten av 2019 kom det ein studie som viste at eit utbrot av meslingar får kroppen sitt immunforsvar til å «gløyme» andre sjukdomar han har hatt før, og miste immuniteten sin.

Aluminium finst i einskilde vaksinar, men i mykje mindre dose enn at det kan utgjere ein risiko. Det blir brukt for å gjere nokre vaksinar meir effektive. Kvikksølv (thiomersal) er blitt fjerna frå barnevaksinar, sjølv om det ikkje er vist nokon helserisiko da det blei brukt. Formaldehyd har vi i blodet frå før, i langt større mengd enn det som blir tilført med ein vaksine.

Kjeldekritikk

Sjekk påstandane og svara på påstandane om vaksinemotstand frå ulike internettkjelder.

Sjekk kva for nokre personar og institusjonar som står bak dei ulike synspunkta.

1. Kven meiner du er mest truverdige?
2. Kva for nokre kjelder oppgjev dei for påstandane sine?
3. Stemmer det dei skriv med det du kan om immunforsvaret frå før?



Oppgåve

1. Kor stor er sannsynet for dei fire ulike alvorlege bivirkningane frå figuren over?
2. Kor stor er sannsynet for å få meslingar i Norge dersom du ikkje er vaksinert?

Programmér og strikk eit smittetal

Oppgåve

Lag eit program som reknar ut kor mange personar som blir smitta med eit valfritt smittetal i 10 smittegenerasjonar. Start med 10 masker og la kvar maske svare til ein person. Kvar nye rad svarer til ein ny smittegenerasjon. Strikk smittelappen for dei 10 første smittegenerasjonane. La kvar gruppe lage ein blome av smittelappane dei lagar (ein «smitteblome»).

HINT

Smittetal svarer til vekstfaktor i eksponentialfunksjonar. Har de programmert vekst for eksponentialfunksjoner før?

Fase 1: Korleis kan ein bruke programmering til å rekne ut smittespreiing? Kva tyder smittegenerasjon? Veit du korleis ein kan auke talet på masker når ein strikkar? Dersom det er noko i oppgåva du ikkje forstår, prøv å finne det ut. Diskuter gjerne med andre på gruppa, søk på internett eller spør kva læraren meiner.

Fase 2: Idémyldre og planlegge

Det er viktig at de er åpne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

1. Tenk sjølv først, og teikne gjerne skisser av både smittelappane og smitteblomen.
2. Forklar og vis ideen din for dei andre på gruppa.
3. Heile gruppa diskuterer dei ulike ideane og lagar ein felles plan for smitteblomen.

Kva er det som gjer at smittelappane vil sjå ulike ut? Kva typar smittelappar vil medverke til ein fin smitteblome? Kor mange smittelappar skal gruppa lage? Kva for nokre fargar og kva type garn skal de nytte? Skal alle bruke dei samme materiala og fargane?



Fase 3: Programmering av populasjonsvekst og lage blome

Lag ei strikkeoppskrift med hjelp av programmering og strikk minst ein lapp kvar. Gjennomfør planen dykkar for å strikke populasjonsendringane, og lag ein plakat med blomen dykkar.

Fase 4: Sidan det er vanskeleg å teste kor god smitteblomen er, kan de hoppe over denne fasen.

Fase 5: Samanlikne smitteblomen med dei andre smitteblomane i klassen. Er det noko de vil endre på?

Fase 6: Gå attende til dei andre fasane for å gjere utbetringane de kom fram til i fase 5.

Fase 7: Dokumenter det som de har gjort og grunnge vala dykkar med ei film- eller lydfil. Presenter gjerne smitteblomen for resten av klassen og ha blomane på utstilling.



Diskuter

1. Korleis heng smittetalet saman med smittelappane?
2. Korleis kan ein bruke smittelappane til å sjå kor mange personar som blir smitta av ein sjukdom?
3. Kva skjer med talet på smitta dersom smittetalet blir dobbelt så stort?

Opplegg 32 - Forbrenningsreaksjonar

Kva er ein forbrenningsreaksjon?

Ein reaksjon mellom eit stoff og oksyngengass, og som frigjev energi.



For at ein rakett skal kunne skytast opp i verdsrommet, må han gjere om ei energiform til kinetisk energi (rørsleenergi). Nå for tida bruker rakettar kjemisk energi lagra i eit drivstoff og eit **oksiderande** stoff i fast form, flytande form eller ei blanding av desse formene. Desse stoffa blir blanda og tende på, slik at vi får ein forbrenningsreaksjon. I desse reaksjonane blir store mengder energi frigjevne, slik at raketten får stor nok fart til å nå verdsrommet.

I organiske forbrenningsreaksjonar er det eit organisk stoff som reagerer med oksygen. Dersom ein har god tilgang til oksygen, vil ein ha ei fullstendig forbrenning. Da blir det frigjeve mykje energi, vatn og karbondioksid. Om det ikkje er nok oksygen, får ein ei ufullstendig forbrenning. Da blir berre litt energi frigjeve, samt vatn og karbonmonoksid. Karbonmonoksid er ein luktfri og usynleg gass som er giftig å puste inn.

Diskusjonsoppgåver

1. Kva for nokre energiformer har vi i ei raketttoppskyting?
2. Kva har forbrenningsreaksjonar med klimaendringar å gjere?
3. Skjer det forbrenningsreaksjonar i menneskekroppen?

Fullstendig forbrenning: Karbonhaldig stoff + $O_2 \rightarrow H_2O + CO_2 + \text{ENERGI}$

Ufullstendig forbrenning: Karbonhaldig stoff + $O_2 \rightarrow H_2O + CO + \text{energi}$

Lag ein utskytingsmekanisme

Oppgåve

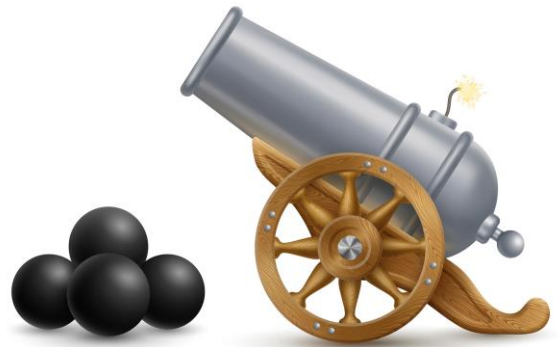
Lag ein utskytingsmekanisme som skyt ei kule lengst mogleg. Han må bruke ein servo som kan stille inn vinkelen på utskytingen i den retninga de programmerer. Mål kor lang tid det tek før kula treff bakken for forskjellige vinklar, og lag ein matematisk modell for samanhengen mellom vinkel og tid.

Fase 1: Gjennomfør informasjonsinnhenting for å få idéar dersom de ønsker det.

Fase 2: Idémyldre og planlegge

Det er viktig at de er åpne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

1. Tenk sjølv først og teikne gjerne skisser.
2. Forklar ideen din for dei andre på gruppa.
3. Heile gruppa diskuterer dei ulike ideane og lager ein felles hypotese for bygging.



Fase 3: Gjennomfør planen dykkar for å lage utskytingsmekanismen, og lag programmet for å stille inn vinkelen.

Fase 4: Test om utskytingsmekanismen verkar som planlagt.

Fase 5: Samanlikne resultata med andre i klassen.

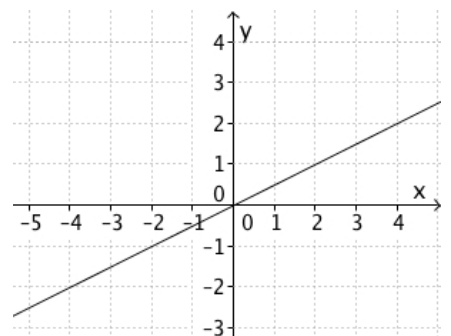
- Fekk nokon andre større avstand?
- Kvifor det?

Fase 6: Gå attende til dei andre fasane for å gjere dei planlagte forbetringane.

Fase 7: Gjennomfør dei siste målingane, desse skal de bruke til å plote ein graf i Geogebra og lage ein matematisk modell, sjå oppgåveboks under. Dokumentér prosjektet med eit valfritt produkt.

Oppgåver

1. Mål kor lang tid kula bruker på å nå bakken med minst fem ulike vinklar.
2. Plott alle dei målte verdiane i Geogebra.
3. Finn ein matematisk modell med å gjere ein regresjon for de målte dataa.
4. Korleis kan de avgjere kva for ein modell som passar best?
5. Kva tyder det når x-verdien er negativ?



Modellering

Passar modellen bra med målingane dykkar?

Kva kan gyldighetsområdet til modellen vere?

- kvifor?

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 11

Opplegg 33 - Geriljabroderi



Therese Susan Isaksen



Mathilde Ellingsen Skjæveland



Irene Skjølberg



Sissel Madsø
Granvang



Ira Pettersen



Ellen Strand Caspersen



Ingrid Langemyhr



Mathilde Ellingsen Skjæveland

Opplegg 33 - Geriljabroderi med teknologi

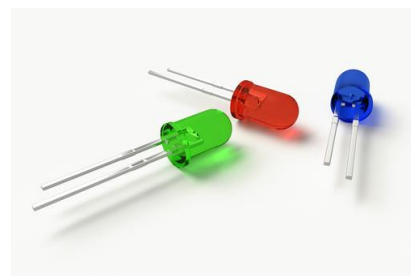
Oppgåve

Lag eit geriljabroderi der du og bruker elektrisk leiande sytråd, minst éi LED-pære og ein micro:bit.

Bruk innovasjonsmetoden frå opplegg 1 for å lage broderiet.



Fase 1: Undersøke og finne informasjon



Fase 2: Idémyldre og planlegge

Det finnes leiande sytråd som leier elektrisitet på akkurat same måten som leidningar.

Korleis kan du bruke han i broderiet ditt?

Korleis kan du dele opp oppgåva? Kva for nokre ulike delar er ho samansett av?



Opplegg 34 - Lag ei teknologisk 17. mai-fane

Oppg ve

Lag ei 17.mai-fane med elektroniske komponentar og minst  en micro:bit.



Bruk innovasjonsmetoden fr  opplegg 1 for   lage fana.

Id myldre

Id ar og plan

Utf re

Teste

Evaluerer

Forbetre

Dokumentere

Fase 1: Unders ke og finne informasjon

Kva slags teknologi kan brukast?



Kvar kan du finne ut om ulike elektroniske komponentar?



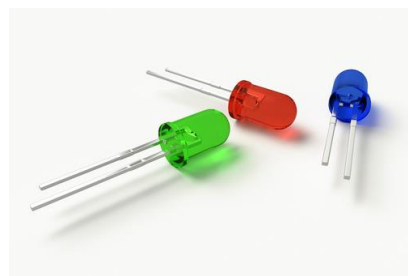
Korleis passar dei saman med 17.mai-designet?



Fase 2: Id myldre og planlegge

Korleis kan du bruke dei ulike elektroniske komponentane i fanedesignet?

Korleis kan du dele opp oppg va? Kva for nokre ulike delar er ho samansett av? Har de gjort noko liknande f r?



Opplegg 35 - Lag eit produkt som bruker viss-kloss(ar)

Oppgåve

Konstruer eit produkt som bruker ei måling til å gjere ei handling. Du kan bruke alle tilgjengelege materiale, inkludert servomotorar og elektronikk.

Bruk innovasjonsmetoden med dei sju fasane frå opplegg 1.

