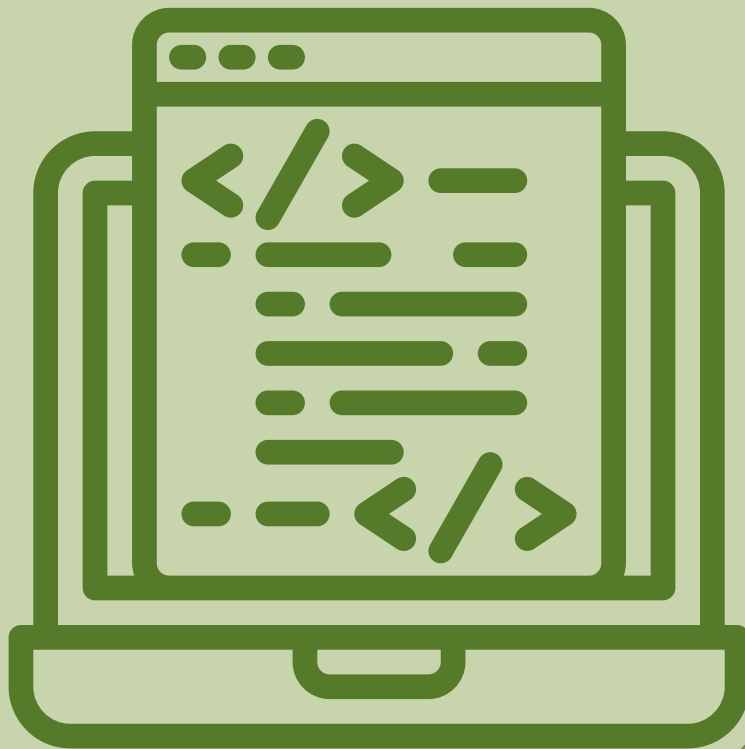


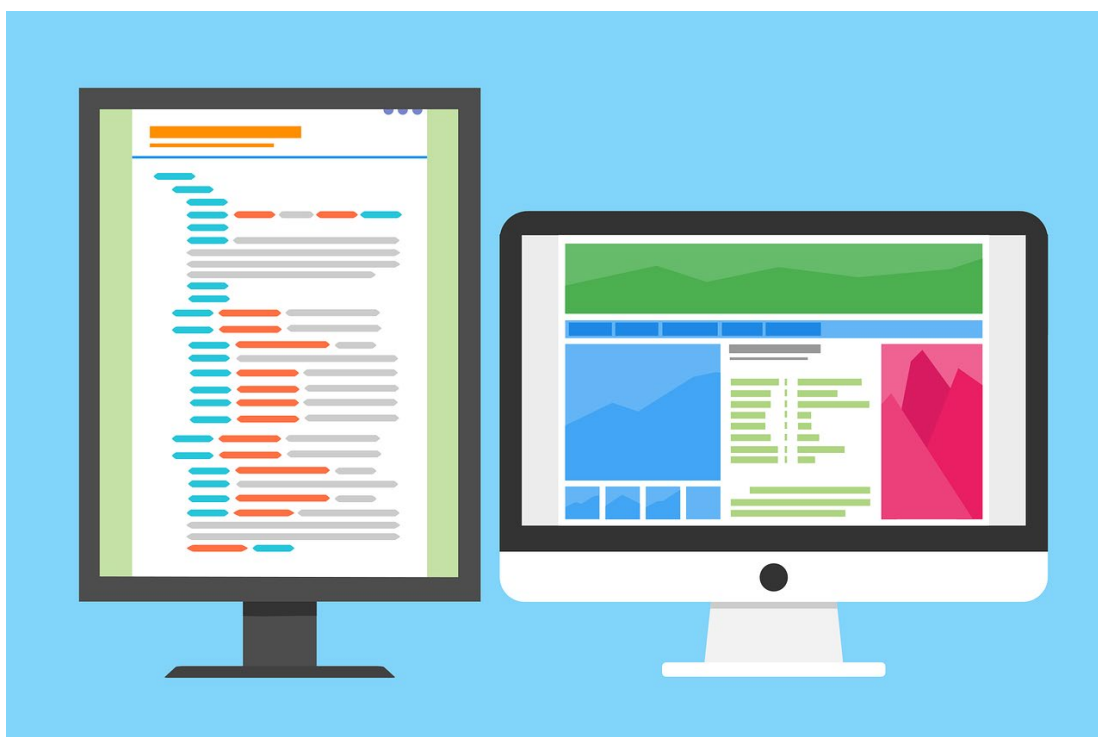
Programmering for elever med nedsatt syn



