

Kobling B - Kreativitet og skaperglede

Lærerveiledning



Innhold

• Tips til bruk av læreverket.....	3
• Rekkefølge/nivå på opplegg.....	4
• Oversikt over alt innhold i elevbok.....	5

Del 1 - Introduksjon til programmering (hører til elevbok kapittel 1)

• Programmering uten datamaskin.....	9
• Introduksjon til noen viktige begreper.....	11
• Funksjoner.....	12
• Miniutfordringer.....	16

Del 2 – Opplegg 1 – 35

• Opplegg 1 – Innovasjonsmetoden.....	22
• Opplegg 2 – Celler.....	25
• Opplegg 3 – Drivhuseffekten.....	28
• Opplegg 4 – Energiformer og –overganger.....	31
• Opplegg 5 – Massetetthet og periodesystemet.....	34
• Opplegg 6 – Friksjon og omforming av energi.....	37
• Opplegg 7 – Solenergi.....	40
• Opplegg 8 – Formler, likninger og fart.....	45
• Opplegg 9 – Rytme.....	47
• Opplegg 10 – Lyd og lydbølger.....	49
• Opplegg 11 – Toner og frekvens.....	53
• Opplegg 12 – Tegne geometriske former med programmering.....	56
• Opplegg 13 – Sirkler, kuler og naturressurser.....	58
• Opplegg 14 – Volum for ulike geometriske former.....	62
• Opplegg 15 – Matematiske mønster.....	64
• Opplegg 16 – Sannsynlighet og representasjoner.....	69
• Opplegg 17 – Uniform sannsynlighet.....	72
• Opplegg 18 – Antall mulige kombinasjoner.....	76
• Opplegg 19 – Sannsynlighet og valgtre.....	79
• Opplegg 20 – Statistikk og sentralmål.....	84
• Opplegg 21 – Frekvenstabell og sektordiagram.....	89
• Opplegg 22 – Statistikk og diagrammer.....	93
• Opplegg 23 – Statistikk og «fake news».....	96
• Opplegg 24 – Statistikk og spredningsmål.....	99
• Opplegg 25 – Statistikk i naturfag og samfunnsfag.....	105
• Opplegg 26 – Laste ned data fra statistisk sentralbyrå.....	108
• Opplegg 27 – Personlig økonomi.....	113
• Opplegg 28 – Statistikk og korrelasjon.....	117
• Opplegg 29 – Varme, isolasjon og energieffektivitet.....	122
• Opplegg 30 – Populasjoner og populasjonsvekt.....	128
• Opplegg 31 – Vaksiner og immunforsvar.....	135
• Opplegg 32 – Forbrenningsreaksjoner.....	138
• Opplegg 33 – Geriljabroderi.....	142
• Opplegg 34 – Lag en teknologisk 17.mai-fane.....	144
• Opplegg 35 – Lag et produkt som bruker hvis-klosser.....	148

Tips til bruk av læreverket

Oppleggene i elevboka inneholder fasene i Innovasjonsmetoden og oppgavene til elevene. For hvert opplegg står det nyttige tips til gjennomføring og utstyrsbehov.

Løsningsforslagene skal være en veiledning for programmeringsdelen av aktivitetene. De skal ikke deles ut til elevene, bruk dem heller selv for å løse elevene trygt gjennom oppgaven dersom de står helt fast. Du kan også bruke løsningsforslaget til å gi elevene hint, men det bør sitte veldig langt inne å dele ut hele forslaget.

Variasjon 1 passer for elever som trenger ekstra utfordringer, og læreren må beregne tid til å veilede elevene.

Variasjon 2 er mer krevende enn variasjon 1, og er beregnet på elever med stort læringspotensial og krever mer enn pensum for ungdomstrinnet. Disse er det fullt mulig å gjennomføre, men selv elever med stort læringspotensial vil trenge grundig veiledning for en del av dem.

Kopioriginaler er gjerne tips som gjør oppgavene enklere for elevene å gjennomføre. Noen er til utfylling. Det er lov å skrive ut, og dele ut hele eller deler av både elevbok og lærerveiledning, da dette er et ikke-kommersielt produkt. Men forfatter må oppgis, og materialet kan ikke utgis for å være egenprodusert.

Viktig!

Introduksjon og grunnleggende arbeid med fagstoffet i boka anbefales å gjøre i forkant av skaperverksted-oppleggene. Elevene bør ha en grunnleggende kjennskap til de ulike temaene de skal jobbe med. Dette gjelder spesielt de oppleggene der det er mange nye begreper og ny informasjon.

Forslag til teknologisk utstyrsbehov for én klasse (en skole kan gjerne ha dobbel mengde)

Det er beregnet at elevene jobber gruppevis med gruppestørrelse på 2 – 4 elever.

30 micro:bit

60 alligatorledninger

100 motstander (gjern 5 ulike typer)

100 LED-pærer (hvit, gul, rød, grønn, blå)

15 + 15 servoer (180° + 360°)

15 koblingsbrett til servoer

30 summere (buzzere) og/eller høyttalere

15 sonarer (3V)

15 lyssensorer

15 temperatursensorer

Ledende sytråd

I tillegg til materialene for hvert oppslag er det beregnet at alle elevene har tilgang til en digital enhet hver.

GAN Aschehoug selger klassesett – se nettsider for mer informasjon:

Rekkefølge/nivå på oppleggene

Hvert undervisningsopplegg består av to sider. Den første siden er et sammendrag av aktuelt fagstoff med diskusjonsoppgaver, spørsmål og andre oppgaver. Den andre siden inneholder selve skaperverkstedopplegget, med oversikt over fasene i innovasjonsmetoden. Dersom man skriver ut oppleggene til elevene, kan det være en fordel å printe begge A4-sidene på samme side av et A3-ark, slik at dette kan ligge åpent på pulten mens elevene jobber.

Stort sett kan boka jobbes gjennom slik rekkefølgen er satt opp. Rekkefølgen er satt opp med tanke på progresjon i programmering i tillegg til i de enkelte fagene. I matematikken er kompetansemålene gitt for hvert enkelt skoleår, dermed har vi valgt å ha de mest matematiske undervisningsoppleggene skilt ut og ordnet etter hvilket trinn kompetansemålene i hovedsak hører til under.

Undervisningsoppleggene som ikke er med i matematikkdelen er mer frie med tanke på hvilket trinn de passer på. Dette vil nok variere alt ettersom hvilken rekkefølge man underviser i de forskjellige naturfagstemaene o.l. Vi har prøvd å lage en tilpasning der de enkleste oppgavene kommer først i boka, dette spesielt med tanke på nivået på programmeringsoppgavene. Generelt er de første undervisningsoppleggene i boka enklest med tanke på programmering. Dermed kan det være greit å gjennomføre noen av disse før man eventuelt går til matematikk-oppleggene på 8.trinn. Generelt er programmeringsoppgavene med micro:bit noe enklere enn programmeringsoppgavene med Scratch/Python.

Det finnes alternative programmeringsoppgaver for alle oppleggene slik at det er mulig å benytte klossbasert eller tekstbasert programmering for alle oppleggene. Det finnes og alternative oppgaver med større/mindre vanskegrad, i denne lærerveiledningen. Bruk gjerne disse som differensiering til ulike elever, det går også an å redigere direkte i oppleggene, og bytte ut tekst, oppgaver o.l. Dette anbefales å gjøre for å tilpasse opplæringen mest mulig.

Oversikt over elevbok:

Kapittel 1 – Generell introduksjon til programmering

Kapittel 2 – Naturfagsopplegg

Kapittel 3 – Kort introduksjon til modellering med Geogebra (lineær og ikke-lineær)

Kapittel 4 – Lineær modellering i naturfag og matematikk (8. trinn)

Kapittel 5 – Musikkopplegg

Kapittel 6 – Geometri, matematiske mønstre og modellering (9.trinn) og naturfag

Kapittel 7 – Sannsynlighet (9.trinn) og naturfag

Kapittel 8 – Statistikk (9.trinn) og naturfag

Kapittel 9 – Modellering og statistikk (10.trinn) (og naturfag)

Kapittel 10 – Naturfag og ikke-lineær modellering (10.trinn)

Kapittel 11 – Kunst og håndverksopplegg

Kapittel 12 – Introduksjon til Pythonprogrammering

Kapittel 13 – Modellering med Python (for de spesielt interesserte)

Det finnes en norsk betaversjon av Scratch som gjør det mulig å skjære ut tegninger fra Scratch direkte på vinylkuttere og laserkuttere. Den er under utprøving og kan finnes her:

https://kristianpedersen.github.io/?fbclid=IwAR2a5s1eXKR3SK2tS0MqJLu_n65D3Qfc7YRrmAB18m11_TYQRQCVawLZ6Sw

Det man tegner kan man skjære ut i vinyl dersom skolen har en vinylkutter. Dette finnes en beskrivelse av i handlenettoppgaven. Dette kan brukes i alle tegneoppgavene, om man vil lage andre produkter.

Samlet innhold for elevbok

Kapittel 1 – Generell introduksjon til programmering

- Analog introduksjon til programmering
- Introduksjon til viktige begreper
- Introduksjon til Scratch
- Introduksjon til micro:bit
- Introduksjon til MakeCode
- Sensorer
- Feilsøking
- Hvordan virker en servo
- Miniutfordringer

Kapittel 2 – Naturfagsopplegg

- Opplegg 1 – Innovasjonsmetoden
 - Demonstrer for ditt standpunkt
- Opplegg 2 – Celler
 - Lag en modell av en celle
- Opplegg 3 – Drivhuseffekten
 - Lag en modell av drivhuseffekten
- Opplegg 4 – Energiformer og –overganger
 - Lag en kjedereaksjon

Kapittel 3 – Kort introduksjon til modellering med Geogebra (lineær og ikke-lineær)

- Lineær modellering
- Ikke-lineær modellering

Kapittel 4 – Lineær modellering i naturfag og matematikk (8. trinn)

- Variabler i Scratch og MakeCode
- Opplegg 5 – Massetetthet og periodesystemet
 - Lag en volum-måler
- Opplegg 6 – Friksjon og omforming av energi
 - Lag et skråplan der vinkelen kan justeres
- Opplegg 7 – Solenergi
 - Lag en effektiv solfanger
- Opplegg 8 – Formler, likninger og fart
 - Lag det raskeste kjøretøyet

Kapittel 5 – Musikkopplegg

- Opplegg 9 – Rytme
 - Lag et programmerbart instrument
- Opplegg 10 – Lyd og lydbølger
 - Lag et musikk-kort
- Opplegg 11 – Toner og frekvens
 - Lag en ringeklokke

Samlet innhold for elevbok

Kapittel 6 – Geometri, matematiske mønstre og modellering (9.trinn) og naturfag

- Opplegg 12 – Tegne geometriske former med programmering
 - Tegning av hus til fargeleggingsbok
- Opplegg 13 – Sirkler, kuler og naturressurser
 - Tegning av sirkler til handlenett
- Opplegg 14 – Volum for ulike geometriske former
 - Utforsk volum med en volummåler
- Opplegg 15 – Matematiske mønstre
 - Lag et putetrek

Kapittel 7 – Sannsynlighet (9.trinn) og naturfag

- Forsøk, simuleringer og sannsynlighet
- Opplegg 16 – Sannsynlighet og representasjoner
 - Lag en klimatilpasset art
- Opplegg 17 – Uniform sannsynlighet
 - Lag en terning eller et lykkehjul
- Opplegg 18 – Antall mulige kombinasjoner
 - Lag en magisk krystallkule
- Opplegg 19 – Sannsynlighet og valgtre
 - Lag et spill

Kapittel 8 – Statistikk (9.trinn) og naturfag

- Opplegg 20 – Statistikk og sentralt mål
 - Lag en gjennomsnittskalkulator
- Opplegg 21 – Frekvenstabell og sektordiagram
 - Lag en spørreundersøkelse
- Opplegg 22 – Statistikk og diagrammer
 - Lag en statistisk fremstilling
- Opplegg 23 – Statistikk og «fake news»
 - Lag en villedende statistisk fremstilling
- Opplegg 24 – Statistikk og spredningsmål
 - Lag en standardavvikskalkulator
- Opplegg 25 – Statistikk i naturfag og samfunnsfag
 - Lag en statistisk fargeleggingsnok

Kapittel 9 – Modellering og statistikk (10.trinn) (og naturfag)

- Opplegg 26 – Last ned data fra statistisk sentralbyrå
 - Lag figurer som illustrerer dataene du bruker
- Opplegg 27 – Personlig økonomi
 - Forsk på personlig økonomi
- Opplegg 28 – Statistikk og korrelasjon
 - Lag en villedende nyhetssak

Samlet innhold for elevbok

Kapittel 10 – Naturfag og ikke-lineær modellering (10.trinn)

- Opplegg 29 – Varme, isolasjon og energieffektivitet
 - Lag et energieffektivt hus
- Opplegg 30 – Populasjoner og populasjonsvekt
 - Lag en øy
- Opplegg 31 – Vaksiner og immunforsvar
 - Programmer og strikk et smittetall
- Opplegg 32 – Forbrenningsreaksjoner
 - Lag en utskytningsmekanisme

Kapittel 11 – Kunst og håndverksopplegg

- Opplegg 33 – Geriljabroderi
 - Lag et teknologisk geriljabroderi
- Opplegg 34 – Lag en teknologisk 17.mai-fane
- Opplegg 35 – Lag et produkt som bruker hvis-klosser

Kapittel 12 – Introduksjon til Pythonprogrammering

- Introduksjon
- Variabler
- Hvis-setninger
- While-løkker
- For-løkker
- Lister
- Tabeller
- Funksjoner
- Tegning
- Python for micro:bit
 - Inndata
 - Utdata
 - Musikk
 - Logging til pc/MAC
 - Sonar
 - Servo
 - Skrive til fil

Kapittel 13 – Modellering med Python (for de spesielt interesserte elevene)

- Modellering og maskinlæring
- Plotte graf
- Lineær regresjon
- Ikke-lineær regresjon
- Korrelasjon
- Valg av regresjonsmodell

Del 1

Kapittel 1 i elevboka

Programmering uten datamaskin

(unplugged programming)

Bygg som jeg sier!

Utstyr: LEGO-klosser eller lignende

Tidsbruk: 0,5 - 1 skoletime

Elevene jobber sammen i par.

Dette gjør du:

- Introduser oppgaven i plenum. Kanskje forbinder elevene programmering foreløpig kun med å teknisk kode. Her er det fornuftig å komme innom at programmering handler mye om å forklare ting så tydelig og enkelt at en datamaskin kan forstå det.
- Del elevene inn i læringspartnere/par. Alle må få utdelt LEGO-/byggeklosser.
- Samle alle på slutten av økta for å oppsummere. Få fram ulike erfaringer fra elevene.

Greit å huske på

Her blir elevene veldig bevisste på hvor presist språket må være for å få til å lage LEGO-figur etter andres instruksjoner. Legg vekt på dette underveis og i oppsummeringen – det bør absolutt kommenteres.

TIPS!

Gi elevene veiledning, både mens de lager oppskrift og når de bygger. Foreslå at de kan bruke vinkler og former i tillegg til farge på klossene.

Til slutt kan du avsløre at slike oppskrifter som de har lagd, egentlig er dataprogrammer!

Programmering uten datamaskin

Gjennom dørå

Utstyr: Post-it-lapper eller lignende

Tidsbruk: 0,5 - 1 skoletime

Elevene jobber sammen i par.

TIPS!

- Marker flere steder i klasserommet (eller ute) med lapper for å fordele parene rundt (det kan fort bli kaotisk hvis alle skal bruke den samme dørå).
- Introduser oppgaven i plenum. Få elevene til å reflektere rundt hva som er viktig når man skal lage en oppskrift. Hva innebærer det å lage en presis oppskrift? (Kort, nøyaktig, forståelig).
- Del elevene inn i læringspartnere/par.
- Samle alle på slutten av økta for å oppsummere. Få fram ulike erfaringer fra elevene.

Greit å huske på

Elevene kommer fort til å oppdage at de ikke har all informasjonen de trenger for å få løst oppgaven, for eksempel lengden på hverandres steg.

Dette kan det være fornuftig å adressere i etterkant: at datamaskiner er «dumme» og ikke kan tenke ut noe selv, derfor vil de heller ikke kunne gjette hva som er en logisk skrittlengde. Det er grunnen til at vi må fortelle datamaskinen absolutt alt den trenger å vite.

Det kan også være lurt å nevne at en datamaskin leser et program fra øverst til nederst. Står ting i feil rekkefølge, så vil resultatet også bli feil.

Variasjon av oppgaven – bør gjennomføres

Etter å ha gjort denne oppgaven første gang, er det en god variasjon å bruke kodeklossene fra påfølgende side i elevboka.

Den viktigste grunnen til å bytte ut elevenes eget språk med kodeklossene, er at man raskt kan trekke parallellen til at forskjellige programmeringsspråk har ferdige klosser (eller tekstkommandoer) man kan bruke.

Heldigvis finnes klosser for det meste man skulle trenge, ellers går det også an å lage egne klosser – det kan man lese mer om på siden om funksjoner i programmering.

Benytt også sjansen til å introdusere noen grunnleggende programmeringsbegreper. Disse går man nærmere inn på i «Introduksjon til noen viktige begreper», men det er alltid lurt å ha knyttet til seg noen erfaringer med dem når de skal læres.

Introduksjon til noen viktige begreper

Tidsbruk: 1 skoletime

Dette gjør du:

- Gjennomgå begrepene i fellesskap.
- La elevene samarbeide om oppgavene til begrepene.
- Ta en felles oppsummering av hver oppgave.

Snakk sammen-oppgaven: Elevene ser et bilde av et program som skal få Scratch-katten til å mjaue. Sammen skal de finne ut av hvilket av vilkårene som oppfyller dette.

Husk: Mjauprogrammet starter bare hvis vilkåret er sant.

Av de tre vilkårene er det bare det midterste vilkåret som er sant (10 er mindre enn 50), og dermed mjauer programmet bare for dette.

Oppgavene: Det er ikke alltid enkelt å vite om et vilkår er sant eller usant. Datamaskiner kan bare forstå vilkår som er skrevet slik at de er sanne eller usanne. Det typiske er å sammenligne om to tall er like store, større eller mindre enn hverandre. Det går også an å sammenligne tekst, men det blir fort mer komplisert.

Grubleoppgaven: Dette programmet inneholder en funksjon som gir et tilfeldig tall (en «random-generator»). Programmet utfører det som står inni løkka («Gå 10 steg»), fram til det tilfeldige tallet ikke lenger er mindre enn 50. Siden de tilfeldige tallene varierer mellom 1 og 100, er det 49/100 sjanse for at løkka utføres, og 51/100 sjanse for at løkka ikke utføres – altså en nesten 50 prosent sannsynlighet for begge alternativene.

Oppfølgingsspørsmål til ekstra interesserte elever: Hva må endres på i programmet for at det skulle blitt akkurat 50 prosent sjanse for hvert av alternativene?

Funksjoner i programmering

Tidsbruk: 1 skoletime

Dette gjør du:

- Gjennomgå funksjonsbegrepet i fellesskap

Funksjoner innen programmering har en annen betydning enn funksjoner i matematikken.

En funksjon i programmering er et lite **program**, og for det finnes det en egen funksjonskloss i Scratch. Det kan for eksempel være en kloss som får Scratch-katten til å bevege seg i en sirkel eller en annen form.

Den viktigste grunnen til å bruke funksjoner, er å forenkle koden. Den gjør at man kan gjenta en del av et program flere ganger. Da kan man begrense antall klosser og gjøre koden mer oversiktlig.

Alle funksjoner må først defineres. Det betyr at man må fortelle hva funksjonsklossen skal gjøre. Deretter må man bruke funksjonen (vår nye kloss) i programmet. Den kan man bruke så mange ganger man vil.

I andre programmeringsspråk kan funksjoner gi en *returverdi*. Det vil si at vi kan bruke slike småprogrammer til å regne ut noe og gi verdien tilbake til programmet. Dette er veldig praktisk, og gjør at programmene ikke blir så store og rotete. I Scratch finnes (per dags dato) ikke denne muligheten, så funksjonene kan heller brukes til å la katten tegne visse former eller bevege seg på bestemte måter.

- Gjør «Snakk sammen»-oppgavene, først i læringspar, deretter i plenum.
- Snakk sammen-oppgaven: Funksjonen til venstre tegner en tilnærmet sirkel (egentlig en 36-kant) med en radius som man må oppgi når man bruker funksjonen. Det er en funksjon, da det er her man definerer innholdet i blokka som kalles «Tegne sirkel».

Programmet til høyre bruker funksjonen til venstre til å tegne en sirkel (36-kant) med radius 15 når man trykker på det grønne flagget. Når man så trykker på mellomromstasten (space-tasten), slettes sirkelen og markøren (katten) går tilbake til opprinnelig startpunkt (origo).

Blokker i Scratch

Utstyr: Digital enhet

Tidsbruk: 1 - 2 skoletimer

Elevene jobber sammen i par i begge oppgavene.

Dette gjør du:

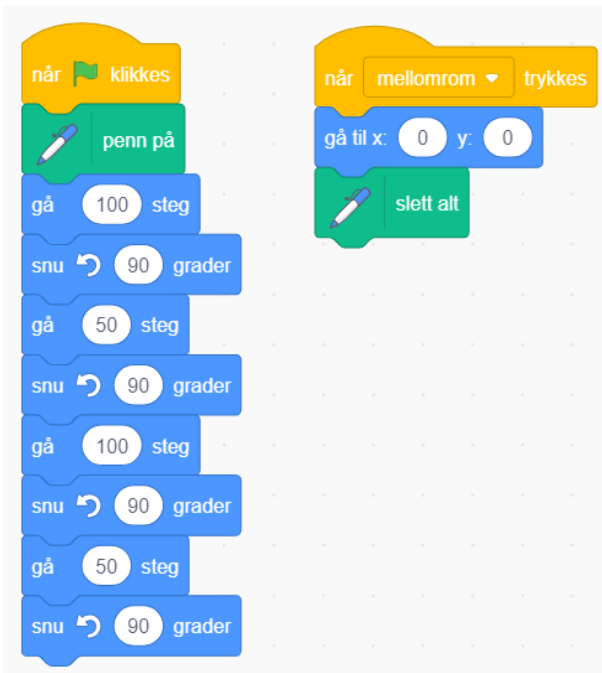
- Gå gjennom de ulike blokkene sammen med elevene.
- Del ut kopioriginal *Klosser i Scratch* og la elevene selv fylle ut feltene for å løse oppgaven.
- Gjennomgå forslagene til slutt. Få elevene til å sette ord på hvorfor de valgte det de gjorde.

Løsningsforslag til å kunne flytte katten tilbake til origo med piltastene



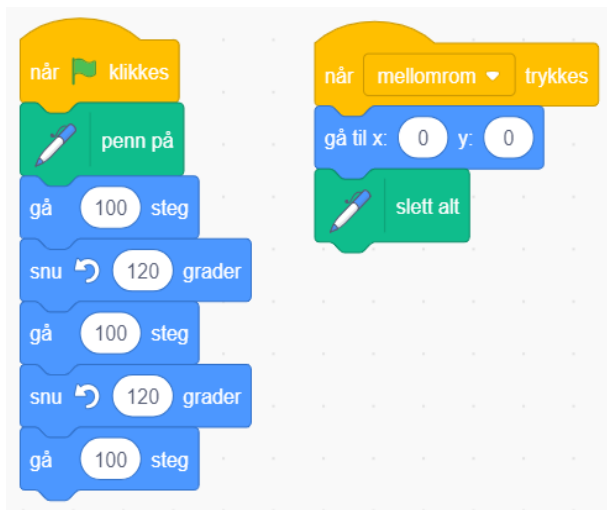
Løsningsforslag til rektangelform

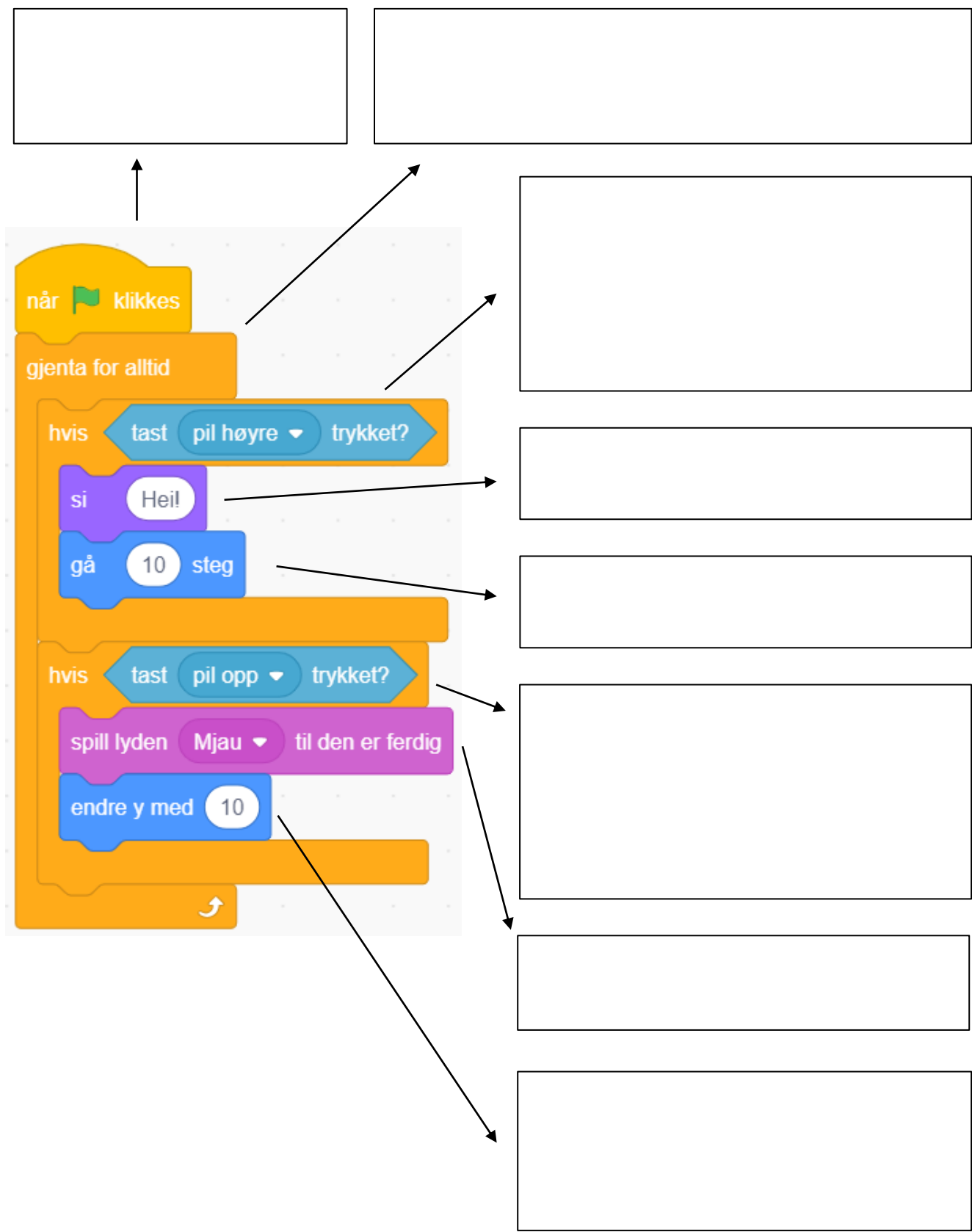
Delen til høyre gjør det enkelt å begynne forfra igjen.



