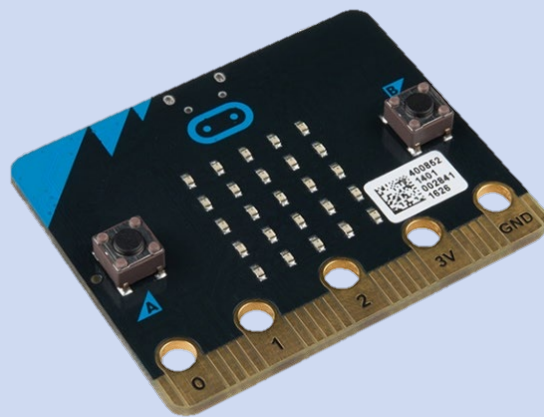


*Johanna Sexe*

# Undervisningsopplegg med micro:bit for naturfag VGS



NTNU

**Skolelaboratoriet**  
for matematikk naturfag  
og teknologi

Juni 2020



# **Undervisningsopplegg med micro:bit for naturfag VGS**

Johanna Sexe

Trondheim 2020

Redigering: Johanna Sexe

Layout: Johanna Sexe, Skolelaboratoriet

Tekst og bilder: Johanna Sexe

Forsidebilde: "[micro:bit Board](#)" av [SparkFun Electronics](#) ([CC BY 2.0](#))

Baksidebilde: "[Code Quest](#)" av [Abington Free Library](#) ([CC BY-NC 2.0](#))

Faglige spørsmål rettes til:

**Skolelaboratoriet ved NTNU**

v/ Johanna Sexe ([johanna.sexe@trfk.no](mailto:johanna.sexe@trfk.no))

evt. Nils Kr. Rossing ([nils.rossing@ntnu.no](mailto:nils.rossing@ntnu.no))

Skolelaboratoriet ved NTNU

Realfagbygget, Høgskoleringen 5

7491 Trondheim

Telefon: 73 55 11 36

<http://www.ntnu.no/skolelab>

Utgave 1 - 24.06.20



## Forord

Hensikten med dette heftet er å presentere noen forslag til aktiviteter i programmering med utgangspunkt i mikrokontrolleren micro:bit i naturfag. Aktivitetene krever lite utstyr utenom et klassesett med micro:bit, USB-kabler og batteripakker.

Læreplanen i naturfag for vg1 studieforbereende utdanningsprogram og vg3 påbygging inneholder kompetansemålet: «Mål for opplæringen er at eleven skal kunne vurdere og lage programmer som modellerer naturfaglige fenomener». Dette kompetansemålet er tenkt å kunne kobles til andre kompetansemål i læreplanen for naturfag, og aktivitetene i heftet forsøker å gjøre nettopp dette. Aktivitetene er presentert med fullstendig løsningsforslag for å støtte lærere og elever, men de kan også brukes som ideer til videre arbeid med programmering i naturfag. Programmene i heftet er skrevet i blokkprogrammering (Microsoft MakeCode), men aktivitetene kan også gjennomføres i et tekstbasert språk som f.eks. Python – og da åpner det seg også flere muligheter for forbedring og videreutvikling av aktivitetene.

Dette heftet inneholder ingen detaljert beskrivelse av hvordan micro:bit er bygget opp. For de som aldri har brukt micro:bit før, kan denne nettsiden være en god start:

<https://support.microbit.org/support/solutions/articles/19000013983-what-is-a-micro-bit->

### Forslag til differensiering av aktivitetene

Differensiering av aktivitetene kan foregå på flere måter. Noen elever kan ha god kjennskap til programmering fra før og trenger større utfordringer, mens andre elever er lite kjent med programmering og trenger mer hjelp. Man kan se for seg tre hovednivåer av differensiering:

- Problemstilling, metode og utstyr er gitt og programløsningen er beskrevet og forklart slik at det er mulig å kopiere med forståelse.
- Problemstilling, metode og utstyr er gitt, men programløsningen er relativt åpen, kanskje kun med noen antydninger om veien å gå.
- Problemstillingen er gitt, men elevene skal velge metode, skaffe tilveie utstyr og skrive programmet selv.

I tillegg inneholder alle aktivitetene forslag til videre arbeid som også kan fungere som ekstra utfordringer for de elevene som har behov for det.

Aktivitet 1 i dette heftet er ment som en enkel opplæring i hvordan man bruker den nettbaserte editoren Microsoft MakeCode og laster det ferdige programmet inn på micro:bit. Det kan derfor være lurt å gjøre denne aktiviteten først. De resterende aktivitetene er sortert omtrentlig etter økende vanskelighetsgrad.

Lykke til med kodinga!