Idémyldre

Idéar og plan

Utføre

Teste

Evaluerer

Utbetere

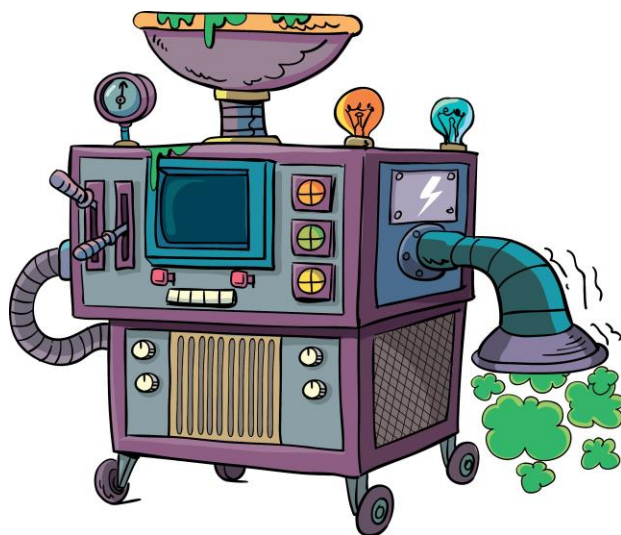
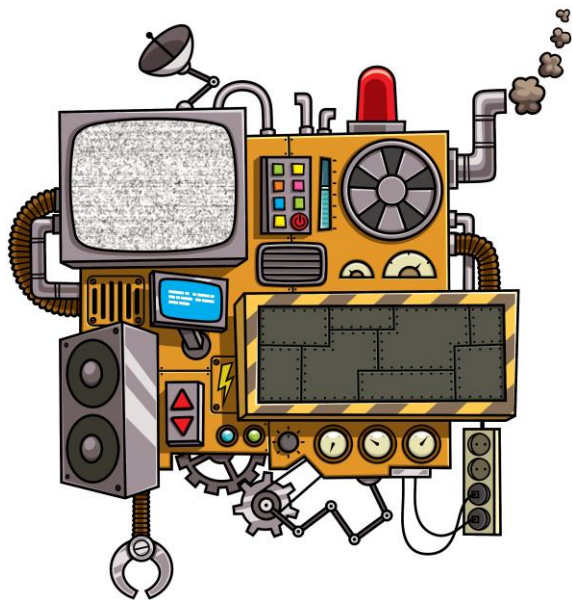
Dokumentere

Moglege målingar:

Temperatur
Lysstyrke
Spennings/motstand
Kople til andre sensorar?

Moglege handlingar:

LED-pære som lyser/blenker
Servo som rører på seg
Lage lydar/spele musikk



Fase 1: Undersøke og finne informasjon

Kan du lage eit produkt som er nyttig for nokon? Eller eit produkt som gjer noko positivt for miljøet? Korleis kan de sjekke kva folk har bruk for? Skal de berre google, eller kan hende spørje folk?

Fase 2: Idémyldre og planlegge

Korleis skal de kombinere sensorar med handlingar?

- Kva slags utstyr skal de bruke?
- Korleis blir programmeringa?

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 12

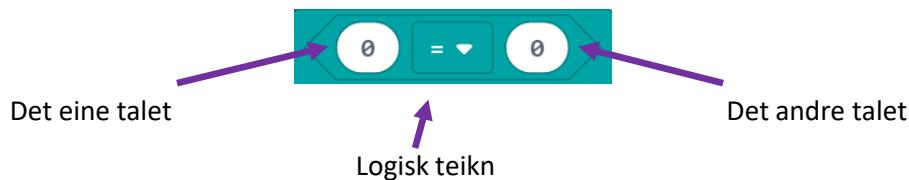


Introduksjon til Python-programmering

Vilkår (conditions)

Vilkår i Python er det same som vilkår i Scratch eller MakeCode. Skilnaden mellom dei er at i staden for å ha ferdige sekskanta klossar med vilkår, må vi skrive vilkåra sjølv i Python. Eit vilkår er ei utsegn som kan vere sann eller usann, og den typen vilkår vi kan bruke i programmering vil alltid samanlikne noko. Ofte samanliknar vi talverdiar og ser om dei er like store, om det første er mindre enn det andre, eller motsett. Det går og an å samanlikne tekstar for å sjå om dei er like eller ikkje. Men for tekstar kan ein ikkje finne ut om den eine teksten er større enn ein anna. Korleis skulle vi målt det?

Eit vilkår i python må vere samansett av tre delar, akkurat som i Scratch og MakeCode. To tal som skal samanliknast på ein eller annan måte, og eit teikn som seier korleis dei skal samanliknast. Dette teiknet kallast ofte for eit logisk teikn eller ein logisk operator. Dersom sammenlikninga stemmer, vil vilkåret gje oss verdien «sann» attende. Dersom samanlikninga ikkje stemmer, vil vi i staden få verdien «usann» attende. Sann og usann kallast boolske verdiar. Det går an å ha boolske variablar, det vil sei at variablen kan lagre verdien sann eller usann.



Under kjem eit oversyn over korleis ein kan skrive ulike vilkår. Sjå om du klarer å finne ut om døma returnerer verdien sann eller usann.

Vilkår		
Teikn	Forklaring	Døme
>	Større enn	4 > 5
<	Mindre enn	2 < 7
>=	Større enn eller lik	-2 >= 0
<=	Mindre enn eller lik	7 <= 12
==	Lik	9 == 19
!=	Ikkje lik	3 != 6
or	Eitt av vilkåra må vere sann	3 == 4 or 3 > 1
and	Alle vilkåra må vere sanne	3 == 4 and 3 > 1

Rekning

Nokre gonger treng vi å gjere reknestykke i Python. Dette er særst nyttig der vi har nokre variablar som vi, til dømes, skal multiplisere eller legge saman. Men det går fint an å gjere reknestykke direkte med tal. Under er eit oversyn over ulike rekneoperatorar for Python. Sjå om du klarer å sjå kva verdien av reknestykke-døma blir.

Rekneoperatorar		
Teikn	Forklaring	Døme
+	Addisjon	(-4) + 5
-	Subtraksjon	(-1) - (-4)
*	Multiplikasjon	3 * 0
/	Divisjon	12 / (-3)
**	Opphøgd i (eksponent)	2**3

Variablar i Python

Ulike typar variablar er viktige innan programmering. Eitt av poenga med å lage program, er at vi skal kunne bruke det same programmet på å løyse fleire liknande oppgåver. Da er det mykje greiare å lagre alle tal vi skal bruke i variablar, slik at vi berre treng byte ut tala ein stad, for at programmet likevel skal gi oss rett svar.

Ein variabel kan romme éin verdi, og kvar gong vi legg ein verdi i ein variabel, erstattar vi den verdien som eventuelt ligg i variabelen frå før. Når vi lagar ein variabel i Python, treng vi ikkje skrive kva type det er, det kjenner Python att ut frå den første verdien vi legg inn i variabelen. Det finnest fire typar variablar:



Heiltal
- integer

1234
5678
90

Desimaltal
- float

0.02

Tekst
- string

Hei!

Boolsk verdi
- boolean

True
False

Heiltal kallast integer i Python. Ein integer er eit positivt eller negativt tall som ikkje har nokon desimalar. 0 er og ein integer.

Desimaltal kallast float i Python. Ein float er eit tal som inneheld desimalar. Det kan vere enten negativt eller positivt. Talet 2,0 er ein float.

Tekst kallast string i Python. (På norsk blir og ordet «streng» brukt.) Ein tekstverdi er eitt eller fleire ord, det kan og vere heile setningar. For at python skal skjønne at det er ein tekstverdi, må vi bruke apostrofar (') rundt teksten. Dersom vi vil legge ordet hei inn i variabelen a, må vi skrive kommandoen slik: `a = 'hei'`

Boolsk verdi kallast boolean value i Python. Denne verdien kan vere enten sann eller usann. Han blir brukt i samband med vilkår, sidan eit vilkår kan vere enten sant eller usant.

Av og til treng vi å gjere om ein type variabel til ein annan type variabel. Slik som når det er en funksjon i Python som berre tek inn heiltall. Dersom vi da har eit desimaltal, må vi gjere det om til eit heiltal, og da må vi avrunde desimaltalet vårt. Da bruker vi kommandoen `round()`. Det går og an å bruke kommandoen `int()`, men denne avrunder ikkje desimaltalet, han berre kuttar ut alt bak kommaet. Det tyder i praksis at desimaltalet alltid blir runda ned til nærmaste heiltal.

For å kunne sende tal mellom to eller fleire micro:bitar, må verdiane vere gjort om til tekstverdier. For å gjere om ein talverdi til ein tekstverdi, kan vi bruke kommandoen `str()`.

Oppgåve

Kva vil bli vist på skjermen for desse programma? Kan du lage nokre program sjølv? Få ein i klassen til å gisse kva som vil skje i programma dine, eller kan hende læraren?

```
a = 10
print(a)
a = a + 1
print(a)
```

```
a = 1.7
a = round(a)
print(a)
```

```
a = '1.7'
a = float(a)
print(a+a)
```

```
a = 1.7
a = str(a)
b = float(a)
print(a+b)
```

```
a = 1.7
a = int(a)
print(a)
```

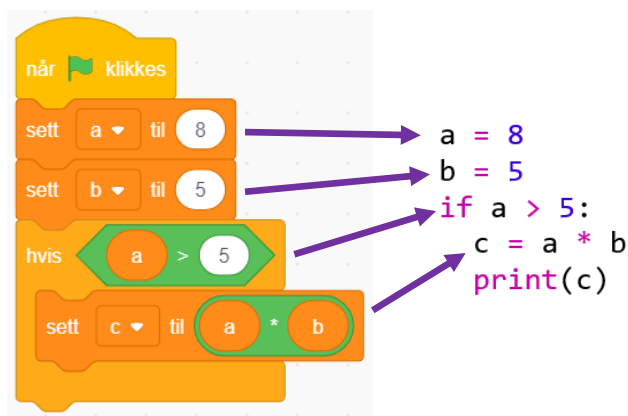
```
a = '1.7'
a = float(a)
print(a)
```

```
a = '1.7'
print(a+a)
```

```
a = 1.7
#a = str(a)
b = float(a)
print(a+b)
```

Viss-setningar (if-setningar)

Viss-setningar svarer til viss-klossar i Scratch og MakeCode. Dei må romme eit krav, og dersom kravet er sant, vil det som «høyrrer til» viss-setningen bli utført. I dei klossbaserte programmeringsspråka puttar vi inn det som skal skje med hjelp av klossane som passar slik som puslespelbrikker. I Python må vi gjere det same, berre at vi må skrive alt sjølv. Sidan vi nå ikkje har klossar for å vise kva som står «inni» viss-setningen, bruker vi innrykk i staden. Alt som står etter ein dersom-setning, med eit innrykk framføre, vil «stå inni» viss-setningen, og blir berre gjort om vilkåret i viss-setningen er sant.



Ofte treng vi å ha fleire alternativ som kan vere sanne. Det finnest to ulike variantar, og dei kallest gjerne «elles» eller «elles hvis».

Elles – det som skal skje, om vilkåret i dersom-setningen er usant.

else:

Elles hvis – det som skal skje om eit anna vilkår er sant.

elif:

Desse to programma gjer det same, og kvar blokk i Scratch tilsvarer éi linje i Python, med to unntak.

I Python treng vi ikkje å trykke på eit flagg, vi trykker på ein «play-knapp» som får koden til å byrje å køyre. Dermed treng vi ikkje noko som svarer til den første blokka i Scratch-koden.

I Python viser ikkje verdien på variablane, med mindre vi skriv dei ut til skjermen med å bruke print-kommandoen. I Scratch visest verdien til alle variablane heile tida, så der treng vi ikkje å få vist fram c på nokon spesiell måte.

Snakk om

Kva måtte du endre i Python-programmet for at det berre skulle rekne ut c når verdien i b var større enn verdien i a?

Kva for ein skilnad hadde det blitt dersom vi flytta linja med print(c) heilt til venstre i Pythonkoden, slik at ho ikkje sto med innrykk?