Programmering for elever med nedsatt syn

Innledning

I fagfornyelsen har programmering kommet inn i flere av fagene i skolen. Matematikkfaget har for eksempel kompetansemål knyttet til koding i alle klassetrinn. Hva betyr dette for elever med synsnedssettelse? Kan de delta i fag som omfatter programmering på lik linje med seende elever? Ofte kan synshemmede programmere som klassen for øvrig, men det krever noen tilpasninger. Enkelte løsninger er derimot utilgjengelige for synshemmede. I disse tilfellene må man finne andre verktøy.



I denne veilederen vil vi ta for oss flere av kodeverktøyene som skolene bruker. Dette omfatter apper eller programmer, webbaserte løsninger, mikrokontrollere og roboter. Vi beskriver hvordan alt dette kan brukes av elever med synsnedssettelse og hvordan det bør tilrettelegges. For enkelte verktøy lenker vi videre til undervisningsopplegg og oppgaver, og også mer detaljert, teknisk informasjon som forklarer hvordan verktøyene kan brukes med skjermleser. Primærmålgruppa for denne ressursen er pedagogisk personale i grunnsopplæringen, men også pedagoger i barnehagen vil kunne ha nytte av denne ressursen.

Begrep i ressursen

Abstraksjon

Når man abstraherer noe skjuler man detaljer som er unødvendige for den oppgaven som skal løses. Et annet ord for dette er å forenkle. Et t-bane-kart er for eksempel en abstraksjon av den virkelige og kompliserte verden. En kalender er en abstraksjon av din tid.

Algoritme

En algoritme er en oppskrift av bestemte instruksjoner som beskriver steg for steg hvordan man kommer frem til løsningen av et problem, eller for å gjennomføre en oppgave. Trinnene i en strikkeoppskrift er en algoritme. I programmering brukes algoritmer til å beskrive løsninger til vanlige oppgaver. Søkemotoren Google bruker søkealgoritmer for å sortere resultatene.

Algoritmisk tenkning

Algoritmisk tenkning er en problemløsningsmetode. Den innebærer at du tilnærmer deg problemer på en systematisk måte, både når du formulerer hva du ønsker å løse og når du foreslår mulige løsninger. Datamaskiner kan ikke tenke algoritmisk, det er det bare mennesker som kan.

Feilsøking (Debugging)

Når man oppdager en feil og retter opp feil (bugs) i programmet/koden, kalles det feilsøking. Du som programmerer møter i hovedsakelig på to typer bugs. Den første er syntakstfeil. En slik bug oppstår når du som programmerer staver feil eller glemmer et semikolon. Den andre type bug er logisk feil, der koden ikke gjør som den skal.

Funksjoner

En funksjon er et sett av instruksjoner som til sammen utfører en oppgave. Noen programmeringsspråk kaller det i stedet en metode. Det er lurt å lage en funksjon om du skal gjenta oppgaven flere ganger underveis i programmet.

Kildekode-editor/editor

En editor er et program som brukes til å skrive tekstbasert program

Løkker

Løkker, eller looper, er kodelinjer som repeteresom og om igjen. Når vi programmerer, så ønsker vi ofte å gjenta identiske instruksjoner i programmet vårt, uten å måtte skrive koden om igjen. Da bruker man løkker. Du kan spesifisere hvor mange ganger du vil løkken skal gjenta seg. Eller kanskje du ønsker en løkke som går i all evighet? Slike løkker kalles «infinite loops».

Microkontroller

Små, kompakte datamaskiner som gjerne er utviklet til et bestemt formål.

Programmeringsspråk

Programmeringsspråk er en betegnelse for et kunstig språk som benyttes for å styre og kontrollere en datamaskin. Selve språket har, akkurat som andre språk, et eget utvalg av ord og regler for "grammatikken" (syntaks). Datamaskinen er ikke i stand til å tolke innholdet slik vi mennesker kan.