Trondheim, 24. juni 2020  
Johanna Sexe



## Innhold

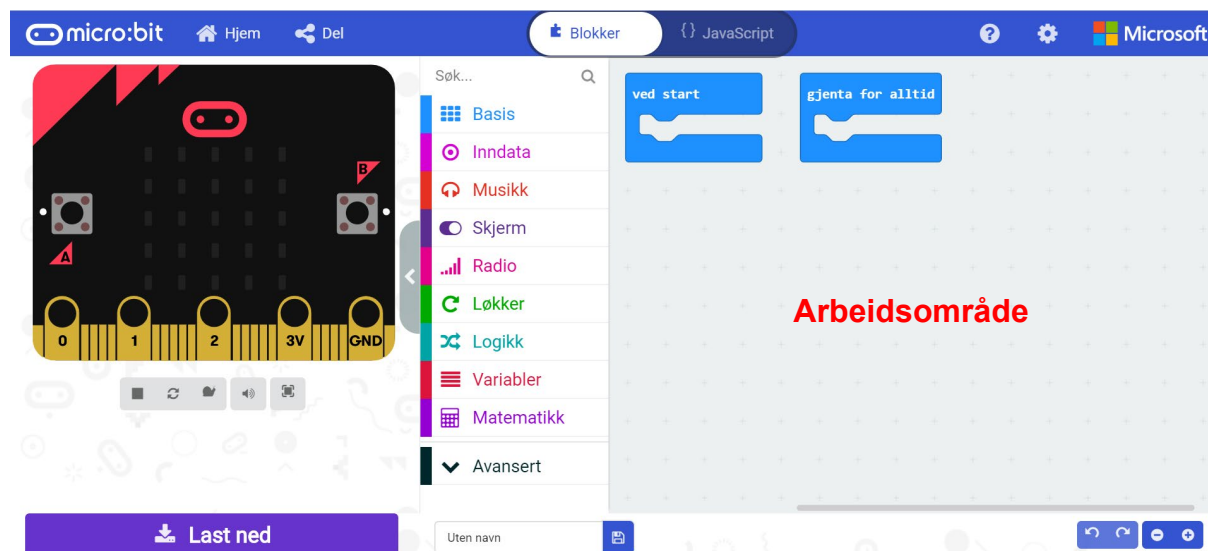
1	Kom i gang med micro:bit – lær å lage og laste inn programmer på micro:bit .....	7
2	Ernæring og helse – lag en skritteller .....	9
2.1	Innledning .....	9
2.2	Oppgave .....	9
2.3	Programmeringskonsepter benyttet i denne aktiviteten.....	9
2.4	Utstyr .....	9
2.5	Fremgangsmåte .....	9
2.6	Forslag til videre arbeid .....	10
3	Trådløs kommunikasjon – lag en trådløs datalogger .....	11
3.1	Innledning .....	11
3.2	Oppgave .....	11
3.3	Programmeringskonsepter benyttet i denne aktiviteten.....	11
3.4	Utstyr .....	11
3.5	Fremgangsmåte .....	11
3.6	Forslag til videre arbeid .....	12
4	Gener og arv – lag et krysningsskjema .....	13
4.1	Innledning .....	13
4.2	Oppgave .....	13
4.3	Programmeringskonsepter benyttet i denne aktiviteten.....	13
4.4	Utstyr .....	13
4.5	Fremgangsmåte .....	14
4.6	Forslag til videre arbeid .....	17
5	Klimaendringer og biologisk mangfold – mål temperaturen i solfangere.....	18
5.1	Innledning .....	18
5.2	Oppgave .....	18
5.3	Programmeringskonsepter benyttet i denne aktiviteten.....	18
5.4	Utstyr .....	18
5.5	Fremgangsmåte .....	18
5.6	Forslag til videre arbeid .....	21
6	Radioaktivitet – simulering av halveringstid .....	22
6.1	Innledning .....	22
6.2	Oppgave .....	22
6.3	Programmeringskonsepter benyttet i denne aktiviteten.....	22
6.4	Utstyr .....	22
6.5	Fremgangsmåte .....	22
6.6	Forslag til videre arbeid .....	24



# 1 Kom i gang med micro:bit – lær å lage og laste inn programmer på micro:bit

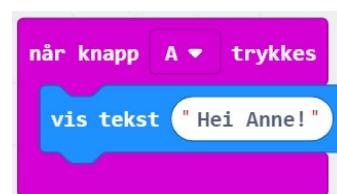
Gå inn på nettsiden [microbit.org](http://microbit.org) og trykk på «Let's code» øverst på siden. Velg «MakeCode editor» for blokkprogrammering.

Velg «Nytt prosjekt». Da skal du ha fått opp en side som ligner bildet under. Bruk tannhjulet øverst til høyre for å velge språk. Bildene i dette heftet bruker norsk språk.

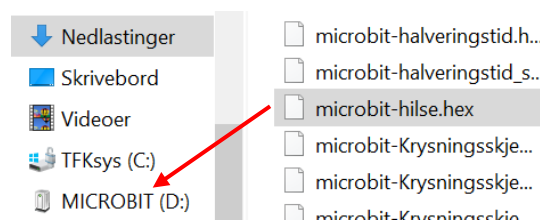


Vi starter med å lage et lite program som gjør at micro:bit hilser på deg:

1. Finn den rosa blokken «når knapp A trykkes» i menyvalget «Inndata» og dra den over til arbeidsområdet.
2. Finn den blå blokken «vis tekst 'Hello!'» i menyvalget «Basis» og sett den inni den lille blokken. Skriv en hilsen som micro:bit skal vise i den hvite bobla, f.eks. som vist på bildet til høyre.
3. Gi kodefila et navn i den hvite ruta nederst på skjermen der det nå står «Uten navn», og klikk deretter på «Last ned»-knappen. Kodefila finner du deretter i mappen «Nedlastinger» i Filutforsker.



4. Koble micro:bit til PC-en med USB-kabelen. I Filutforsker drar og slipper du programfila du laget (med etternavnet .hex) fra mappen «Nedlastinger» og over til MICROBIT (D:). Det gule lyset på baksiden av micro:bit skal nå blinke for å indikere at programfila overføres.





5. Etter at fila er ferdig overført kan du trykke på knapp A på micro:bit, og hilsenen du valgte skal rulle over skjermen. Husk at micro:bit må ha strøm for å fungere. Det får den enten ved å være koblet til PC med USB-kabelen, eller ved å være koblet til batteripakken.
  
6. *Hvis du bruker Mac: Micro:bit-fila finner du ved hjelp av Finder, og drar og slipper den til micro:bit-mappen.*

Gratulerer – du har skrevet ditt første program for micro:bit!





## 2 Ernæring og helse – lag en skritteller

### 2.1 Innledning

Micro:bit har flere innebygde sensorer, og en av disse er et akselerometer som kan registrere bevegelse. Her utnytter vi denne sensoren for å lage en enkel skritteller.

Denne aktiviteten kan knyttes til i læreplanmålene «gjøre rede for funksjonene til noen næringsstoffer og diskutere hvorfor et variert kosthold er viktig i et helse- og bærekraftsperspektiv» og «drøfte aktuelle helse- og livsstilsspørsmål og vurdere pålitelighet i informasjon fra ulike kilder».

Utvidelsen av aktiviteten kan også knyttes til læreplanmålet «drøfte hvordan utvikling av naturvitenskapelige hypoteser, modeller og teorier bidrar til at vi kan forstå og forklare verden».

### 2.2 Oppgave

Lag et program som gjør micro:bit til en skritteller som viser antallet skritt du har gått på skjermen, og som kan nullstilles.

### 2.3 Programmeringskonsepter benyttet i denne aktiviteten

Input, output og variabler.

### 2.4 Utstyr

En micro:bit med batteripakke, en USB-kabel og PC.