Oppgåve

Kva vil bli vist på skjermen for desse programma? Kan du lage nokre program sjølv? Få ein i klassen til å gisse kva som vil skje i programma dine, eller kan hende læraren?

```
a = 8
b = 5
if a > 5:
    c = a * b
    print(c)
else:
    print(a)
    print(b)
```

```
a = 8
b = 5
if a > b:
    c = a * b
    print(c)
elif a < b:
    print(a)
    print(b)
```

```
a = 5
b = 8
if a > b:
    c = a * b
    print(c)
elif a == b:
    print(a)
else:
    print(a)
    print(b)
```

```
a = 5
b = a
if a > b:
    c = a * b
    print(c)
elif a == b:
    print(a)
else:
    print(a)
    print(b)
```

```
a = 5
b = 8
if a > b:
    c = a * b
    print(c)
elif a == b:
    print(a)
elif a < b:
    print(a)
    print(b)
```

Korleis skrive viss-setningar – ei oppskrift

if betingelse:

kode som utføres når betingelsen er sann

elif betingelse:

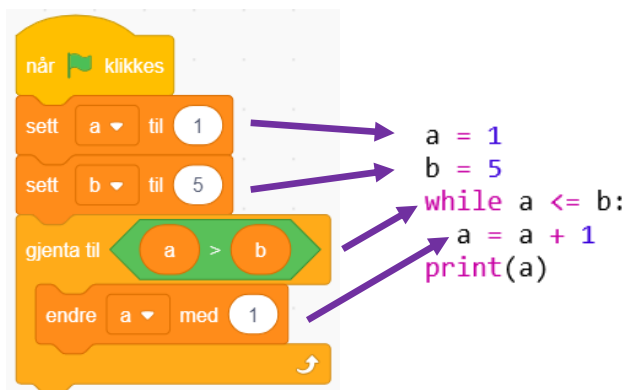
kode som utføres når betingelsen er sann

else:

kode som utføres hvis ingen av de andre betingelsene er sanne

While-lykkje i Python

While-lykkjer (vilkårlykkjer) svarer til «gjenta til-lykkjene» i Scratch. Dei må ha med eit vilkår som fortel kor mange gonger det som står inni lykkja skal gjerast. For Scratch er det slik at det som står inni lykkja blir gjort heilt til vilkåret blir sant (altså at lykkja går medan vilkåret er usant). I Python er det motsett, at det som står inni lykkja blir gjort heilt til vilkåret ikkje lenger er sant. Det gjer at sjølve vilkåret må skrivast på ulike måtar dersom ein skal oppnå det same med Scratch og Python.



Snakk om

Kva vil vise på skjermen for Python-programmet?

Vilkåret i dei to programma er ikkje like. Kva ville Pythonprogrammet vist dersom ein brukte vilkåret frå Scratch-programmet direkte?

Kva for ein verdi ville a hatt, dersom vi brukte vilkåret frå Pythonprogrammet direkte i Scratch-programmet?

Desse to programma gjer det same, og kvar blokk i Scratch svarer til éi linje i Python, men med to unntak.

I Python treng vi ikkje trykke på eit flagg, vi trykker på ein «play-knapp» som får koden til å byrje å køyre. Dermed treng vi ikkje noko som svarer til den første klossen i Scratch-koden.

I Python viser ikkje verdien på variablane, med mindre vi skriv dei ut til skjermen med å bruke print-kommandoen. I Scratch visest verdien til alle variablane heile tida, så der treng vi ikkje å få vist fram a på nokon særskilt måte, men vi rekk ikkje å sjå kva verdiar a har undervegs, sidan programmet går så raskt.

Korleis skrive while-lykkjer – ei oppskrift

while vilkår:

kode som utføres mens vilkåret er sant

Pass på å sette alt som skal vere inni while-lykkja med innrykk. Dersom det ikkje står med innrykk, vil det ikke skje før lykkja er ferdig, og programmet held fram.

Oppgåve

1. Hva vil bli vist på skjermen for desse programma?

Skriv ned kva du trur vil bli vist på skjermen, og skriv inn programma med Pythonkode etterpå. Køyr koden, og sjekk om du får det svaret du trudde du skulle få.

- Dersom du ikkje fikk svaret du trudde, kva var grunnen til det, trur du? Diskuter gjerne med andre.
- Kan du lage nokre program sjølv? Få ein i klassen til å gisse kva som vil skje i programma dine, eller kan hende læraren?

```
a = 1
b = 5
while a < b:
    a = a + 1
    print(a)
```

```
a = 1
b = 5
while a > b:
    a = a + 1
    print(a)
```

```
a = 10
b = 0
while a > b:
    a = a - 1
    print(a)
```

```
a = 10
b = 0
while a > b:
    a = a - 1
    print(a)
```

```
a = 1
b = 5
while a < b:
    a = a + 1
    print(a)
    print(b)
```


For-lykkjer i Python

For-lykkjer (teljelykkjer) svarer til «repetere eit visst tal gonger-lykkjene» i Scratch. Desse lykkjene gjentek det som står inni dei så mange gonger som vi ønsker. I Scratch kan vi berre skrive direkte kor mange gonger vi vil gjenta lykkja, men i Python er det litt annleis. Der blir det oppretta ein variabel som heiter `i`, vi kan gjerne kalle det ein teljevariabel. Han tek til på den talverdien som står først i parentesen til for-lykkja, i dømet vårt er det 0. Det andre talet i parentesen viser kva tal teljervariabelen i skal telje seg opp til (men ikkje inkludert). I situasjonen vår skal i telje seg opp til verdien 5. Det siste talet inni parentesen seier kor store steg vi skal auke verdien til teljervariabelen i med. I dømet vårt skal vi telje med 1, altså seier vi at teljervariabelen tek til på 0, vi tel opp til 5 med å auke med 1 kvar gong. Da vil i vere 0, 1, 2, 3, 4 før lykkja blir avslutta.



```
for i in range (0, 5, 1):  
    print(i)
```

Snakk om

Kva ville skjedd om vi brukte variabel `a` i staden for talet 4 i lykkja i Scratch?

Går både Scratch-programmet og Pythonprogrammet gjennom lykkja like mange gonger?

Korleis skrive for-lykkjer – ei oppskrift

```
for i in range (fra og med-verdi, til-verdi, steglengde):  
    kode som utføres mens i teller gjennom verdiene den skal ha
```

Desse to programma gjer det same, men her svarer ikkje kvar kloss i Scratch til éi linje i Python. Her er tre av blokkene i Scratch innbakte i for-lykkja i python. I tillegg er det to klossar i Scratch som ikkje svarer til nokon linjer i Python.

I Python treng vi ikkje trykke på eit flagg, vi trykker på ein «play-knapp» som får koden til å byrje å køyre. Dermed treng vi ikkje noko som svarer til den første klossen i Scratch-koden.

I Python viser ikkje verdien på variablane med mindre vi skriv dei ut til skjermen med å bruke print-kommandoen. I Scratch visest verdien til alle variablane heile tida, så der treng vi ikkje å få vist fram a på nokon særskilt måte, men vi rekk ikkje å sjå kva verdier han har undervegs sidan det går så raskt. For å kunne sjå kva for nokre verdier variabelen i Scratch har, har vi denne gongen lagt inn ein kloss som seier «vent 1 sekund», og denne gjer at programmet tek ein pause i eitt sekund. Da rekk vi å sjå verdien på variabelen før programmet går vidare att.

Oppgåve

1. Kva vil bli vist på skjermen for desse fem programma?

Skriv ned kva du trur vil visest på skjermen, og skriv inn programma med Pythonkode etterpå. Køyr koden, og sjekk om du får det svaret du trudde du skulle få.

2. Dersom du ikkje fekk svaret du trudde, hva var grunnen til det, trur du? Diskuter gjerne med andre.
3. Kan du lage nokre program sjølv? Få ein i klassen til å gisse kva som vil skje i programma dine, eller kan hende læraren?

```
for i in range (0, 5, 2):  
    print(i)
```

```
for i in range (0, 5, 0):  
    print(i)
```

```
for i in range (1, 10, 2):  
    print(i+i)
```

```
for i in range (1, 10, 2):  
    print(i-i)
```

```
for i in range (1, 10, 1):  
    print(i**2)
```

Lister i Python



Ofte treng vi å bruke mange variablar, og da kan det vere greit å lage ei liste eller ein tabell (array i Python). Dersom éin variabel svarer til éin skuff som har eit namn og ein verdi, er ei liste som ein kommode med fleire skuffar. Kvar variabel er eitt element i lista, og kvart element har eit namn og ein verdi. Namnet på elementet er namnet på heile lista saman med plasseringa på lista, altså kva for ei rad det høyrer til.

Dermed treng vi ikkje å lage et eige namn til kvart element. Det er og enkelt å bruke ei lykkje for å gå gjennom alle elementa for å bruke dei, til dømes i eit reknestykke.

Ei liste har eit namn og element med kvart sitt nummer. Det som er litt spesielt med lister innan programmering, er at det første elementet er element nummer 0. Det er fort gjort å gløyme når ein programmerer, og er alltid lurt å sjekke som ein del av feilsøkinga. Ei liste har i utgangspunktet berre element med éin datatype. Det går ikkje an å bruke nokon heiltal og nokon tekstverdiar i same liste. Ei liste er i utgangspunktet ei lang rekke med element, og er med det eindimensjonal.

Namnet på lista må vere eitt samanhengande ord for at Python skal skjønne det. Innan programmering er det vanleg å bruke `_` som mellomrom for namn på variablar, lister og tabellar. Bokstavane æ, ø og å bør vi unngå, sidan nokre Python-editorar ikkje taklar namn som bruker andre enn engelske teikn.

Ønskeliste

0. Hund
1. Mobil
2. Sko
3. Bukse
4. Hagenisse

Eit element har både et namn og ein verdi.

Kva er namnet og kva er verdien til dette elementet?

Oversyn over nokre nyttige kommandoar i Python

```
liste = []
```

Lagar ei tom liste.

```
liste = [1, 3, 5, 7]
```

Lagar ei liste med verdiane 1, 3, 5, 7.

```
liste[3] = 7
```

Legg inn verdien 7 i element tre, i ei liste med minst fire element.

```
del liste[3]
```

Slettar element tre, i ei liste med minst fire element.

```
liste.append(4)
```

Legg til verdien fire som eit nytt element i ei liste som heiter liste. Det har ikkje noko å seie kor mange element som er i lista frå før, for verdien blir alltid lagt i eit nytt element etter det siste elementet i den opprinnelege lista.

```
print(liste[2])
```

Hentar verdien til element to i lista, og skriv det på skjermen.

```
len(liste)
```

Finn kor mange element det er i ei liste.

Diskusjonsoppgåver

1. Kvifor er det liten vits i å bruke ei lykkje for å legge saman mange variablar?
2. Kvifor er det enkelt å bruke ei lykkje for å gjere det same dersom du har alle verdiane i ei liste?

Tabellar i Python

Ein tabell (array) kan innehalde fleire typer data, og han kan vere todimensjonal. Det vil seie at han har meir enn éin kolonne med data. Da treng vi to tal for å forklare kva for eit element i tabellen vi snakkar om. Det blir på same måten som eit punkt i eit koordinatsystem, der vi må bruke både x- og y- koordinaten for å plassere eit punkt. Det første elementet tek til på 0 for tabellar og, både for radene og for kolonnane. Tabellar finnest i numpy-biblioteket, og må importerast `from numpy import *`

Reine tekstfiler (.txt-filer) frå statistisk sentralbyrå gir todimensjonale (2D) tabellar i Python, og er med det viktige når vi skal analysere data og lage matematiske modellar. Det er ganske enkelt å gjere om ein 2D- array til ein 1D-array som er det `fit_function()` tek inn av data (sjå kapittel 13).