Sekvens

En sekvens er en serie beskrivelser som følger hverandre i riktig rekkefølge. I en sekvens må man følge hvert skritt i riktig rekkefølge

Blokkprogrammering og tekstprogrammering

- Kodeverktøyene i skolen kan deles i tre grupper;
 - programmering av roboter
 - blokkprogrammering på skjerm
 - tekstprogrammering

Det kan være utfordrende å vite hvilke verktøy som vil passe til elever med redusert syn, og det er mye å velge mellom. For de aller yngste barna, gjerne fra barnehagen, er det vanligst med programmering av fysiske roboter. Roboter programmeres ved å trykke på knapper på roboten eller ved at brikker settes i en bestemt rekkefølge.

Enkelte roboter kan også programmeres med PC/iPad.

I blokkprogrammering på skjerm lages kode ved å flytte brikker på skjermen med musepeker, såkalt drag-n-drop. I skolen er dette den vanligste formen for programmering, og forlagene har gjerne ressurser på dette. Siden eleven må navigere med musepeker, er ikke blokkprogrammering på skjerm funksjonelt for elever med blindhet. Det resulterer i at gjerne må se etter alternativer for disse elevene, først og fremst

Programmering av robot passer til alle elever med synshemming, så fremt roboten blir merket. Roboten kan enten styres med fysiske knapper på roboten eller fysiske kodebrikker. For elever med blindhet kan det være aktuelt å merke med punktskrift eller annen taktil merking. For elever med moderat synshemming kan merkingen bestå av å fargelegge knapper på roboten slik at det blir gode kontraster. For elever med lite behov for forstørrelse, vil det ikke alltid være nødvendig med tilpasninger. De vil kunne få tilstrekkelig forstørrelse ved å redusere avstanden til roboten.

Tekstprogrammering.

Tekstprogrammering betyr at kode skrives i et programmeringsspråk etter helt bestemte regler. Følges ikke reglene, vil ikke koden fungere. Tekstprogrammering brukes mest fra ungdomsskolen og oppover, men elever med blindhet har ofte for å lære dette tidligere da det er begrenset med alternativer.



```
describe("jQuery plugin", function() {
  it("$.fn.serializeObject is setup", function() {
    assert.strictEqual(typeof $.fn.serializeObject, 'function');
    assert.strictEqual($.fn.serializeObject, $.fn.serializeObject);
  });
  it("$.fn.serializeJSON is setup", function() {
    assert.strictEqual(typeof $.fn.serializeJSON, 'function');
    assert.strictEqual($.fn.serializeJSON, $.fn.serializeJSON);
  });
});
```

Når elever med blindhet programmerer roboter fungerer ofte selve programmeringen med den omtalte taktile merkingen. Det som derimot kan være krevende, er å følge med hva som skjer når roboten kjører. Et råd kan være å legge hånda forsiktig på roboten mens den kjører. Da vil eleven kunne registrere hvor langt roboten kjører fram, og når den snur. Det kan også være en fordel om roboten produserer lyd. Mange vil i tillegg ha nytte av et taktilt kart som de kan bli kjent med på forhånd. Dette kommer vi tilbake til under beskrivelse av [underlag til roboter](#).

Blokkprogrammering på skjerm vil ofte fungere for barn som har lite behov for forstørring. Det vil ofte være gunstig med en stor skjerm (vanlig skjerm i stedet for skjerm på bærbar PC). I tillegg bør du vurdere om det er mulig og hensiktsmessig og forstørret skjermbildet i nettleser (Ctrl + pluss/musehjul) eller ved innstillinger i operativsystemet. Enkelte vil være tjent med høykontrastfarger.



Trenger eleven ytterligere forstørring må forstøringsprogram vurderes; enten innebygde forstøringsprogram i enheten eller eksterne programmer. Dersom forstøringsbehovet er stort, kan det være krevende å navigere i programmet. Derfor bør det alltid testes om den valgte løsningen passer eleven.

Som nevnt tidligere, vil ikke blokkprogrammering være tilgjengelig for elever med blindhet, og følgelig må man se etter alternativet i tekstprogrammering.

Tekstprogrammering vil fungere for alle synshemmede så fremt man får riktig tilrettelegging. Har man lite behov for forstørring, så holder det kanskje å bruke en ekstern skjerm og kort arbeidsavstand til skjermen. Er behovet for forstørring større, må det kanskje suppleres med et forstøringsprogram - enten innebygde forstøringsprogram (PC, Mac, Chromebook, iPad) eller med et eksternt forstøringsprogram på PC som Supernova eller Zoomtext.