### 2.5 Fremgangsmåte

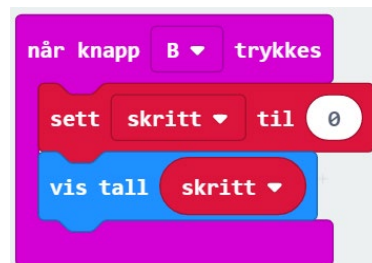
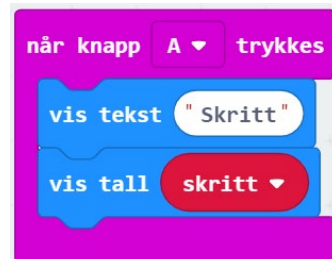
1. Velg input-blokken «når ristes» i menyvalget «Inndata» for å kunne registrere om micro:bit ristes.
2. For at micro:bit skal kunne telle skrittene må vi lage en variabel som kan lagre antallet skritt. I menyvalget «Variabler» lager du en variabel du kaller «skritt».
3. Blokken «endre skritt med 1» setter vi deretter inn i «når ristes»-blokka, da vil antallet skritt øke med en for hver gang micro:bit ristes når vi går.



I tillegg ønsker vi at microbit skal vise på skjermen hvor mange skritt vi har tatt:



4. Velg blokken «vis tall» i menyvalget «Basis» og sett inn i «når ristes» blokka som vist på bildet til høyre.
5. Fra menyvalget «Variabler» henter du den runde blokka «skritt» og legger inn i «vis tall»-blokka.
6. Det kan være vanskelig å lese av antallet skritt underveis, så vi kan legge inn en kode som viser oss sluttresultatet når vi trykker på knapp A som vist på bildet i midten til høyre.
7. For å nullstille skrittelleren når vi ønsker å starte en ny måleserie kan vi legge inn kode som gjør at antallet skritt nullstilles når vi trykker på knapp B. Dette kan vi gjøre som vist på bildet nederst til høyre.



Gi programmet ditt et fornuftig navn og last det over på micro:bit. Nå kan du koble microbit fra PC-en, koble til en batteripakke og ta med deg micro:bit på en spasertur. Fest den i buksa eller legg den i en lomme og se etterpå hvor mange skritt du har gått!

## 2.6 Forslag til videre arbeid

- Skrittelleren kan brukes til å registrere aktivitet gjennom f.eks. et døgn. Hvordan er sammenhengen mellom aktivitet og fysisk og psykisk helse? Hva kan vi gjøre for å øke aktivitetsnivået?
- Hvor nøyaktig måler skrittelleren? Planlegg og gjennomfør flere forsøk og undersøk hvor nøyaktig skrittelleren er. Hvorfor gjør forskere alltid mange målinger av det som skal undersøkes? Hva kan vi gjøre for at vi bedre skal kunne stole på resultatene fra målingene våre?
- Hvordan kan skrittelleren gjøres mer pålitelig? Kan du endre programmet eller utformingen av skrittelleren slik at målingene blir mer nøyaktige?



## 3 Trådløs kommunikasjon – lag en trådløs datalogger

### 3.1 Innledning

I hverdagen omgir vi oss med trådløs kommunikasjon hele tiden som f.eks. Wi-Fi, radio, mobiltelefoner, utstyr med blåtannteknologi og mye mer. Micro:bit har en innebygd radiosender/mottaker som gjør at to eller flere micro:bit kan kommunisere trådløst med hverandre (på frekvensen 2,4 GHz). Dette kan vi bruke til å lage fjernkontroller til radiostyrte roboter, spille spill mot andre micro:bit og mye annet.

Denne aktiviteten kan knyttes til læreplanmålet «forklare hovedprinsippene for trådløs kommunikasjon og gi eksempler på hva slik teknologi brukes til».

### 3.2 Oppgave

Lag en trådløs datalogger der en micro:bit foretar målinger og sender måleverdiene til en annen micro:bit som viser verdiene på skjermen.

### 3.3 Programmeringskonsepter benyttet i denne aktiviteten

Input, output og løkker.

### 3.4 Utstyr

To micro:bit med batteripakker, USB-kabel og PC.

### 3.5 Fremgangsmåte

Først lager vi programmet til den micro:biten som skal foreta målinger og sende måleverdiene. Vi velger her å la micro:bit måle temperatur, men det er også mulig å måle f.eks. akselerasjon eller lysintensitet.

1. I menyvalget «Radio» finner vi blokka «sett gruppe...». Denne legger vi inn i «ved start»-blokka for å angi hvilken 'kanal' vi ønsker at micro:bit skal sende på. For å unngå at dine micro:bit forstyrres av signaler fra andre må alle elevgrupper velge sin egen 'kanal'. Du kan velge kanalnummer fra 0 til 255. Her har vi satt kanalnummeret til 10.



2. I blokka «gjenta for alltid» lager vi hovedprogrammet. All kode inne i denne blokka gjentas så lenge micro:bit er skrudd på. Her velger vi at micro:bit skal sende temperaturverdier ved å bruke blokka «radio send tall», og inne i den hvite bobla setter vi inn blokka «temperatur» som vi finner i menyvalget «Inndata». I tillegg kan vi legge inn en pause mellom hver gang en måling





sendes, her er det valgt 30 000 millisekunder (dvs. 30 sekunder) mellom hver måling.

3. Nå er programmet som skal foreta målingene ferdig. Gi programfila et fornuftig navn og overfør det til den ene micro:biten. Koble micro:bit til en batteripakke.

Deretter lager vi programmet til den micro:biten som skal motta måledataene, du må åpne et nytt prosjektvindu i nettleseren din:

4. Dette programmet må også starte med å velge en 'kanal' for mottak av radiosignaler. Dette gjør du på samme måte som vist for det forrige programmet (punkt 1). Sørg for å velge samme kanalnummer som sist.

5. For å gi micro:bit beskjed om hva den skal gjøre når den mottar et radiosignal velger vi blokka «når radio mottar...» som vi finner i menyvalget «Radio». Vi ønsker at micro:bit skal vise måleverdien den mottar på skjermen, derfor velger vi blokka «vis tall...» fra «Basis»-menyen og holder og drar den røde bobla med «receivedNumber» over i den hvite bobla med tallet 0. Det er også mulig å legge inn litt tekst som skal vises på skjermen som vist på bildet. Blokka «tøm skjermen» er valgfri. Hva skjer hvis du ikke tar den med?



6. Gi programmet ditt et fornuftig navn (ikke det samme navnet som forrige program), og last det over på micro:bit. La micro:bit være koblet til PC-en for å få strøm, eller koble den til en batteripakke.

Nå kan du legge dataloggeren din (micro:biten med programmet du laget i punkt 1 - 3) på det stedet der du ønsker å foreta målingene, og følge med på målingene på den andre micro:biten.