Tabellen under har namnet budsjett. Akkurat som for lister, byrjar vi å telle på null. Dette gjeld for både radene (vassrett) og kolonnane (loddrett). Rad null i budsjett-tabellen inneheld tekstverdiane: Månad, Inntekt, Klede og snacks, Mobilutgifter, Diverse utgifter, SUM. Kolonne null inneheld Månad, Januar, Februar, Mars, April, Mai, Juni.

Månad	Inntekt	Klede og snacks	Mobilutgifter	Diverse utgifter	SUM
Januar	1500	1000	300	100	100
Februar	1500	1000	300	100	100
Mars	3000	1900	300	100	700
April	1500	1000	300	100	100
Mai	1500	1000	300	100	100
Juni	2500	1000	300	300	900

Dette er element [3,5] og har verdien 700.

Diskusjonsoppgåver

1. Kva for ein verdi har element [0,0]?
2. Kva for eit element har verdien 3000?

Oversyn over nokre nyttige kommandoar

```
budsjett[7][2] = 2000
```

Endrar verdien frå 1500 til 2000 i elementet i rad sju og kolonne to i budsjett-tabellen på denne sida.

```
print(budsjett[3][1])
```

Skriv ut verdien til elementet i rad tre og kolonne ein, altså 1500 frå budsjettet på denne sida.

```
tabell = zeros([2,3])
```

Lagar ein tabell med to rader og tre kolonnar, der alle verdiene er null.

```
tabell = array([(1, 2, 3),(4, 5, 6)])
```

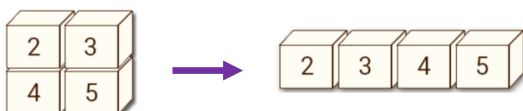
Lagar ein tabell med to rader og tre kolonnar. Tala innanfor parentesane gjev verdiane til elementa i kvar rad.

```
tabell = linspace(0, 5, 11)
```

Lagar ein 1D-tabell med 11 verdier jamnt fordelt frå 0 og med null til og med fem.

```
a = tabell.flatten()
```

Gjer om en 2D-tabell til ein 1D-tabell. Alle radene i 2D-tabellen blir lagt etter kvarandre til ei lang rad.



Funksjonar i Python

Funksjonar i Python svarer til «lag ein ny kloss» i Scratch, og er eigentleg berre småprogram. Faktisk treng ein funksjon ikkje vere veldig liten heller, men det som skil han frå eit vanleg program, er at han må bli definert før han kan brukast. Når funksjonen er definert, kan vi bruke han så mange gonger vi ønsker utan å skrive all koden på ny, berre med å skrive éi kodelinje.



```
def Funksjon(inndata):  
    for i in range(0, inndata+1, 1):  
        print(i)
```

Funksjon(10)

Desse to programma gjer det same, men her svarar ikkje kvar kloss i Scratch til éi linje i Python. Funksjonen som blir definert, inneheld ei for-lykkje som ikkje er bygd opp på heilt lik måte i Scratch og Python (se for-lykkje-sida).

I Python treng vi ikkje trykke på eit flagg, men vi må legge inn ei linje for å vise variabelen på skjermen. For å kunne sjå alle verdiane på skjermen, har vi lagt inn ein pause på eitt sekund i Scratch-programmet.

Både i Scratch og Python kan funksjonane ta imot innverdiar. Det vil seie at når vi bruker funksjonen, må vi oppgi éin eller fleire verdiar som blir plasserte i variablar. Desse verdiane blir da brukt til noko i funksjonen, gjerne ei utrekning. I dømet heiter variabelen til innverdien i Pythonprogrammet inndata.

Ein viktig skilnad mellom funksjonar i Scratch og Python, er at i Python kan funksjonane gi oss ein utverdi. Det vil seie at vi til dømes kan bruke ein funksjon til å rekne ut eit reknestykke for oss, og gi oss eit tal attende. Dette talet kan vi til dømes sette inn i nye reknestykke, i vissetningar eller i lykkjer. I Scratch går ikkje dette an.

Snakk om

Kvifor må vi ha med at vi skal sette variabelen a til 0 i Scratch-programmet?

Går både Scratch-programmet og Pythonprogrammet gjennom lykkja like mange gonger?

Oppgåve

1. Kva vil bli vist på skjermen for desse tre programma?
Skriv ned kva du trur vil visast på skjermen, og skriv inn programma med Pythonkode etterpå.
Køyr koden, og sjekk om du får det svaret du trudde du skulle få.
2. Dersom du ikkje fekk svaret du trudde, kva var grunnen til det, trur du? Diskuter gjerne med andre.
3. Kan du lage nokre program sjølv? Få ein i klassen til å gisse kva som vil skje i programma dine, eller kan hende læraren?

```
def Funksjon(inndata):  
    return inndata*2  
  
Funksjon(1)
```

```
def Funksjon(inndata):  
    return inndata*2  
  
print(Funksjon(8.5))
```

```
def Funksjon(inndata):  
    return inndata**2  
  
for i in range(1, 11, 1):  
    print(Funksjon(i))
```

Korleis definere og bruke funksjonar – ei oppskrift

```
def navn(variabler):  
    kode som utføres av funksjonen  
    return returverdien til funksjonen
```

navn(variabler)

Teikning i Python - turtle



Python har eit bibliotek som heiter turtle, som kan brukast når ein ønsker å bruke Python til å teikne figurar eller mønster. For å importere dette biblioteket må ein skrive: `from turtle import *` øverst i programmet.

Døme på nokre grunnleggande funksjonar frå turtle-biblioteket

```
t = Turtle()
```

Denne funksjonen lagar eit «turtle-objekt» som ein treng når ein bruker dei andre funksjonane som lagar teikninga. Vi kan tenke på det som ein blyant eller penn.

```
t.forward(100)
t.backward(100)
```

Desse funksjonane flytter turtle fram eller attende 100 steg.

```
t.left(60)
t.right(60)
```

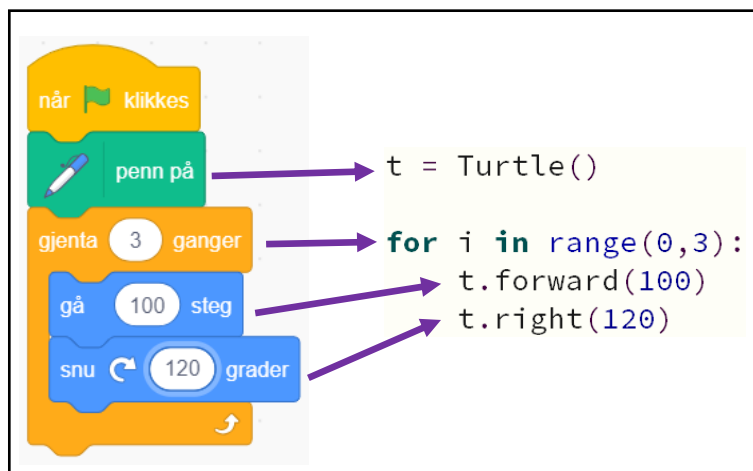
Desse funksjonane snur turtle 120 grader til venstre eller til høgre (det er det same som at den ytre vinkelen er 60 grader).

```
t.circle(50)
```

Denne funksjonen teikner ein sirkel med radius 50 steg.

```
t.goto(100,150)
```

Denne funksjonen får turtle til å flytte seg til punktet med x-koordinat 100 og y-koordinat 150.



Oppgåve

1. Kva vil visast på skjermen for desse programma?

Skriv ned kva du trur vil visast på skjermen, og skriv inn programma med Pythonkode etterpå. Kjør koden, og sjekk om du får det svaret du trudde du skulle få.

2. Dersom du ikkje fekk svaret du trudde, kva var grunnen til det, trur du? Diskuter gjerne med andre.

3. Kan du lage nokre program sjølv? Få ein i klassen til å gisse kva som blir teikna med programma dine, eller kan hende læraren?

```
t = Turtle()
for i in range(0,4):
    t.forward(100)
    t.right(90)
```

```
t = Turtle()
for i in range(0,36):
    t.forward(10)
    t.left(10)
```

```
t = Turtle()
for i in range(0,36):
    t.right(10)
    for i in range(0,36):
        t.forward(10)
        t.left(10)
```

Oversyn over nokre kommandoar for micro:bit i Python Mu

Når vi skal bruke Python for å programmere ein micro:bit, finst det ein del ferdige kommandoar vi kan nytte. Her viser vi eit oversynt over nokre av de mest nyttige. Andre døme kan du finne på denne nettsida: <https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/introduction.html>

Vi må alltid ta til med å importere det biblioteket som inneheld micro:bit-kommandoane. Det gjer ein ved å skrive øvst:

```
from microbit import *
```

Inndata

Knappar

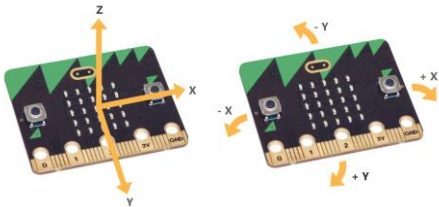
Sidan micro:biten har to knappar, A og B, kan den delen av koden der det står button_a i døma bytast ut med button_b i staden.

`button_a.is_pressed()`
`button_a.get_presses()`

Denne kommandoen tel talet på gonger ein knapp har vore trykt sidan programmet starta.

Innebygde sensorar

Vi kan bruke micro:bitens innebygde sensorar med Python-programmering og. For akselerasjonen kan denne målast i alle dei tre romlige retningane vi har. Tenk på det som eit koordinatsystem med 3 aksar. Dei vanlege, x-aksen og y-aksen, men og ein z-akse som står loddrett på begge dei to andre aksane.



`accelerometer.get_x()`
`temperature()`
`display.read_light_level()`
`running_time()`

Påkopla sensorar – sjå og sida om sonar

Ein digital verdi kan berre vere enten 1 eller 0, der 1 tyder på og 0 tyder av. Den digitale verdien 1 er eigentleg ei høg spenning, medan den digitale verdien 0 er ei veldig låg spenning. Digitale verdiar bruker vi sjelden for sensorar, berre dersom vi skal sjekke om noko er avskrudd eller påskrudd. Sensorar vi koplar på micro:biten gir som regel verdiar mellom 0 og 1023, og da må vi lese av ein analog verdi i staden. Ein analog verdi kan vere mykje anna enn berre 0 eller 1, og passar med det best for å lese av kva for ein verdi sensoren gir.

`pin0.read_analog()`
`pin0.read_digital()`

Utdata

Skrive til skjermen

Det går an å skru på dei einiskilde lysdiodane på skjermen til micro:biten, med verdier mellom 0 og 9 for kor sterkt dei skal lyse.

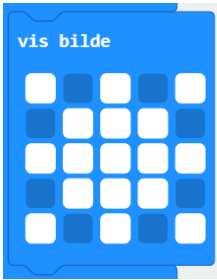
```
display.show()
```



```
display.show(temperature())
```



```
display.show(Image("90909:09990:99999:09990:90909"))
```



Påkopla lysdiodar

Når du koplar på ein lysdiode til micro:biten, kan du programmere han til å skru seg av eller på. Da må ein bruke kommandoen som sender ut digital data, altså verdien 1 eller 0, der 1 svarer til på og 0 svarer til av.

Kva for ein utgang lysdioden er tilkopla (0, 1 eller 2).

```
pin0.write_digital(1)  
pin0.write_digital(0)
```

→ På (1) eller av (0).



Sende data over radio-forbindelsen

På same måten som med å bruke klossar, kan vi programmere micro:bitar til å sende data til kvarandre med å bruke Python.



```
import radio  
radio.on()
```

Import radio-kommandoen importerer biblioteket som gjer at vi kan bruke radio-sambandet mellom micro:bitane. I tillegg må vi bruke radio.on() som skur på radiosambandet.