Løsningsforslag til trekantform

Legg merke til vinklene i koden, Scratch-katten må svinge ($180 - v$) for at den indre vinkelen i trekanten skal bli v .





Mini-utfordringer

Hovedpoenget med disse utfordringene er å trene elevene på å jobbe i grupper. Gjennom øvelsene får de oppleve en god dose frustrasjon i nærmest «umulige» oppgaveformuleringer. Uklare kriterier hjelper heller ikke!

La elevene kjenne på disse følelsene, og ikke minst hvilke roller de tar i denne type arbeid. Hvordan tenker og agerer de i slike situasjoner?

Dere trenger: Papp, tape, strikker, hyssing, bomull, papir, klinkekuler, doruller, linjaler, svamper mv.

Tidsbruk: Maksimalt 10-12 minutter for elevenes gjennomføring (per oppgave). Hele opplegget tar mye lenger tid, da poenget er samtalen i etterkant av hver oppgave. Det går for eksempel an å ta to av oppgavene og samtale i løpet av en skoletime. Disse oppgavene kan også være fine å introdusere timer med, og kan dermed fordeles over oppstarten på seks forskjellige timer.

Elevene jobber sammen i grupper på tre til fire elever. Det må være en ekstra elev per gruppe som observerer dem. Denne eleven får et ark med punkter han/hun skal følge med på, som læreren forklarer og gjennomgår med observatørene uten at resten av klassen vet hva observatørene skal gjøre. Læreren er også observatør, men følger med på alle gruppene. Rollen som observatør går på omgang, slik at alle elevene får prøve seg som gruppedeltagere.

NB: Ikke fortell de andre elevene konkret hva oppgaven til observatøren er.

Det skjeve tårn i Pisa

I denne oppgaven skal elevene bygge et tårn utfra noen kravspesifikasjoner. Læreren demonstrerer med en papplate som står skrått oppå to treklosser. Denne skal være basen for byggverket, og være i kontakt med klossene hele tiden. NB: Det er likevel ikke et strengt krav at plata må stå på skrå, poenget er å få det til å fremstå slik siden læreren demonstrerer det på den måten. Dette er et viktig poeng og lærer elevene noe om å gjøre antagelser, som kanskje ikke er riktige. Lærdommen er at det er alltid lurt å sjekke at man har klart for seg hva oppgaven innebærer. Og om det er mulig å angripe oppgaven på andre måter enn det man tror ved første øyekast, kanskje kan man være litt mer kritisk til hva oppgaven faktisk spør om?

Utskytningsmekanisme

Elevene skal lage en utskytningsmekanisme som skal få f.eks. en svamp til å fly minst 5 meter. Her vil du fort oppdage at elevene blir frustrert over hvor vanskelig det er å få svampen til å fly langt, pga. stor luftmotstand. En løsning på dette er å komprimere svampen så mye dere klarer og deretter teipe den, for så å prøve på nytt.

Uidentifisert flygende objekt (UFO)

Her kan det fort bli diskusjon om hva det vil si «å være i luften». Må gjenstanden fly eller holder det at den henger? Elevene bør få lov til å oppdage selv at det ikke er skikkelig definert i oppgaven. Det er lov med smutthull. Men det bør tas opp i fellesskap etterpå – hva mener resten av klassen?

Bakkeløp

I dette opplegget er ikke poenget å gjøre noe raskest, men å være mest mulig nøyaktig. De bør ha tilgang til stoppeklokke (for eksempel på mobilen). Elevene kan både justere kjøretøyet sitt eller hellingen på bakken (papplata), men det kan være fint å bare demonstrere justering av kjøretøy, slik at elevene kan oppdage andre måter selv. Dette kan føre til frustrasjon hos noen av de som ikke oppdager denne måten å gjøre det på, og man kan ta det opp i samtalen etter gjennomføring.

Brobygging

Denne oppgaven kan være veldig enkel, eller fryktelig vanskelig, avhengig av hvilke materialer elevene har tilgang på. Man kan tilpasse hvor langt brospennet skal være ut fra hvilke materialer man har tilgang til. Husk at hovedpoenget er ikke at de skal få til oppgaven, men lære noe om hvordan de jobber under press.

Bordbasket

Det kan være greit å avslutte med en oppgave som er lettere å gjennomføre, slik at elevene også får kjenne på ren mestringsglede!

Forslag til oppsummerende refleksjoner

Refleksjonsdelen bør ta lengre tid enn selve oppgavene.

Ta for dere én og en gruppe, og gå gjennom spørsmålene fra observasjonsarket på neste side (kopioriginal).

Spør gjerne elevene på gruppa de samme spørsmålene før observatøren kommer med sine observasjoner. Stemmer det gruppa sier overens med det observatøren så?

Koble observasjonene opp mot hvordan elevene kan oppføre seg i gruppearbeid. Kom med forslag til strategier for hvordan takle frustrasjon – var det noen grupper som taklet det bedre enn andre? Hvordan taklet de det?

Observasjon av mini-utfordringene

Kven tek styringa?

- Er det dei same heile tida?

Blei nokon frustrerte?

- Kva gjorde at dei blei frustrerte?

Var nokon kreative på gruppa?

- Korleis kunne du sjå det?

Kva for nokre rollar hadde dei ulike på gruppa?

- Leiar
- Følgar
- Ho/han med idéane
- Byggaren
- Ho/han som spurte læraren
- Andre roller?

Del 2

Undervisningsopplegg 1 – 35

Kapittel 2 – 13 i elevboka

Undervisningsopplegg

Kapittel 2 – 11 inneholder opplegg med fagstoff for å arbeide direkte med fagenes kompetansemål. Dette blir gjort gjennom kreative prosjekter der elevene bruker innovasjonsmetoden for å lage en fysisk gjenstand. Disse oppleggene kalles også gjerne for skaperverkstedprosjekter.

Før dere tar fatt på oppleggene, bør dere ha gjennomgått de grunnleggende øvelsene med analog programmering (programmering uten datamaskin), og introduksjon til forskjellige programmeringsbegreper. I tillegg er det en forutsetning at du og elevene har brukt tid på å lære dere grunnleggende klosser i Scratch og MakeCode eller grunnleggende Pythonkoder (se kapittel 12 i elevboka).

Gå selv gjennom programmeringsdelen i oppleggene før du setter i gang, og vurder behovet for å repetere programmeringsbegreper eller klosser for de ulike oppleggene. Utover dette er det ikke slik at du som pedagog trenger å sitte med absolutt alle svarene – båten blir til mens man ror!

Opplegg 1 – Innovasjonsmetoden

Kompetansemål: Det er ingen spesifikke kompetansemål for dette opplegget da det er ment som en grunnleggende introduksjon til arbeidsmåten i de andre undervisningsoppleggene. Det er naturligvis mulig å inkludere kompetansemål fra mange ulike fagområder – kanskje er samfunnsfag og språkfag de mest aktuelle, i tillegg til de tverrfaglige temaene.

Materialer: A3-ark, fargede ark, lim, sakser, div. hobbyutstyr som fjær, silkepapir, avisapapir, kvister, kongler, småstein osv.

TIPS!

Generelt om tidsbruk i innovasjonsmoden: Elevene vil trenge hjelp med tidsbruken når de jobber med innovasjonsmetoden. Spesielt i starten er det viktig at du tegner opp start- og sluttider. Deretter kan du slippe gradvis mer opp. Pass likevel på de gruppene du vet trenger hjelp til å disponere tiden. Det er en klar fordel for deg som lærer at elevene jobber med ulike faser til ulike tider, da det blir litt mer spredt i tid at de forskjellige gruppene jobber med fasene de trenger mest hjelp til.

Gå gjennom innovasjonsmetoden med elevene og diskuter gjerne hvordan de ulike punktene hører sammen med naturvitenskaplig metode:

1. Undersøke og finne informasjon

Denne fasen ligger ikke eksplisitt i naturvitenskaplig metode, men omfatter all undervisning før elevene skal gjøre et forsøk innen temaet. For naturvitenskaplige forskere, gjelder dette punktet generelt i alt arbeidet de gjør. De samler kontinuerlig informasjon ved å lese andre forskeres resultater og observere ulike fenomener (kan dra paralleller til utviklingen av teorien om Big Bang for eksempel).

2. Idémyldre og planlegge

I planleggingsfasen setter elevene opp en hypotese dersom det er et opplegg som krever det.

Oppleggene i skaperverkstedet krever som regel ikke at man lager en hypotese. Likevel jobber elevene etter de samme prinsippene, ved å lage antakelser om hvordan de tror gjenstanden eller programmet bør se ut før de starter. Disse antagelsene tilsvarer hypoteser.

Det er i denne fasen du lar elevene møte algoritmisk tenkning i praksis. Det innebærer blant annet at dere trener på å bryte en problemstilling ned i mindre deler for å lettere utvikle et produkt (eller en prototyp). Det kan dere gjøre ved å sette opp en rekke forskningsspørsmål, eller en liste med punkter for kjennetegn til sluttproduktet - før man så går baklengs og ser hva man trenger for å oppnå disse.

I boka vil elevene stadig møte setningen «Ha en idémyldring for deg selv». Det er viktig at du legger til rette for dette, selv om de først og fremst skal jobbe i grupper. Gi gjerne denne idémyldringen i lekse, slik at de får litt tid alene til å tenke i fred og ro før de skal samarbeide. Dette er nyttig for alle, men vil være spesielt verdifullt for de elevene som er litt tilbakeholdne i gruppearbeid, spesielt de mer introverte elevene som da får mulighet til å forberede seg grundig på forhånd.

3. Utføre

Denne fasen er parallell med gjennomføringen av et forsøk. Man utfører det man har planlagt for å få testet hypotesen.

4. Teste

Denne fasen inngår også i gjennomføringen av et forsøk, men innebærer også at man dokumenterer målingene man gjør. Dersom man kobler det til naturvitenskaplig metode og forsøk, så er dette den objektive datainnsamlingen man beskriver i resultatdelen i en forsøksrapport.

5. Evaluere

Denne fasen tilsvarer diskusjonsdelen i en naturvitenskaplig forsøksrapport, og her skal elevene vurdere resultatene de fikk i punkt 4. Fase 4 og 5 er adskilt for å trene elevene i objektiv datainnsamling, slik at ikke alt vurderes direkte og det er større fare for at subjektive meninger påvirker resultatene. Man vurderer også hvilke feilkilder som kan påvirke resultatene, i hvilken retning de påvirker resultatene, og hvordan de kan minimeres eller elimineres.

6. Forbedre

Denne fasen ser man vanligvis ikke i enkeltstående naturvitenskaplige forsøk, men i utviklingen av nye teorier og modeller, som forsøkene er en del av. Her tar man med seg vurderingen rundt feilkilder, og kan gå tilbake til alle foregående punkter for å minimere feilkildene. I tillegg ønsker man generelt å optimalisere forsøksdesignet eller prototypen man har utviklet, og går tilbake til de fasene man vurderer at trengs. Denne fasen legger til rette for mange iterasjoner, der man gradvis forbedrer innovasjonen sin.

Dette er et sentralt poeng, da det kan hjelpe elevene til å utvikle en tro på at de kan lære og mestre, samt redusere usunn perfektjonisme ved at man *må* gjøre feil underveis og det er det man lærer av. Det er fint om læreren kan modellere dette selv, og påpeke egne feil som man har lært av, dette ufarliggjør denne prosessen overfor elevene, at til og med læreren kan gjøre feil – og det er helt greit. Dette fasen kan også hjelpe å redusere elevens prestasjonspress, da det å ikke ha alt riktig med *en* gang, blir ufarliggjort.

7. Dokumentere

I denne fasen gjennomfører elevene den siste målingen og fullfører dokumentasjon av prosessen. Det er viktig å dokumentere alle stegene i prosessen. Bruk gjerne vedlagte skjema for å bevisstgjøre elevene på de forskjellige fasene. Dette tilsvarer å skrive rapport for et forsøk, men her dokumenterer de hele prosessen med å planlegge forsøket, og dokumenteringen har ikke så strenge formkrav, det er opp til læreren å bestemme. Det går fint an at elevene dokumenterer alt med en film, bildeserie, presentasjon, rapport, podcast eller tilsvarende.

Generelt:

Dette er også en generell problemløsningsmetode som også fungerer å ha i bakhodet for hvordan forskere jobber med å drive vitenskapen fremover ved å designe egne forsøk og studier for å teste hypotesene de også lager.

Bruk gjerne skjemaet til noen av de andre prosjektene i boka også.

Ekstraoppgave:

- Finn et eksempel på bruk av innovasjonsmetoden i utviklingen av et produkt eller teoretisk modell.
- Identifiser hvilke hendelser som hører til hvilken fase (her er det ikke sikkert at man kan identifisere alle fasene for alle eksemplene).

1.	Undersøke og finne informasjon
2.	Idémyldre og planlegge
3.	Utføre
4.	Teste
5.	Evaluerer
6.	Forbetre
7.	Dokumentere
Generelle kommentarer:	

Opplegg 2: Celler

Kompetansemål

Naturfag:

- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener
- sammenligne celler hos ulike organismer og beskrive sammenhenger mellom oppbygning og funksjon

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer:

Micro:bit, Alligatorledninger, motstander og LED-pærer

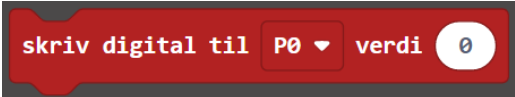
Byggematerialer: papp, papir, limpistoler, sakser, tape ++

Dekorasjonsmaterialer: maling, silkepapir, glitter +++

TIPS!

Bruk gjerne maskeringstape for å feste micro:bit, LED-pærer, alligatorledninger o.l. med modellen. Maskeringstape er enkel å fjerne etter bruk slik at elektronikken ikke blir ødelagt. Pass på at elevene ikke limer det fast med lim!

Alternative programmeringsoppgåver



Puslespel-programmering

Bruk disse klossane for å lage eit program som gjer at LED-pæra blenker. Du må nok bruke nokon av klossane fleire gonger.

Løysingsframlegg for blenkande LED-pære



Programmeringsoppgåver med python

```
from microbit import *  
  
while True:  
    pin0.write_digital(1)
```

Døme på program

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

Puslespel-programmering

Bruk desse kodelinjene for å lage eit program som gjer at LED-pæra blenker.

```
pin0.write_digital(1)          sleep(500)
```

```
sleep(500)    pin0.write_digital(0)
```

```
from microbit import *      while True:
```

Løysingsframlegg for blenkande LED-pære

```
from microbit import *  
  
while True:  
    pin0.write_digital(1)  
    sleep(500)  
    pin0.write_digital(0)  
    sleep(500)
```

Opplegg 3: Drivhuseffekten

Kompetansemål

Naturfag:

- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener
- beskrive drivhuseffekten og gjøre rede for faktorer som kan forårsake globale klimaendringer
- gjøre rede for hvordan fotosyntese og celleånding gir energi til alt levende gjennom karbonkretsløpet

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer:

Micro:bit, Alligatorledninger, motstander og LED-pærer

Byggematerialer: papp, papir, limpistoler, sakser, tape ++

Dekorasjonsmaterialer: maling, silkepapir, glitter +++

TIPS!

Bruk gjerne maskeringstape for å feste micro:bit, LED-pærer, alligatorledninger o.l. med modellen. Maskeringstape er enkel å fjerne etter bruk slik at elektronikken ikke blir ødelagt. Pass på at elevene ikke limer det fast med lim!

Alternative programmeringsoppgåver



Puslespel-programmering

Bruk disse klossane for å lage eit program som gjer at LED-pæra blenker. Du må nok bruke nokon av klossane fleire gonger.

Løysingsframlegg for blenkande LED-pære



Programmeringsoppgåver med python

```
from microbit import *  
  
while True:  
    pin0.write_digital(1)
```

Døme på program

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

Puslespel-programmering

Bruk desse kodelinjene for å lage eit program som gjer at LED-pæra blenker.

```
pin0.write_digital(1)        sleep(500)  
  
sleep(500)    pin0.write_digital(0)  
  
from microbit import *      while True:
```

Løysingsframlegg for blenkande LED-pære

```
from microbit import *  
  
while True:  
    pin0.write_digital(1)  
    sleep(500)  
    pin0.write_digital(0)  
    sleep(500)
```

Opplegg 4: Energiformer og -overgangar

Kompetansemål

Naturfag:

- bruke programmering til å utforske naturfaglige fenomener
- gjøre rede for energibevaring og energikvalitet og utforske ulike måter å omdanne, transportere og lagre energi på
- drøfte hvordan energiproduksjon og energibruk kan påvirke miljøet lokalt og globalt

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materiale:

Micro:bit, 2 alligatorledninger, servomotor og krinskort med batteri.

Byggemateriale: papp, papir, limpistolar, sakser, tape ++

Dekorasjonsmateriale: maling, silkepapir, glitter +++

TIPS!

Dette opplegget kan bli gjort med begge typene servoar, berre pass godt på å merke kva for ein type dei er.

Ekstraoppgåve

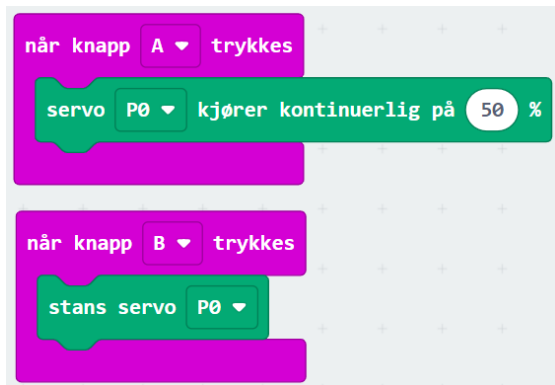
Finne ut korleis dei ulike måtane å produsere elektrisitet på påverkar området der det blir produsert, og i resten av verda.

Sjå gjerne på:

- Vindmøller
- Solceller
- Vasskraftverk
- Kolkraftverk
- Kjernekraftverk
- Bioetanol eller biodiesel

Løsningsforslag

programmeringsoppgave



Fra disse tre eksemplene, kan man se at det finnes mange måter å variere hvordan servoen beveger seg på. Her er noen eksempler:

For 180-graders-servoer der dreiehastigeten er den samme uansett kan vi variere:

- Hvor i sirkelbanen servoen treffer instrumentet
- Hvor lenge servoen går mot en vinkel (om den rekker å gå «ferdig», eller ikke)
- Hvor stor vinkel den beveger seg til
- Er det en pause etter servoen har nådd en vinkel?

For en kontinuerlig servo der vi bestemmer dreiehastighet og ikke vinkel:

- Hvor i sirkelbanen servoen treffer instrumentet
- Hvilken dreiehastighet som velges
- Hvor lenge den hastigheten beholdes
- Snur den retning på noe tidspunkt

Programmering med Python

For å programmere servoer med Python bør ein importere eit bibliotek inn på sjølve micro:biten. Dette står forklart i elevboka under «Servo – med Python Mu» i kapittel 12. Når dette er importert, kan ein bruke oppgåva under.

```
from microbit import *

while True:
    if button_a.was_pressed():
        servovinkel(pin0,-90)

from microbit import *

while True:
    if button_b.was_pressed():
        servovinkel(pin0,90)
```

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 kjedereaksjonen dykkar?

Løysingsframlegg

```
from microbit import *

while True:
    if button_a.was_pressed():
        servovinkel(pin0,-90)
    elif button_b.was_pressed():
        servovinkel(pin0,90)
```

Opplegg 5: Massetetthet og periodesystemet

Kompetansemål

Matematikk:

- lage og forklare rekneuttrykk med tal, variablar og konstantar knytte til praktiske situasjonar
- utforske, forklare og samanlikne funksjonar knytte til praktiske situasjonar
- representere funksjonar på ulike måtar og vise samanhengar mellom representasjonane
- utforske korleis algoritmar kan skapast, testast og forbeholdt ved hjelp av programmering

Naturfag:

- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener
- bruke atommodeller og periodesystemet til å gjøre rede for egenskaper til grunnstoffer og kjemiske forbindelser

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer:

Micro:bit, alligatorledningar, sonar (3V-typen) og batteripakke. 500 ml begerglas (evt 250 ml – det bør ikkje vere for lite), vekt, gjenstandar av ulike materiale (for å måle massetetthet).

Byggematerialer: papp, papir, sakser, limpistoler, tape ++

Dekorasjonsmateriale: maling, silkepapir, glitter +++

TIPS!

Bruk høge og smale begerglass da målingane blir meir nøyaktige desto meir vasstanden endrar seg, for avstanden blir målt i heile cm.

For meir nøyaktige svar kan ein endre til å lese av tid, som er tida det tek for lyden å røre seg mellom sensoren og overflata av vatnet. Da kan man bruke at lydfarten i luft i romtemperatur er ca 340 m/s. $s=v \cdot t$.

Det er viktig å diskutere målefeil i dette opplegget.

Ekstraoppgåve

1. Lag eit program til micro:biten som reknar ut massetettleiken for det stoffet de måler volumet på.
2. Lag eit program som kan gjere om mellom ml og cm³.

Krevjande ekstraoppgåve

1. Kople til ein ultralydsensor som kan brukast til å måle avstanden frå sensoren til vatnet. Da kan ein kalibrere han med hjelp av regresjon, slik at de finn ut kva verdien de les av frå ultralydsensoren svarer til i auka vasshøgde utan å bruke sonar-pakken til micro:bit (nullhøgde+x=måletal). Da kan de rekne ut volumet automatisk og skrive det ut på micro:biten dykkar.
2. Finn ut korleis ein ultralydsensor verkar. Sjå gjerne:
<https://www.teachwithict.com/hcsr043v3.html>

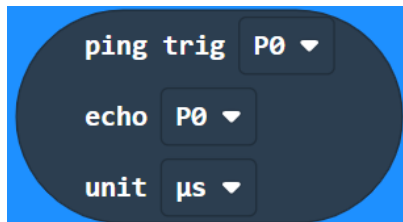
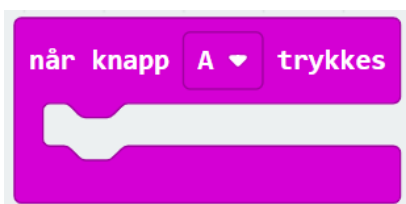
TABLE 1.4 The Density of Some Common Substances at 20 °C

Substance	Density (g/cm ³)
Charcoal (from oak)	0.57
Ethanol	0.789
Ice	0.917 (at 0 °C)
Water	1.00 (at 4 °C)
Sugar (sucrose)	1.58
Table salt (sodium chloride)	2.16
Glass	2.6
Aluminum	2.70
Titanium	4.51
Iron	7.86
Copper	8.96
Lead	11.4
Mercury	13.55
Gold	19.3
Platinum	21.4

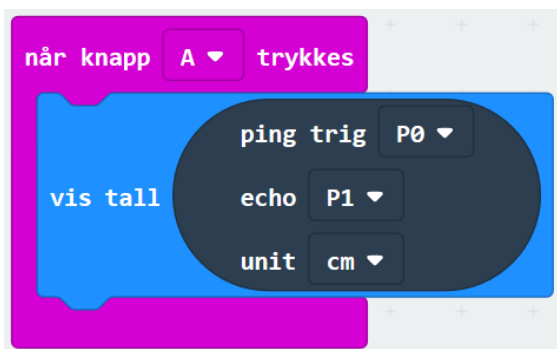
Programmeringsoppgåver

Puslespel-programmering

Bruk desse klossane for å lage eit program som les av avstanden frå sensoren til vatnet si overflate..