### 3.6 Forslag til videre arbeid

- Rekkevidden til radiosignalene er ganske gode, kan du teste ut hvor stor rekkevidde signalene har? Hva kan hindre signalene i å nå fram til mottakeren? Hva er et Faradaybur – og kan du kanskje lage et?
- Micro:bit måler antakelig noe høyere temperatur enn forventet. Dette er fordi temperatursensoren blir litt påvirket av varmeutviklingen i prosessoren til micro:bit. Prøv å lage kode som korrigerer for dette!



## 4 Gener og arv – lag et krysningskjema

### 4.1 Innledning

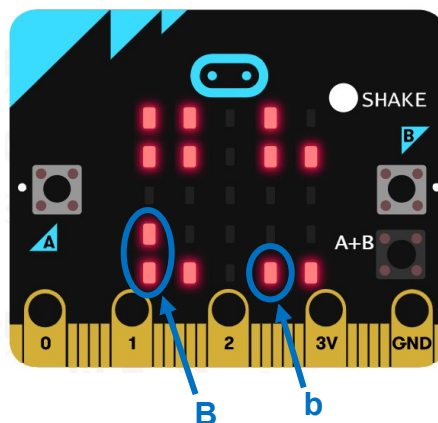
For å holde oversikt over hvilke genkombinasjoner som er mulig for en bestemt egenskap når vi skal studere arv fra foreldre til avkom er et krysningskjema nyttig. I et slikt skjema gir vi genene bokstavsymboler der stor bokstav symboliserer et dominant gen, og liten bokstav symboliserer et recessivt gen. Skjemaet viser oss hvilke kjønnsceller som dannes, og hvilke egenskaper krysningsen kan gi. Som eksempel kan vi tenke oss at vi skal lage en oversikt over hvilke genkombinasjoner to foreldre med samme øyefarge kan få. Vi tenker oss at far har genkombinasjonen Bb der B er genet for brune øyne, og b er genet for blå øyne. Mor har også genkombinasjonen Bb, og et krysningskjema viser oss hvilke genkombinasjoner som er mulig for barna deres.

		Far	
		B	b
Mor	B	BB	Bb
	b	Bb	bb

Denne aktiviteten kan knyttes til læreplanmålet «beskrive DNA og hvordan egenskaper arves, og gjøre rede for hvordan arv er en forutsetning for evolusjon».

### 4.2 Oppgave

Lag et program som viser et krysningskjema på skjermen til micro:bit etter at du har registrert fars og mors genkombinasjoner. Du skal altså først kunne registrere fars og mors genkombinasjoner ved hjelp av f.eks. knappene på micro:bit, og deretter skal krysningskjemaet som passer til disse genkombinasjonene vises på skjermen. Et slikt skjermbilde kan se ut som vist til høyre (her har både far og mor fått genkombinasjonen Bb) der skjermen er delt inn i fire felt. To led-lys symboliserer genvarianten B, mens et led-lys symboliserer genvarianten b.



### 4.3 Programmeringskonsepter benyttet i denne aktiviteten


Input, output, variabler, betingelser og løkker.

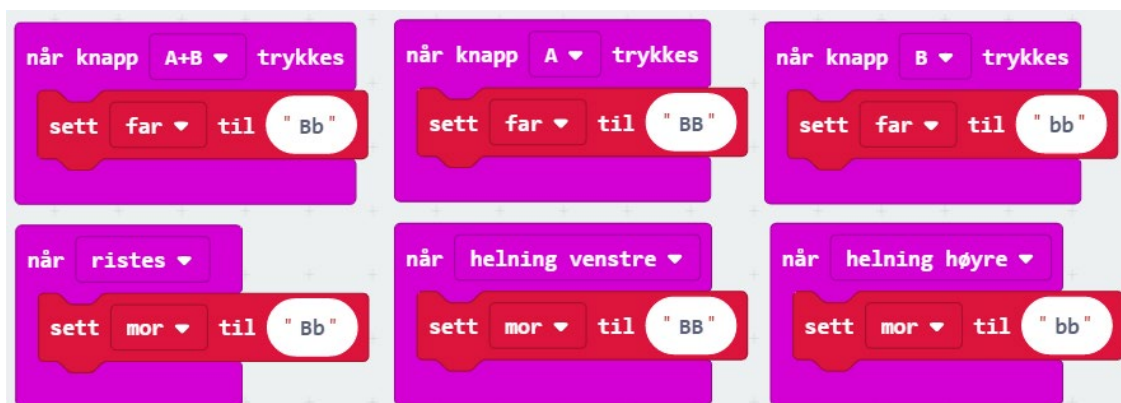
### 4.4 Utstyr

En micro:bit, en USB-kabel og PC.



## 4.5 Fremgangsmåte

1. Først lager vi variablene som senere skal inneholde fars og mors genkombinasjoner:  
I menyvalget «Variabler» velger du «Lag en variabel» og skriver inn variabelnavnet, f.eks. «far». Lag en variabel for «mor» også. Velg blokken «sett far til 0» og legg den inn i «ved start»-blokka som vist på bildet. Bytt ut den hvite sirkelen som viser «0» med en sirkel som viser «' », den finner du i menyvalget «Tekst» (klikk på «Avansert» nederst i menyen for å få det opp). Dette må vi gjøre fordi vi senere har tenkt å fylle variablene med tekst og ikke tall, altså bokstavkombinasjonen for gener. Ved å legge disse variabelblokkene i «ved start»-blokka vil nå variablene resettes når programmet startes på nytt.  

2. Deretter lager vi muligheter til å registrere genkombinasjoner for far og mor. Dette kan vi gjøre ved å endre innholdet i variablene vi laget i punkt 1 når vi gjør noe med micro:bit, som f.eks. ved å trykke på knapper eller riste den. Blokkene vi trenger finner vi i menyvalget «Inndata», og i hver blokk legger vi inn den genkombinasjonen vi ønsker at variabelen skal få når dette skjer. Det kan se slik ut, men her kan du velge det du selv synes er hensiktsmessig:



3. Nå skal vi få skjermen på micro:bit til å vise et krysningskjema basert på de genkombinasjonene vi har registrert. Dette legger vi inn i «gjenta for alltid»-blokka slik at skjermbildet hele tiden oppdateres etter hvert som vi gjør nye genregistreringer. Denne blokka gjentar all kode som ligger inne i den i en uendelig løkke.