Til slutt må vi definere kva for ein kanal micro:bitane skal kommunisere på. Da må vi passe på å velge same kanal for alle micro:bitane som skal «snakke saman».



```
radio.config(channel=1)
```



```
x = str(temperature())  
radio.send_bytes(x)
```

Når ein programmerer i Python, må ein gjere om tal til tekst (strenger) før det går an å sende dei over radio-sambandet. Det blir gjort med str()-kommandoen.



```
x = radio.receive_bytes()
```

Denne kommandoen tek imot teksten som blir sendt og legg han i ein variabel kalla x.

Musikk

- med Python Mu



En micro:bit kan enkelt koplatt til ein summer (også kalt buzzer) eller høgtalar for å spele av lyd. Sidan ein summer berre kan spele av ein enkelt tone, vil det bli mykje finare lyd frå ein høgtalar, da han kan spele dei overtonane vi er vane med å høyre som ein del av dei reine tonane i musikk.

For å kunne bruke en høgtalar eller ein summer i Python Mu, må vi importere eit ferdig bibliotek som inneheld metodar for å lage og kontrollere lyd.

```
from music import *
```

Når vi skal spele av ein tone i Python Mu, må vi oppgi tre ting som tekst (strenger). Tonen sitt namn, kva for ein oktav han høyrer til og kor lang tid han skal spelast. Eit piano dekker om lag sju oktav, MakeCode for micro:bit dekker tre oktav og Python Mu dekker ni oktav. Oktav 0 er den lågaste oktaven, som gir dei mørkaste tonane. Oktav nummer fire er den som inneheld kammertonen A med frekvens 440 Hz. I MakeCode tilsvarer låg oktav (liten) oktav nummer tre, midtre oktav (einstroken) oktav nummer fire, og høg oktav (tostroken) oktav nummer fem i Python Mu. Lengda på tonen vert oppgitt som eit tal som svarar til talet på slag multiplisert med 4, der lengda på kvart slag er definert av Mu. Funksjonen under spelar av fire tonar.

```
music.play("C3:4", "03:4", "E3:4", "C#3:8")
```

Namn på tonen Kva oktav tonen tilhøyrer Kor mange slag tonen skal spelast

Dersom vi ønsker å skrive litt mindre, kan play()-funksjonen hugse kva for ein oktav vi bruker og kor mange slag tonen varer. Dette er særst nyttig å bruke når vi skriv heile melodiar, da blir det ikkje fullt så mykje å skrive. Som regel vil ein melodi i hovudsak vere del av berre éin oktav. Når vi nyttar Mus minne, ser kommandoen over slik ut:

```
music.play("C3:4", "D", "E", "C# :8")
```

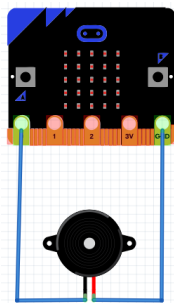
Dersom du vil avgjere sjølv akkurat kor lenge kvar tone skal spelast, kan du bruke ein funksjon som blir kalla pitch(). Da må du oppgi frekvensen til tonen, og ikkje namnet på han. Nokre frekvensar er oppgitt i tabellen i opplegg 11. Speletida blir oppgitt i millisekund, og med det kan du avgjere tempoet (bpm) med stor presisjon.

```
music.pitch(440, 1000, pin1)
```

Frekvensen Varighet i millisekunder Kva for ein utgang høgtalaren (summeren) er kopa til. Denne treng du ikkje skrive noko på dersom du nyttar pin0, da den ligg inne som standardverdi.

Det finnest og ei rekke ferdige melodiar. Du kan finne dei på denne sida: <https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/music.html>

Det har ikkje noko å seie kva utgang på høgtalaren eller summeren du koplar til kva for ein utgang på micro:biten.



Oppgave

Lag eit program som spelar av «Lisa gikk til skolen».

Lag ein eigen melodi som du spelar av på micro:biten.

Logge til PC/MAC med Python Mu

Noko det kan være veldig fint å bruke micro:bit til, er som ein dataloggar. Det vil seie at vi nyttar han til å samle inn data, slik vi har gjort mykje av, og så får vi datamaskinen til å skrive opp målingane på skjermen og lage ein graf av dei medan vi måler.

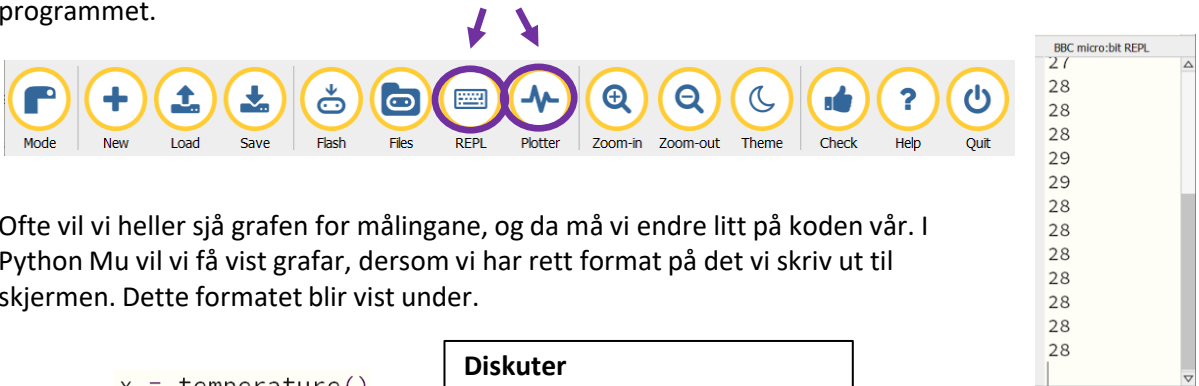
For å vise noko på skjermen bruker vi `print()`-kommandoen. Det som står inni parentesen er det som kommandoen skriv til skjermen, altså viser på skjermen. Dersom vi berre er interesserte i å få ut ei liste med målingar etter kvart som dei blir målte, kan vi berre bruke ein kodesnutt som ser slik ut:

```
x = temperature()
print(x)
```

Måler temperaturen og legg verdien i variabel x.

Skriv ut verdien av variabel x til skjermen.

Etter at vi har ført heile programmet over til micro:biten, må vi trykke på to knappar for å få fram tabellen med målingane. Trykk først på «Plotter», deretter på «REPL». Det er viktig å gjere dette i denne rekkefølga, sjølv om vi ikkje skal vise nokon graf (plottaren viser grafen), elles krasjar programmet.



Ofte vil vi heller sjå grafen for målingane, og da må vi endre litt på koden vår. I Python Mu vil vi få vist grafar, dersom vi har rett format på det vi skriv ut til skjermen. Dette formatet blir vist under.

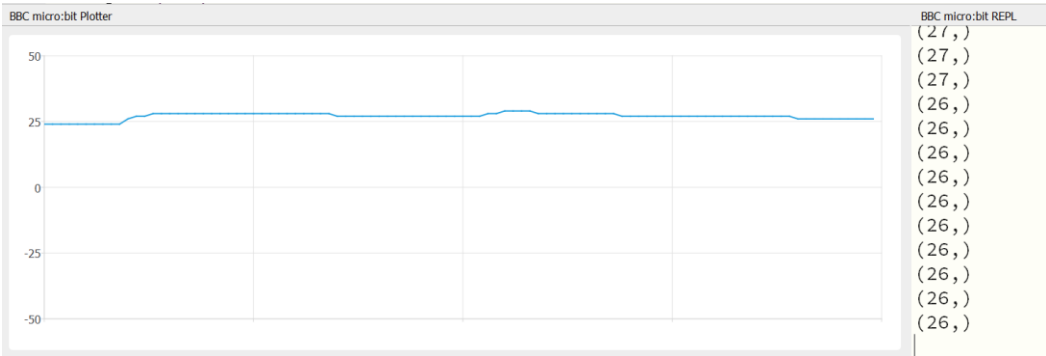
```
x = temperature()
print((x,))
```

Diskuter

Kva er skilnaden på dei to kodesnuttane på denne sida?

Slik ser da lista med målingane ut.

Når det nye programmet er ført over, og du åpnar plottar, deretter REPL, vil vi sjå både grafen og ei liste over verdiane. Denne lista er litt meir tungvint å lese ut frå, men så lenge ein klarer å sjå bort frå ekstra parentesar og komma, står verdiane frå temperaturmålingane som på figuren under.



Oppgåve

Lag heile programmet som måler temperaturen og plottar han som ein graf på skjermen.

Kva skjer om du skriv inn ein annan verdi i staden for eit tomt felt i `print()`-kommandoen?

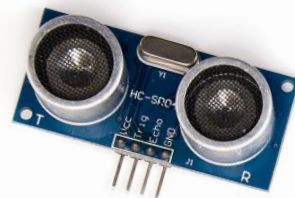
Diskuter

Å logge data kan vi gjere med alle dei innebyggede sensorane til micro:bit, men og med alle sensorar vi kopler på sjølv.

Kva må du endre i koden for å få til dette?

Avstandsmålar (sonar)

- med Python Mu

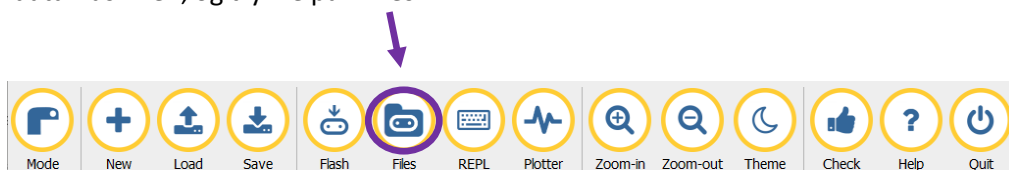


Ein sonar sender ut eit lydsignal, og måler kor lang tid det tek før signalet blir reflektert attende til sonaren. Dersom ein veit kva lydfarten er, kan vi enkelt rekne ut avstanden til det som reflekterer lydsignalet. Sonaren veit kva lydfarten er, og vi kan dermed bruke han til å måle avstandar med. For å kunne bruke ein sonar i Python Mu, må vi importere eit bibliotek på sjølv micro:biten. Dette biblioteket er eigentleg eit program som gjer det enklare for oss å programmere servoen, utan å måtte tenke på alle detaljane i målingane.

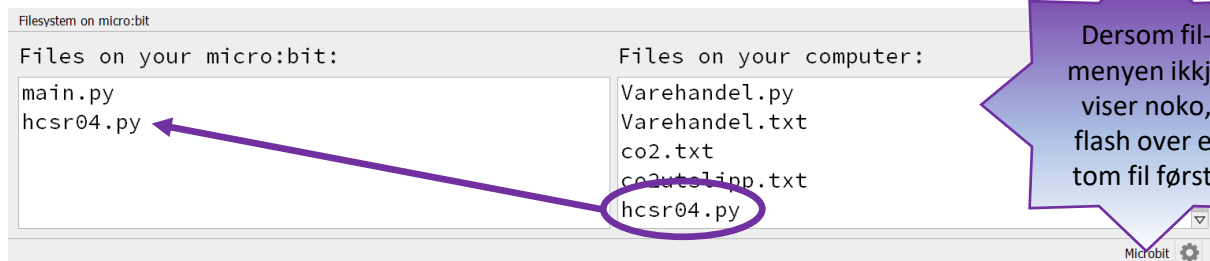
Det første vi må gjere er å få tak i biblioteket til sonaren, og det kan finnast på www.kunnskapsfilm.no.....

Kopier koden som står på sida i ei ny fane (fil) i Python Mu og lagre denne som «hcsr04.py» med å trykke på «Save» og lagre fila i katalogen som dukkar direkte opp.

Så må denne fila leggst over på sjølv micro:biten, og det kan gjerast med å kople micro:biten til datamaskinen, og trykke på «Files».