For elever som bruker skjermleser, vil tekstprogrammering fungere fint med de programmene som støtter leselist, men det kan selvsagt være krevende å lære seg programmeringsspråket for de yngste elevene. De øvrige elevene lærer som regel blokkprogrammering først, og begynner med tekstprogrammering på ungdomsskolen.

I tabellen under har vi laget en grovmasket oversikt over hvilken type programmering som vil passe til elever med ulik grad av synsnedsettelse.

	Blokkprogrammering av robot	Blokkprogrammering mot skjerm	Tekstprogrammering
Barn med lite forstørreingsbehov	Ok. Eventuell merking av robot og underlag.	Ok. Stor skjerm, skjerminnstilling er.	Ok. Vurder større skjerm og bruk av forstørreingsprogram i operativsystem.
Barn med stort forstørreingsbehov	Ok. Merking av robot og underlag.	Ok, men ikke mulig for alle.	Ok. Større skjerm og forstørreingsprogram.
Punktskriftbrukere	Ok når det brukes fysiske brikker som er taktilt tilpasset eller programmerer direkte på roboten. Underlag må merkes. Kart av underlag må vurderes.	Ikke tilgjengelig. Tekstprogrammering eller kodebrikker kan være alternativer.	Ok, men undersøk først om programmet støtter leselist.

Tilpasninger - moderat synshemmede

Man snakker gjerne om 3 synspedagogiske prinsipper; forstørrelse, forsterking og forenkling. Har man nedsatt skarpsyn eller kontrastsyn hjelper det å forstørre det som står på skjermen. For enkelte kan også forsterkning av farger være gunstig, f.eks. høykontraste farger. Forenkling av skjermbilder er også nyttige - komplekse skjermbilder er gjerne mer krevende å forholde seg til hvis man har et redusert syn.

Forstørrelse

Forstørrelse kan man oppnå ved å redusere avstand til skjermen, større skjerm og innstillinger på enheten. Vi anbefaler følgende fremgangsmåte:

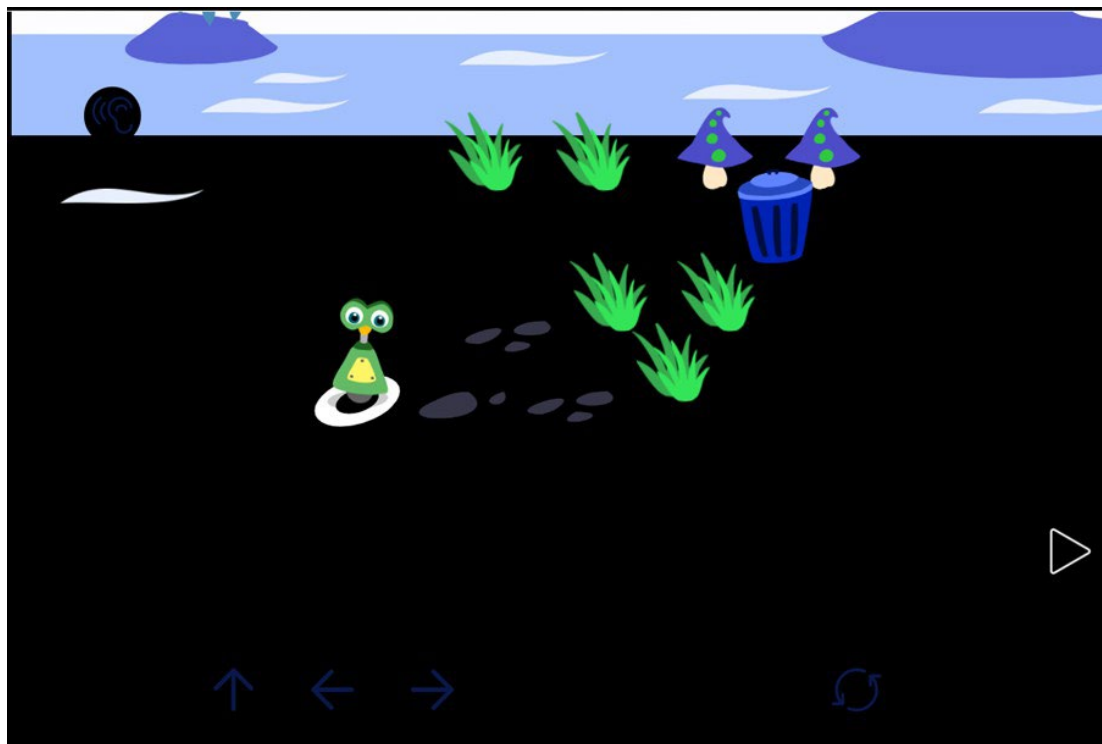
- Se etter innstillinger for skriftstørrelse i koderessursen.
- Forstørre ytterligere med standard nettleseforstørrelse, ctrl + pluss eller ctrl + musehjul.
- Forstørreingsprogram på enhet, for eksempel Win + pluss på PC.