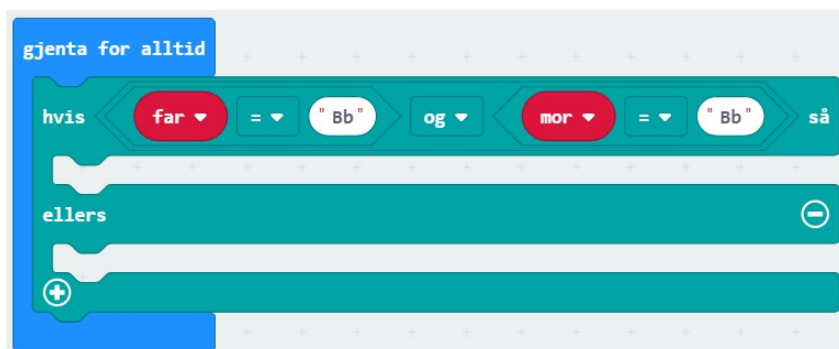
For at programmet vårt skal kunne velge mellom ulike krysningskjemaer må vi bruke blokka «hvis – ellers» som ligger i menyvalget «Logikk». Programmet vårt må velge to genkombinasjoner, en fra far og en fra mor. For å få til dette må vi bytte ut den sekskantede mørke ruten («sann») med en blokk som kan sammenligne to «utsagn», denne finner vi også i menyvalget «Logikk».



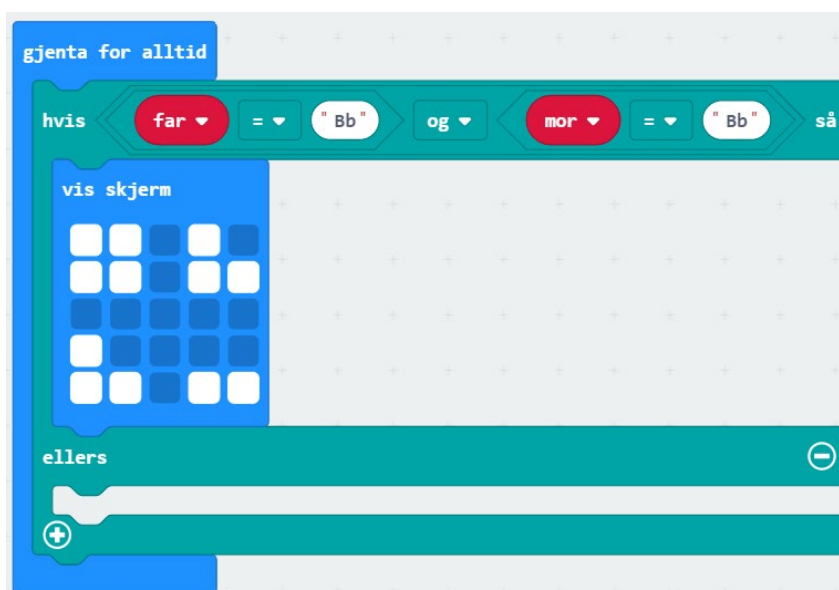
Vi tenker oss nå at hvis far har genkombinasjonen Bb og mor har genkombinasjonen Bb skal vi få fram et bilde på skjermen som



illustrerer kryssingsskjemaet denne genkombinasjonen gir. Først må vi legge inn sammenligningsblokker i de to mørke rutene der vi sjekker om genkombinasjonen Bb er registrert i variablene for far og mor. Sammenligningsblokkene (med «'») finner vi også i menyvalget «Logikk», og den røde variabelblokka henter vi fra menyvalget «Variabler». Etter å ha lagt inn disse blokkene skal det se slik ut:



Menyvalget «Basis» inneholder blokka «vis skjerm» der vi selv kan velge hvilke led-lys på micro:bit som skal skrues på. Denne blokka legger vi inn i «hvis – ellers»-blokka og tegner kryssingsskjemaet som passer til genkombinasjonen. Dette kan f.eks. se slik ut:



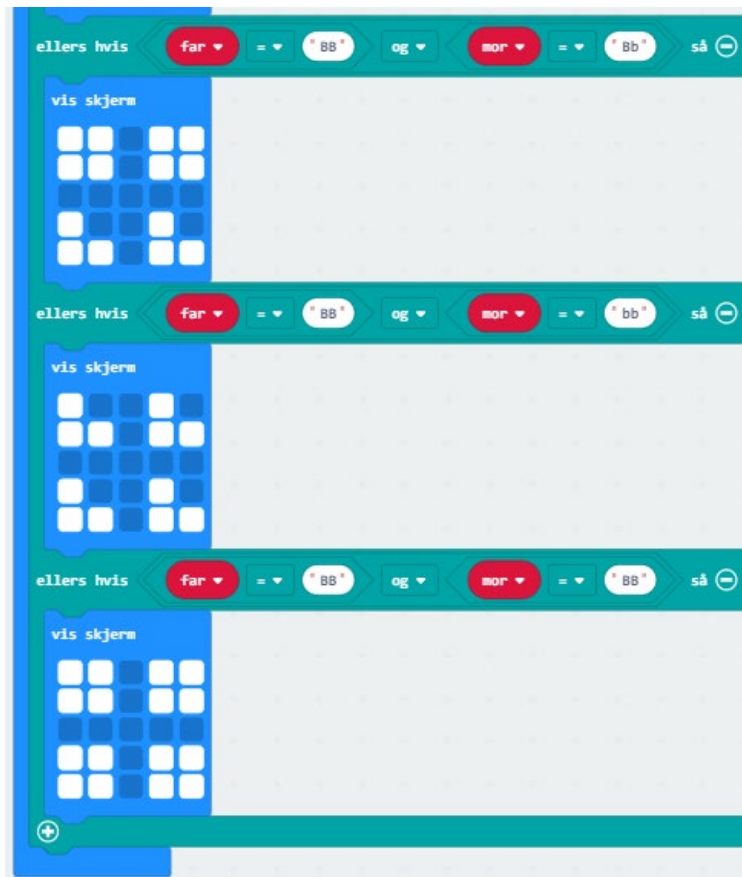
4. Det som gjenstår er å legge inn alle andre mulige genkombinasjoner med tilhørende visning av kryssingsskjema. Ved å trykke på pluss-tegnet nederst på den grønne «hvis – ellers»-blokka dukker det opp en ny linje som heter «ellers hvis», og vi kan få fram så mange slike linjer vi ønsker ved å trykke flere ganger på pluss-tegnet. Skriv ned alle de 9 mulige kryssingsskjemaene på et ark slik at du får oversikt før du lager resten av koden. Til slutt bør det se omtrent slik ut:



The image shows a Scratch script for a micro:bit project. The script is titled "gjenta for alltid" (Repeat forever). It consists of a loop of six blocks:

- Block 1:** "hvis" (if) block with conditions "far" = "Bb" and "mor" = "Bb". The "så" (then) block contains a "vis skjerm" (show screen) block with a 5x5 grid of blue squares.
- Block 2:** "ellers hvis" (else if) block with conditions "far" = "Bb" and "mor" = "BB". The "så" block contains a "vis skjerm" block with a 5x5 grid of blue squares.
- Block 3:** "ellers hvis" (else if) block with conditions "far" = "bb" and "mor" = "bb". The "så" block contains a "vis skjerm" block with a 5x5 grid of blue squares.
- Block 4:** "ellers hvis" (else if) block with conditions "far" = "bb" and "mor" = "bb". The "så" block contains a "vis skjerm" block with a 5x5 grid of blue squares.
- Block 5:** "ellers hvis" (else if) block with conditions "far" = "bb" and "mor" = "Bb". The "så" block contains a "vis skjerm" block with a 5x5 grid of blue squares.
- Block 6:** "ellers hvis" (else if) block with conditions "far" = "bb" and "mor" = "BB". The "så" block contains a "vis skjerm" block with a 5x5 grid of blue squares.





Gi programmet et fornuftig navn, last det inn på micro:bit og sett i gang med å kombinere gener!

#### 4.6 Forslag til videre arbeid

- Finn en måte å legge inn genparene til far og mor som er enklere å bruke enn det som er foreslått her.
- Kan du endre programmet slik at micro:bit angir sannsynligheten for de ulike resultatene i kryssingsskjemaet?
- Prøv å lage et program som viser sykdomsrisiko ved kjønnsbundet arv.



## 5 Klimaendringer og biologisk mangfold – mål temperaturen i solfangere

### 5.1 Innledning

Energi fra sola kan brukes til å varme opp vann. Enkle solfangere laget av plastflasker kan modellere forskjellen i oppvarming av havet med eller uten is (albedoeffekt) eller forskjellen i ismelting hos ren og forurenset is. Dette kan gi et innblikk i hvordan klimaendringer påvirker havtemperatur, ismelting og videre fra dette endring i utbredelsen av arter. Micro:bit har en innebygd temperatursensor som kan brukes til å registrere temperaturendringer over tid, og hvis man ønsker ekstra informasjon er det mulig å måle lysintensitet (led-lysene på skjermen til micro:bit kan fungere som enkle lyssensorer).

Denne aktiviteten kan knyttes til læreplanmålet «gjøre rede for hvordan klimaendringer påvirker evolusjon, utbredelse av arter og biologisk mangfold».

### 5.2 Oppgave

Du skal lage to enkle solfangere og registrere temperaturendringene i solfangerne over et selvvalgt tidsrom ved hjelp av en micro:bit i hver solfanger. Den ene solfangeren skal ha hvit bakgrunn, og den andre skal ha svart bakgrunn. Du skal undersøke hvor store forskjellene i temperaturendring er i de to solfangerne. Blir det forskjeller mellom solfangerne hvis det er overskyet?

### 5.3 Programmeringskonsepter benyttet i denne aktiviteten

Input, output, variabler, betingelser, lister og løkker.

### 5.4 Utstyr

- To micro:bit med batteripakker
- USB-kabel
- PC
- To 1,5-litersflasker
- Plastfolie/plastpose
- Svarte og hvite papirark
- Saks/kniv og teip

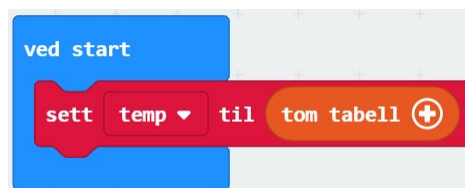
### 5.5 Fremgangsmåte

Først lager vi programmet som skal brukes til å registrere temperaturendringene i flaskene. Vi bruker det samme programmet på begge micro:bitene.

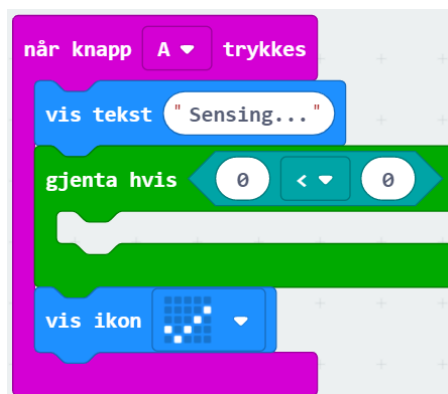
1. For å lagre måleverdiene underveis i forsøket lager vi en tabell som skal fylles med måleverdiene etter hvert som forsøket pågår. For å få til dette må vi først lage en variabel med navnet «temp» i menyvalget «Variabler».
2. I menyvalget «Tabeller» (klikk på «Avansert» for å få fram dette valget) velger vi blokka «sett tabell til...» og legger inn i «ved start»-blokka. Vi endrer «tabell» til



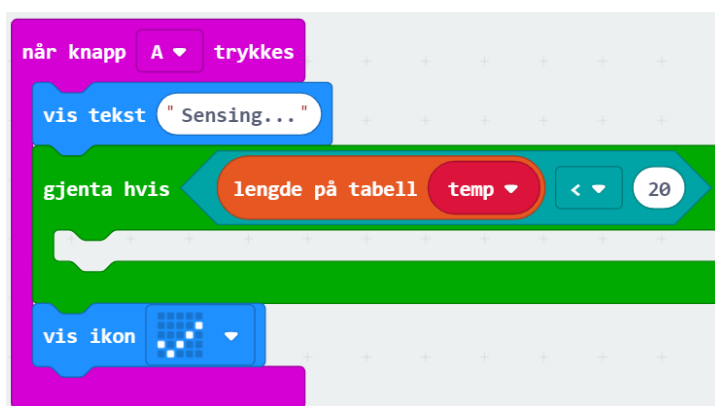
«temp», og deretter bytter vi ut den runde delen der det står «tabell med 1 2...» med den runde blokka «tom tabell» som vist i bildet over. Dette gjør vi for å fortelle programmet at variabelen «temp» skal være en tom tabell.