Løysingsframlegg programmering



For å programmere sonarar med python, bør ein importere eit bibliotek inn på sjølve micro:biten. Dette står forklart i elevboka under «Sonar – med python mu», s... Når dette er importert, kan ein nytte oppgåva under.

Opplegg 6: Friksjon og omforming av energi

Kompetansemål

Matematikk (oppgitte mål for 8. trinn, men kan brukes for modellering på 10.trinn også):

- lage og forklare rekneuttrykk med tal, variablar og konstantar knytte til praktiske situasjonar
- lage, løyse og forklare likningar knytte til praktiske situasjonar
- utforske, forklare og samanlikne funksjonar knytte til praktiske situasjonar
- representere funksjonar på ulike måtar og vise samanhengar mellom representasjonane
- utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering

Naturfag:

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener
- gjøre rede for energibevaring og energikvalitet og utforske ulike måter å omdanne, transportere og lagre energi på

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materiale:

Micro:bit, 2 alligatorleider, 180°-servomotor og kretskort med batteri.

Byggemateriale: papp, papir, sakser, limpistolar, tape ++

Dekorasjonsmateriale: maling, silkepapir, glitter +++

TIPS!

Dette opplegget kan berre gjerast med 180°-servoar, da de må kunne stille inn kva for ein vinkel de skal ha.

Ekstraoppgåve

Kvilefriksjon og glidefriksjon er ikkje like store – kvifor ikkje?

Hint: Elektrisk kraft mellom molekyla i dei to overflatene.

Programmeringsoppgåver

Dei elevane som ikkje treng hjelp med programmeringa, kan få berre informasjon om kjøretid-funksjonen/blokka. Be dei ikkje sjå på puslespel-oppgåva i elevboks. Dei kan evt gå over til Python.

Tidsmåleblokk

Denne blokka måler kor lang tid det har gått sidan programmet tok til å køyre. Ho byrjar å telje med ein gong programmet er ført over til micro:biten som fins under kategorien inndata – meir... (som dukkar opp rett under inndata når du trykker på inndata).

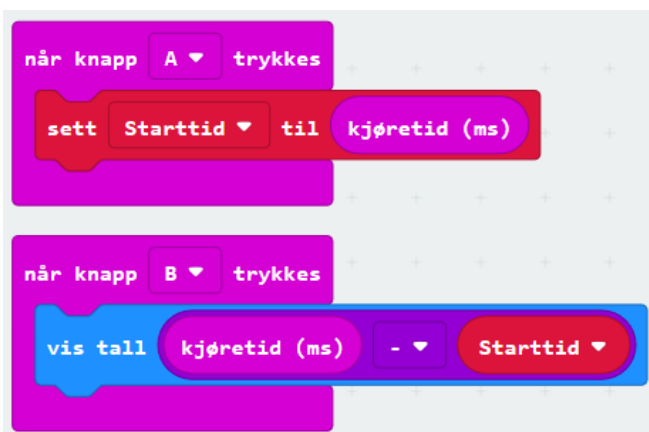
kjøretid (ms)

Gjer-ferdig-programmering

Bruk denne starten av eit program og legg til kodeblokker slik at micro:biten verkar som ei stoppeklokke.



Løysingsframlegg programmering



Programmeringsoppgåver Python

Puslespel-programmering

Bruk disse kodelinjene for å lage eit program som får micro:biten til å verke som ei stoppeklokke

```
if button_a.was_pressed():  
if button_b.was_pressed():  
from microbit import *  
starttid = running_time()  
sluttid = running_time() - starttid  
display.scroll(sluttid)  
while True:
```

Gjer-ferdig-programmering

Bruk denne starten av eit program og legg til kodelinjer slik at micro:biten verkar som ei stoppeklokke.

```
from microbit import *  
  
while True:  
    if button_a.was_pressed():  
        starttid = running_time()
```

Løysingsframlegg programmering

```
1 from microbit import *  
2  
3 while True:  
4     if button_a.was_pressed():  
5         starttid = running_time()  
6  
7     if button_b.was_pressed():  
8         sluttid = running_time() - starttid  
9         display.scroll(sluttid)
```

Opplegg 7 - Solenergi

Kompetansemål

Matematikk (oppgitte mål for 8. trinn, men kan brukes for modellering på 10.trinn også):

- lage og forklare rekneuttrykk med tal, variablar og konstantar knytte til praktiske situasjonar
- lage, løyse og forklare likningar knytte til praktiske situasjonar
- utforske, forklare og samanlikne funksjonar knytte til praktiske situasjonar
- representere funksjonar på ulike måtar og vise samanhengar mellom representasjonane
- utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering

Naturfag:

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener
- gjøre rede for energibevaring og energikvalitet og utforske ulike måter å omdanne, transportere og lagre energi på

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materiale:

Micro:bit med batteripakke.

Papp, avispapir, limpistolar, sakser, tape, svart maling, plastfolie, aluminiumsfolie +++

TIPS!

Det går an å måle temperaturen ved å halde micro:biten opp mot begerglasets med vatn, men sidan glas leier varme ganske dårlig, blir målingane ikkje veldig gode. Det går og an å ta to lag med ein kraftig plastpose (frysepose) rundt micro:bit ved måling oppi vatnet i begerglasset.

Det går an å diskutere fordelar og ulemper med dei ulike måtane med elevane.

Ekstraoppgåve

Finn gjennomsnittleg stigningstal for dei første 5 minutta. Kva tyder dette talet?

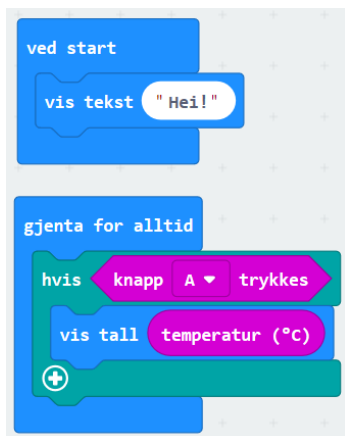
Ekstra programmeringsoppgåve

Sjå opplegg 10 – Lydbølger og musikkort for intro til å programmere radio-signal for micro:bit

Snakk om

Skilnaden på solceller og solfangarar?

PRIMM-klossoppgåve



1. Giss kva programmet ved sida av gjer. Skriv ei kort utgreiing på baksida av arket og diskuter med andre.
2. Gå inn på nettadressa: <https://makecode.microbit.org/#editor> og skriv av koden. Du må leite etter dei rette blokkene i menyane til venstre. Dei ulike kategoriane har lik farge som blokkene sine.
3. Last ned programmet til micro:biten. Stemte det du gissa? Kva observerer du, om du blåser på baksida av micro:biten?

4. Forklar koden:

Blokk	Forklaring

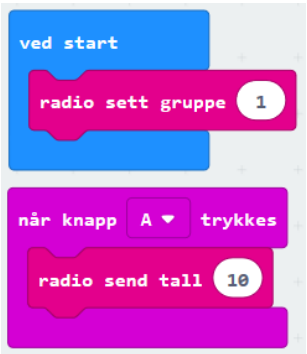
5. Endre koden

- a) Byt ut temperatur med lysstyrke. Kva for nokre verdier får du? Prøv å lyse på LED-lysa med lommelykta på mobilen din. Kva skjer da?
- b) Trykk på pluss-teiknet i viss-blokka og legg inn ei melding. Kva tid kjem meldinga opp?

6. Utfordring: Klarer du å lage eit program som skriv «mørkt» på micro:bit skjermen om du ikkje lyser på han, og «lyst» når du lyser på han med mobil-lommelykta?

PRIMM-oppgåve for sendar og mottakar

Micro:bit 1



Micro:bit 2



1. Giss kva programma ved sida av gjer. Skriv ei kort utgreiing på baksida av arket og diskuter med andre.
2. Skriv av kodane på MakeCode-nettsida. Du må leite etter dei rette klossane i menyane til venstre. Dei ulike kategoriane har lik farge som klossane sine.
3. Last ned programmet til micro:bitene. Stemte det du gissa?

4. Forklar koden:

Blokk	Forklaring

5. Endre koden
 - a) Kva skjer om du byter talet i linje 4 i tabellen frå 10 til 5?
 - b) Korleis kan du få den eine microbiten til å spele ein heil melodi?
6. Utfordring: Klarer du å lage eit program som speler ein melodi?

PRIMM-Pythonoppgåve for micro:bit

```
1 from microbit import *
2
3 while True:
4     x = temperature()
5     print(x)
6     sleep(500)
```

1. Giss kva programmet ved sida av gjer. Skriv ei kort utgreiing på baksida av arket og diskuter med andre.
2. Gå inn på nettadressa: <https://codewith.mu/en/download> og last ned Python Mu. Skriv av koden og overfør han til micro:biten ved å trykke på flash-knappen. Kva hender?

3. Forklar koden:

Kodelinje	Forklaring
<code>from microbit import *</code>	
<code>while True:</code>	
<code>x = temperature()</code>	
<code>print(x)</code>	
<code>sleep(500)</code>	

4. Endre koden
 - a) Byt rekkefylgja på linje 3 og 5. Kva trur du vil endre seg når du køyrer programmet?
 - b) Kvifor MÅ vi ha med sleep-kommandoen?
 - c) Kan du slå saman to av kodelinjene og få same resultat?
5. Utfordring: Bruk kapittel 12 i boka for å finne koden du treng for å lage eit program som sender akselerasjonen målt med ein micro:bit til ein annan micro:bit. Hugs at begge micro:bitane må programmerast. Spør gjerne om hint.

Programmering med Python

Puslespel-oppgåve

Måle temperatur med micro:bit som kan loggest på PC – micro:bit nummer 1

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

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

Micro:bit nummer 2

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

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

Opplegg 8: Formler, likninger og fart

Kompetansemål

Matematikk (oppgitte mål for 8. trinn, men kan brukes for modellering på 10.trinn også):

- lage og forklare rekneuttrykk med tal, variabler og konstanter knytte til praktiske situasjoner
- lage, løse og forklare likninger knytte til praktiske situasjoner
- utforske, forklare og samanlikne funksjonar knytte til praktiske situasjoner
- representere funksjonar på ulike måtar og vise samanhengar mellom representasjonane
- utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering

Naturfag:

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer:

Micro:bit, alligatorledninger, 360°-servomotor og kretskort med batteri.

Papp, papir, limpistoler, tape, sakser, blomsterpinner, maling, silkepapir, glitter +++

TIPS!

Dette opplegget kan bare gjøres med 360°-servoer, da de må kunne få et kjøretøy til å bevege seg skikkelig.

Programmeringsoppgaver

Se opplegg 4 – Energiformer- og overganger for programmeringsoppgaver for å programmere servoer.

Se opplegg 6 – Friksjon og energiomforming for ekstraoppgaven der elevene skal lage en stoppeklokke av micro:biten.

Opplegg 9: Rytme

Kompetansemål:

Musikk:

- skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder
- utforske og formidle musikalske opplevelser og erfaringer, og reflektere over bruk av musikalske virkemidler
- lytte og prøve ut ulike uttrykk og begrunne valg i skapende prosesser fra idé til ferdig resultat
- bruke gehør og notasjonsteknikker som støtte i skapende arbeid
- bruke relevante fagbegreper i skapende arbeid og i refleksjon over prosesser og resultater

Matematikk:

- utforske korleis algoritmar kan skapast, testast og forbeholdt ved hjelp av programmering

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materiale:

Micro:bit, alligatorledningar, servomotor og krinskort med batteri.

Papp, papir, limpistol, tape, plastflasker, sakser, maling, silkepapir, glitter +++

TIPS!

I denne oppgåva blir det gjerne ei form for tromme som blir laga. Det kan gjerast med begge typene servoar, men er nok enklast å få til med en 180-graders servo, da ein enkelt kan setje desse til å ta eitt og eitt «slag», og kan da variere rytmen. Med ein kontinuerleg servo derimot, vil han gå rundt og rundt i ein viss fart, da kan det bli vanskelegare å variere rytmen slik ein vil. Men det går fint an! Det blir berre meir prøving og feiling undervegs.

Ekstraoppgåve

1. Kan du bruke noter for å skrive kva for ein rytme instrumentet ditt speler?
2. Få instrumentet til å spele med ein annan rytme.
3. Kan du skrive denne rytmen med noter?

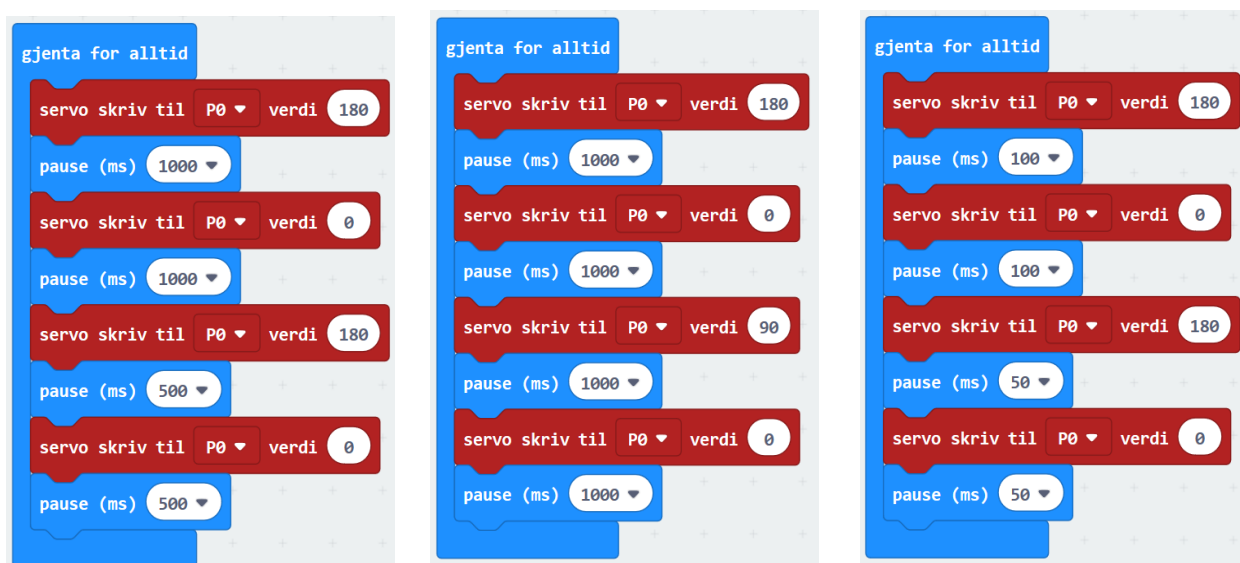
Krevjande ekstraoppgåve

Kan du bygge om instrumentet frå eit rytmeinstrument til et strengeinstrument?

Programmeringsoppgaver

Se opplegg 4 – Energiformer- og overganger for programmeringsoppgaver for å programmere servoer.

Løsningsforslag programmering



Fra disse tre eksemplene, kan man se at det finnes mange måter å variere rytmen på. Her er noen eksempler:

For 180-graders-servoer der dreiehastigeten er den samme uansett kan vi variere:

- Hvor i sirkelbanen servoen treffer instrumentet
- Hvor lenge servoen går mot en vinkel (om den rekker å gå «ferdig», eller ikke)
- Hvor stor vinkel den beveger seg til
- Er det en pause etter servoen har nådd en vinkel?

For en kontinuerlig servo der vi bestemmer dreiehastighet og ikke vinkel:

- Hvor i sirkelbanen servoen treffer instrumentet
- Hvilken dreiehastighet som velges
- Hvor lenge den hastigheten beholdes
- Snur den retning på noe tidspunkt

Opplegg 10: Lyd og lydbølger

Kompetansemål:

Musikk:

- skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder
- utforske og formidle musikalske opplevelser og erfaringer, og reflektere over bruk av musikalske virkemidler
- lytte og prøve ut ulike uttrykk og begrunne valg i skapende prosesser fra idé til ferdig resultat
- bruke gehør og notasjonsteknikker som støtte i skapende arbeid
- bruke relevante fagbegreper i skapende arbeid og i refleksjon over prosesser og resultater

Matematikk:

- utforske korleis algoritmar kan skapast, testast og forbeholdast ved hjelp av programmering

Naturfag:

- bruke programmering til å utforske naturfaglige fenomener

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer:

Micro:bit med batteripakke, 2 alligatorledninger og en summer (buzzer) eller høyttaler

Ark, tynn kartong, fargede ark, lim, sakser, div. hobbyutstyr som fjær, silkepapir, avis, papir, kvister, kongler, småstein osv.

TIPS!

Bruk gjerne maskeringstape for å feste micro:bit og tilbehør med kortet. Maskeringstape er enkel å fjerne etter bruk slik at elektronikken ikke blir ødelagt. Pass på at elevene ikke limer det fast med lim!

Programmeringsoppgåver

For dette opplegget fins det to alternative «gjer-ferdig» oppgåver. Den første vil vere den enklaste, da det er med tonar som skal haldast i ulike tider. Den nyttigste hjelpa for elevar vil nok heller vere å hinte om å bruke noteark, og lese notene, heller enn å lytte seg fram. Men begge deler vil kunne fungere fint. Nedst på denne sida ligg eit heilt enkelt noteark som går an å dele ut til elevar som ønsker/treng det. Men helst bør dei kunne skaffe dette fram sjølv.

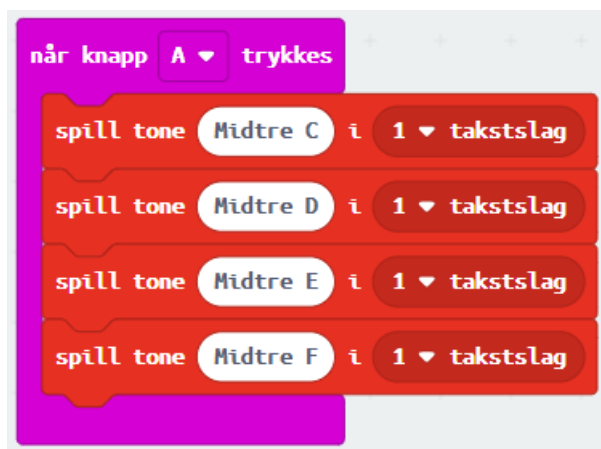
Gjer-ferdig-programmering

Bruk denne byrjinga av eit program og legg til kodeklossar slik at micro:biten speler «Lisa gikk til skolen».



Gjer-ferdig-programmering

Bruk denne byrjinga av eit program og legg til kodeklossar slik at micro:biten speler «Lisa gikk til skolen».



Lisa gikk til skolen

anon., arr. Frank Nordberg (Norway)



Løsningsforslag programmering

når knapp A trykkes

spill tone

Midtre C

i

1

▼

takstsløp

spill tone

Midtre D

i

1

▼

takstsløp

spill tone

Midtre E

i

1

▼

takstsløp

spill tone

Midtre F

i

1

▼

takstsløp

spill tone

Midtre G

i

2

▼

takstsløp

spill tone

Midtre G

i

2

▼

takstsløp

spill tone

Midtre A

i

1

▼

takstsløp

spill tone

Midtre A

i

1

▼

takstsløp

spill tone

Midtre A

i

1

▼

takstsløp

spill tone

Midtre A

i

1

▼

takstsløp

spill tone

Midtre G

i

4

▼

takstsløp

spill tone

Midtre F

i

1

▼

takstsløp

spill tone

Midtre F

i

1

▼

takstsløp

spill tone

Midtre F

i

1

▼

takstsløp

spill tone

Midtre F

i

1

▼

takstsløp

spill tone

Midtre E

i

2

▼

takstsløp

spill tone

Midtre E

i

2

▼

takstsløp

spill tone

Midtre D

i

1

▼

takstsløp

spill tone

Midtre D

i

1

▼

takstsløp

spill tone

Midtre D

i

1

▼

takstsløp

spill tone

Midtre D

i

1

▼

takstsløp

spill tone

Midtre C

i

4

▼

takstsløp

Programmeringsoppgåver Python

Puslespel-programmering

Bruk disse kodelinjene for å lage eit program som får micro:biten til å spele «Lisa gikk til skolen».

```
if button_a.was_pressed():
    music.play("A:1", "A", "A", "A", "G:4")
    music.play("F:1", "F", "F", "F", "E:2", "E")
    music.play("C4:1", "D", "E", "F", "G:2", "G")
    music.play("D:1", "D", "D", "D", "C:4")
from microbit import *
while True:
```

Gjer-ferdig-programmering

Bruk denne byrjinga av eit program og legg til kodelinjer slik at micro:biten speler «Lisa gikk til skolen»..

```
1 from microbit import*
2 from music import*
3
4 while True:
5     if button_a.was_pressed():
```

Løysingsframlegg programmering

```
1 from microbit import*
2 from music import*
3
4 while True:
5     if button_a.was_pressed():
6         music.play("C4:1", "D", "E", "F", "G:2", "G")
7         music.play("A:1", "A", "A", "A", "G:4")
8         music.play("F:1", "F", "F", "F", "E:2", "E")
9         music.play("D:1", "D", "D", "D", "C:4")
```

Opplegg 11: Toner og ringeklokke

Kompetansemål:

Musikk:

- skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder
- utforske og formidle musikalske opplevelser og erfaringer, og reflektere over bruk av musikalske virkemidler
- lytte og prøve ut ulike uttrykk og begrunne valg i skapende prosesser fra idé til ferdig resultat
- bruke gehør og notasjonsteknikker som støtte i skapende arbeid
- bruke relevante fagbegreper i skapende arbeid og i refleksjon over prosesser og resultater

Matematikk:

- utforske korleis algoritmar kan skapast, testast og forbeholdt ved hjelp av programmering

Naturfag:

- utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker
- bruke programmering til å utforske naturfaglige fenomener

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer:

Micro:bit med batteripakke, 2 alligatorledninger og en summer (buzzer) eller høyttaler

Papp, tynn kartong, fargede ark, lim, sakser, div. hobbyutstyr som fjær, silkepapir, avis, kvister, kongler, småstein osv.

TIPS!

Bruk gjerne maskeringstape for å feste micro:bit og tilbehør med resten av ringeklokka.

Maskeringstape er enkel å fjerne etter bruk slik at elektronikken ikke blir ødelagt. Pass på at elevene ikke limer det fast med lim!

Programmeringsoppgaver

Se opplegg 10 – Lyd og lydbølger for programmeringsoppgaver for å programmere musikk med micro:bit.

Musikkoppgåve

Kan du skrive tonenamnet og varigheita for alle notene under?



Tone							
Varigheit							



Tone							
Varigheit							

Opplegg 12: Teikne geometriske former med programmering

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske eigenskapane ved ulike polygonar og forklare omgrepa formlikskap og kongruens
- utforske og argumentere for korleis det å endre føresetnader i geometriske problemstillingar påverkar løysingar
- utforske og argumentere for formalar for areal og volum av tredimensjonale figurar

K&H:

- utvikle løysningar gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materiale: Ark eller vinyl

TIPS!

- Ta ein «screenshot» av den delen av skjermen som inneheld teikninga og skriv ut biletet på papir. Dette arket kan da inngå i klassen si matematiske fargeleggingsbok.
- Sjå gjerne opplegg 13 - Handlenett for korleis kunne få teikninga fra Scratch skoren ut i vinyl, dersom skulen har ein vinylkuttar.

Ekstraoppgåve

1. Gjer om alle måla på teikninga til verkelege mål for eit stort hus.
2. Finn ut kva målestokk er.
3. Finn målestokken for husteikninga di.

Krevjande ekstraoppgåve

1. Berekne arealet av det verkelege huset.
2. Kva blir tilhøvet mellom arealet til husteikinga og det store huset?
3. Berekne volumet av det verkelege huset.
4. Kva blir tilhøvet mellom volumet til husteikninga og det store huset?
5. Kva har dette med omgjerding mellom lengder, areal og volum å gjere?

Programmeringsoppgaver

I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Løsningsforslag på ulike geometriske figurer vises under.



Kvadrat



Likesidet trekant



Sirkel (tilnærmet)



Sekskant



Sletter tidligere figurer, og katten går til origo



Skruer av pennen og flytter katten bort fra tegningen

For tegning med Python, se kapittel 12.

Opplegg 13: Sirkler, kuler og naturressurser

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske og argumentere for korleis det å endre føresetnader i geometriske problemstillingar påverkar løysingar
- utforske og argumentere for formalar for areal og volum av tredimensjonale figurar

Naturfag:

- gi eksempler på og drøfte aktuelle dilemmaer knyttet til utnyttelse av naturressurser og tap av biologisk mangfold

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer: Ferdig handlenett eller stoff og gjordebånd som vist på mønsterarket per elev, samt symaskin og strykejern. Enten nål og tråd til å brodere mønster på veska, eller vliselin, stoff og stofflim, eller HTV-vinyl, vinylkutter og strykejern/strykepresse/varmepresse.