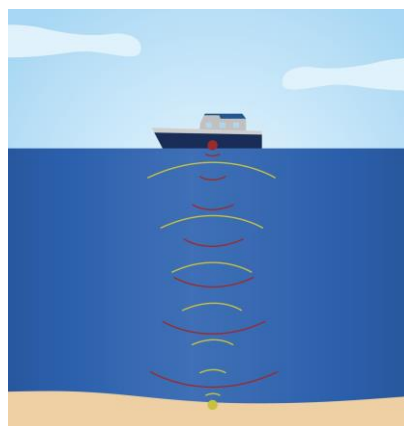
Da kjem denne menyen opp i Mu, og du må leite i feltet på høgre side til du finn fila som du nettopp lagra: «hcsr04.py». Når du finn henne, dreg du henne over i det venstre feltet. Da vil lyset på baksida av micro:biten blenke til programmet er ført over.



Da står det att å lage sjølv programmet som bruker sonaren til å måle avstandar med. Under er det ei puslespill-oppgåve der kodelinjene må ryddast i rett rekkefølge for å lage dette programmet.

Puslespill-oppgave

```
print((HCSR04(),))  
from hcsr04 import HCSR04  
sleep(1000)  
while True:  
    from microbit import *
```



Servo

- med Python Mu



Ein servo mottek eit signal frå micro:biten slik at han rører seg slik vi ønsker. Det finnest to typar servoar, 180° og kontinuerlege (360°), og de blir styrte på litt ulike måtar. 180°-servoen kan røre seg frå ein -90°-vinkel til ein 90°-vinkel, altså 180° til saman. For ein kontinuerleg servo gir vi opp kor raskt han skal snurre rundt i prosent. Sjølve farten kan variere litt med kor mykje straum som er att i batteria, for servoar trekker ganske mykje straum. Så dersom du har ein servo på 100 % fart i byrjinga av skuletimen, vil han snurre ein del saktere på slutten av timen.

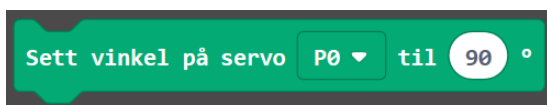
For å kunne bruke ein servo i Python Mu, må vi importere eit bibliotek på sjølve micro:biten. Dette biblioteket er eigentleg eit program som gjer det enklare for oss å programmere servoen, utan å måtte tenke på alle detaljane i signala som styrer han, på same måten som for sonaren.

Det første vi må gjere er å få tak i biblioteket til servoen, og det kan finnast på www.kunnskapsfilm.no.....

Kopier koden som står på sida i ei ny fane (fil) i Python Mu og lagre fila som «servo.py» med å trykke på «Save» og lagre henne i katalogen som dukker direkte opp.

Så må denne fila leggest over på sjølve micro:biten, og det kan gjerast slik som for sonaren på førre sida.

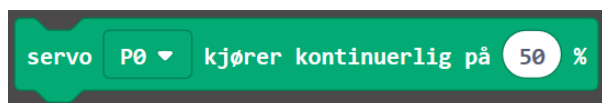
180°-servo



```
servovinkel(pin0, 90)
```

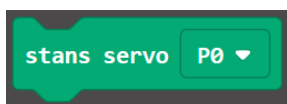
Med å bruke denne kommandoen, kan vi stille inn kva vinkel servoen skal røre seg til. Kommandoen tek inn kva for ein utgang vi har kopla servoen til, og kva vinkel servoen skal røre seg til. Vinkelen kan variere mellom -90° og 90°.

Kontinuerlig servo (360°)



```
servohastighet(pin0, 50)
```

Med å bruke denne kommandoen, kan vi stille inn kva for ein fart servoen skal snurre med. Denne kommandoen tek og inn kva utgang vi har kopla servoen til, i tillegg til farten i prosent. Farten kan variere mellom -100 % og 100 %, der forteiknet avgjer i kva retning servoen skal snurre.



```
servostopp(pin0)
```

Denne kommandoen får ein kontinuerlig servo til å slutte å snurre rundt. Det går og an å bruke kommandoen `servohastighet()` til dette.

Skrive til fil på micro:bit

- med Python Mu



I slutten av kapittel 9 brukte vi datasett i form av tekstfiler for å lage matematiske modeller. Hadde det ikke vore kult om vi kunne fått micro:biten til å lage slike filer av målingene sine? Da kan vi bruke micro:biten til å samle inn data over ein viss periode, utan å vere kopla til ein datamaskin. Datasettet blir lagra i micro:biten som ei .txt-fil, og vi kan legge fila over i datamaskinen, før vi plottar resultata eller lager ein matematisk modell med Python.

Det aller første vi må gjere, er å opprette ei tekstfil, det vil seie ei fil som sluttar på .txt, og det er enklast å gjere i Mu. Lagre fila i same mappe som resten av Python Mu-filene dine.

Deretter må den fila du har laga, kopierast over på micro:biten. Dette kan du gjere på same måten som for sonaren.

```
with open ('filnavn.txt', 'w') as fil:
```

Åpner fila som heiter filnavn.txt og tildeler henne til eit objekt som vi kallar fil. Sidan vi har 'w' (write) i kommandoen, blir det gjort klart til å skrive inn i fila.

```
with open ('filnavn.txt', 'r') as fil:
```

Tilsvarende som linja over, men sidan vi har 'r' (read) i kommandoen, gjerast det klart til å lese fila. Det er valfritt å ta med 'r'.

```
fil.write(y+'\n')
```

Skriv variabelen y til fila vår. '+n' gjer at det blir eit linjeskift etter kvart tal, og det må vi ha for at python skal kunne lese filene. Må stå med innrykk.

```
y = str(x)
```

Gjer om variabelen x til ein streng (tekstverdi). Må stå med innrykk.

```
z = fil.read()
```

```
print(z)
```

Skriv variabelen z til skjermen på PC. Må IKKJE stå med innrykk.

Les det som står i fila vår, og legg det inn i variabel z. Må stå med innrykk.

Da står det att å lage sjølv programmet. Under er eit døme der micro:biten måler temperaturen og legg målinga i eim.txt-fil ved namn datasett.txt. Det kan vere greitt å kunne sjekke korleis datasett.txt-fila ser ut utan å overføre til datamaskinen og åpne henne, difor er dette og med i puslespel-oppgåva.

Puslespel-oppgåve

```
print(innhold)          t = temperature()
while not button_a.is_pressed():
    sleep(1000)          fil.write(str(t)+'\n')
with open('datasett.txt') as fil:
with open('datasett.txt', 'w') as fil:
    innhold = fil.read()  from microbit import *
```

Ekstraoppgåve

Lag eit program som bruker ein sensor du koplar på micro:biten for å samle inn data som du lagrar i ei fil på micro:biten. Kva må du endre på?

Kva må du endre i programmet for å lage ei fil med to kolonner der den første kolonna er kor lang tid det har gått før målinga er gjort?

Kopling B

Programmering og skaperverkstad for 8. - 10. klasse

Kapittel 13

Modellering og maskinlæring med Python



Maskinlæring – kva er det?

Maskinlæring er ein del av fagområdet kunstig intelligens, men det er mindre avansert enn ein skulle tru. Den mest grunnleggande formen for maskinlæring er å lage matematiske modellar med hjelp av enkle regresjonar slik som de tidlegare har gjort i Geogebra.

Etter kvart kan ein bruke meir og meir avanserte modellar, og gjerne system som er samansette av mange modellar som heng saman med kvarandre. Eit døme på eit slikt gigantisk system er vermodellering. Der blir det henta inn alle slags data om temperaturar, vindstyrke og lufttrykk frå mange stader i verda. Desse dataa blir nytta for å lage matematiske modellar, altså ulike matematiske funksjonar, slik de har gjort tidlegare. Vidare kan meteorologane sette inn nye verdier i funksjonane sine for å rekne ut kva slags ver det er størst sannsyn for at det blir i ulike område på eit seinare tidspunkt. Korleis heng dette saman med kva område som er gyldig og kva som er usikkert i målingar?

Mykje modellering skjer utan at menneske aktivt programmerer alle delane, programma har «fått lov til» å lage sine egne matematiske modellar grunna på ulike datasett. Dette kan gjere det vanskelegare å vite kva som ligg bak avgjerder som desse dataprogramma gjer, da ein del av algoritmane kan vere vanskeleg å få tilgang til. På ein måte er det fint at maskinar kan gjere dette for oss, på den andre sida kan det til dømes kunne føre til problem med diskriminering.

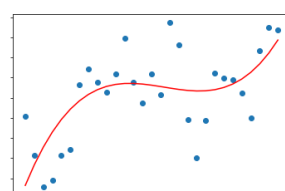
Diskriminering på grunn av maskinlæring har ein sett fleire døme på. Dersom eit maskinlæringsprogram for ansiktsattkjenning får tilgang til datasett som nesten berre er samansett av kvite menn, vil dette programmet bli mykje dårlegare på å kjenne att ansikta til kvinner og ikkje-kvite menneske. Dette har ein kunna merke på enkelte mobilmerke sin ansiktsattkjenning for å låse opp telefonen. Det er og funne døme på diskriminering i tilsettingar, til dømes der eit maskinlæringsprogram føreslo Amazon å heller tilsette menn enn kvinner, bygd på inndata om at det var veldig mange fleire menn enn kvinner tilsette der i utgangspunktet. Les gjerne meir her: <https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scrap-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G>



Frå data



Via regresjon



Til modellar

Lage graf med Python

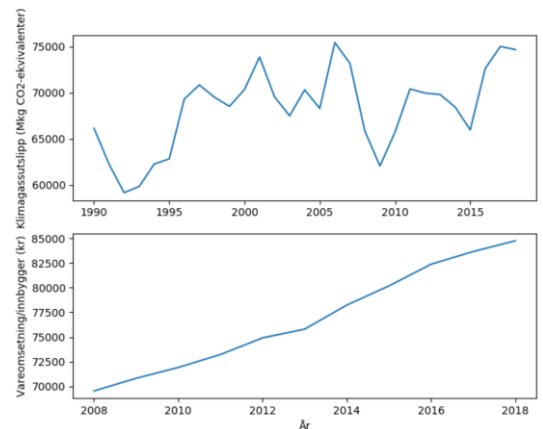


Kvifor Python?

De kan alt teikne grafar med Geogebra, så kva er poenget med å bruke Python i staden?

I Geogebra kan vi plote ein funksjon, eller legge inn punkt og få plotta ein graf frå punkta. Sidan vi må skrive inn alle punkta for hand, kan det vere tungvint dersom vi har veldig mange målingar. Dette gjeld særst om vi har ferdige filer med mange målingar. I Python kan ein lage sub-plot, det vil seie bilete med fleire grafar som viser ulike målingar i same bilete. Men hovudgrunnen til å bruke Python til plotting av grafar, er at det er byrjinga på å bruke statistikk for å analysere data. Det er slik forskarar jobbar når dei har samla inn data!

I Python kan du bruke data som ligg i ferdige filer. Du slepp å skrive dei inn! Det som er viktig, er at filene er rett type. Filene kan berre innehalde tal og lagrast i tekstformat (.txt) for dei døma vi ser på her.



Plotting av grafar i Python er samansett av seks deler:

1. Importere pylab-biblioteket i programmet
2. Importere data frå ein .txt-fil
3. Legge dataa i ein tabell (array) for x-verdiane og ein for y-verdiane
4. Skrive kva for nokre data du skal plote
5. Skrive inn tittel og merkelappar på aksane
6. Få plottet til å visast på skjermen

I tillegg kan du skrive inn korleis grafen skal sjå ut med til dømes farge, linjestil og kor tjukk linja skal vere.

`from pylab import *` → Importerer det naudsynte biblioteket

`plot(x, y)` → Lagar eit plott av alle verdiane i tabellane x og y

`show()` → Viser figuren av plottet på skjermen

`tabell = loadtxt("data.txt")` → Importerer fila data.txt og legg henne inn i ein tabell med namn tabell

`x = tabell[:,0]`
`y = tabell[:,1]` → Plukkar alle x-verdiane frå tabellen og legg dei i ein tabell som heiter x. Tilsvarende for y-verdiane.