3. Vi ønsker å starte målingene ved å trykke på knapp A. I menyvalget «Inndata» finner vi blokka som passer. For at det skal være enkelt å se at målingene begynner kan vi vise teksten «Sensing...» på skjermen før selve målesekvensen starter. Målingene skal gjøres flere ganger, derfor går vi til menyvalget «Løkker» og finner blokka «gjenta hvis sann». All kode inne i denne blokka vil gjentas så mange ganger som vi bestemmer. Vi bytter ut den sekskantede ruten «sann» med en sammenligningsblokk fra menyvalget «Logikk». Til slutt kan vi legge inn et skjermbilde som skal vises når målingene er ferdige, her er det valgt «vis ikon √».

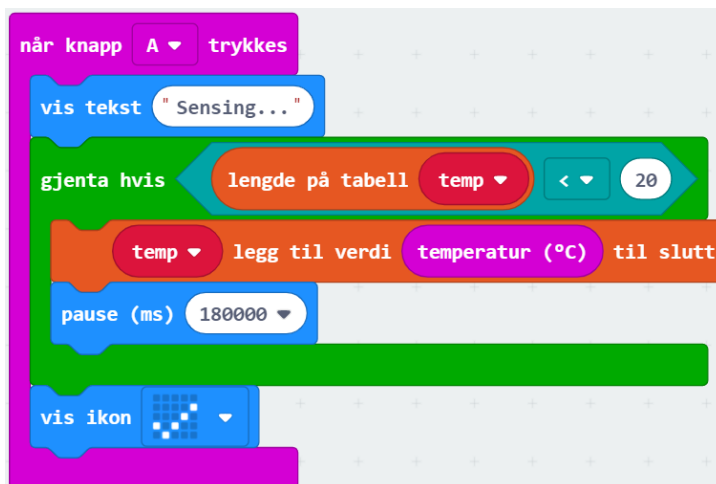


4. I dette tilfellet ønsker vi å foreta 20 målinger. Det vil si at når målingene er ferdige skal tabellen «temp» bestå av 20 tallverdier. Dette kan vi bruke for å styre hvor mange ganger koden inne i «gjenta hvis»-blokken skal repeteres. Vi finner blokka «lengde på tabell» i menyvalget «Tabeller» og setter den inn på den første plassen i sammenligningsblokka. Koden i bildet til høyre sier nå at så lenge tabellen inneholder færre enn 20 verdier skal målingene gjentas. Når tabellen har lagret 20 målinger vil programmet gå videre til neste blokk, her «vis ikon».



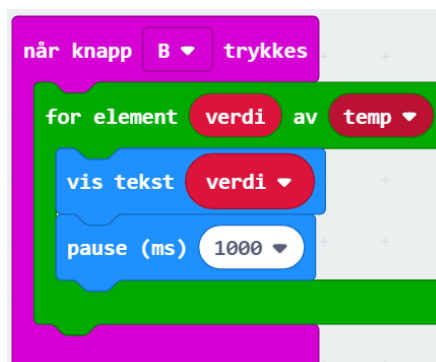


5. Hver gang «gjenta hvis» blokken repeteres skal det legges inn en temperaturmåling i tabellen «temp». Dette får vi til ved å legge inn blokka «list legg til verdi til slutt» fra menyvalget «Tabeller». I den tomme sirkelen legger vi inn blokka «temperatur» fra menyvalget «Inndata». Blokka «pause (ms)» finner vi i menyvalget «Basis» og legger inn en pause i programmet før det kjører videre. Her er det valgt en pause på 180 000 millisekunder (dvs. 3 minutter), altså vil det bli foretatt en temperaturmåling hvert 3. minutt.



Til sammen tar det 60 minutter å foreta de 20 målingene vi ønsker, men her står du fritt til selv å velge antall målinger og et tidsintervall mellom hver måling som passer til ditt forsøk.

6. Etter at forsøket er ferdig må vi ha mulighet til å lese av de målte temperaturverdiene. Dette kan vi gjøre ved å be micro:bit vise måleverdiene på skjermen når vi trykker på knapp B. Vi velger blokka «for element...» fra menyvalget «Løkker». Koden i denne blokka vil gå gjennom tabellen «temp»; vise første tallverdi på skjermen, vente 1 sekund og så vise neste tallverdi i tabellen helt til alle verdiene er vist på skjermen.



Det er ikke mulig å stoppe programmet mens tallverdiene ruller over skjermen, så det kan være lurt å filme micro:bit med mobilen mens dette pågår slik at det blir lettere å notere ned tallverdiene til senere bruk.

7. Gi programmet ditt et fornuftig navn, last det inn i begge micro:bitene, koble fra USB-kabelen og koble en batteripakke til hver micro:bit.

Deretter lager vi solfangerne:

8. Lag to solfangere ved å skjære av toppen på de to flaskene. Teip et hvitt ark som bakgrunn på den ene flasken, og et svart ark som bakgrunn på den andre.
9. Pakk hver micro:bit godt inn sammen med batteripakka i plastfolie/plastpose. Det må bli helt tett slik at micro:biten ikke blir våt!



10. Fyll opp flaskene med like mye vann i hver. Start programmet på micro:bit ved å trykke på knapp A, legg en micro:bit i hver flaske og lukk igjen åpningen øverst med plastfolie. Sett flaskene utendørs i sola og la dem stå helt til registreringene er fullført.
11. Les av resultatene fra målingene ved å trykke på knapp B og notere verdiene fra skjermen ned på et ark. Tips: Film med mobilen mens verdiene ruller over skjermen på micro:bit, da blir det enklere å notere tallverdiene fra filmen etterpå. Hva forteller resultatene dine om de to solfangerne du har laget?

## 5.6 Forslag til videre arbeid

Prøv å utvide programmet ditt slik at micro:bit også måler lysintensiteten gjennom forsøket.

### Tips:

Solfangerne som er beskrevet her vil antakelig ikke gi så stor forskjell i temperaturøkning mellom de to flaskene. Prosjektet kan videreutvikles på ulike måter der elevene kan undersøke hvilke utforminger av solfangerne som gir mest pålitelige resultater. Plastflaskene kan f.eks. dekkes helt med svart og hvitt papir, eller de kan males med svart og hvit maling (eller også andre farger). Aluminiumsfolie kan også benyttes for å få bedre refleksjon av sollyset. For å unngå at micro:bitene blir våte kan dere også prøve luftfylte solfangerer, f.eks. syltetøyglass med lokk.