TIPS!

På neste side ligger mønster og fremgangsmåte for syng av handlenett. Dette er valgfritt, man kan heller kjøpe ferdige handlenett fra en rekke ulike fysiske- eller nettbutikker, f.eks: SKG emballasje, Kreativshop, CChobby, Søstrene Grene m.fl.

Man kan bare bruke tegningen som mønster og brodere det på handlenettet eller klippe ut stoff utfra mønsteret og lime på med stofflim. Da er det viktig å la Scratch-katten bevege seg bort fra tegningen uten at den tegner strek over den, ved å legge til eksempelkode under på slutten av tegnekode.

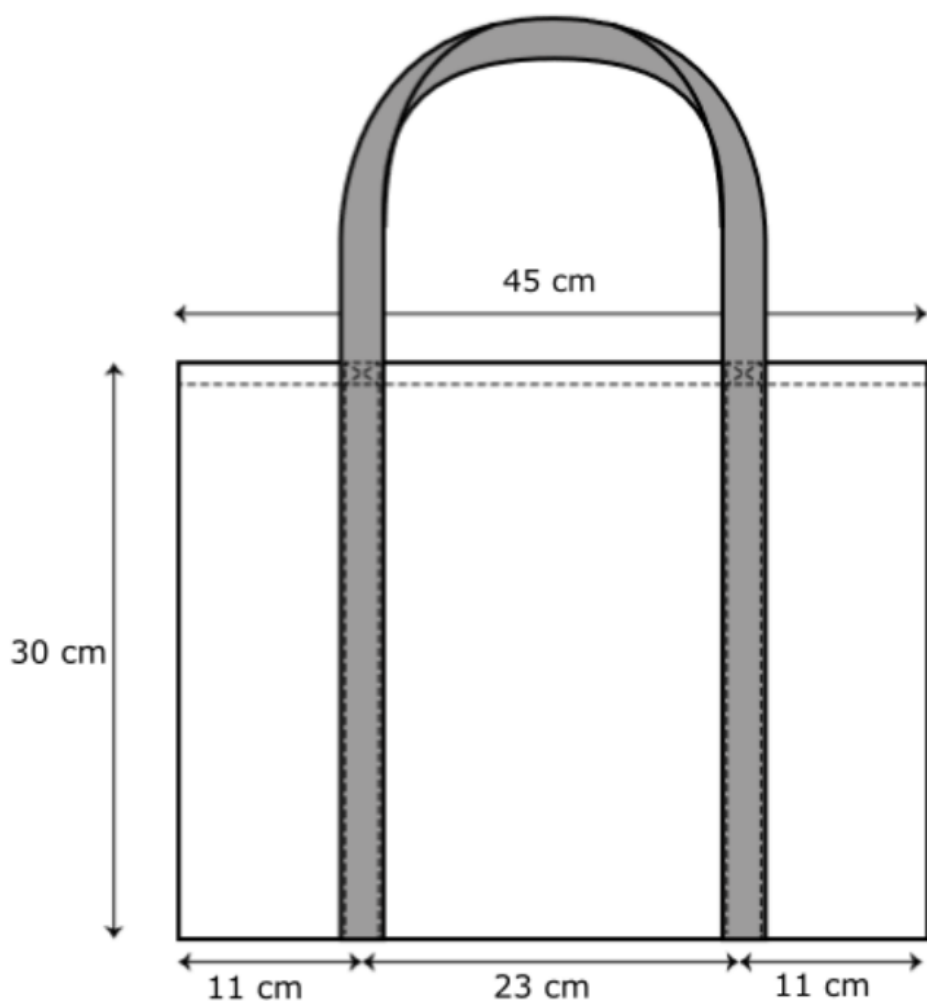
Deretter tar eleven et «screenshot» av den delen av skjermen sin som inneholder tegningen, og skriver ut bildet på papir.

Dersom man har tilgang på en vinylkutter, vil man ta en tilsvarende «screenshot» og lagre den som et bilde. Dette bildet åpnes i et vektorprogram, gjerne det som følger med vinylkutteren og man sporer (tracer) bildet for å få en kuttefil.

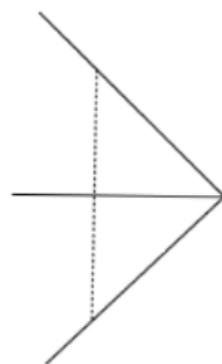
Dette kan også gjøres i programmet Inkscape (for PC og MAC som lastes ned her:

<https://inkscape.org/release/inkscape-0.92.3/>

Mønster og framgangsmåte for sying av handlenett



Figur 1



Figur 2

Framgangsmåte:

1. Klipp to stoffbitar på 45 cm × 35 cm.
2. Klipp to biter gjordebånd eller anna kraftig bånd på 120 cm kvar (eventuelt berre 50 cm kvar for å ha kort hank utan «striper» på handlenettet).
3. Brett over mot vranga og sy ein kant øvst på begge stoffstykkja på ca 2 cm.
4. Sy gjordebånda på slik figur 1 viser.
5. Sy eit kryss øvst på kvart gjordebånd (4 stader) for at hanken skal bli sterk nok.
6. Sy botnen i handlenettet saman frå vranga og stryk saumen flat.
7. Sy sidene på handlenettet saman frå vranga og stryk saumane flate.
8. Sy hjørna i botnen, som vist i figur 2 og stryk dei flate mot botnen.

Ekstraoppgåver

Ekstraoppgåve:

Kva tyder det når x- eller y-verdien er negativ?
Kva kan det aktuelle området for modellen vere?

Ekstraoppgåve:

Lag ei strekteikning med geometriske figurar (for hånd eller i Scratch). La sidemannen teikne denne (for hånd eller i Scratch) utan å sjå teikninga. Du skal forklare teikninga berre med å bruke geometriske begrep.

- Den som teiknar får ikkje stille spørsmål.
- Den som forklarar får ikkje sjå teikninga før ho er ferdig.
- Vis til lærar og forklar korleis de tenkte.

Programmeringsoppgaver

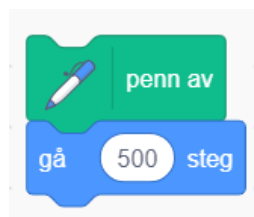
I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Løsningsforslag på ulike geometriske figurer vises under.



Sirkel (tilnærmet)



Sletter tidligere figurer, og katten går til origo



Skruer av pennen og flytter katten bort fra tegningen

For tegning med Python, se kapittel 12.

Opplegg 14: Volum for ulike geometriske former

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske og argumentere for korleis det å endre føresetnader i geometriske problemstillingar påverkar løysingar
- utforske og argumentere for formalar for areal og volum av tredimensjonale figurar

Naturfag:

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materiale:

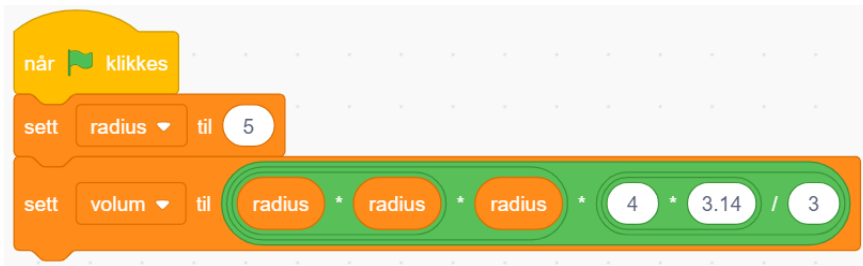
Leire eller play-doh eller tilsvarande og ein stor målesylinder.

Ekstraoppgåve

1. Korleis kan de avgjere kva for ein modell som passar best?
2. Passar modellen bra med målingane?
3. Kva tyder det når x- eller y-verdien er negativ?
4. Kva kan det aktuelle området for modellen vere?

Programmeringsoppgaver

I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Løsningsforslag for en kule vises under.



For regning med Python, se kapittel 12. Løsningsforslag for kule vises under.

```
1 radius = 5
2 volum = (4 * radius**3 * 3.14)/3
3 print(volum)
```

Opplegg 15: Matematiske mønster

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- beskrive, forklare og presentere strukturar og utviklingar i geometriske mønster og i talmønster
- utforske eigenskapane ved ulike polygonar og forklare omgrepa formlikskap og kongruens
- utforske og argumentere for korleis det å endre føresetnader i geometriske problemstillingar påverkar løysingar
- utforske og argumentere for formlar for areal og volum av tredimensjonale figurar

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer: Ferdig putetrekk (eller stoff og materialer for å sy et putetrekk). Enten nål og tråd til å brodere mønster på veska, eller vliselin, stoff og stofflim, eller HTV-vinyl, vinylkutter og strykejern/strykepresse/varmepresse.

TIPS!

Man kan bare bruke tegningen som mønster og brodere det på putetrekket eller klippe ut stoff utfra mønsteret og lime på med stofflim. Da er det viktig å la Scratch-katten bevege seg bort fra tegningen uten at den tegner strek over den, ved å legge til eksempelkode under på slutten av tegnekode.

Deretter tar eleven et «screenshot» av den delen av skjermen sin som inneholder tegningen, og skriver ut bildet på papir.

Dersom man har tilgang på en vinylkutter, vil man ta en tilsvarende «screenshot» og lagre den som et bilde. Dette bildet åpnes i et vektorprogram, gjerne det som følger med vinylkutteren og man sporer (tracer) bildet for å få en kuttefil.

Dette kan også gjøres i programmet Inkscape (for PC og MAC som lastes ned her:

<https://inkscape.org/release/inkscape-0.92.3/>

Deloppgave a) til den krevende ekstraoppgaven er ganske omfattende og kan gjøres på flere måter, men det ligger et løsningsforslag her. Det kan tenkes at elevene kan prøve selv først, og dersom de ikke kommer i mål med veiledning, så kan de få se på løsningsforslaget.

Refleksjonsspørsmål: Hva vil skje dersom det står noe annet enn 0 for flere av verdiene?

Ekstraoppgåve:

1. Lag eit program som gjer om frå meter til nokre andre lengdeiningar.
2. Lag eit program som gjer om frå kvadratmeter (m^2) til nokre andre flateiningar (areal).
3. Lag eit program som gjer om frå kubikkmeter (m^3) til nokre andre volumeiningar.

Krevjande ekstraoppgåve:

- a) Lag eit program som tek inn ei lengd og kan gjere om til dei andre lengdeiningane.
- b) Lag eit program som kan ta inn eit areal og gjere om til alle de andre einingane for flate.
- c) Lag eit program som kan ta inn eit volum og gjere om til alle dei andre einingane for volum.
- d) Lag ei forklaring eller vis matematisk kvifor ein med omgjerung mellom:
 - lengdeiningar må multiplisere/dividere med 10 per eining
 - flateiningar må multiplisere/dividere med 100 per eining
 - volumeiningar må multiplisere/dividere med 1000 per eining
- e) Kan du lage ein god hugseregul for når ein skal multiplisere eller dividere for å gjere om på einingane?

Programmeringsoppgaver

I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Løsningsforslag på ulike geometriske figurer vises under.



Kvadrat



Likesidet trekant



Sirkel (tilnærmet)



Sekskant



Sletter tidligere figurer, og katten går til origo



Skruer av pennen og flytter katten bort fra tegningen

For tegning med Python, se kapittel 12.

Programmeringsoppgaver ekstraoppgaver

- løsningsforslag

Det blir tilsvarende programmer for areal og volum, man må bare endre til at man multipliserer/dividerer med 100 for hver enhet innen areal og 1000 for volum. Her må man sette inn hvilken verdi man har i meter før man kjører programmet. Det går også an å definere det som en funksjon som tar inn lengden i meter.



Løsningsforslag krevende ekstraoppgave



Opplegg 16: Sannsynlighet og representasjoner

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- berekne og vurdere sannsyn i statistikk og spel
- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe, ved å bruke programmering

Naturfag:

- bruke programmering til å utforske naturfaglige fenomener
- beskrive drivhuseffekten og gjøre rede for faktorer som kan forårsake globale klimaendringer
- beskrive hvordan forskere har kommet fram til evolusjonsteorien og bruke denne til å forklare utvikling av biologisk mangfold
- gi eksempler på og drøfte aktuelle dilemmaer knyttet til utnyttelse av naturressurser og tap av biologisk mangfold

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer: Leire/play-doh/trolldeig, papp, papir, sakser, limpistoler, tape, fjær, maling, silkepapir, glitter +++

TIPS!

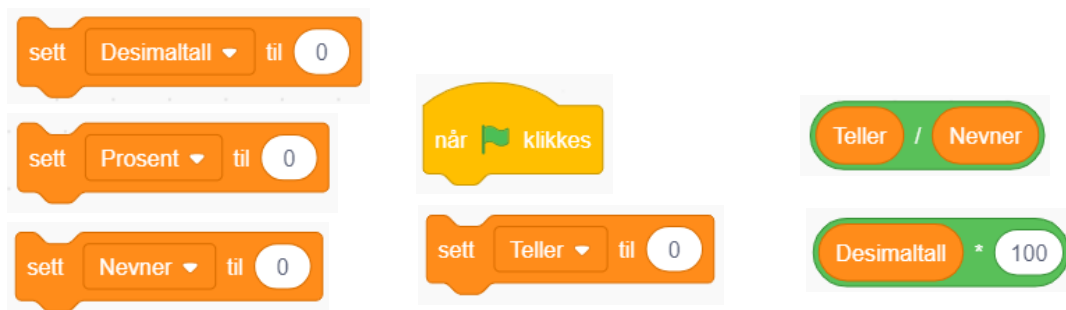
Assister gjerne elevene med å resonnerer fram til hvordan arten kan se ut for å være bedre tilpasset klimaendringene. Først, hvilke konsekvenser får klimaendringene? Jo, mer ekstremvær som mer tørke, flere stormer, flere og større oversvømmelser, flere skogbranner osv. Videre, kan man tenke på hvilke egenskaper som kan være nyttige for dyr å ha i de tilfellene. Kanskje vinger, slik at de kan fly bort fra naturkatastrofer? Eller svømmehud for å kunne navigere om det blir flom?

Det er fint om elevene tar utgangspunkt i en kjent art, og tilpasser den, det blir tettest opptil den faktiske evolusjonen. Men elevene kan velge selv, og som en del av oppsummeringen, kan man te en liten diskusjon på hvorfor det er mest realistisk å ta utgangspunkt i en eksisterende art.

Programmeringsoppgåver

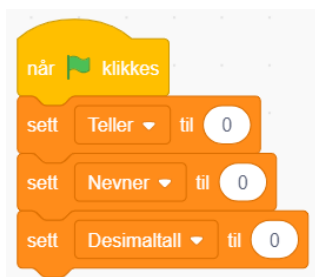
Puslespel-programmering

Bruk desse blokkene for å lage eit program som gjer om frå ein brøk til desimaltal og prosent.



Gjer ferdig-programmering

Bruk desse blokkene og hald fram med koden slik at det blir eit program som gjer om frå ein brøk til desimaltal og prosent.



Løysingsframlegg programmering

I Scratch viser alltid verdiane til alle variablane, så ein treng ikkje å bruke blokka «vis variabel». Men skal ein over til tekstprogrammering, må ein skrive verdiane av variablar til skjerm for å sjå dei. Dermed er det første alternativet enklare å setje om til tekstprogrammering.



Programmeringsoppgåver Python

Puslespel-programmering

Bruk disse kodelinjene for å lage eit program som gjer om frå ein brøk til desimaltal og prosent.

```
print(Prosent)

print(Desimaltall)

Desimaltall = Teller / Nevner

Prosent = Desimaltall * 100

Teller = 2

Nevner = 5
```

Gjer ferdig-programmering

Bruk disse kodelinjene og hald fram med koden slik at det blir eit program som gjer om frå ein brøk til desimaltall og prosent.

```
Teller = 2
Nevner = 5

Desimaltall = Teller / Nevner
print(Desimaltall)
```

Løysingsframlegg programmering

```
1 Teller = 2
2 Nevner = 5
3
4 Desimaltall = Teller / Nevner
5 print(Desimaltall)
6
7 Prosent = Desimaltall * 100
8 print(Prosent)
```

Opplegg 17: Uniform sannsynlighet

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- berekne og vurdere sannsyn i statistikk og spel
- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe, ved å bruke programmering

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Utstyr: Splittbinders, papir, sakser, lim, papp (gjerne farget) til lykkehjul, playdoh/leire/trolldeig til terning

TIPS!

Det er lurt å passe på at elevene ikke lager terninger med altfor mange sider eller lykkehjul med utrolig mange valg, i så fall bør oppgaven med valgtre og spill tilpasses litt slik at det ikke blir altfor mange mulige utfall de skal simulere og beregne sannsynligheten for.

Programmeringsoppgåver

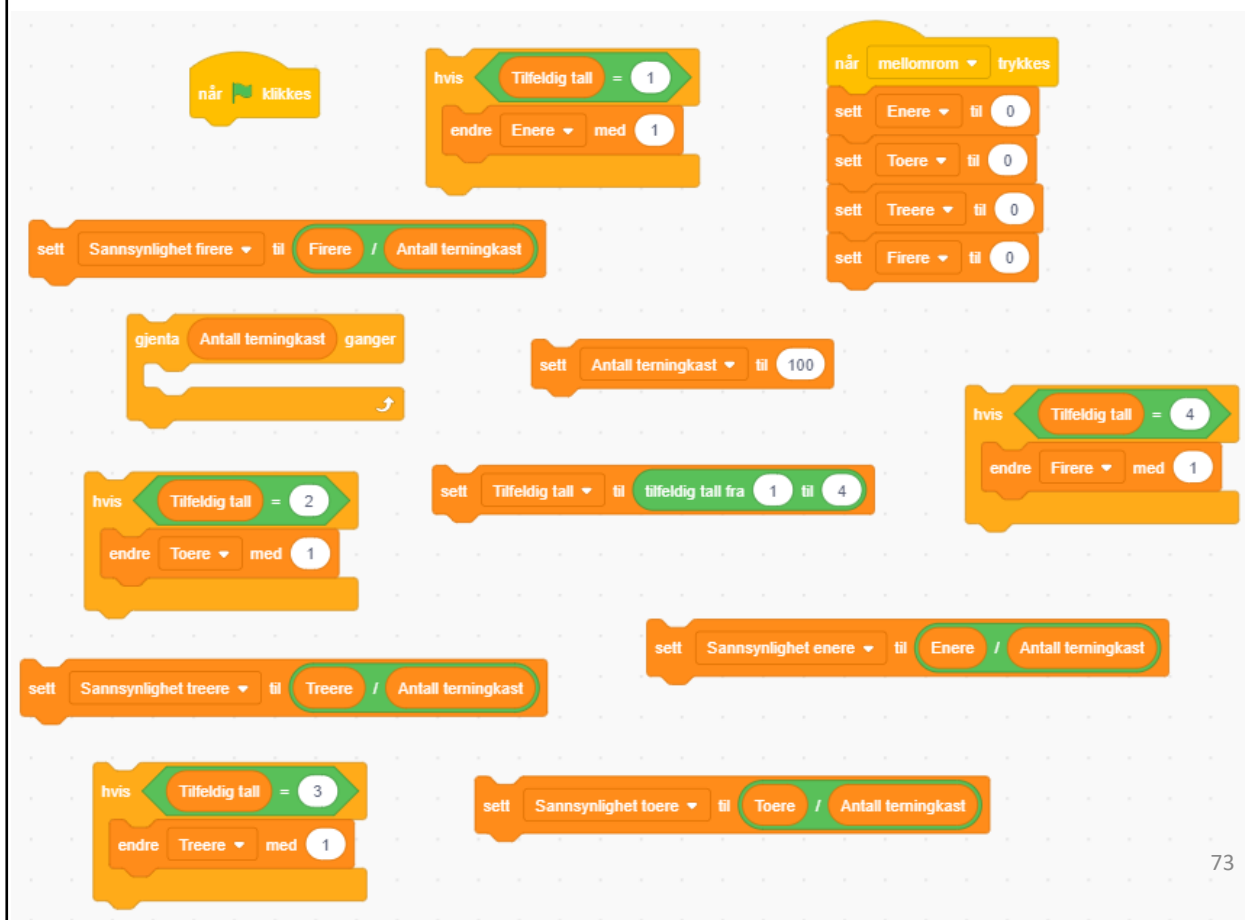
Gjer ferdig-programmering

Bruk desse klossane og hald fram koden slik at det blir eit program som simulerer og berekner sannsyn.



Puslespel-programmering

Bruk desse klossane for å lage eit program som simulerer og berekner sannsyn.



Løsningsforslag

– hvorfor bør den lille delen til høyre være med?



Programmeringsoppgåver i Python

Puslespel-programmering

Bruk desse kodelinjene for å lage eit program som simulerer og berekner sannsyn.

```
femmere = 0
femmere = femmere + 1
from pylab import *
terning = randint(1, 7)
kast = 100000
for i in range(0, kast):
    if terning == 5:
```

Gjer ferdig-programmering

Bruk desse kodelinjene og hald fram med koden slik at det blir et program som simulerer og berekner sannsyn.

```
from pylab import *

kast = 100000
femmere = 0

for i in range(0, kast):
    terning = randint(1, 7)
```

Løysingsframlegg programmeringsoppgåve

```
1 from pylab import *
2
3 kast = 100000
4 femmere = 0
5
6 for i in range(0, kast):
7     terning = randint(1, 7)
8     if terning == 5:
9         femmere = femmere + 1
10
11 sannsynlighet = femmere / kast
12 print(sannsynlighet)
```

Her blir berre femmarar talt opp, men ein kan legge til variablar for dei andre utfalla og.

Det er og berre sannsynet for å få femmar som blir vist på skjermen.

Opplegg 18: Antall mulige kombinasjoner

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- berekne og vurdere sannsyn i statistikk og spel
- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe, ved å bruke programmering

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Utstyr:

Micro:bit med batteripakke.

Papir, sakser, lim, papp, play-doh/leire/trolldeig +++

TIPS!

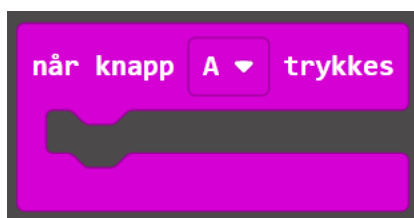
Bruk gjerne maskeringstape for å feste micro:bit, LED-pærer, alligatorledninger o.l. med modellen. Maskeringstape er enkel å fjerne etter bruk slik at elektronikken ikke blir ødelagt. Pass på at elevene ikke limer det fast med lim!

Programmeringsoppgåver

Sjå opplegg 17 – uniformt sannsyn, for å lage eit program som simulerer og berekner sannsyn.

Puslespel-programmering

Bruk desse klossane for å lage eit program som simulerer trekning av spådom med hjelp av micro:biten



Løysingsframlegg programmeringsoppgave for micro:bit



Programmeringsoppgåver i Python

Puslespel-programmering

Bruk desse kodelinjene for å lage eit program som simulerer trekninga av ein spådom med micro:biten.

```
if button_a.was_pressed():  
    display.scroll(randint(1, 10))  
from microbit import *  
while True:
```

Gjer ferdig-programmering

Bruk desse kodelinjene og hald fram med koden slik at det blir eit program som simulerer trekninga av ein spådom med micro:biten.

```
from microbit import *  
  
while True:
```

Løysingsframlegg programmeringsoppgåve

```
from microbit import *  
  
while True:  
    if button_a.was_pressed():  
        display.scroll(randint(1, 10))
```

Opplegg 19: Sannsynlighet og valgtre

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- berekne og vurdere sannsyn i statistikk og spel
- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe, ved å bruke programmering

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Utstyr:

Papir, sakser, lim, papp, play-doh/leire/trolldeig (til brikker) +++

Terning eller lykkehjul fra opplegg 17.

TIPS!

Dersom elevene sliter med å tenke ut hvilke typer spill de kan lage, så er yatzy, stigespill og terningspill der man skal komme først i mål (men har ulike hindringer underveis) være greie forslag.

Programmeringsoppgåver

Sjå opplegg 17 – uniformt sannsyn, for å lage eit program som simulerer og bereknar sannsyn.

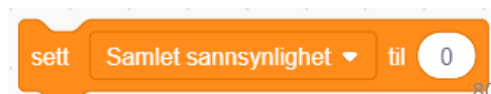
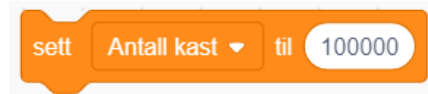
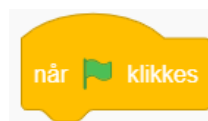
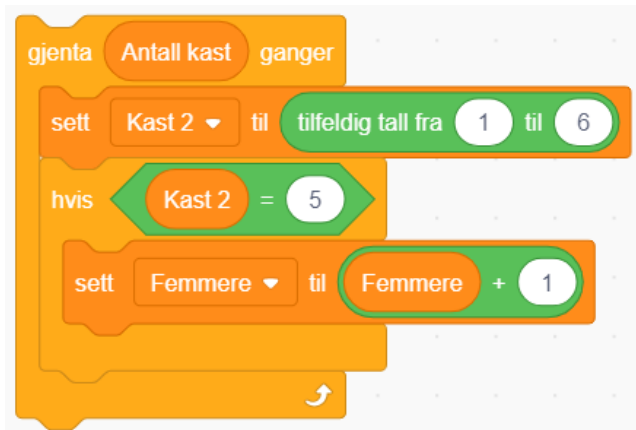
Gjer ferdig-programmering

Bruk desse klossane for å lage eit program som simulerer og bereknar sannsyn for to terningkast eller lukkehjul-snurrar.