`title("Tittel")`
`xlabel("x-verdier")`
`ylabel("y-verdier")` → Lager tittel og merkelappar til aksane.

Oppgåver

1. Bruk kodelinjene på venstre side for å lage eit program som plottar ein graf frå ei fil som heiter data.txt.
2. Skriv ein omtale for kvar del av programmet ditt utan å sjå på denne sida. Legg dei gjerne inn som kommentarar i koden din. Sammenlikne så omtalane dine med dei på denne sida. Kva for nokre skilnader fann du?
3. Finn eit valfritt datasett frå statistisk sentralbyrå, og bruk Python til å lage ein graf av dataa. Kva må du endre i dømet på denne sida?

Lineær regresjon med Python



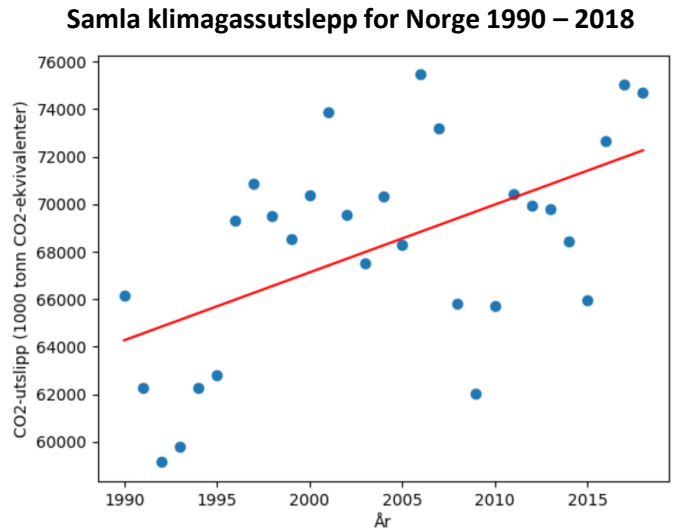
Kvifor Python?

De har at gjort mange lineære regresjonar med Geogebra, så kva er poenget med å bruke Python i staden?

I Geogebra må vi skrive inn alle målingane våre for hand, eller kopiere frå til dømes Excel. Da kan det vere vanskeleg å gjere ein regresjon dersom vi har mykje data. Tenk deg at du har 100 målingar, da tek det veldig lang tid å skrive inn alle målingane i reknearket i Geogebra. Det kan fort bli ein feil eller to undervegs og. Tenk om du har endå fleire målingar, kan hende fleire enn 1000! Da blir det ganske kjedeleg å skrive inn alle tala.

I Python kan du bruke data som ligg i ferdige filer. Du slepp å skrive dei inn! Det som er viktig, er at filene er rett type. Slik som med plotting av grafar i Python.

Ein annen fordel med å bruke Python, er at det er eit programmeringsspråk som blir nytta av forskarar som jobbar med modellering. Andre forskararere kan bruke andre programmeringsspråk som er spesialtilpassa akkurat den modelleringen dei jobbar med, men de følger dei samme prinsippa som Python. Så dersom du kan modellere med Python, er det enklare å modellere i eit anna programmeringsspråk enn om du berre har brukt Geogebra tidlegare.



Regresjon i Python er samansett av seks delar:

1. Importere dei biblioteka som trengst
2. Importere data frå ei .txt-fil
3. Legge dataa i ein tabell for x-verdiane og ein for y-verdiane
4. Lage ein programmeringsfunksjon som seier at vi skal bruke ein lineær modell i regresjonen
5. Utføre regresjonen, og vise på skjermen kva den matematiske modellen blir, stigningstalet (a) og skjæringspunktet med y-aksen (b)
6. Teikne grafen med resultatet – tilpass koden for plotting av graf
 - Plotte punkta med eit scatterplot
 - Plotte regresjonsgrafen – korleis få programmet til å rekne ut y-verdiane for modellen?

Puslespeloppgåve

Sorter kodelinjene etter nummera i tekstboksen over, om regresjon i Python. Da vil du få den rette rekkefølgen i programmet ditt.

- | | |
|---|---|
| 1. <code>co2data = loadtxt("co2utslipp.txt")</code> | 5. <code>x_verdier = co2data[:,0]</code>
<code>y_verdier = co2data[:,1]</code> |
| 2. <code>from pylab import *</code>
<code>from scipy.optimize import curve_fit</code> | 6. <code>def funksjon(x, a, b):</code>
<code> return (a * x) + b</code> |
| 3. <code>scatter(x_verdier, y_verdier)</code>
<code>plot(x_verdier, funksjon(x_verdier, a, b), color="red")</code>
<code>title("Samlet klimagassutslipp for Norge 1990 - 2018")</code>
<code>xlabel("År")</code>
<code>ylabel("CO2-utslipp (1000 tonn CO2-ekvivalenter)")</code>
<code>show()</code> | |
| 4. <code>koeffisienter, kovarianser = curve_fit(funksjon, x_verdier, y_verdier)</code>
<code>a, b = koeffisienter</code>
<code>print(a,b)</code> | |

Ikkje-lineær regresjon med Python

Skilnad på lineær og ikkje-lineær regresjon i Python

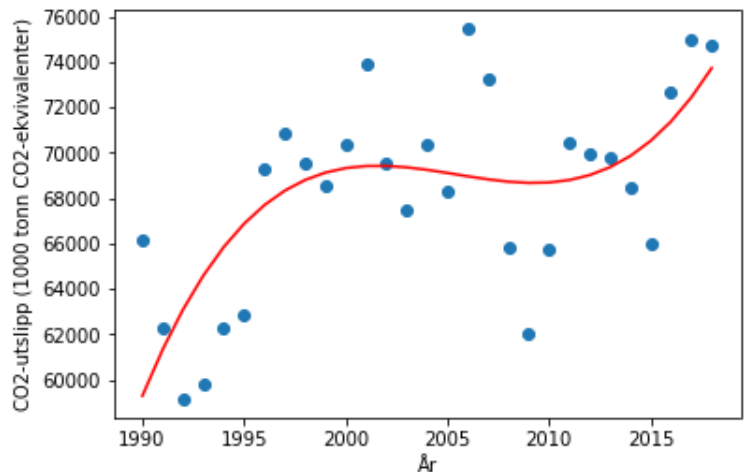
Det er liten skilnad på korleis programma for lineær og ikkje-lineær regresjon ser ut. Den viktigaste skilnaden er å endre på kva for ein datasettet skal tilpassast. Det står eit oversyn over nokre funksjonstypar, og korleis ein skriv dei i python, på dei to neste sidene.

Men korleis skal vi vite kva for ein funksjonstype som passar til datasettet vårt? Går det an å finne det ut på førehand? Svaret på det spørsmålet kjem an på datasettet ditt. Du kan lage eit

scatterplot av datasettet ditt, og sjå om du kjenner att ein av funksjonstypene. Som regel er det likevel vanskeleg å gisse seg til ein funksjonstype på førehand, og vi kan prøve oss fram med ulike funksjonstypar.

Programmeringa er veldig lik for lineær og ikkje-lineær regresjon. Vi må berre endre den lineære funksjonen til en ikkje-lineær funksjon. Da vil modellen vår tilpasse seg datasettet vårt best mogleg, og vi får vite koeffisientane som avgjer funksjonen for modellen vår. I tillegg må vi endre til rett tal koeffisientar i programmet vårt. Sjå dømet under på kva som må endrast.

Samla klimagassutslepp for Norge 1990 – 2018



```
def funksjon(x, a, b): → def funksjon(x, a, b, c, d):  
    return (a * x) + b      return (a*x**3)+(b*x**2)+(c*x)+d
```

```
koeffisienter, kovarianser = curve_fit(funksjon, x_verdier, y_verdier)  
a, b = koeffisienter  
print(a,b)
```

```
koeffisienter, kovarianser = curve_fit(funksjon, x_verdier, y_verdier)  
a, b, c, d = koeffisienter  
print(a,b,c,d)
```

```
scatter(x_verdier, y_verdier)  
plot(x_verdier, funksjon(x_verdier, a, b), color="red")  
title("Samlet klimagassutslepp for Norge 1990 – 2018")  
xlabel("År")  
ylabel("CO2-utslepp (1000 tonn CO2-ekvivalenter)")  
show()
```

```
scatter(x_verdier, y_verdier)  
plot(x_verdier, funksjon(x_verdier, a, b, c, d), color="red")  
title("Samlet klimagassutslepp for Norge 1990 – 2018")  
xlabel("År")  
ylabel("CO2-utslepp (1000 tonn CO2-ekvivalenter)")  
show()
```

Vi nytter data frå statistisk sentralbyrå over klimagassutslepp i Norge for åra 1990 – 2018, og ønsker å lage ein matematisk modell over utviklinga over tid. Vi prøver ut ulike funksjonstypar i regresjonsprogrammet, og ser kva for nokre modellar vi får.

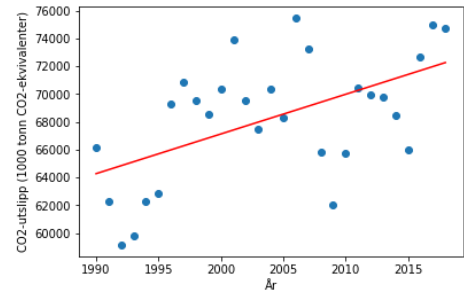
Vi bruker ein lineær modell i programmet vårt

```
def funksjon(x, a, b):
    return (a * x) + b
```

Programmet gir oss den lineære funksjonen

$$f(x) = 286x - 504059$$

som den beste tilpassinga til datasettet vårt.



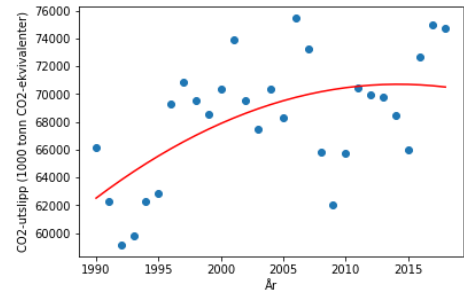
Vi bruker ein andregradsmodell i programmet vårt

```
return (a*x**2)+(b*x)+c
```

Programmet gir oss andregradsfunksjonen

$$f(x) = -14x^2 + 56298 - 56627608$$

som den beste tilpassinga til datasettet vårt.



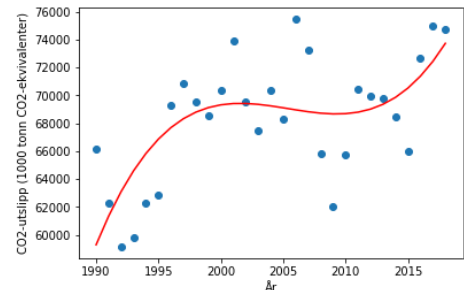
Vi bruker ein tredjegradsmodell i programmet vårt

```
return (a*x**3)+(b*x**2)+(c*x)+d
```

Programmet gir oss tredjegradsfunksjonen

$$f(x) = 3x^3 - 19753x^2 + 39612328x - 26479502215$$

som den beste tilpassinga til datasettet vårt.



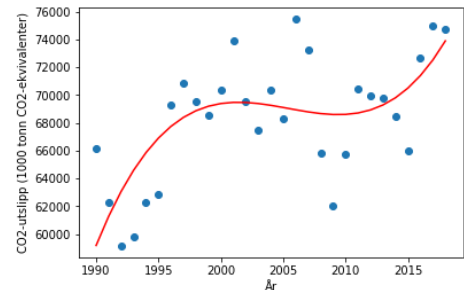
Vi bruker ein fjerdegradsmodell i programmet vårt

```
return (a*x**4)+(b*x**3)+(c*x**2)+(d*x)+e
```

Programmet gir oss fjerdegradsfunksjonen

$$f(x) = 0,004x^4 - 30x^3 + 79206x^2 - 92029574x + 39188643912$$

som den beste tilpassinga til datasettet vårt.