Forsterking

Forsterking av skjermbildet kan man få ved å bruke høykontraste farger som ligger tilgjengelig i både PC, Mac, Chromebook og iPad. Imidlertid bør man teste ut dette grundig i den konkrete ressursen og med eleven. I verste fall kan man erfare at ressurser blir umulig å bruke med høykontraste farger. Eksemplet under er fra Gyldendals skolestudio, Multi, 2. trinn. Her skal eleven velge knapper i en rekkefølge slik at figuren Robi kommer til søppelbøtta.



Når vi velger høykontrast i Windows ser vi i bildet under at både knapperaden og brettet hvor knappene skal plasseres i blir usynlig. Akkurat dette valget vil ikke fungere, men det kan være at andre fargekombinasjoner vil fungere. I andre koderessurser som Salaby fra samme forlag, erfarer vi derimot det fungerer svært godt med de samme høykontraste valgene i Windows.



Forenkling

Forenkling er det tredje synspedagogiske prinsippet som kan være nyttig på skjerm. I noen koderessurser kan man forenkle skjermbilder ved å fjerne menyer. Er man i en nettleser på PC eller Chromebook kan man alltid fjerne menylinjen med F11. skjermbilder inneholder så mye informasjon at det blir svært krevende å bruke ressursen, bør man derfor se etter alternative ressurser.

Analog programmering (uten datamaskin)

Hvordan virker det?

Ofte kan det være en fordel om elevenes første møte med programmering er helt uten datamaskiner og roboter. Da kan du jobbe med de grunnleggende prinsippene for hvordan kommunikasjon med en datamaskin foregår. Disse ligger til grunn for programmeringsspråk, og kan utforskes gjennom aktiviteter som ikke trenger å involvere teknologi.

Elevene kan lære å sette sammen en serie av instruksjoner til en algoritme som løser en gitt oppgave. Da kan de bruke grunnleggende prinsipper i programmering, slik som kommando, symbol, løkke, variabel, funksjon, sekvens og kode. Det kan være lurt å gi elevene lapper med de ulike instruksjonene de kan lage programmer av. På den måten er det enklere å få oversikt over det de programmerer.

Tilrettelegging for elever med synshemming

For elever med nedsatt syn er det viktig at språket brukes presist, akkurat slik det også er i programmering. Det er mange begreper som elevene må kjenne til i grunnleggende programmering, og mange av dem handler om egen kroppsbevissthet. Begreper som høyre og venstre må beherskes, og hva er det egentlig å snu seg 90 grader?

Når man bruker lapper for å symbolisere programmet, er det viktig å tilpasse disse til elevens synsfunksjon. For punktskriftbrukere kan man bruke [Statpeds kodebrikker](#).

Forslag til undervisningsopplegg

Programmer hverandre!

La elevene jobbe sammen to og to. Den ene får i oppgave å lage en rute den andre skal gå. De bestemmer hva målet for ruta er.

- Hvordan må eleven stå ved start?
- Hvor mange skritt må eleven gå fremover?
- Når må eleven snu?
- Hvor mange grader må eleven snu seg?
- Må eleven over noen hindre?

Elevene bytter på å programmere hverandre, og vanskelighetsgraden kan øke etter hvert. Skriftliggjør gjerne koden.

Tips: Hvis elevene må følge instruksene i blinde stiller det økte krav til at koden er presis. Instruksjoner som for eksempel "gå ut på gangen" blir da for upresise.

Lag en dans!

Å lage en danserutine ligner svært mye på det å lage en algoritme til en datamaskin. Vi planlegge hva vi skal gjøre, når vi skal gjøre det, og hvor fort det skal gå, og hvor mange ganger vi skal gjenta det.

- Hvordan skal dansetrinnet se ut?
- Skal vi bruke både bein og arm samtidig?

- Skal vi snu oss rundt? Hvor mange grader for hvert steg?
- Skal vi gjøre det samme flere ganger etter hverandre? (løkke)

La elevene jobbe sammen to og to. Hver gruppe kan prøve sin dansekode på en litt større gruppe.