Puslespel-programmering

Bruk desse klossane for å lage eit program som simulerer og bereknar sannsyn for to terningkast eller lukkehjul-snurrar.



Løsningsforslag



Grunnen til at det er brukt to omganger med simulering for de to terningene i løsningsforslaget, er at de to terningkastene er uavhengige hendelser. Dersom man bare tar en simulering og «plukker ut» to forskjellige utfall (for eksempel firere og femmere), så vil ikke antall firere og femmere være uavhengige av hverandre. Dette vil føre til minimale konsekvenser dersom antall terningkast er høyt, men kan få store konsekvenser om man velger å kaste bare 100 ganger for eksempel. Diskuter gjerne med elever. Spesielt hvordan koden kan forenkles ved å bruke bare en løkke for å gjennomføre begge simuleringene. Det blir da essensielt å bruke `randint()`-funksjonen to ganger i samme løkke, for at hendelsene skal være uavhengige.

Programmeringsoppgåver i Python

Puslespel-programmering

Bruk disse kodelinjene for å lage et program som simulerer og berekner sannsyn for to terningkast eller lukkehjul-snurrar.

```
femmere = 0
femmere = femmere + 1
from pylab import *
terning = randint(1, 7)
kast = 100000
for i in range(0, kast):
    if terning == 5:
        sannsynlighet1 = firere / kast
        samletsannsynlighet = sannsynlighet1 * sannsynlighet2
        sannsynlighet2 = femmere / kast

firere = 0
if terning == 4:
    firere = firere + 1
terning = randint(1, 7)
terning = randint(1, 7)
print(samletsannsynlighet)
```

Gjer ferdig-programmering

Bruk disse kodelinjene for å lage eit program som simulerer og berekner sannsyn for to terningkast eller lukkehjul-snurrar.

```
from pylab import *

kast = 100000
firere = 0
femmere = 0

for i in range(0, kast):
    terning = randint(1, 7)
    if terning == 4:
        firere = firere + 1

sannsynlighet1 = firere / kast
```

Løsningsforslag programmering i Python

```
1 from pylab import *
2
3 kast = 100000
4 firere = 0
5 femmere = 0
6
7 for i in range(0, kast):
8     terning = randint(1, 7)
9     if terning == 4:
10         firere = firere + 1
11
12 sannsynlighet1 = firere / kast
13
14 for i in range(0, kast):
15     terning = randint(1, 7)
16     if terning == 5:
17         femmere = femmere + 1
18
19 sannsynlighet2 = femmere / kast
20 samletsannsynlighet = sannsynlighet1 * sannsynlighet2
21 print(samletsannsynlighet)
```

Grunnen til at det er brukt to omganger med simulering for de to terningene i løsningsforslaget, er at de to terningkastene er uavhengige hendelser. Dersom man bare tar en simulering og «plukker ut» to forskjellige utfall (for eksempel firere og femmere), så vil ikke antall firere og femmere være uavhengige av hverandre. Dette vil føre til minimale konsekvenser dersom antall terningkast er høyt, men kan få store konsekvenser om man velger å kaste bare 100 ganger for eksempel. Diskuter gjerne med elever. Spesielt hvordan koden kan forenkles ved å bruke bare en løkke for å gjennomføre begge simuleringene. Det blir da essensielt å bruke randint()-funksjonen to ganger i samme løkke, for at hendelsene skal være uavhengige.

Opplegg 20: Statistikk og sentralmål

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- finne og diskutere sentralmål og spreiingsmål i reelle datasett

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Materialer:

- Eventuelt materialer til å bygge solfanger (og termometer, om dere ikke har gjort solfangeropplegget før, eller micro:bit som programmeres for å måle temperatur)
- Materialer til å lage utstilling (Ark, tusj og tape, saks, papp, garn, klistermerker, silkepapir, glitter++).

TIPS!

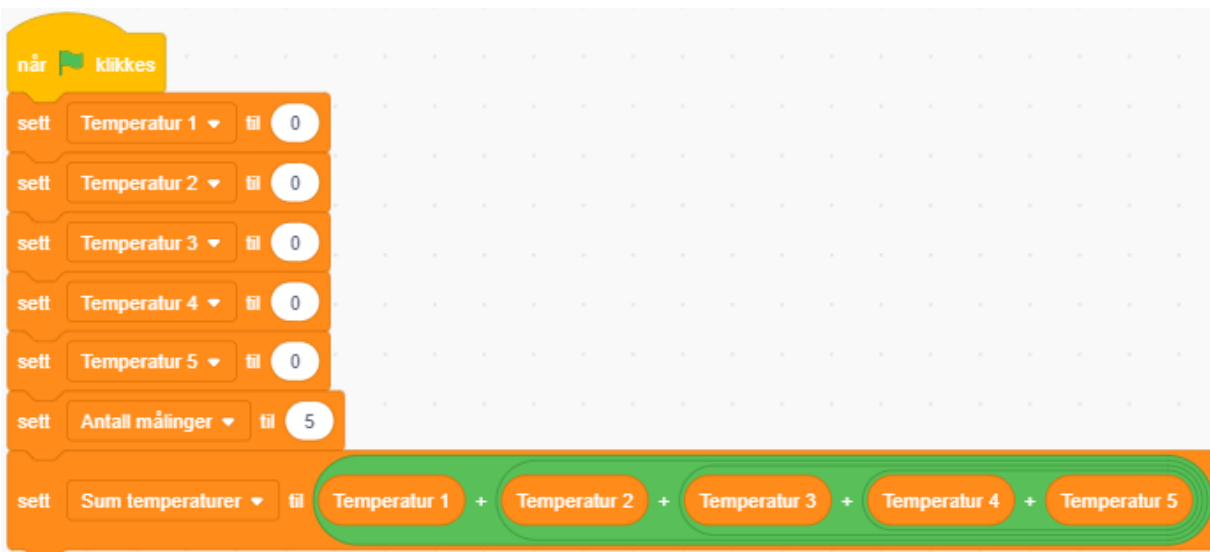
I denne oppgaven bør man ha gjort solfanger-opplegget først.

Snakk sammen-oppgaven: Her er poenget å få elevene til å forstå at når man finner median i et datasett, så får verdiene som avviker mye fra resten av materialet mindre å si enn for gjennomsnittet. Man kan også bruke skolefravær som eksempel. Dersom alle i klassen har et fravær på mindre enn 2 dager i løpet av et skoleår, men en elev var uheldig og ble syk. Eleven fikk et fravær på 23 dager. For gjennomsnittet vil den uheldige eleven trekke gjennomsnittet ganske mye opp, spesielt om det er få elever i klassen. Dersom vi heller ser på medianen, så vil den maksimalt forskyves ett hakk opp mot en elev med bare lite grann høyere fravær.

Programmeringsoppgåver

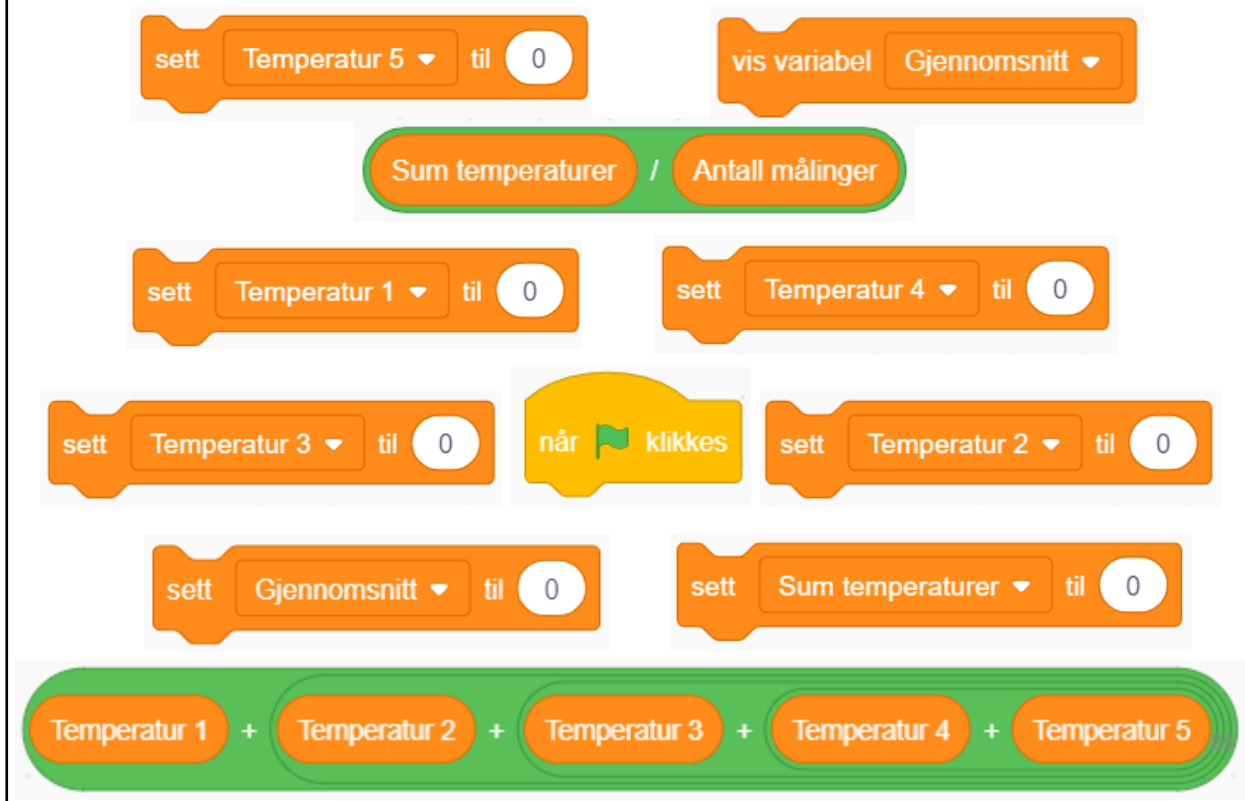
Gjer ferdig-programmering

Bruk desse klossane og hald fram med koden slik at det blir eit program som reknar ut gjennomsnittet for fem temperaturmålingar.



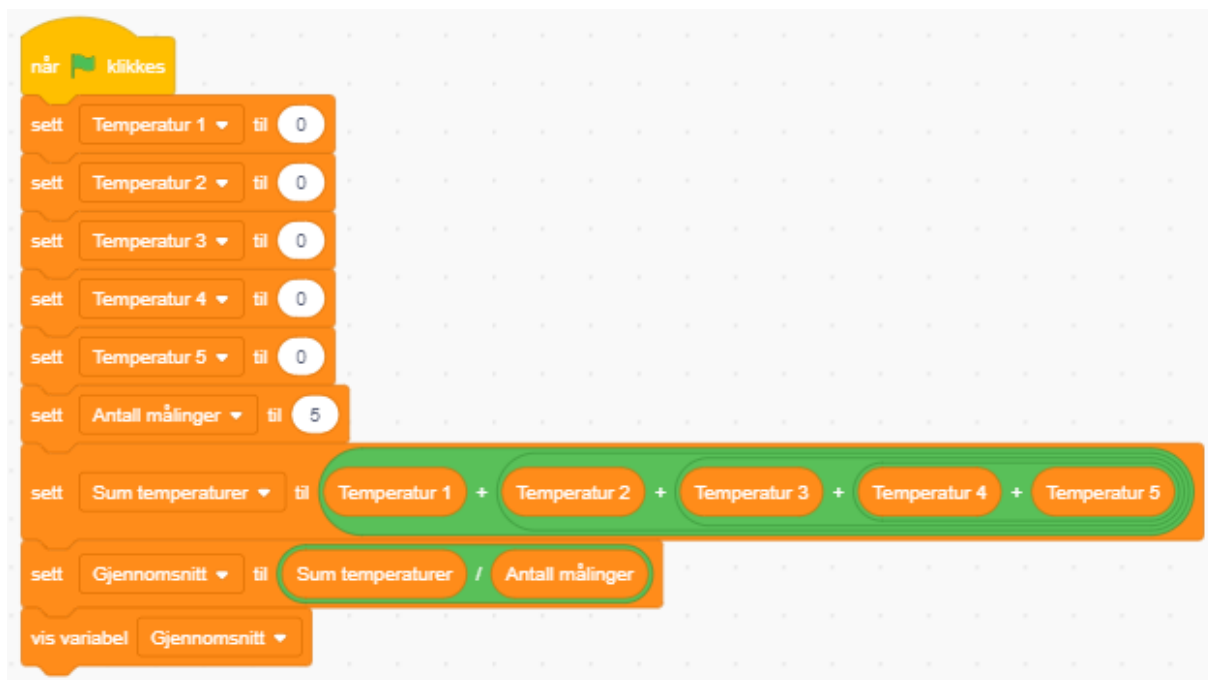
Puslespel-programmering

Bruk desse klossane for å lage eit program som reknar ut gjennomsnittet for fem temperaturmålingar.

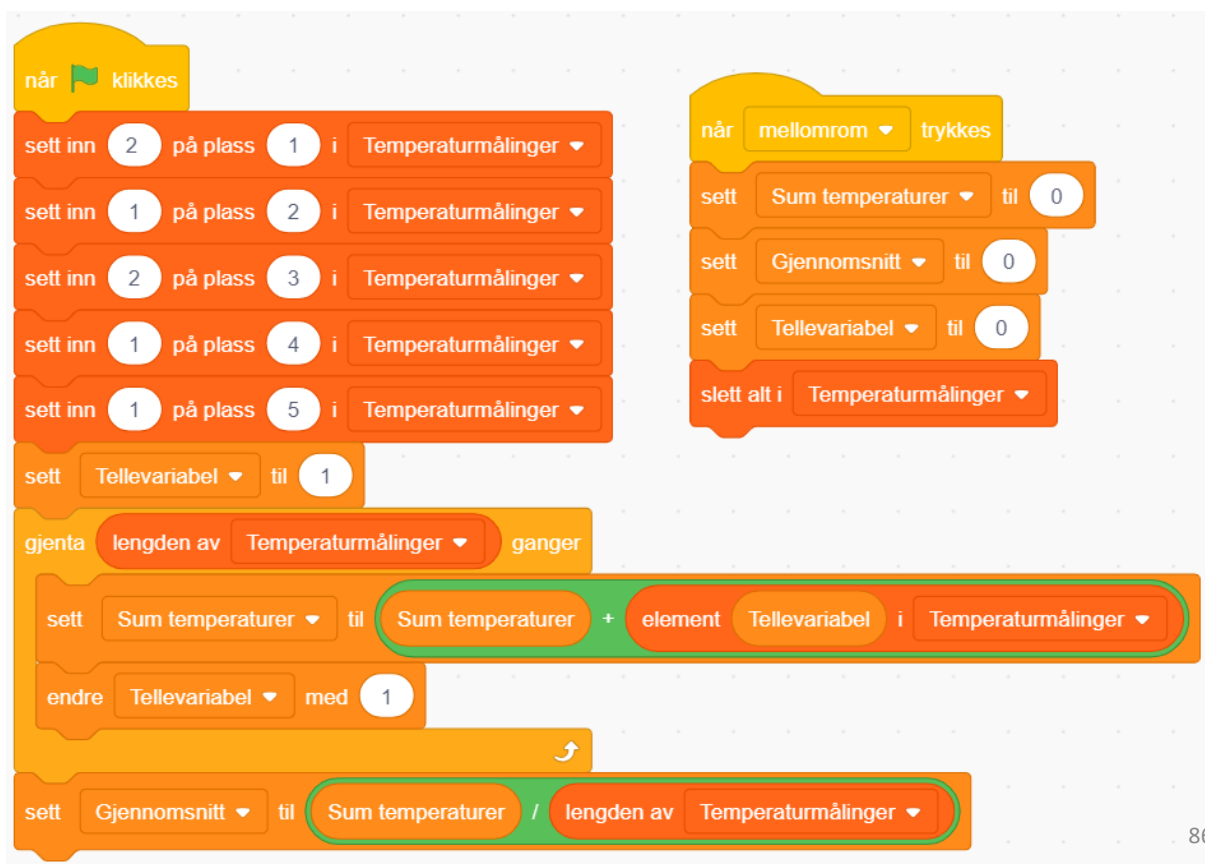


Programmering løsningsforslag

Løsningsforslag (variant uten bruk av lister)



Løsningsforslag (variant med bruk av lister)



Programmeringsoppgåver i Python

Puslespel-programmering

Bruk desse kodelinjene for å lage eit program som reknar ut gjennomsnitt av 10 temperaturmålingar.

```
i = 0
i = i+1
summen = 0
print(gjennomsnitt)
while i < len(inndata):
    summen = summen + inndata[i]
    gjennomsnitt = summen/len(inndata)
inndata = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Gjer ferdig-programmering

Bruk desse kodelinjene og hald fram med koden slik at det blir eit program som reknar ut gjennomsnittet av 10 temperaturmålingar.

```
inndata = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
i = 0
summen = 0

while i < len(inndata):
```

Løsningsforslag programmering i Python

```
1 inndata = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2 i = 0
3 summen = 0
4
5 while i < len(inndata):
6     summen = summen + inndata[i]
7     i = i+1
8
9 gjennomsnitt = summen/len(inndata)
10 print(gjennomsnitt)
```

Her er det tatt med 10 målinger for å vise elevene hvor enkelt det er å ta med mange målinger. Det er også bare tallene 1 til 10 man regner ut gjennomsnittet av, så elevene må endre tallene til sine målinger for å få det oppgaven spør etter.

Opplegg 21: Frekvenstabell og sektordiagram

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske og argumentere for korleis framstillingar av tal og data kan brukast for å fremje ulike synspunkt

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

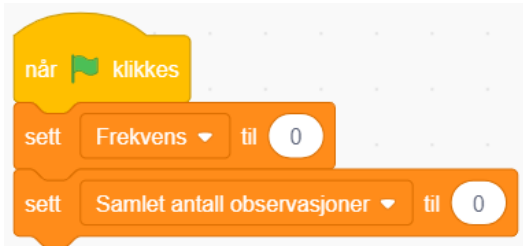
Materialer:

- Materialer til å lage utstilling (Ark, tusj og tape, saks, papp, garn, klistermerker, silkepapir, glitter++).

Programmeringsoppgåver

Gjer ferdig-programmering

Bruk desse klossane for å lage eit program som reknar ut vinkelen i sektordiagrammet.



Puslespel-programmering

Bruk desse klossane for å lage eit program som reknar ut vinkelen i sektordiagrammet.



Løysingsframlegg sektordiagram
(vinkelberegning)



Programmeringsoppgåver Python

Puslespel-programmering

Bruk desse kodelinjene for å lage eit program som reknar ut alle vinklane som svarer til dei ulike frekvensane, til bruk i sektordiagrammet.

```
frekvenser = [1, 4, 7, 3, 5]      print(vinkel[j])

summen = 0                        vinkel = frekvenser
                                j = j+1                        i = 0
j = 0
                                while i < len(frekvenser):
summen = summen + frekvenser[i]    i = i+1
                                while i < len(frekvenser):

vinkel[j] = (frekvenser[j]/summen)*360
```

Gjer ferdig-programmering

Bruk desse kodelinjene og hald fram med koden slik at det blir eit program som reknar ut alle vinklane som svarer til dei ulike frekvensane, til bruk i sektordiagrammet.

```
frekvenser = [1, 4, 7, 3, 5]
summen = 0
vinkel = frekvenser
i = 0
j = 0

while i < len(frekvenser):
    summen = summen + frekvenser[i]
    i = i+1
```

Løsningsforslag programmering i Python

```
1 frekvens = 1
2 samlet_antall = 20
3
4 vinkel = (frekvens/samlet_antall)*360
5
6 print(vinkel)
```

Med bruk av lister for frekvensene:

```
1 frekvenser = [1, 4, 7, 3, 5]
2 summen = 0
3 vinkel = frekvenser
4 i = 0
5 j = 0
6
7 while i < len(frekvenser):
8     summen = summen + frekvenser[i]
9     i = i+1
10
11 while j < len(frekvenser):
12     vinkel[j] = (frekvenser[j]/summen)*360
13     print(vinkel[j])
14     j = j+1
```

Opplegg 22: Statistikk og diagrammer

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske og argumentere for korleis framstillingar av tal og data kan brukast for å fremje ulike synspunkt

Naturfag

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

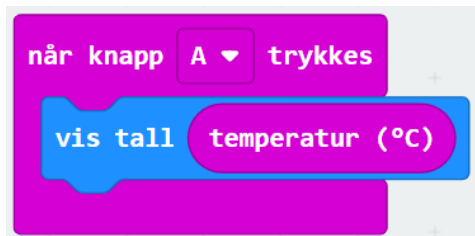
- Micro:bit med batteripakke
- Materialer til å lage utstilling (Ark, tusj og tape, saks, papp, garn, klistermerker, silkepapir, glitter++).

TIPS!

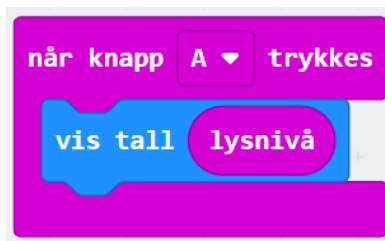
Eksempler på mulig datainnsamling med micro:bit: Temperaturendring i termosier eller solfangeren, temperatur etter en viss tid i mange forskjellige termosier, romtemperatur i forskjellige rom, utetemperatur i løpet av en dag, lysstyrke inne/ute, som tellehjelp (teller +1 hver gang man trykker inn en knapp) for: Antall biler av en viss farge, antall steg på en viss avstand, antall mennesker med langt/mellomlangt/kort hår osv.

Programmeringsoppgaver

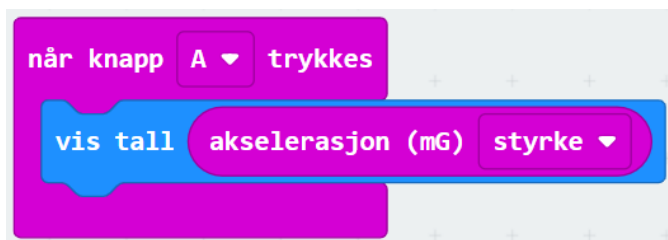
I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Løsningsforslag på ulike målinger ved hjelp av micro:bit vises under.



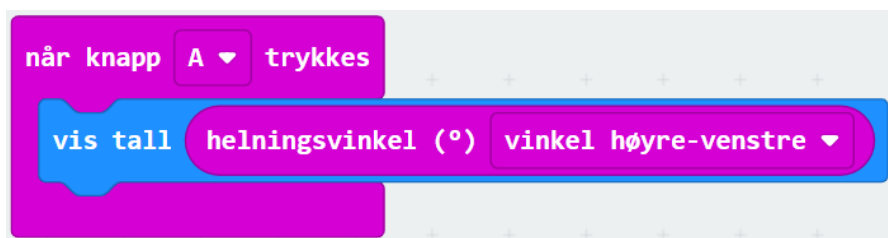
Viser temperatur når knapp A trykkes.



Viser lysnivå når knapp A trykkes.



Viser akselerasjonen (summert for alle retninger) når knapp A trykkes.



Viser helningsvinkelen til micro:biten når knapp A trykkes.



Eksempel på en «tellemaskin» der man øker antall ved å trykke på knapp A, og viser antallet på micro:bitens skjerm ved å trykke på knapp B.

Løsningsforslag programmering i Python

I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Se løsningsforslag for programmer for målinger ved hjelp av micro:bit i elevboka i kapittel 12.

Opplegg 23: Statistikk og «fake news»

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- tolke og kritisk vurdere statistiske framstillinger frå media og lokalsamfunnet
- utforske og argumentere for korleis framstillingar av tal og data kan brukast for å fremje ulike synspunkt

Naturfag (dette kommer an på hvilke data elevene samler inn)

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

- Micro:bit med batteripakke
- Materialer til å lage utstilling (Ark, tusj og tape, saks, papp, garn, klistermerker, silkepapir, glitter++).

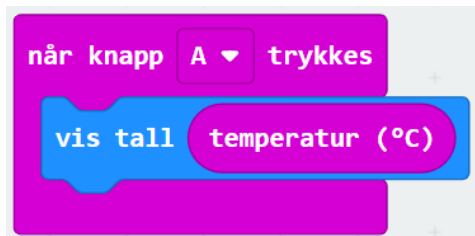
TIPS!

Eksempler på mulig datainnsamling med micro:bit: Temperaturendring i termosier eller solfangeren, temperatur etter en viss tid i mange forskjellige termosier, romtemperatur i forskjellige rom, utetemperatur i løpet av en dag, lysstyrke inne/ute, som tellehjelp (teller +1 hver gang man trykker inn en knapp) for: Antall biler av en viss farge, antall steg på en viss avstand, antall mennesker med langt/mellomlangt/kort hår osv.

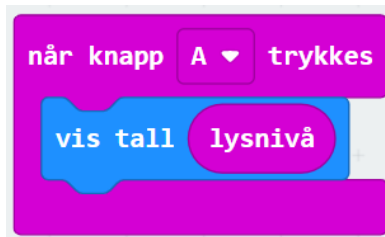
Størsteparten av poenget med denne oppgaven er å trene elevene i å kritisk vurdere informasjon fra forskjellige kilder, spesielt med tanke på statistiske framstillinger. I utgangspunktet i denne boka skal elevene lage en utstilling basert på de tre forskjellige statistikk-oppleggene. Blant utstillingen skal det da være en ekte nyhetssak som elevene har laget og en nyhetssak basert på dårlig-/juksestatistikk. En viktig del av denne oppgaven blir det å la elevene se hverandres utstillinger når de er ferdige og prøve å identifisere hvilke nyhetssaker som er ekte, og hvilke som er falske. Deretter bør man ta en diskusjon i klassen der man får fram hva elevene brukte som kriterier for å identifisere de falske sakene. Var dette gode kriterier – virket de? Vil de være gode å bruke generelt? Få med elevene på en «tenke-høyt» diskusjon rundt temaet, og bruk gjerne tidsaktuell villedende statistikk. Bruk gjerne den tilhørende «statistikk og kildekritikk» filmen.

Programmeringsoppgaver

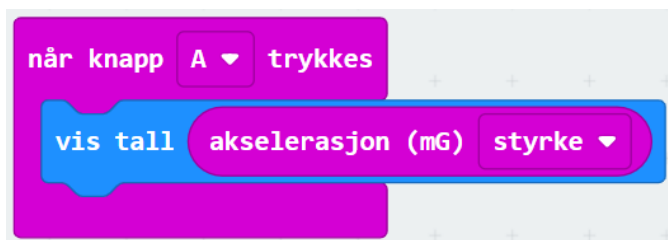
I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Løsningsforslag på ulike målinger ved hjelp av micro:bit vises under.



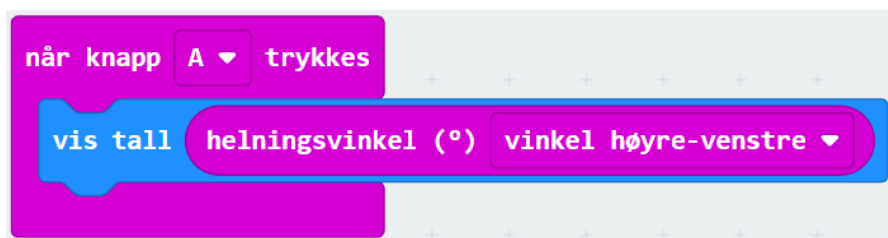
Viser temperatur når knapp A trykkes.



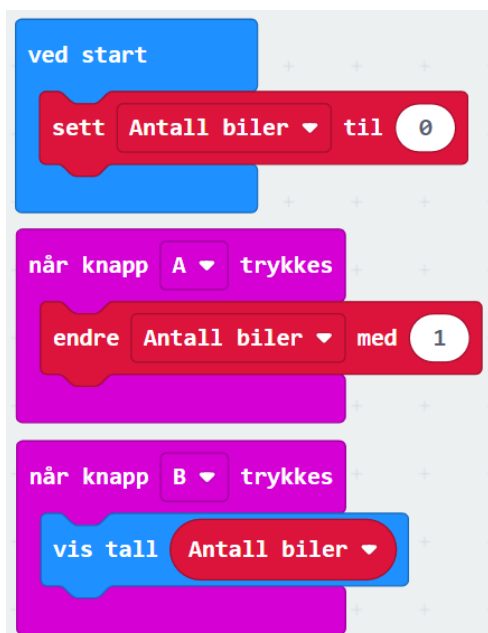
Viser lysnivå når knapp A trykkes.



Viser akselerasjonen (summert for alle retninger) når knapp A trykkes.



Viser helningsvinkelen til micro:biten når knapp A trykkes.



Eksempel på en «tellemaskin» der man øker antall ved å trykke på knapp A, og viser antallet på micro:bitens skjerm ved å trykke på knapp B.

Løsningsforslag programmering i Python

I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Se løsningsforslag for programmer for målinger ved hjelp av micro:bit i kapittel 12.

Opplegg 24: Statistikk og spredningsmål

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- finne og diskutere sentralmål og spreiingsmål i reelle datasett
- utforske og argumentere for korleis framstillingar av tal og data kan brukast for å fremje ulike synspunkt

Naturfag (dette kommer an på hvilke data elevene samler inn)

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

- Micro:bit med batteripakke
- Materialer til å lage utstilling (Ark, tusj og tape, saks, plast, trolldeig, papp, garn, klistermerker, silkepapir, glitter++).

Programmeringsoppgåver

Gjer ferdig-programmering

Bruk desse klossane for å lage eit program som reknar ut standardavvik.

Scratch code blocks for calculating standard deviation:

- gjernta Antall målinger ganger
- endre Sum med element Tellevariabel i Målinger - Gjennomsnitt * element Tellevariabel i Målinger - Gjennomsnitt * element Tellevariabel i Frekvenser
- endre Tellevariabel med 1
- sett Standardavvik til kvadratrota av Sum / Antall målinger

Puslespel-programmering

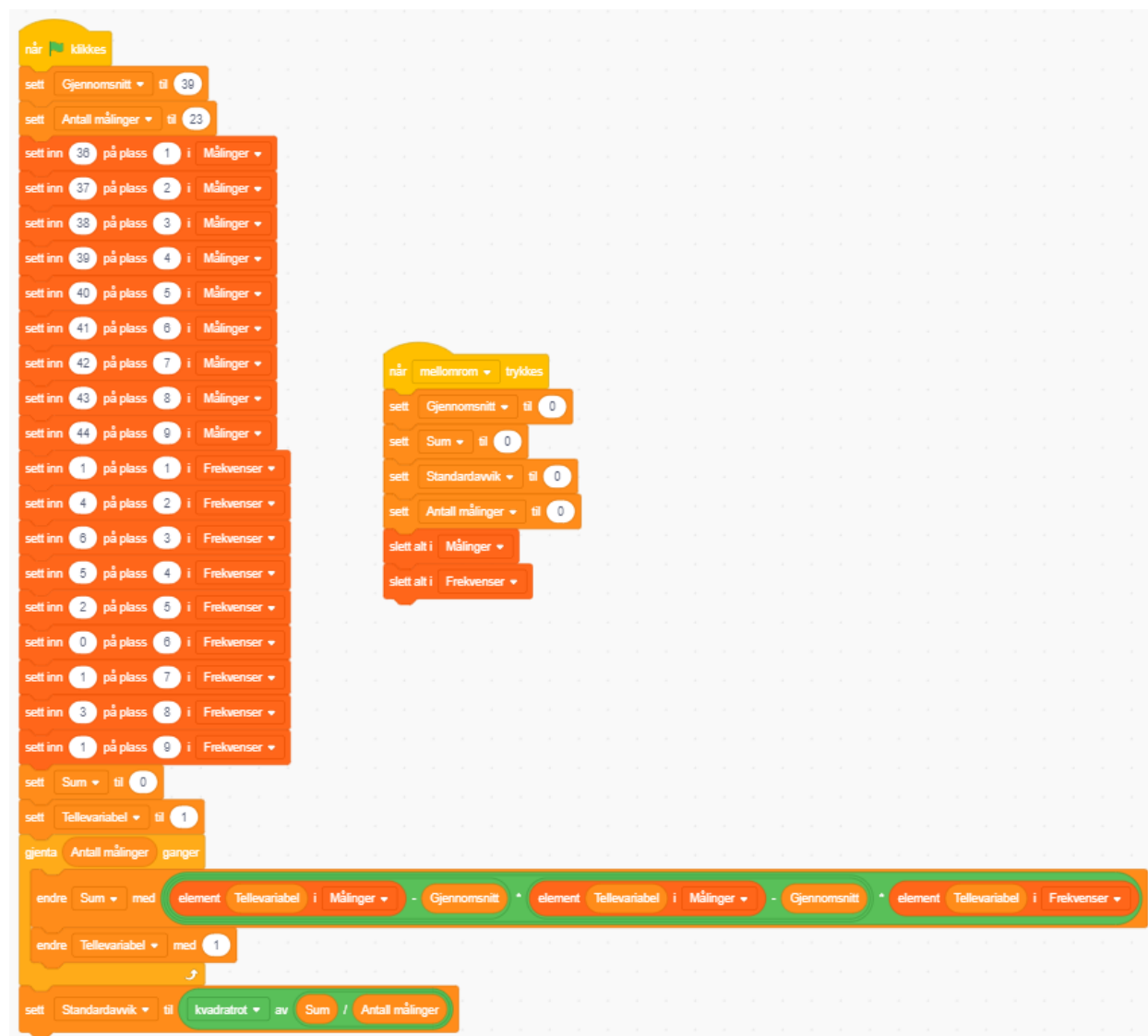
Bruk desse klossane for å lage eit program som reknar ut standardavvik

Scratch code blocks for a puzzle game program:

- endre Sum med element Tellevariabel i Frekvenser
- gjernta Antall målinger ganger
- element Tellevariabel i Målinger - Gjennomsnitt * element Tellevariabel i Målinger - Gjennomsnitt
- endre Tellevariabel med 1
- kvadratrota av Sum / Antall målinger
- sett Standardavvik til 0
- når flagget klikkes
- sett Gjennomsnitt til 39
- sett Antall målinger til 23
- sett inn 36 på plass 1 i Målinger
- sett inn 37 på plass 2 i Målinger
- sett inn 38 på plass 3 i Målinger
- sett inn 39 på plass 4 i Målinger
- sett inn 40 på plass 5 i Målinger
- sett inn 41 på plass 6 i Målinger
- sett inn 42 på plass 7 i Målinger
- sett inn 43 på plass 8 i Målinger
- sett inn 44 på plass 9 i Målinger
- sett inn 1 på plass 1 i Frekvenser
- sett inn 4 på plass 2 i Frekvenser
- sett inn 6 på plass 3 i Frekvenser
- sett inn 5 på plass 4 i Frekvenser
- sett inn 2 på plass 5 i Frekvenser
- sett inn 0 på plass 6 i Frekvenser
- sett inn 1 på plass 7 i Frekvenser
- sett inn 3 på plass 8 i Frekvenser
- sett inn 1 på plass 9 i Frekvenser
- sett Sum til 0
- sett Tellevariabel til 1

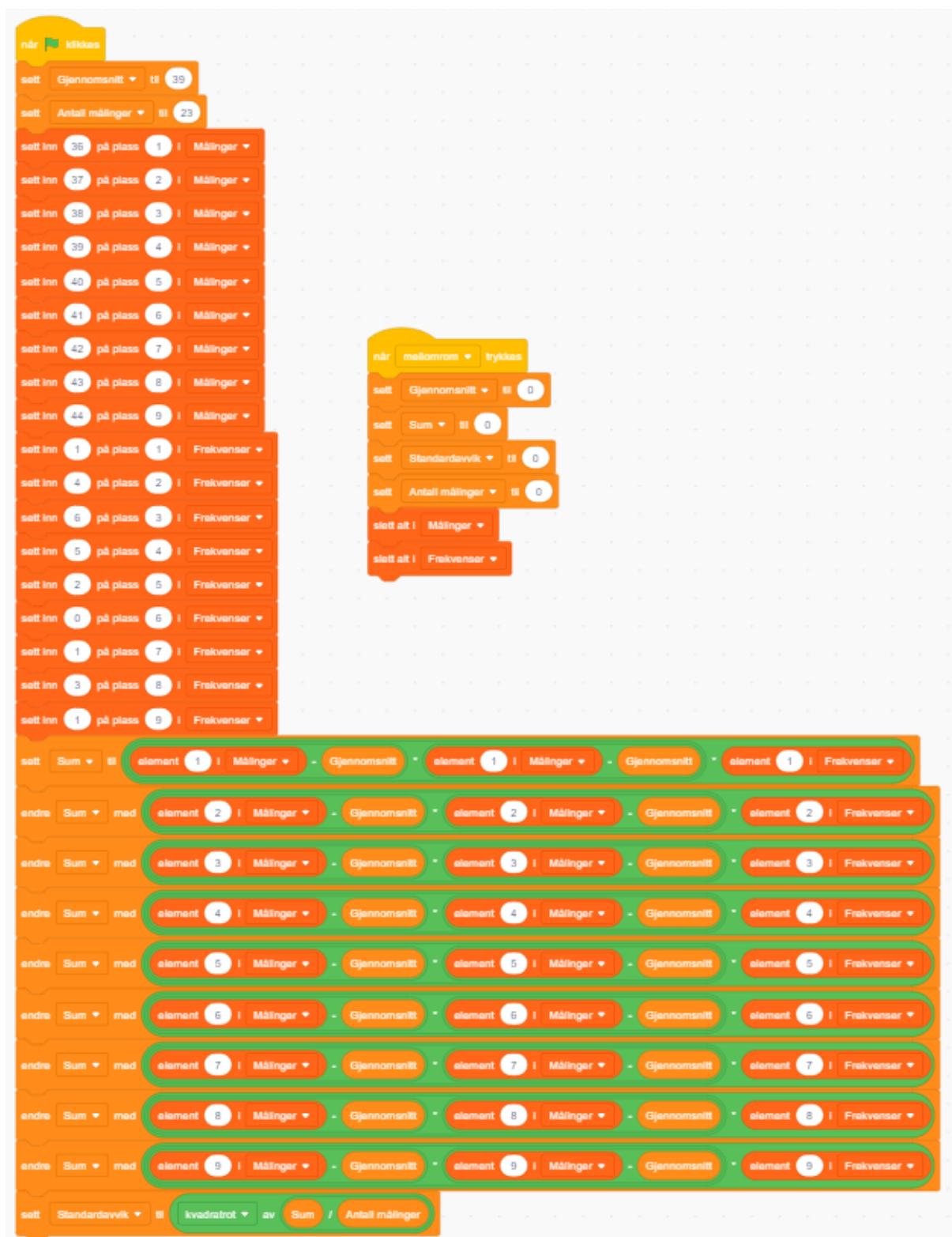
Løsningsforslag programmeringsoppgaver

Dette programmet går det ikke an å bruke funksjoner for å gjøre mer effektivt, da det i skrivende stund ikke går an å lage funksjoner som returnerer verdier i Scratch. Det er også valgt å ta inn gjennomsnitt som en innverdi, siden programmet allerede blir så stort at det er relativt uoversiktlig.



Alternativt løsningsforslag

- uten løkke



Programmeringsoppgåver i Python

Puslespel-programmering

Bruk disse kodelinjene for å lage eit program som reknar ut standardavvik.

```
from pylab import *
inndata = [36, 37, 38, 39, 40, 42, 43, 44]
frekvens = [1, 4, 6, 5, 2, 1, 3, 1]
i = 0
j = 0
antall = 0
summen = 0
a = 0

while i < len(frekvens):
    while j < len(frekvens):
        gjennomsnitt = summen/antall
        standardavvik = sqrt(a/antall)
        print(standardavvik)

        antall = antall + frekvens[i]
        summen = summen + (inndata[i]*frekvens[i])
        a = a + (((inndata[j]-gjennomsnitt)**2)*frekvens[j])

        i = i+1
        j = j+1
```

Gjer ferdig-programmering

Bruk disse kodelinjene og hald fram med koden slik at det blir eit program som reknar ut standardavvik.

```
from pylab import *
inndata = [36, 37, 38, 39, 40, 42, 43, 44]
frekvens = [1, 4, 6, 5, 2, 1, 3, 1]
i = 0
j = 0
antall = 0
summen = 0
a = 0

while i < len(frekvens):
    summen = summen + (inndata[i]*frekvens[i])
    antall = antall + frekvens[i]
    i = i+1

gjennomsnitt = summen/antall
```

Løsningsforslag programmering i Python

```
1 from pylab import *
2 inndata = [36, 37, 38, 39, 40, 42, 43, 44]
3 frekvens = [1, 4, 6, 5, 2, 1, 3, 1]
4 i = 0
5 j = 0
6 antall = 0
7 summen = 0
8 a = 0
9
10 while i < len(frekvens):
11     summen = summen + (inndata[i]*frekvens[i])
12     antall = antall + frekvens[i]
13     i = i+1
14
15 gjennomsnitt = summen/antall
16
17 while j < len(frekvens):
18     a = a + (((inndata[j]-gjennomsnitt)**2)*frekvens[j])
19     j = j+1
20
21 standardavvik = sqrt(a/antall)
22 print(standardavvik)
```


Opplegg 25: Statistikk i naturfag og samfunnsfag

Kompetansemål

Matematikk (beregnet for 9.trinn, men kan kobles til kompetansemål på andre trinn også):

- finne og diskutere sentralt mål og spreiingsmål i reelle datasett
- utforske og argumentere for korleis framstillingar av tal og data kan brukast for å fremje ulike synspunkt

Naturfag (dette kommer an på hvilke data elevene samler inn)

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

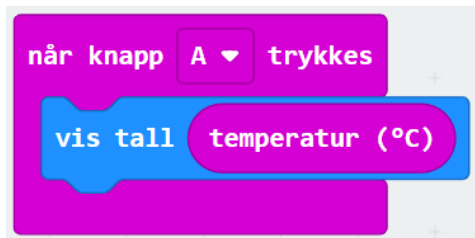
- Micro:bit med batteripakke
- Papir til fargeleggingsbok

TIPS!

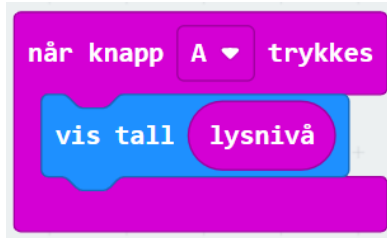
Eksempler på mulig datainnsamling med micro:bit: Temperaturendring i termoser eller solfangeren, temperatur etter en viss tid i mange forskjellige termoser, romtemperatur i forskjellige rom, utetemperatur i løpet av en dag, lysstyrke inne/ute, som tellehjelp (teller +1 hver gang man trykker inn en knapp) for: Antall biler av en viss farge, antall steg på en viss avstand, antall mennesker med langt/mellomlangt/kort hår osv.

Programmeringsoppgaver

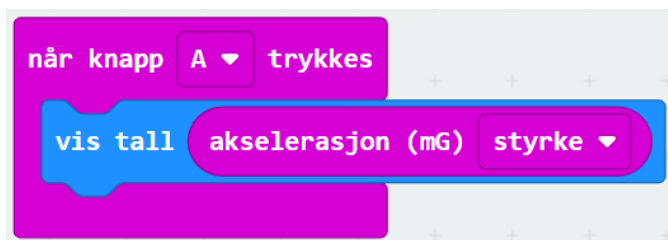
I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Løsningsforslag på ulike målinger ved hjelp av micro:bit vises under.



Viser temperatur når knapp A trykkes.



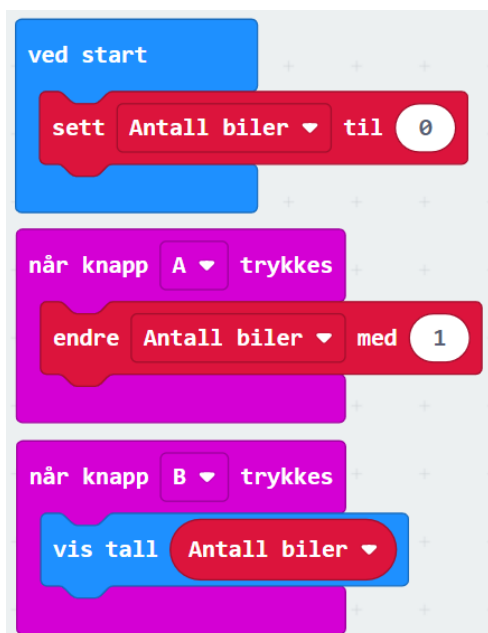
Viser lysnivå når knapp A trykkes.



Viser akselerasjonen (summert for alle retninger) når knapp A trykkes.



Viser helningsvinkelen til micro:biten når knapp A trykkes.



Eksempel på en «tellemaskin» der man øker antall ved å trykke på knapp A, og viser antallet på micro:bitens skjerm ved å trykke på knapp B.

Løsningsforslag programmering i Python

I dette opplegget blir programmeringsoppgaven gitt som en åpen oppgave, dermed finnes det ikke et spesifikt løsningsforslag. Se løsningsforslag for programmer for målinger ved hjelp av micro:bit i kapittel 12.

Opplegg 26: Laste ned data fra statistisk sentralbyrå

Kompetansemål

Matematikk (beregnet for 10.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Naturfag

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

Trolldeig, sugerør, ark, tusj og tape, saks, papp, garn, klistermerker, silkepapir, glitter++ (Evt bakeingredienser – elevene kan bake figurer som illustrere dataene.)

TIPS!

Denne oppgaven kan ikke programmeres med klossprogrammering, bare med Python. Alternativet er å la elevene bruke Geogebra til å plote grafer. Se kapittel 13 i elevboka for intro til plotting med Python.

Ekstraoppgåve:

1. Korrelasjon – kva er det?
2. Kan du finne kor stor korrelasjon det er for datasettet ditt? Sjå gjerne kapittel 13 for tips.

Krevjande ekstraoppgåve:

1. Kva er avhengige og uavhengige variablar?

Vi har ofte mange av kvar type variabel, sær med dei datasetta de har brukt her.

2. I datasettet du brukte, kva for nokre variablar er avhengige, og kva for nokre er dei uavhengige variablane? (Det kan hende at det ikkje er meir enn éin av kvar.)

Neste steg innan forskning, etter å finne ein korrelasjon, er å isolere éin og éin variabel, og så teste samanhangen med å gjere eit forsøk. Dette er vanleg å gjere i naturfag, og ganske enkelt å teste.

3. Sjekk dei andre modelleringsoppgåvene de gjorde på 8.trinn, i kapittel 4. Testa de berre ein variabel om gongen der?
4. Korleis heng uavhengige og avhengige variablar saman med feilkjelder?

Innan samfunnsfag, slik som denne typen statistikk frå SSB, er det ofte sær mange variablar, og vanskelegare å isolere dei, sidan vi har med menneske å gjere. Da må vi og ha med mange nok menneske – for å kontrollere for flest mogleg andre variablar som kan påvirke resultatet. Det finst fagfelt som blandar realfag og samfunnsfagleg modellering, slik som medisin.

5. Kan du kome på nokre døme på modellering med sær mange variablar som er aktuelle nå for tida?

Programmeringsoppgåver for Python

Puslespel-programmering

Bruk desse kodelinjene for å lage eit program som plotter ein graf frå ei .txt-fil.

```
xlabel("x-verdier")          from pylab import *
plot(x, y)                   ylabel("y-verdier")
x = tabell[:,0]              tabell = loadtxt("data.txt")
show()                       y = tabell[:,1]
```

Gjer ferdig-programmering

Bruk desse kodelinjene og hald fram med koden slik at det blir eit program som plottar ein graf frå ei .txt-fil.

```
from pylab import *

tabell = loadtxt("data.txt")

x = tabell[:,0]
y = tabell[:,1]
```

Løsningsforslag programmering i python

```
from pylab import *

tabell = loadtxt("data.txt")

x = tabell[:,0]
y = tabell[:,1]

plot(x, y)

xlabel("x-verdier")
ylabel("y-verdier")

show()
```

Løsningsforslag ekstraoppgave i Python

- enkel korrelasjon

```
1 from pylab import *
2 from scipy.stats import pearsonr
3
4 co2data = loadtxt('co2utslipp.txt')
5
6 x = co2data[:,0]
7 y = co2data[:,1]
8
9 a = pearsonr(x, y)[0]
10 print(a)
11 print(a**2)
```


Opplegg 27: Forsk på personleg økonomi

Kompetansemål

Matematikk (beregnet for 10.trinn, men kan kobles til kompetansemål på andre trinn også):

- rekne ut stigingstalet til ein lineær funksjon og bruke det til å forklare omgrepa endring per eining og gjennomsnittsfart
- hente ut og tolke relevant informasjon frå tekstar om kjøp og sal og ulike typar lån og bruke det til å formulere og løyse problem
- planleggje, utføre og presentere eit utforskande arbeid knytt til personleg økonomi
- bruke funksjonar i modellering og argumentere for framgangsmåtar og resultat
- modellere situasjonar knytte til reelle datasett, presentere resultata og argumentere for at modellane er gyldige
- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materiale:

Trolldeig, sugerør, ark, tusj og tape, saks, papp, garn, klistermerke, silkepapir, glitter++ (Evt bakeingrediensar – elevane kan bake figurar som illustrerer dataa.)

TIPS!

Denne oppgåva kan ikkje programmerast med klossprogrammering, berre med Python. Alternativet er å la elevane bruke Geogebra til å gjere regresjonen. Sjå kapittel 13 i elevboka for intro til regresjon med Python.

Krevjande ekstraoppgåve 1:

1. Kryssvalidering – kva er det?
2. Kan du dele opp datasettet ditt for å kryssvalidere modellen din? Sjå gjerne her for tips: https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html#sphx-gl-auto-examples-linear-model-plot-ols-py

Krevjande ekstraoppgåve 2:

1. Finn ut koreis du kan ha ikkje-lineær regresjon i Python, og prøv ut ein regresjon med ein ikkje-lineær funksjon.
2. Er den lineære eller den andre best?
3. Korleis kan du bruke Python til å finne det ut?

Sjå i kapittel 13 i for ikkje-lineær regresjon med løysingsframlegg.

Programmeringsoppgåver i Python

Desse oppgåvene bruker `curve_fit()`-funksjonen da denne og kan brukast ved ikkje-lineær regresjon. Men han kan og kjennast som vanskelegare å forstå for elevane, difor står og løyningsframlegget der ein nyttar `LinearRegression()`-funksjonen i staden. Kva for ein versjon som vil fungere best i si elevgruppe får læraren vurdere.

Puslespel-programmering

Bruk desse kodelinjene for å lage eit program som utfører ein lineær regresjon med data frå ei .txt-fil.