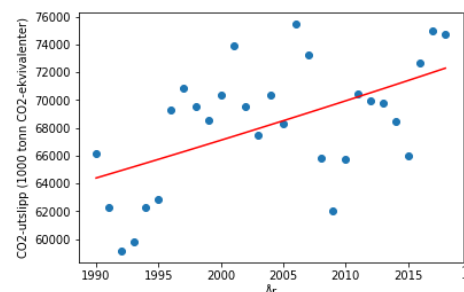
Vi bruker ein eksponentiell modell i programmet vårt

```
return a * (b**x)
```

Programmet gir oss eksponentialfunksjonen

$$f(x) = 17 \cdot 1,004^x$$

som den beste tilpassinga til datasettet vårt.



Vi bruker ein potensmodell i programmet vårt

```
return a*(x**b)
```

RuntimeError: Optimal parameters not found: Number of calls to function has reached maxfev = 600.

Når vi prøver å lage ein potens-modell for CO₂-dataa våre, får vi det ikkje til. For nokre datasett vil nokon funksjonstypar passe veldig dårlig. Da kan det hende at vi får ei feilmelding i programmet, om at vi har brote ei tidsavgrensing. Da har det teke programmet for lang tid å tilpasse modellen for oss, og vi får ikkje noko resultat. Det kan vere eit teikn på at modellen (funksjonstypen) vi har valt passer dårleg med datasettet vårt.

Vi ser på nokre andre døme med eit datasett for kumulativt tal Covid 19-smitta i Norge for veke 8 til veke 32 i 2020.

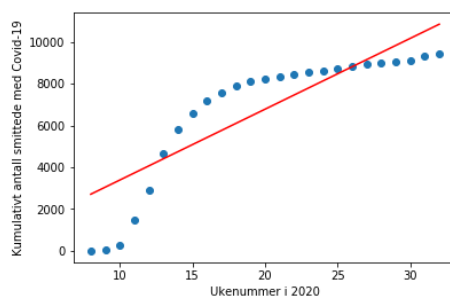
Vi nyttar ein potensmodell i programmet vårt

```
return a*(x**b)
```

Programmet gir oss potensfunksjonen

$$f(x) = 0,800 \cdot x^{1,00}$$

som den beste tilpassinga til datasettet vårt.



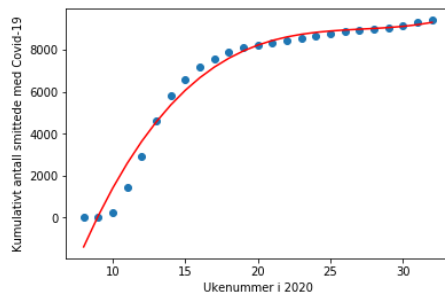
Vi nyttar ein tredjegradsmodell i programmet vårt

```
return (a * x**3) + (b * x**2) + (c * x) + d
```

Programmet gir oss tredjegradsfunksjonen

$$f(x) = 1,37x^3 - 112x^2 + 3080x - 19578$$

som den beste tilpassinga til datasettet vårt.



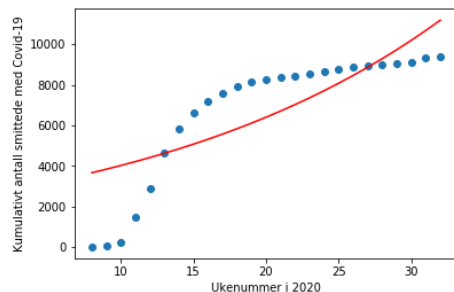
Vi nyttar ein eksponentialmodell i programmet vårt

```
return a * (b**x)
```

Programmet gir oss eksponentialfunksjonen

$$f(x) = 2521 \cdot 1,048^x$$

som den beste tilpassinga til datasettet vårt.



Nå veit vi korleis vi kan bruke data for å tilpasse mange ulike funksjonstypar. Men korleis skal vi vite kva for ein modell som passar best med datasettet vårt? Må vi berre sjå kva for ein funksjonstype som ser ut til å passe best med datasettet vårt, eller kan vi gjere det på ein meir objektiv måte? Bla om til neste side for å finne ut meir.



Korrelasjon i Python

Når vi skal rekne ut korrelasjonen, er det som regel Pearson sin korrelasjonskoeffisient vi meiner, og den blir kalla R . Det er og denne vi skal bruke her. Vi er mest interesserte i denne korrelasjonskoeffisienten opphøgd i andre, altså R^2 , fordi den verdien seier kor stor andel av variasjonen i målingane/datasettet vårt som heng direkte saman. Verdien kan vere mellom null og ein. Jo høgare verdi, desto større del av variasjonen i y kan forklarast av variasjonen i x . R^{2b} blir oppgitt som eit desimaltal, men multipliserer vi det med 100 %, får vi svaret i prosent.

Den enklaste forma for korrelasjon er mellom x og y -verdiane i eit datasett. Da finn vi kor stor del av variasjonen i y som kan forklarast av variasjonen i x . Dette blir gjort med at Python gjer ein lineær regresjon, slik at R^2 viser kor god den lineære samanhengen mellom x og y -verdiane er. Da er R^2 eit mål på kor godt den lineære funksjonen passar med datasettet vårt.

Det går og an å gjere tilsvarande for ikkje-lineære regresjonar. Da får vi et objektivt mål på kva for ein modell som passar best med datasettet vårt. Dersom ein vil finne R^2 for ein regresjon der $f(x)$ er ein eksponentialfunksjon, tek vi til med å utføre sjølve regresjonen. Når han er ferdig, veit vi kva koeffisientane i eksponentialfunksjonen er, altså veit vi kva $f(x)$ er. Da bruker vi den estimerte funksjonen til å putte inn x -verdiene frå datasettet vårt, for så å rekne ut dei tilsvarande $f(x)$ verdiane. For å finne R^2 bruker vi y -verdiane frå datasettet vårt og $f(x)$ -verdiane som vi rekna ut ved å bruke den eksponentielle modellen vi fann. Da gjeld framleis det at den modellen som gir ein verdi nærmast $R^2 = 1,00$ er den beste modellen.

Korrelasjon i Python er samansett av fem delar:

1. Importere biblioteka som trengst
2. Importere data frå ei .txt-fil
3. Legge dataa i ein tabell (array) for x -verdiane og ein for y -verdiane
4. Rekne ut korrelasjonskoeffisienten
5. Vise korrelasjonskoeffisienten på skjermen

Datasetta vi bruker, kan vi til dømes hente frå statistisk sentralbyrå. Korleis ein gjer dette står i opplegg 26. Eller ein kan nytte ein micro:bit for å samle inn data. Dersom vi har mange målingar, er det fint å lagre det som ei fil på micro:biten. Korleis dette kan bli gjort står i kapittel 12.

```
from pylab import *  
from scipy.stats import pearsonr  
  
data = loadtxt("filnavn.txt")  
  
x = data[:,0]  
y = data[:,1]  
  
R = pearsonr(x, y)[0]  
  
print(R)  
print(R**2)
```

Importerer dei naudsynte biblioteka.

Legg inn data frå fil i ein tabell.

Legg x -verdiane frå første kolonne inn i ein tabell og y -verdiene frå den andre kolonnen i ein tabell.

Reknar ut korrelasjonskoeffisienten.

Skriv korrelasjonskoeffisienten og R^2 til skjermen.

R^2 svarer til R^2 -verdien i Geogebra.

Oppgåver

1. Finn eit valfritt datasett frå statistisk sentralbyrå, og bruk Python til å finne korrelasjonen for datasettet. Kva må du endre i dømet på denne sida?
2. Korleis kan ein rekne ut korrelasjonen mellom to ulike datasett? Kva for nokre kriterium må vere oppfylte for at det skal kunne gå an?

Val av regresjonsmodell i Python



Vi kan ikkje *berre* sjå på R^2 når vi skal velge modell (velge mellom dei ulike funksjonstypene). Vi bør nemlig ha ein idé ut frå teori, om kva type modell som burde fungere best. Det er til dømes slik at ein polynomfunksjon vil passe betre og betre, desto høgare grad han er. Det vil sei at ein fjerdegradsfunksjon alltid vil passe like godt eller betre enn en tredjegradsfunksjon. Eller at ein andregradsfunksjon alltid vil passe like godt eller betre enn ein førstegradsfunksjon, som er det same som ein lineær funksjon.

Regresjon med korrelasjon i Python er samansett av sju delar:

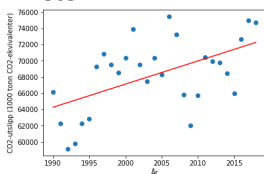
1. Importere biblioteka som trengst
2. Importere data frå ei .txt-fil
3. Legge dataa i ein tabell for x-verdiane og ein for y-verdiane
4. Lage ein funksjon som seier kva for ein modell som skal brukast i regresjonen
5. Gjere regresjonen, og vise på skjermen kva den matematiske modellen blir
6. Teikne grafen med resultatet
7. Rekne ut R og R^2 og skrive dei ut til skjermen

```
from pylab import *  
from scipy.optimize import curve_fit  
from scipy.stats import pearsonr
```

```
R = pearsonr(y_verdier, funksjon(x_verdier, a, b, c))[0]  
print(R)  
print(R**2)
```

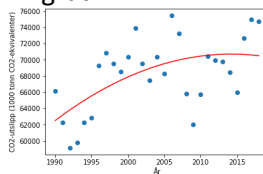
Nå kan vi sjekke kva for ein matematisk modell som skildrar datasetta våre best. Vi sjekker kva for ein modell som passar best for både CO₂-utsleppa og kumulativt tal på covid-19 smitta. Sjekk om du får same verdiar for dei ulike modellane.

Lineær



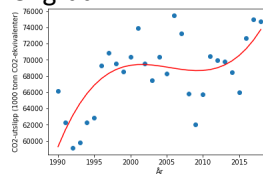
R = 0,5428
 $R^2 = 0,2947$

2. grad



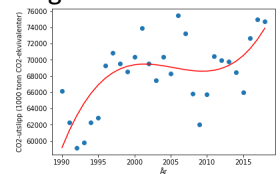
R = 0,5780
 $R^2 = 0,3340$

3. grad



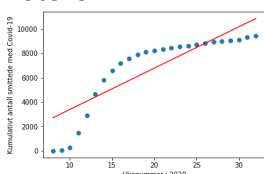
R = 0,6695
 $R^2 = 0,4483$

4. grad



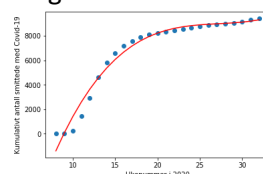
R = 0,6710
 $R^2 = 0,4502$

Potens



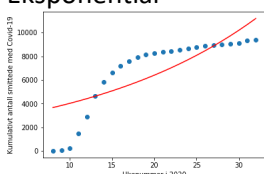
R = 0,8730
 $R^2 = 0,7620$

3. grad



R = 0,9869
 $R^2 = 0,9740$

Ekspontial



R = 0,8004
 $R^2 = 0,6407$

Diskusjonsoppgåver

1. Kva for ein modell skildrar CO₂-utsleppa best? Grunngi svaret.
2. Kva for ein modell skildrar kumulativt tal på covid-19 smitta best? Grunngi svaret.
3. Kan det potensielt finnast andre modellar som skildrar datasetta våre betre?
4. Dersom du har laga eit program som alt utfører ein regresjon, kva må du legge inn for at det og skal finne og skrive ut R og R^2 ?