Tips og ideer

- [Uten datamaskin](#) | [Kodeklubben](#) (kidsakoder.no)
- [CS Fundamentals Unplugged](#) | Code.org
- [Bokserien om Ruby](#) har både fengende historier og oppgaver som kan brukes som introduksjon. Ikke alle er tilpasset elever med synssvekkelse og man må derfor påregne tid til tilrettelegging.
- Perkins har laget et opplegg der koding introduseres ut ifra boken "vi skal på bjørnejakt". Her ligger det også [ferdige maler](#) som kan skrives ut visuelt, på svellpapir eller i punktskrift.

Mobilitetsopplæring

Analog programmering henger tett sammen med mobilitetsopplæring. En mobilitetsrute kan være programmeringsinstrukser. "Gå til stokken treffer veggen. Snu 90 grader til høyre. Gå til stokken treffer veggen. osv." Elevene kan gjerne lage ruter til hverandre som de skal gjøre med blindfold for å ha et inkluderende opplegg, og programmering kan sees i sammenheng med å lære eleven å finne fram på skolen på egen hånd.

Progresjon i opplæringsforløpet

Det er i hovedsak fire hovedmåter å programmere på.

Her er et eksempel på en typisk progresjonsplan for opplæringsløpet, og det er vanligvis stigende vanskelighetsgrad på disse:

1. **Analog programmering**
2. **Blokkprogrammering med robot, med eller uten fysiske kodebrikker**
3. **Blokkprogrammering på skjerm**
4. **Tekstprogrammering**

Den største utfordringen for punktskriftbrukere er at steg 3 blokkprogrammering er en visuell måte å presentere tekstprogrammeringen på, og er laget nettopp for å være til visuell hjelp. Blokkprogrammering fungerer ikke med lese-list, og kan være utfordrende for andre synshemmede. Derfor mangler dette steget i progresjonen for elever som ser for dårlig til å nyttiggjøre seg av det. Når det gjelder steg 4 er det ansett som det vanskeligste for normaleleven, men for en punktskriftbruker er det den programmeringsformen der gapet til de seende er minst.

Dersom en elev har for dårlig syn til å beherske blokkprogrammering på skjerm, men evner å bruke tekstprogrammering, bør man vurdere om det er hensiktsmessig å bruke Statpeds kodebrikker eller om man bør la eleven bruke tekstprogrammering og lære det ordentlig. Statpeds kodebrikker kan være til hjelp som alternativ til steg 3 på veien til tekstprogrammering, men er ikke et mål i seg selv.

[Code Jumper](#) er utviklet som et alternativ til blokkprogrammering.

Statpeds kodebrikker



Hvordan virker de?

Statpeds kodebrikker består av et brett med langsgående kanter og med markerte kanter som avgrenser brettet øverst og mot venstre. Selve kodebrikkene er utformet med spor på undersiden som passer inn i kantene på brettet. Både brett og brikker produseres på en vanlig 3D-printer.

Kodebrikker kan bestilles av Statped gjennom din kontaktperson. Du kan også produsere kodebrikker selv på egen 3D-printer. Hør med din kontaktperson for mer informasjon om dette. Du bestemmer selv hva som skal stå på brikkene.

Kodebrikkene gjør ingenting av seg selv, men fungerer som et støtteverktøy for forskjellige typer programmering. Det egner seg godt til både fysisk programmering, programmering av roboter og tekstprogrammering. Kodebrikkene viser hvordan programmering handler om å bruke ulike instruksjoner i en bestemt rekkefølge.

Kodebrikkene vil danne kontinuitet gjennom hele skoleløpet.



Brikkene har tekst i punktskrift og varierer i lengde avhengig av lengden på teksten på de enkelte

brikkene. Kantene på brettet brukes til å holde brikkene på plass, og er også til hjelp når enkelte brikker skal ha innrykk fra venstre kant. Brikkene kan produseres i både 6- og 8-punkt. 6-punkt er ofte tilstrekkelig, men 8-punkt er en stor fordel hvis du skal bruke brikker med programkode (for eksempel Python).

Det er også mulig å lage blanke brikker uten punktskrift. Da kan du i stedet lime på egen punktskrift med dymotape.

Tilrettelegging for elever med synshemming

Hensikten med kodebrikkene er å lette overgangen til tekstprogrammering for elever med alvorlig synssvekkelse som leser visuelt og for elever som bruker punktskrift.