```
x_verdier = co2data[:,0]
y_verdier = co2data[:,1]

def funksjon(x, a, b):
    return (a * x) + b

koeffisienter, kovarianser = curve_fit(funksjon, x_verdier, y_verdier)
a, b = koeffisienter
print(a,b)

co2data = loadtxt("co2utslipp.txt")

from pylab import *
from scipy.optimize import curve_fit

scatter(x_verdier, y_verdier)
plot(x_verdier, funksjon(x_verdier, a, b), color="red")
title("Samlet klimagassutslipp for Norge 1990 - 2018")
xlabel("År")
ylabel("CO2-utslipp (1000 tonn CO2-ekvivalenter)")
show()
```

Gjer ferdig-programmering

Bruk desse kodelinjene og hald fram med koden slik at det blir eit program som utfører ein lineær regresjon med data frå ei .txt-fil.

```
from pylab import *
from scipy.optimize import curve_fit

co2data = loadtxt("co2utslipp.txt")

x_verdier = co2data[:,0]
y_verdier = co2data[:,1]

def funksjon(x, a, b):
    return (a * x) + b
```

Løsningsforslag programmering i python

Løsningsforslaget bruker `curve_fit()`-funksjonen da denne også kan brukes ved ikke-lineær regresjon. Men den kan også oppleves som vanskeligere å forstå for elevene, derfor står også løsningsforslaget der man bruker `LinearRegression()`-funksjonen i stedet. Det står en puslespilloppgave i elevboka i kapittel 13 i teksten om lineær modellering i Python med bruk av `curve_fit()`.

```
from pylab import *
from scipy.optimize import curve_fit

co2data = loadtxt("co2utslipp.txt")

x_verdier = co2data[:,0]
y_verdier = co2data[:,1]

def funksjon(x, a, b):
    return (a * x) + b

koeffisienter, kovarianser = curve_fit(funksjon, x_verdier, y_verdier)
a, b = koeffisienter
print(a,b)

scatter(x_verdier, y_verdier)
plot(x_verdier, funksjon(x_verdier, a, b), color="red")
title("Samlet klimagassutslipp for Norge 1990 - 2018")
xlabel("År")
ylabel("CO2-utslipp (1000 tonn CO2-ekvivalenter)")
show()
```

Alternativt løsningsforslag programmering i Python

Her brukes `linearRegression()`-funksjonen i stedet for den mer generelle `curve_fit()`. I tillegg går det an å bruke `flatten()`-funksjonen i stedet for `reshape(-1, 1)`.

```
from pylab import *
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model

co2data = loadtxt("co2utslipp.txt")

x = co2data[:,0].reshape(-1, 1)
y = co2data[:,1].reshape(-1, 1)

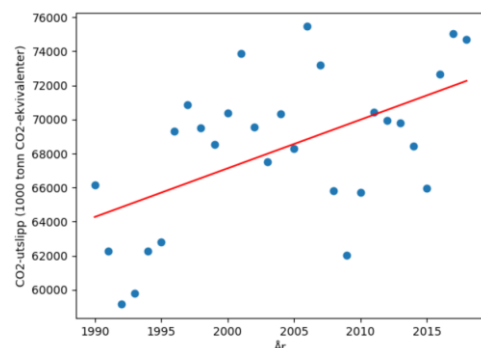
modell = linear_model.LinearRegression()
modell.fit(x, y)

print(modell.coef_ , modell.intercept_)

plt.scatter(x, y)
plot(x, (x*modell.coef_+modell.intercept_), color="red")
xlabel("År")
ylabel("CO2-utslipp (1000 tonn CO2-ekvivalenter)")
show()
```

Løsningsforslag med kommentarer i selve programmet.

```
1 from pylab import *
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from sklearn import datasets, linear_model
5
6 # Importerer data og gjør de om til riktig format
7 co2data = loadtxt("co2utslipp.txt")
8 co2aar = co2data[:,0].reshape(-1, 1)
9 co2utslipp = co2data[:,1].reshape(-1, 1)
10
11 # Lager en lineær modell (som det ikke er noe i foreløpig)
12 regr = linear_model.LinearRegression()
13
14 # Utføre regresjonen
15 regr.fit(co2aar, co2utslipp)
16
17 # Koeffisientene
18 print('Koeffisientene: \n', regr.coef_ , regr.intercept_)
19
20 # Plotting av datapunkter og lineær modell
21 plt.scatter(co2aar, co2utslipp)
22 plot(co2aar, (co2aar*regr.coef_+regr.intercept_), color="red")
23 xlabel("År")
24 ylabel("CO2-utslipp (1000 tonn CO2-ekvivalenter)")
25 show()
```



Koeffisientene:
[[285.59211823]] [-504058.81182266]

Opplegg 28: Korrelasjon og «fake news»

Kompetansemål

Matematikk (beregnet for 10.trinn, men kan kobles til kompetansemål på andre trinn også):

- bruke funksjonar i modellering og argumentere for framgangsmåtar og resultat
- modellere situasjonar knytte til reelle datasett, presentere resultata og argumentere for at modellane er gyldige
- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

Materialer til nyhetssak (papir)

TIPS!

Denne oppgaven kan ikke programmeres med klossprogrammering, bare med Python.

Alternativet er å la elevene bruke Geogebra til å finne korrelasjon. Se kapittel 13 i elevboka for intro til korrelasjon med Python.

Når elevene er kommet skikkelig i gang med programmeringen, evt i en oppsummering etter prosjektet er ferdig, kan man ta opp hvorfor x-verdiene i begge datasettene må være like, og det må være like mange i hver datafil. Det er også fint å kunne diskutere hvorfor man finner korrelasjonen mellom y og y1, og hva forskjellen er med korrelasjonene mellom x og y, og x1 og y1.

Størsteparten av poenget med denne oppgaven er å trene elevene i å kritisk vurdere informasjon fra forskjellige kilder, spesielt med tanke på statistiske fremstillinger. I utgangspunktet i denne boka skal elevene lage en utstilling basert på de tre forskjellige statistikk-oppleggene. Blant utstillingen skal det da være en ekte nyhetssak som elevene har laget og en nyhetssak basert på dårlig-/juksestatistikk. En viktig del av denne oppgaven blir det å la elevene se hverandres utstillinger når de er ferdige og prøve å identifisere hvilke nyhetssaker som er ekte, og hvilke som er falske. Deretter bør man ta en diskusjon i klassen der man får fram hva elevene brukte som kriterier for å identifisere de falske sakene. Var dette gode kriterier – virket de? Vil de være gode å bruke generelt? Få med elevene på en «tenke-høyt» diskusjon rundt temaet, og bruk gjerne tidsaktuell villedende statistikk.

Ekstraoppgåver

Ekstraoppgåve:

Korrelasjon og kausalitet – kva er skilnaden? Bruk gjerne konkrete døme når du forklarar.

Lyøingsframlegg/HINT til ekstraoppgåve:

Korrelasjon tyder eigentleg samvariasjon, at noko varierer på same måten. Til dømes finst det ei nettside med rare ting som korrelerer. Sjå her: <https://www.tylervigen.com/spurious-correlations>

Kausalitet heng saman med å finne ut kva som er årsaka til at noko skjer. Dersom eg dyttar deg, fell du. Da er det at jeg dyttar deg årsaka til at du fell.

Det at noko varierer på same måten (korrelerer) tyder ikkje at vi har med ei årsak å gjere. Det er slik at kor overvektig ein er, korrelerer med kor mykje lettbrus ein drikk. Dersom ein er overvektig drikk ein gjennomsnittlig meir lettbrus enn ein som ikkje er overvektig. Det tyder ikkje at det er lettbrusen ein drikk som gjer ein overvektig, det treng ikkje vere årsaka. Årsak kan vere at dei som er overvektige, er meir bevisste på å ikkje drikke sukkerhaldig brus, og dermed drikk lettbrus i staden. I forskning er det ofte sær s vanskeleg å finne ut *årsaka* til at noko hender, men enkelt å finne ut *at* det hender.

Krevjande (omfattande) ekstraoppgåve:

1. Lag eit forskingsprosjekt der du måler noko innan naturvitskap, gjerne med hjelp av micro:bit.
2. Definer kva for nokre variablar du skal måle.
3. Finn teori om kva slags matematisk samanheng det er mellom dei.
4. Gjennomfør forskingsprosjektet, og gjer dei planlagte målingane.
5. Analyser resultata med hjelp av programmering i Python.
6. Oppsummer alle delane av prosjektet, inkludert resultata og diskusjon, på ein forskingsplakat.

Eventuelt ei lik ekstraoppgåve med stort datasett frå SSB (meir samfunnsfagleg perspektiv).

Programmeringsoppgåver i Python

Her må ein passe på å ha to .txt-filer tilgjengelege for python-programmet. Det gjer ein stort sett med å legge filene i same mappe som ein lagrar python-programmet, men med bruk av programmering i ein nettlesar, må ein som regel laste opp .txt-filene slik det er vist i løysingsframlegget for Aschehougs trinket-baserte nettlesarprogrammering.

Puslespel-programmering

Bruk desse kodelinjene for å lage eit program som finn korrelasjonen mellom to sett med data frå to .txt-filer.

```
x = co2data[:,0]
y = co2data[:,1]
```

```
a = pearsonr(y, y1)[0]
print(a)
print(a**2)

from pylab import *
from scipy.stats import pearsonr

x1 = varedata[:,0]
y1 = varedata[:,1]

co2data = loadtxt('co2utslipp.txt')
varedata = loadtxt('Varehandel.txt')
```

Gjer ferdig-programmering

Bruk desse kodelinjene og hald fram med koden slik at det blir eit program som finn korrelasjonen mellom to sett med data frå to .txt-filer.

```
from pylab import *
from scipy.stats import pearsonr

co2data = loadtxt('co2utslipp.txt')
varedata = loadtxt('Varehandel.txt')

x = co2data[:,0]
y = co2data[:,1]
```

Løsningsforslag i Python

Løsningsforslag for å finne korrelasjonen mellom x-verdier og y-verdier fra en .txt-fil ligger under ekstraoppgaven i opplegg 26 i denne lærerveiledningen. Under vises løsningsforslag for korrelasjonen mellom y-verdiene for to forskjellige datasett, så lenge datasettene inneholder like mange målinger, og for de samme årstallene. X-verdiene er tiden, og med data lastet ned fra SSB, er det år som er typisk enhet.

```
1 from pylab import *
2 from scipy.stats import pearsonr
3
4 #Pass på at de to .txt-filene har like mange målinger og har like x-verdier
5 co2data = loadtxt('co2utslipp.txt')
6 varedata = loadtxt('Varehandel.txt')
7
8 x = co2data[:,0]
9 y = co2data[:,1]
10
11 x1 = varedata[:,0]
12 y1 = varedata[:,1]
13
14 a = pearsonr(y, y1)[0]
15 print(a)
16 print(a**2)
```

Løsningsforslag under inkluderer også en lineær regresjon, plotting av graf og punkter og selve beregningen av korrelasjonen for y-verdiene til de to datasettene.

main.py	co2utslipp.txt	varehandel.txt
---------	----------------	----------------

```
from pylab import *
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model
from scipy.stats import pearsonr

# Importerer data og gjør de om til riktig format
co2data = loadtxt("co2utslipp.txt")
varedata = loadtxt("varehandel.txt")

x = co2data[:,0].reshape(-1, 1)
y = co2data[:,1].reshape(-1, 1)

x1 = varedata[:,0].reshape(-1,1)
y1 = varedata[:,1].reshape(-1,1)

# Lager en lineær modell (som det ikke er noe i foreløpig) og utfører regresjonen
modell = linear_model.LinearRegression()
modell.fit(y, y1)

# Koeffisientene
print(modell.coef_ , modell.intercept_)

# Korrelasjonskoeffisient, R^2
a = pearsonr(y, y1)[0]
print(a**2)

# Plotting av datapunkter og lineær modell
plt.scatter(y, y1, color="green")
plot(y, (y*modell.coef_+modell.intercept_), color="red")
xlabel("CO2-utslipp (1000 tonn CO2-ekvivalenter)")
ylabel("Varehandel per innbygger (kr)")
show()
```


Løsningsforslag i Python for ekstraoppgave

- lineær regresjon med enkel korrelasjon

```
from pylab import *
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model
from scipy.stats import pearsonr

# Importerer data og gjør de om til riktig format
co2data = loadtxt("co2utslipp.txt")

x = co2data[:,0].reshape(-1, 1)
y = co2data[:,1].reshape(-1, 1)

# Lager en lineær modell (som det ikke er noe i foreløpig) og utfører regresjonen
modell = linear_model.LinearRegression()
modell.fit(x, y)

# Koeffisientene
print(modell.coef_ , modell.intercept_)

# Korrelasjonskoeffisient, R^2
a = pearsonr(x, y)[0]
print(a**2)

# Plotting av datapunkter og lineær modell
plt.scatter(x, y)
plot(x, (x*modell.coef_+modell.intercept_), color="red")
xlabel("År")
ylabel("CO2-utslipp (1000 tonn CO2-ekvivalenter)")
show()
```

```
from pylab import *
from scipy.stats import spearmanr
from scipy.stats import pearsonr

co2data = loadtxt("co2utslipp.txt")

x = co2data[:,0]
y = co2data[:,1]

a = spearmanr(x, y)
b = pearsonr(x, y)

print(a)
print(b)
```

Opplegg 29: Varme, isolasjon og energieffektivitet

Kompetansemål

Matematikk (beregnet for 10.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske og samanlikne eigenskapar ved ulike funksjonar ved å bruke digitale verktøy
- bruke funksjonar i modellering og argumentere for framgangsmåtar og resultat
- modellere situasjonar knytte til reelle datasett, presentere resultata og argumentere for at modellane er gyldige
- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Naturfag

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- delta i risikovurderinger knyttet til forsøk og følge sikkerhetstiltakene
- utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker
- bruke programmering til å utforske naturfaglige fenomener
- gjøre rede for energibevaring og energikvalitet og utforske ulike måter å omdanne, transportere og lagre energi på

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

- Micro:bit med batteripakke
- Kakelys med holder, fyrstikker, papp, limpistol, saks, tape, avispapir, plast- og aluminiumsfolie, garn/stoff, plast +++

TIPS!

Pass på at det er plass til å tenne på et lys inni huset elevene bygger. Det må være en viss avstand til tak og vegger for at det ikke skal risikere at det tar fyr, gjerne delvis åpent tak akkurat når lyset brennes. Dette bør diskuteres før elevene setter i gang med oppgaven, og hvilke sikkerhetstiltak man kan gjøre. (For eksempel gjennomføre lysbrenning og temperaturmåling utendørs på asfalt eller grusflate med en bøtte vann i nærheten.)

Dersom man ikke ønsker å bruke kakelys for å øke temperaturen inni huset, går det an å bruke et begerglass med vann med høy temperatur i stedet, men da vil det være mulig å brenne seg på vannet. Det vil også ta noe lenger tid å øke temperaturen inni huset på den måten. Kanskje har elevene andre forslag? Diskuter gjerne ulike løsninger med dem.

Ikke sett inn micro:biten i huset før kakelyset er slokket og fjernet fra huset.

Ekstraoppgåve

Nå skal de kalibrere termometeret på micro:biten. Å kalibrere tyder å samanlikne målingane med nokre andre målingar for å passe på at målingane viser ein mest mogleg rett verdi.

For å kalibrere temperaturen må de ha eit vanleg termometer som og kan måle temperaturen, slik at de får nokre målingar de kan samanlikne med.

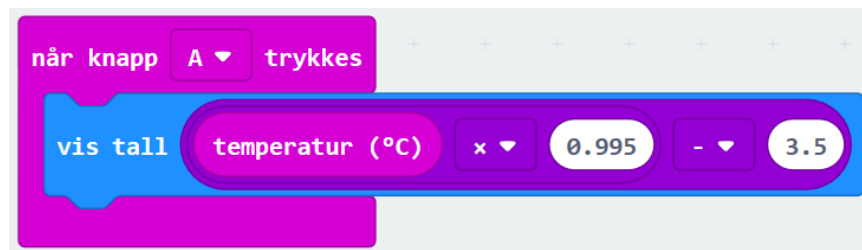
Noter kva for nokre verdier micro:biten dykkar måler for vatn med 5 ulike temperaturar. Gjer det same for det vanlege termometeret. Bruk verdiane for micro:biten sitt termometer som x-verdiar og verdiane frå det vanlege termometeret som y-verdiar.

Utfør ein lineær regresjon der de finn den rette linja som passar best med punkta dine.

1. Kva tyder denne matematiske modellen de har fått?
2. Kva kan han brukast til?
3. Kunne de brukt den i programmet på micro:biten for å vise den kalibrerte temperaturen på skjermen til micro:biten?
4. Korleis ville programmet blitt? Lag det.

Løysingsframlegg til programmeringa for ekstraoppgåva

Dersom funksjonen skulle bli $y=0,995x-3,5$, vil programmet kunne sjå slik ut:



Programmeringsoppgaver

For klossprogrammering, se opplegg 7 – Solenergi og solfanger. Der finnes flere varianter med klossprogrammering. Men elevene bør måle temperatur med en micro:bit og sende resultatet til en annen micro:bit, for å jobbe med de kompetansemålene som står oppgitt i opplegget.

Python-oppgåve for micro:bit

```
1 from microbit import *
2
3 while True:
4     x = temperature()
5     print(x)
6     sleep(500)
```

1. Giss kva programmet ved sida av gjer. Skriv ei kort forklaring på baksida av arket.
2. Finn fram pythoneditoren din. Skriv av koden og før han over til micro:biten med å trykke på flash-knappen. Kva hender?

3. Forklar koden:

Kodelinje	Forklaring
<code>from microbit import *</code>	
<code>while True:</code>	
<code>x = temperature()</code>	
<code>print(x)</code>	
<code>sleep(500)</code>	

4. Endre koden

- a) Byt rekkefølga på linje 3 og 5. Kva trur du vil endre seg når du køyrer programmet?
- b) Kvifor MÅ vi ha med sleep-kommandoen?
- c) Kan du slå saman to av kodelinjene og få same resultat?

5. Utfordring: Bruk denne nettressursen: <https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/introduction.html> for å finne koden du treng for å lage eit program som sender temperaturen målt med en micro:bit til ein annan micro:bit. Hugs at begge micro:bitane må bli programmerte
Spør gjerne om hint.

Programmeringsoppgåve i Python

Gjer ferdig-programmering – micro:bit 1

Bruk desse kodelinjene og hald fram med koden slik at det blir eit program som måler temperaturen og sender målinga til micro:bit nummer to..

```
from microbit import *  
import radio  
  
radio.on()  
radio.config(channel=40)
```

Gjer ferdig-programmering – micro:bit 2

Bruk desse kodelinjene og hald fram med koden slik at det blir eit program som finn temperaturen frå micro:bit nummer ein, og viser han på skjermen.

```
from microbit import *  
import radio  
  
radio.on()  
radio.config(channel=40)
```

Løsningsforslag programmering i Python

```
1 from microbit import *
2 import radio
3
4 radio.on()
5 radio.config(channel=40)
6
7 while True:
8     temperatur = temperature()
9     x = str(temperatur)
10    radio.send_bytes(x)
11    sleep(200)
```

```
1 from microbit import *
2 import radio
3
4 radio.on()
5 radio.config(channel=40)
6
7 while True:
8     x = radio.receive_bytes()
9     print(x)
10    sleep(200)
```

Opplegg 30: Populasjoner og populasjonsvekst

Kompetansemål

Matematikk (beregnet for 10.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske og samanlikne eigenskapar ved ulike funksjonar ved å bruke digitale verktøy
- utforske samanhengen mellom konstant prosentvis endring, vekstfaktor og eksponentialfunksjonar
- bruke funksjonar i modellering og argumentere for framgangsmåtar og resultat
- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Naturfag

- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener
- drøfte hvordan energiproduksjon og energibruk kan påvirke miljøet lokalt og globalt
- utforske sammenhenger mellom abiotiske og biotiske faktorer i et økosystem og diskutere hvordan energi og materie omdannes i kretsløp
- gi eksempler på og drøfte aktuelle dilemmaer knyttet til utnyttelse av naturressurser og tap av biologisk mangfold

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

Papp, leire/trolldeig, limpistol, bomull, saks, tape, avispapir, plast- og aluminiumsfolie, garn, stoff, plast, maling, silkepapir +++ (Mest mulig forskjellige materialer – elevene kan velge å lage en spiselig modell om de ønsker det, og det er tilgjengelig rom på skolen.)

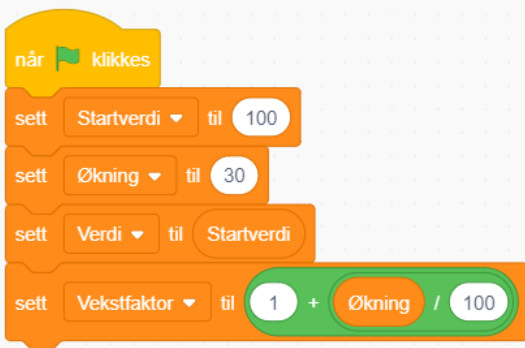
TIPS!

Elevene skal lage både biotiske og abiotiske faktorer til øya si, og spesielt de abiotiske faktorene kan være krevende å lage på en fornuftig måte. Ulike temperaturer kan vises ved et termometer eller ulike værphenomener som snø, is eller en stor sol. Elevene kan lage skyer, skyer med papirdråper, oppsprukket og tørr bakke o.l. La gjerne elevene bruke tid på å diskutere dette, hva som er abiotiske faktorer, og hvordan de kan vises fram på øymodellen.

Programmeringsoppgåver

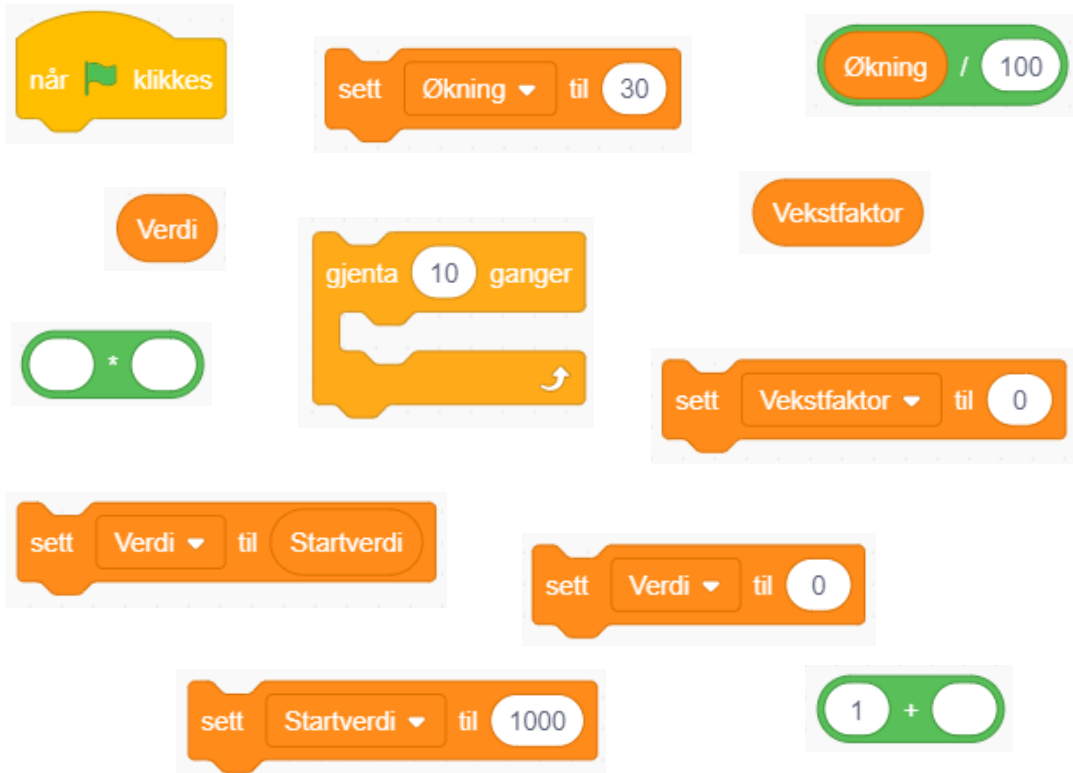
Gjer ferdig-programmering

Bruk desse klossane for å lage eit program som rekner ut populasjonsveksten på øya.



Puslespel-programmering

Bruk desse klossane for å lage eit program som rekner ut populasjonsveksten på øya.



Løysingsframlegg programmering

Eksponentiell vekst



Python-oppgåve populasjonar

```
1 økning = 20
2 startverdi = 500
3
4 vekstfaktor = 1 + økning/100
5 verdi = startverdi
6
7 for i in range(0,10):
8     verdi = verdi*vekstfaktor
9     print(verdi)
```

1. Giss kva programmet ved sida av gjer. Skriv ei kort forklaring på baksida av arket.
2. Gå inn på nettadressa: <https://tjenester.lokus.no/open/programmering/python.html> og skriv av koden. Trykk på play-knappen for å køyre koden. Kva skjer?

3. Forklar koden:

Kodelinje	Forklaring
økning = 20	
startverdi = 500	
vekstfaktor = 1 + økning/100	
verdi = startverdi	
for i in range(0,10):	
verdi = verdi*vekstfaktor	
print(verdi)	

4. Endre koden

- a) Byt rekkefølga på linje 8 og 9. Kva trur du vil endre seg når du køyrer programmet?
- b) Finn ut kor mange dyr det er etter 10 år dersom auken er 4 % årlig.
- c) Kva må du endre i programmet ditt dersom populasjonen **minkar** med 20 % årlig?
- d) Kan du fjerne ei linje i programmet og framleis få same resultat som i oppgåve b? Må du endre noko meir i tillegg?

5. Utfordring:

- a) Finn ut kor lang tid det tek før ein pingvinpopulasjon på 10 000 pingvinar blir dubla dersom auken er på 2 %.
- b) Endre koden slik at du lagar ein graf over talet på pingvinar for 5000 år.
- c) Kva type matematisk modell har vi for pingvinpopulasjonen? Er denne realistisk?



Python-oppgåve renter

```
1 rente = 2
2 startverdi = 10000
3
4 vekstfaktor = 1 + rente/100
5 verdi = startverdi
6
7 for i in range(0,10):
8     verdi = verdi*vekstfaktor
9     print(verdi)
```

1. Giss kva programmet ved sida av gjer. Skriv ei kort forklaring på baksida av arket.
2. Gå inn på nettadressa: <https://tjenester.lokus.no/open/programming/python.html> og skriv av koden. Trykk på play-knappen for å køyre koden. Kva hender?

3. Forklar koden:

Blokk/kodelinje	Forklaring
rente=2	
startverdi=10000	
vekstfaktor=1+rente/100	
verdi=startverdi	
for i in range(0,10):	
verdi=verdi*vekstfaktor	
print(verdi)	

4. Endre koden

- a) Byt rekkefølga på linje 8 og 9. Kva trur du vil endre seg når du køyrer programmet?
- b) Finn ut kor mykje du har etter 10 år dersom renta er 4 %.
- c) Kan du fjerne ei linje i programmet og framleis få same resultat som i oppgåve b? Må du endre noko meir i tillegg?

5. Utfordring:

- a) Finn ut kor lenge du må ha 10 000 kr i banken med ei rente på 2 % for at dei skal vekse til 15 000 kr.
- b) Endre koden slik at du ser kor mykje du har i banken ved slutten av kvart år dersom du startar med 0 kr og set inn 2000 kr kvart år i 10 år på starten av året. Renta er fast på 2,5 %.

Programmeringsoppgåve i Python

Gjer ferdig-programmering

Bruk disse kodelinjene og hald fram med koden slik at det blir et program som reknar ut populasjonsveksten på øya.