Foto: Johanna Sexe



## 6 Radioaktivitet – simulering av halveringstid

### 6.1 Innledning

Når en atomkjerne i et radioaktivt stoff sender ut alfa- eller betastråling blir atomkjernen omdannet til et annet grunnstoff. Omdanningen av et enkelt atom skjer på et tilfeldig tidspunkt. Tiden det tar til halvparten av det radioaktive stoffet har blitt omdannet kalles halveringstiden. For å forstå bedre hva halveringstid betyr kan vi bruke et forsøk med terninger, der vi kaster f.eks. 100 terninger og tenker oss at alle terningene som viser sekser tilsvarer en atomkjerne som har blitt omdannet og tas ut av «spillet» før vi kaster terningene på nytt. En variant av denne terningsimuleringen kan også programmeres på micro:bit der vi tenker oss at hvert led-lys på skjermen tilsvarer en atomkjerne. Vi starter med at alle led-lysene på skjermen lyser, og etter hvert som de tenkte atomkjernene omdannes slukkes lysene. Ved å registrere antall gjenværende atomkjerner (eller led-lys) gjennom simuleringen kan vi til slutt tegne opp halveringskurven for micro:bit-atomene.

0, 0 1, 0 2, 0 3, 0 4, 0  
0, 1 1, 1 2, 1 3, 1 4, 1  
0, 2 1, 2 2, 2 3, 2 4, 2  
0, 3 1, 3 2, 3 3, 3 4, 3  
0, 4 1, 4 2, 4 3, 4 4, 4

Denne aktiviteten kan knyttes til læreplanmålet «utforske og beskrive elektromagnetisk og ioniserende stråling, og vurdere informasjon om stråling og helseeffekter av ulike strålingstyper».

### 6.2 Oppgave

Lag et program til micro:bit som simulerer halveringstid på skjermen.

### 6.3 Programmeringskonsepter benyttet i denne aktiviteten

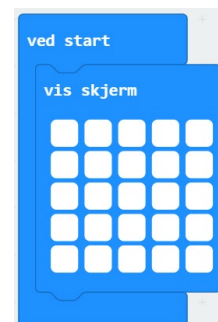
Input, output, variabler, betingelser og løkker.

### 6.4 Utstyr

En micro:bit, en USB-kabel og PC.

### 6.5 Fremgangsmåte

1. Programmet skal starte med at alle led-lysene på skjermen skrus på. Dette kan vi gjøre ved å hente «vis skjerm»-blokka fra «Basis»-menyen og legge den inn i «ved start»-blokka. Deretter klikker vi på alle rutene for å markere at led-lysene skal skrus på.





- Hvert enkelt lys på skjermen har en koordinat på formen (x, y) som vist på figuren øverst på forrige side. For å kunne velge hvilke led-lys som skal skrus av trenger vi to variabler som kan hjelpe oss å angi koordinatene til hvert lys. I menyvalget «Variabler» lager vi derfor to nye variabler som vi kaller «x» og «y».

- For å simulere halveringstid ønsker vi at et tilfeldig antall led-lys skal skrus av hver gang vi trykker på knapp A. I kodeblokka «når knapp A trykkes» fra menyvalget «Inndata» legger vi inn to løkker ved hjelp av blokka «gjenta for...fra...» fra menyvalget «Løkker». Alt som står inne i disse blokkene gjentas så mange ganger vi ønsker, her vil vi at hver løkke skal gjentas 5 ganger (fra og med verdien 0 til og med verdien 4). Ved å gjøre dette vil vi kunne gå gjennom alle koordinatene til led-lysene på skjermen.

- For hver koordinat tenker vi oss at vi kaster en terning, og hvis terningen viser sekser skal led-lyset skrus av. Vi trenger derfor en «hvis sann»-blokk fra menyvalget «Logikk». Inne i denne legger vi inn blokka «slukk x...y» fra menyvalget «Skjerm», og bytter ut de hvite boblene med variabelblokkene «x» og «y».

- Vi må også lage kode for selve «terningkastet», og til dette bruker vi en sammenlikningsblokk fra menyvalget «Logikk» og en «velg tilfeldig...»-blokk fra menyvalget «Matematikk». Den ferdige koden for hva som skal skje når knapp A trykkes ser du til høyre. For hver koordinat på skjermen skal vi velge et tilfeldig tall mellom 1 og 6. Hvis det valgte tallet blir 6 skal led-lyset til denne koordinaten skrus av.

- Til slutt kan vi legge inn en funksjon der alle gjenværende lys slukkes hvis vi trykker på knapp B.

- Gi programmet ditt et fornuftig navn og last det inn på micro:bit. Nå er du klar til å simulere halveringstid.



Når knapp A på micro:bit trykkes inn vil noen av lysene skrus av. Hvis knapp A trykkes inn flere ganger og det registreres mellom hver gang hvor mange lys som fortsatt lyser kan antallet lys plottes som funksjon av antall trykk på knappen og «halveringstiden» for micro:bit kan leses av grafen.

## 6.6 Forslag til videre arbeid

- Undersøk hvordan halveringstiden kan variere fra forsøk til forsøk og undersøk spredningen i forsøkene.
- Hvilken parameter må du endre på for å endre halveringstiden til «grunnstoffet»?
- Endre programmet slik at det «trykker» på knappen en gang i sekundet og automatisk måler tiden det tar til antallet atomer som er blitt omdannet er halvert. Når det skjer skal skjermen vise halveringstiden i sekunder.





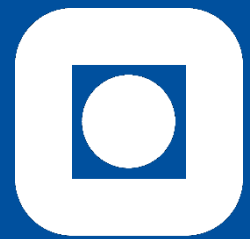






Vi omgir oss med mye programvare i hverdagen, og programmering blir en stadig viktigere del av undervisningen i skolen. Dette heftet gir forslag til seks undervisningsopplegg i naturfag for VGS med aktiviteter med micro:bit knyttet opp mot ulike kompetansemål i faget.

Johanna Sexe  
Realfaglærer ved Orkdal vgs  
E-post: [johanna.sexe@trfk.no](mailto:johanna.sexe@trfk.no)



NTNU

**Skolelaboriet**  
for matematikk, naturfag  
og teknologi  
[www.ntnu.no/skolelab](http://www.ntnu.no/skolelab)