Disse elevgruppene vil ikke klare å bruke visuelle, blokkbaserte løsninger som for eksempel Scratch, Code Studio og MakeCode. Mange av dem har allerede erfaring med fysisk programmering (programmering med kroppen) og med diverse roboter som Blue-Bot, Cubetto og MatataLab. Flere av disse robotene programmeres ved hjelp av klosser eller brikker der hver kloss representerer en instruksjon. Statpeds kodebrikker bygger videre på dette.

Video: [bluebot kodebrikker](#)

En stor fordel med kodebrikkene er fleksibiliteten. Selv om de opprinnelig ble laget for å lette overgangen til tekstprogrammering, er det en fordel å innføre brikkene før eleven skal i gang med dette. Da kan brikkene brukes i forbindelse med for eksempel roboter og ha instruksjoner som tilsvarer instruksjonene på roboten. Eleven vil da venne seg til å bruke brikkene tidlig.

Eksempel 1: Kodebrikker med Blue-Bot og underlag



Roboten Blue-Bot er utbredt i skolen og er et godt egnet verktøy for elever med synsnedsettelse.

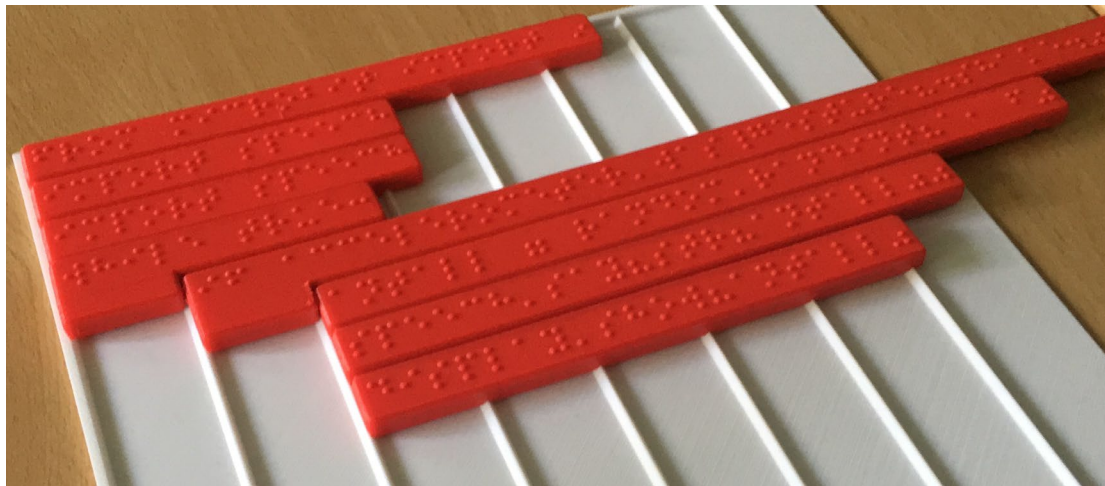
Bruk brikker med instruksjoner som tilsvarer instruksjonene på Blue-Bot (fremover, bakover, høyre, venstre). Dette vil da altså være teksten i punktskrift på brikkene. Her er det tilstrekkelig med 6-punkt.

En ulempe med Blue-Bot, er at man kun får én sjanse til å programmere den. Gjør man feil, må man begynne helt på nytt. Da kan det være bedre å la eleven lage programmet sitt med kodebrikkene først. Eleven vil da til enhver tid ha full oversikt over alle instruksjonene på brettet. Og det vil være enkelt å gjøre endringer. Når eleven er tilfreds med programmet på brettet, kan hun kopiere det til Blue-Bot.

Hvis Blue-Bot brukes sammen med et underlag med rutemønster, kan en oppgave være at roboten skal programmeres til å kjøre fra én rute til en annen. Når eleven har laget programmet for dette med kodebrikkene, kan eleven lese programmet fra brettet med en hånd og simulere hva som vil skje på underlaget med den andre. Hvis resultatet blir riktig, kan eleven programmere Blue-Bot og deretter kjøre den samme ruta med roboten.

Med kodebrikkene kan eleven få en bedre forståelse for at et program består av flere instruksjoner som kommer i en helt bestemt rekkefølge.

Eksempel 2: Kodebrikker med Python og micro:bit



Bruk brikker med instruksjoner som tilsvarer instruksjoner i et lite Python-program for micro:bit. Her er det en fordel om brikkene er i 8-punkt.