```
populasjon = 1000  
økning = 20  
vekstfaktor = 1 + økning/100
```

Løysingsframlegg programmering i Python

På arket til elevane «Python-oppgave populasjoner», i oppgave 5b, skal elevane teikne grafen i Python. Det står korleis dette blir gjort i kapittel 13.

Løysingsframlegg for populasjonsvekst slik oppgåva står i elevboka.

```
1 populasjon = 1000
2 økning = 20
3 vekstfaktor = 1 + økning/100
4
5 for i in range(0,10):
6     populasjon = populasjon * vekstfaktor
7     print(populasjon)
```

Løysingsframlegg for populasjonsvekst inkludert plotting av graf.

```
1 from pylab import *
2
3 endring_i_prosent = 20
4 populasjon = [1000]
5 måneder = 12
6
7 vekstfaktor = 1 + endring_i_prosent/100
8
9 for i in range(0, måneder):
10     populasjon.append(populasjon[i]*vekstfaktor)
11
12 plot(linspace(0, måneder, måneder+1),populasjon)
13 show()
```

Løysingsframlegg for bankinnskott med utsikt for eit årleg innskott.

```
1 rente = 3
2 innskudd = 1000
3
4 vekstfaktor = 1 + rente/100
5 verdi = 0
6
7 for i in range(0,10):
8     verdi = verdi*vekstfaktor + innskudd*vekstfaktor
9     print(verdi)
```

Opplegg 31: Vaksiner og smittetall

Kompetansemål

Matematikk (beregnet for 10.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske og samanlikne eigenskapar ved ulike funksjonar ved å bruke digitale verktøy
- utforske samanhengen mellom konstant prosentvis endring, vekstfaktor og eksponentialfunksjonar
- bruke funksjonar i modellering og argumentere for framgangsmåtar og resultat
- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Naturfag

- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- bruke programmering til å utforske naturfaglige fenomener
- beskrive kroppens immunforsvar og hvordan vaksiner virker, og gjøre rede for hva vaksiner betyr for folkehelsen

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

Garn og strikkepinner, plakater, limpistol evt nål og tråd (for å sy smitteblomsten sammen) og materialer for å dekorere plakaten.

TIPS! I dette opplegget skal elevene programmere en eksponentialfunksjon som skal være en enkel modell for smitte. Dette programmet vil lage «strikkeoppskrifter» der antall masker per runde tilsvarer antall smittede per uke. Smittetallet kan elevene variere som de vil, men greit at det ikke blir altfor høyt, da blir det veldig mye å strikke etter hvert. Det kan være greit å diskutere dette med inkubasjonstid, og hvor lenge en person er smittsom. I modellen til elevene, så går vi ut fra at hver syke person smitter x antall personer mens de er syke, og at dette skjer på en uke. Mer nøyaktig vil det være å si at x antall personer blir smittet i hver smittegenerasjon, men det kan også gjøre modelleringen unødvendig komplisert å forstå for elevene.

Programmeringsoppgave

I dette opplegget vil programmeringsoppgavene være like som i Opplegg 30 – Populasjoner og populasjonsvekst, da det er bruk av eksponentialfunksjoner i begge tilfeller.

Programmeringsoppgåve i Python

Gjer ferdig-programmering

Bruk disse kodelinjene og hald fram med koden slik at det blir eit program som bereknar smitteutvikling (og lagar ei strikkeoppskrift med tal på masker per runde).

```
populasjon = 10
smittetall = 1.2
uker = 10
```

Puslespel-programmering

Bruk disse kodelinjene for å lage et program som bereknar smitteutvikling (og lagar ei strikkeoppskrift med tal på masker per runde).

```
populasjon = 10      smittetall = 1.2
print(populasjon)    uker = 10
for i in range(0, uker):
    populasjon = populasjon * smittetall
```

Løysingsframlegg for berekning av smitteutvikling
– og strikkeoppskrift med tal på masker per runde.

```
1 populasjon = 10
2 smittetall = 1.2
3 uker = 10
4
5 for i in range(0, uker):
6     populasjon = populasjon * smittetall
7     print(populasjon)
```

Opplegg 32: Forbrenningsreaksjoner

Kompetansemål

Matematikk (beregnet for 10.trinn, men kan kobles til kompetansemål på andre trinn også):

- utforske og samanlikne eigenskapar ved ulike funksjonar ved å bruke digitale verktøy
- bruke funksjonar i modellering og argumentere for framgangsmåtar og resultat
- modellere situasjonar knytte til reelle datasett, presentere resultata og argumentere for at modellane er gyldige
- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Naturfag

- stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere avhengige og uavhengige variabler og samle data for å finne svar
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- delta i risikovurderinger knyttet til forsøk og følge sikkerhetstiltakene
- bruke programmering til å utforske naturfaglige fenomener
- utforske kjemiske reaksjoner, forklare massebevaring og gjøre rede for betydninger av noen forbrenningsreaksjoner
- beskrive drivhuseffekten og gjøre rede for faktorer som kan forårsake globale klimaendringer
- gjøre rede for energibevaring og energikvalitet og utforske ulike måter å omdanne, transportere og lagre energi på

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

Micro:bit, 2 alligatorledninger, 180°-servomotor og kretskort med batteri.

Byggematerialer: papp, papir, sakser, limpistoler, sugerør, strikk, tape ++

TIPS!

Elevene kan godt ha kruttlapper tilgjengelig, så kan de prøve seg på forbrenningsreaksjoner. Det fungerer bra med gummistrikk som veke som tapes på kruttlappen. Ellers er det greit å bruke flere mekaniske utskytningsmekanismer, som «pusterør», ulike typer strikkanoner/spretttert-mekanismer o.l.

I dette opplegget kan det også være bra å diskutere sikkerhetstiltak. Dersom man bruker noe litt «eksplosivt», kan dette være farlig? Hva med eventuell brannfare? Hva om «raketten» når skikkelig høyt – og treffer noe når det kommer ned igjen? Bør man helst gjennomføre testene ute?

Elever som ønsker mer avansert programmering kan bruke Python for å gjennomføre regresjonen. Dette står beskrevet i oppleggene 25 – 28 i elevboka og lærerboka.

Ekstraoppgåver

Ekstraoppgåve

1. Korleis kan vi finne ut om telys brenn fullstendig eller ufullstendig?
2. Kva har fullstendige/ufullstendige forbrenningsreaksjonar med kroppen vår å gjere?

Krevande ekstraoppgåve

1. Finn ut kva oksidasjon er. Forklar for dei andre i klassen.
2. Sjå meir på rakett drivstoff. Kva for nokre stoff blir oksiderte? Kva tyder det at dei blir oksiderte?

HINT til ekstraoppgåve:

Fullstendig forbrenning skjer i kroppen ved aerob trening, medan ufullstendig forbrenning skjer under anaerob trening.

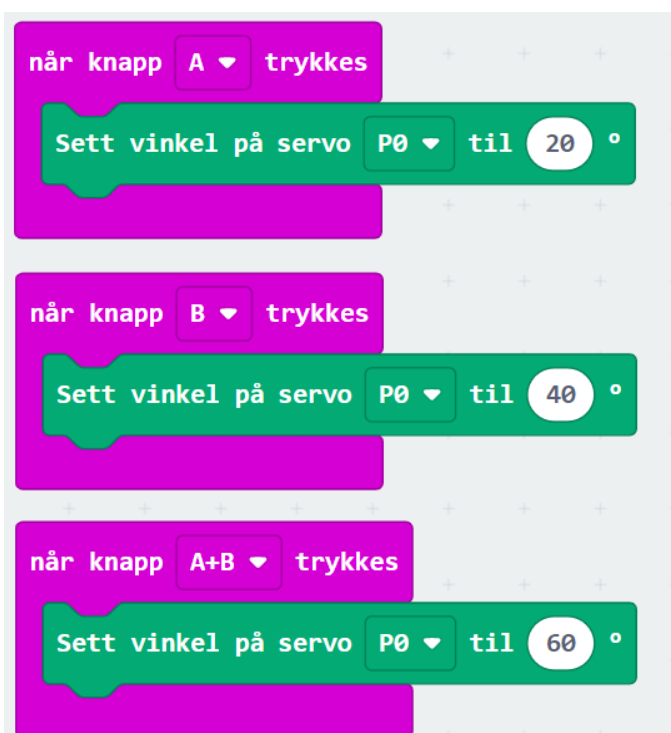
Programmeringsoppgåver

I dette opplegget vil programmeringsoppgåvene vere særse enkle, ein skal berre bruke éin servo for å stille inn ein vinkel, og treng da berre to klossar. Dermed er det liten vits i å ha andre typar programmeringsoppgåver enn ei helt åpen oppgåve, med hint om kva for nokre klossar elevane skal bruke dersom dei står fast.

Dersom elevane ønsker det, kan dei bruke Python for å gjennomføre regresjonen. Dette står beskrevet i kapittel 13.

Dersom nokon elevar ønsker å programmere servoen med hjelp av Python, kan dei opplegg 4 – Energiformer og -overgangar.

Løysingsframlegg for innstilling av vinkel med servo



Servo med Python

- krevende programmering

Løsningsforslag for servobiblioteket som må legges over som et bibliotek på micro:biten. Fremgangsmåte står beskrevet i kapittel 12.

```
from microbit import *

def servovinkel(pin, vinkel):
    y = round((0.11378 * vinkel) + 29.72)
    #set_analog_period(50)
    pin.write_analog(y)

def servohastighet(pin, prosent):
    y = round((0.1024 * prosent) + 29.72)
    #set_analog_period(50)
    pin.write_analog(y)
```

Løsningsforslag for innstilling av vinkel med servo – må bruke 180°-servo

```
1 from servo import servovinkel
2 from microbit import *
3
4 if button_a.is_pressed():
5     servovinkel(pin1, 20)
6
7 if button_b.is_pressed():
8     servovinkel(pin1, 40)
```

Løsningsforslag for innstilling av dreiehastighet med servo – må bruke kontinuerlig servo

```
from servo import servohastighet
from microbit import *

if button_a.is_pressed():
    servohastighet(pin1, 50)
if button_b.is_pressed():
    servohastighet(pin1, -75)
```

Opplegg 32: Geriljabroderi

Kompetansemål

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer:

- Micro:bit med batteripakke og LED-pærer (evt annen elektronikk som summere o.l.)
- Ledende sytråd, nål, broderitråd, stoff (AIDA)

TIPS!

Det er lurt å la elevene bruke et program som kan lage mønster for dem, og det finnes veldig mange tilgjengelig. Disse kan også konvertere bilder til mønster.

For apple-produkter er Stitch Sketch et fint valg

<http://www.iktsoft.net/kgchart-en/StitchSketch/>

For windows er KG-chart for cross stitch et fint valg

<https://www.iktsoft.net/kgchart-en/kgchart/>

Programmeringsoppgave

I dette opplegget vil programmeringsoppgavene være enkle, man kan f.eks. bruke LED-pærer, en høyttaler/buzzer (summer) som gir lyd eller en servo for bevegelse, eller en kombinasjon.

Eksempler på programmeringsoppgaver står i oppleggene 2 -4, i tillegg til oppleggene 9 – 11 for musikk.

Opplegg 33: Lag en teknologisk 17.mai-fane

Kompetansemål

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

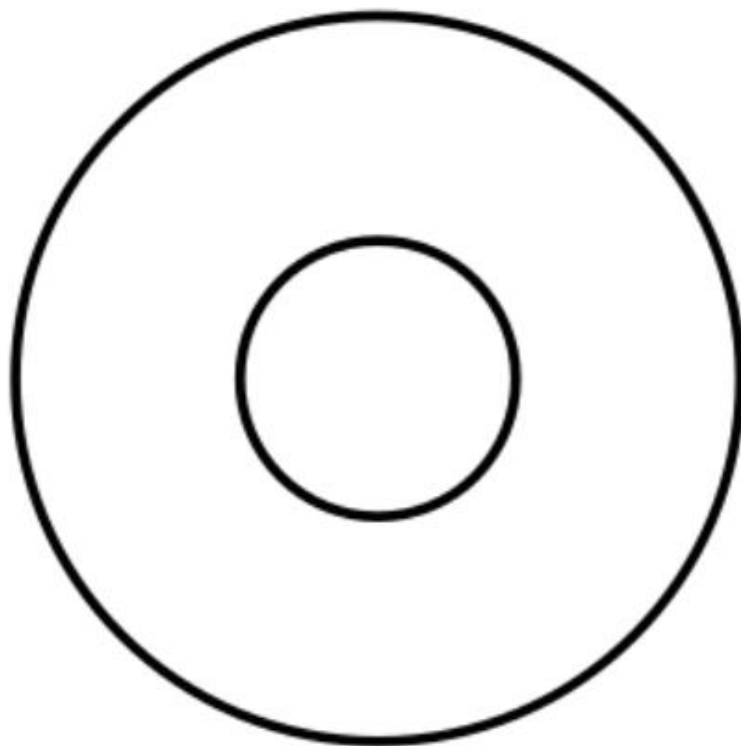
Materialer:

- Micro:bit med batteripakke, alligatorledninger og LED-pærer (evt annen elektronikk som summere o.l.)
- Ark, tusj og tape, papp, garn, saks, limpistol, klistermerker, silkepapir, garn, glitter +++

TIPS!

Elevene kan gjerne lage papirvindmøller eller garndusker til å dekorere fanen, maler ligger på de to neste sidene.

Mønster til garnduskar



Framgangsmåte

1. Klipp ut to stykke av malen i litt tjukk papp og legg dei oppå kvarandre.
2. Snurr garn i ønska farge rundt pappmalane og ned i midten slik at du får ei tjukk smultringform til slutt. Hald fram så lenge du får plass til garn i midten.
3. Tre garn gjennom ei stor nål, stikk henne mellom dei to pappmalene og før henne rundt midten og attende til der du stakk henne inn. Ta nåla ut, men la garntråden vere i ei lykkje mellom dei to pappmalane. Knyt garntråden stramt samstundes som nokon klipper over trådane mellom dei to pappmalane langs heile ytterkanten.
4. Fjern pappmalane og klipp av garnbitar med ujamn lengd.

Framgangsmåte til papirvindmølle



Programmeringsoppgave

I dette opplegget vil programmeringsoppgavene være enkle, man kan f.eks. bruke LED-pærer, en høyttaler/buzzer (summer) som gir lyd eller en servo for bevegelse, eller en kombinasjon.

Eksempler på programmeringsoppgaver står beskrevet i oppleggene 2 -4, i tillegg til i oppleggene 9 – 11 for musikk.

Opplegg 34: Lag et produkt som bruker viss-kloss(ar)

Kompetansemål

K&H:

- utvikle løsninger gjennom en stegvis designprosess og vurdere holdbarhet, funksjon og estetisk uttrykk
- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy
- reflektere kritisk over visuelle virkemidler og eksperimentere med ulike visuelle uttrykk i en skapende prosess

Materialer: Alt mogleg – mest mogleg!

TIPS!

Det kan vere fint å ta ein klasseromdiskusjon om kva skilnaden på ein digital- og ein analogverdi. Den digitale kan berre vere enten av eller på, det vil seie 1 eller 0, der 1 tyder at spenninga er på, og 0 tyder at spenninga er skrudd av. Ein analog verdi kan variere mellom fleire verdier. For ein sensor er det ofte vanleg at det varierer mellom 0 og 255 eller mellom 0 og 1023.

Framlegg til produkt:

- Automatisk lyspåskruar – skru på ei tilkopla LED-pære når det blir mørkt i rommet og skru henne av når det blir lyst att eller etter ei viss tid (hugs å legge inn ei forsinking i programmet, elles vil pæra skru seg av att med ein gong.)
- Nattlys – éi eller fleire LED-pærer som skru seg automatisk av etter ein time (når ein har sovna).
- Brannalarm som piper med for høg temperatur
- Innbrotsalarm - ein alarm piper om ei dør blir åpna

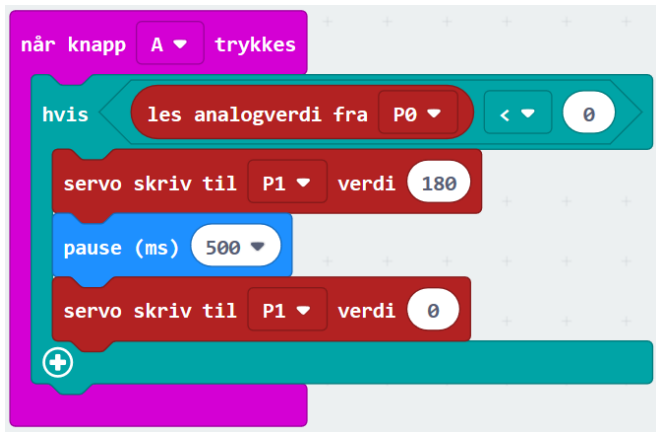
Ekstraoppgåve

Lag ein liten film der du forklarar korleis produktet ditt verkar.
Hugs å forklare programmet og koplingene du har laga.

Programmeringsoppgave

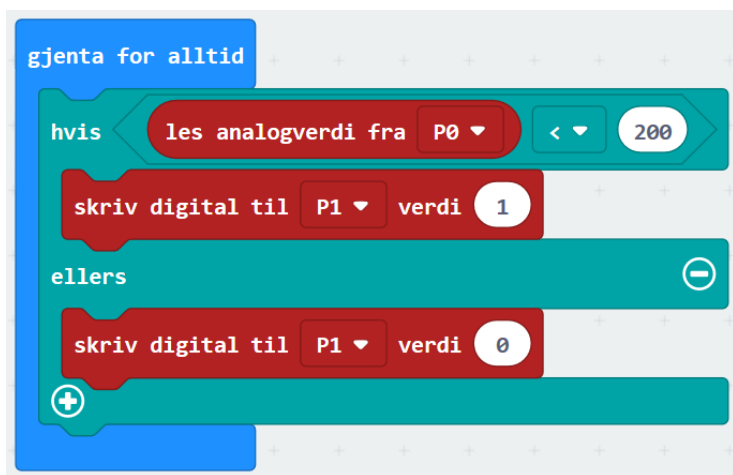
- løsningsforslag

Programmet blir en variant av denne typen program. Her leser vi av en sensor som er koblet til P0-porten på micro:biten. Ut fra hvilken verdi vi leser av, så skjer det noe. I dette tilfellet, så beveger den armen til en 180-graders servo frem og tilbake.

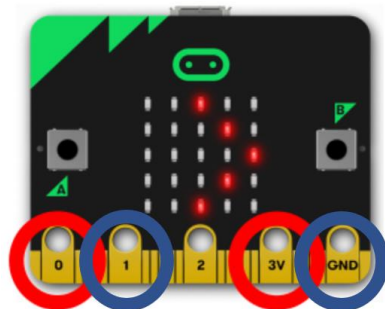


Alternativt løsningsforslag

Her må verdien som sammenlignes med verdien som leses fra P0 finnes ved å først ta en (helst noen flere) felles avlesninger i klasserommet. Ta gjerne en måling først med vanlig springvann og noter hvilken verdi dere får for spenningen. Deretter kan dere ta noen målinger med forskjellig saltkonsentrasjon, også kan klassen i fellesskap bestemme en grenseverdi. Eventuelt kan lærer nevne at man må finne ut hvilken verdi man skal sammenligne med, men ikke gi løsningen før elevene kommer så langt at de lur.



Dette programmet vil slå på det som er festet til P1-porten på micro:biten når den leser en verdi som er mindre enn 200 fra P0-porten. Da må tilkoblingen se slik ut:



Her må alligatorledningene med sensoren være festet til 0 og 3V, siden vi skal måle hvor stor del av spenningen som er igjen etter at strømmen har gått gjennom sensoren. Det er P0-porten vi leser av data fra. Til port P1 og GND (jord) må vi koble det vi skal skru på når vi har en liten strøm ut fra sensoren. Det kan være en LED-pære eller noe annet. Husk at LED-pæra må ha det lengste beinet til P1-porten, ellers vil den ikke kunne skrue seg på.

Se tips til løsningsforslag for Python-programmer i kapittel 12.

Sonar med Python

Løysingsframlegg for sonarbiblioteket som må leggst over som eit bibliotek på micro:biten. Framgangsmåten er forklart i kapittel 12.

```
from microbit import *
from machine import time_pulse_us

def HCSR04():
    trig = pin0
    echo = pin1

    trig.write_digital(0)
    echo.read_digital()

    trig.write_digital(1)
    trig.write_digital(0)

    micros = time_pulse_us(echo, 1)
    t_echo = micros / 1000000

    dist_cm = (t_echo / 2) * 34300
    return dist_cm
```

Løysingsframlegg for eit program som måler avstanden i cm og viser verdien på skjermen til micro:biten

```
from hcsr04 import HCSR04
from microbit import *

while True:
    print((HCSR04(),))
    sleep(100)
```

Program for lage .txt-fil med Python Mu

Fremgangsmåte står beskrevet i kapittel 12.

```
from microbit import *

with open('measured_data.txt', 'w') as min_fil:
    while not button_a.is_pressed():
        t = temperature()
        min_fil.write(str(t)+"\n")
        sleep(1000)
with open('measured_data.txt') as min_fil:
    innhold = min_fil.read()
print(innhold)
```

```
from microbit import *

with open('datasett.txt', 'w') as fil:
    while not button_a.is_pressed():
        t = temperature()
        fil.write(str(t)+'\n')
        sleep(1000)
with open('datasett.txt') as fil:
    innhold = fil.read()
print(innhold)
```

<https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/storage.html>