Når kodebrikkene skal brukes til Python-kode, er det viktig at brikkene ikke bare har tekst som tilsvarer de enkelte Python-instruksjonene. Det er like viktig at brikkene kan vise innrykkene som brukes i koden. I Python er innrykk en veldig viktig del av syntaksen. Kodebrikker med instruksjoner som krever at de neste kodebrikkene skal ha innrykk, har et hakk i underkant. Når Python krever innrykk, flyttes den aktuelle kodebrikken ett hakk mot høyre på brettet.

På denne måten kan elever som bruker kodebrikkene øve på å lage programmer i Python der de ikke bare tar hensyn til selve Python-instruksjonene, men også øver på å bruke riktig syntaks. Når de senere skal programmere Python i en kodeeditor, kan de bygge videre på dette.

Forslag til undervisningsopplegg

Introduksjon av kodebrikker



Tidsbruk

30 minutter

Alder

Alle som begynner med koding bør teste analog koding først. Dette opplegget kan brukes fra 1.klasse.

Hva trenger du?

Kodebrikker og brett til analog koding, for eksempel: hoppe, klappe, venstre, høyre, fremover, bakover. Brikkene man bruker avhenger av alder og hvor krevende oppgaver man ønsker å lage.

Visuelle kodebrikker med de samme kommandoene

Mål

Se sammenhengen mellom instruksjon gitt ved kodebrikker og bevegelser som roboten/elevene utfører.

Hva gjør du?

Introduksjon: Datamaskiner er gode til å oppfatte det vi sier. Men de er veldig til å tenke selv. Hva skjer for eksempel om du setter en robot til å gå utenfor et bord? Vil den stoppe når den ser faren, slik som oss mennesker?

For at blant annet roboter skal fungere må vi være de smarte og programmere godt! I dag skal vi late som vi er roboter og programmerere, og styre hverandre. Vi skal øve på være presise og gi gode instruksjoner. Som roboter har vi bare lov å gjøre det programmereren sier.

1. **Lag en kode på kodebrettet i fellesskap. Skriv den både på tavlen og lag den på kodebrettet i punktskrift. La en av elevene være robot og "gjøre koden".**
2. **Del klassen inn i grupper på minst to og to. La dem lage koder til hverandre uten noen kriterier.**

3. Utvidelse av opplegget:

1. **Lag flere brikker:** du kan variere antall grader (vend 90 grader, 180 grader etc.), lengde på skritt eller ha brikker med flere kommandoer.
2. **Kodene kan ha mål de skal nå:**
 1. **Lag en kode til klasseromsdøra**
 2. **Lag en kode til en skatt som er gjemt ute**
 3. **Lag en kode til en medelev**

Oppsummering

Snakk om:

- hva lærte vi om viktigheten av å være presis?
- hvordan en robot og programmerer jobber?
- kan vi dele noen tips og triks om hvordan det er lurt å lage en kode?



Introduksjon av robot

Statped har testet ulike roboter til bruk i programmering. Disse finner du mer informasjon om her ([lenke](#)). Det finnes også mange andre på markedet som kan fungere. Opplegget under bygger på koderoboten Blu-Bot, men kan enkelt endres til andre roboter.

Tidsbruk

90 minutter

Alder

Alle som begynner med koding bør teste analog koding først. Dette opplegget kan brukes fra 1.klasse.

Hva trenger du?

- Kodebrikker og brett som samsvarer med det roboten utfører: venstre, høyre, fremover, bakover.

- Blue-Bot
- Taktile plansjer/kart.
- Målebånd med taktile kjennetegn.

Mål

Se sammenhengen mellom instruksjon gitt ved kodebrikker og bevegelser so roboten/elevene utfører.

Hva gjør du?

- Beskriv roboten:**
 1. Hvordan er knappene plassert?
 2. Hva slags lyder lager roboten og hva betyr de?
 3. Navngi knappene på Bluebot med ordene som tilsvarer kodebrikkene i punkt: fremover, bakover, venstre og høyre. Du gjør dette ved å holde knappen inne til det kommer en lyd.
- Lag en kode på kodebrettet i fellesskap.** Skriv den både på tavlen og lag den på kodebrettet i punktskrift. La Blue-Bot kjøre koden på gulvet. Eleven som har synsnedsettelse kan kjenne på Blue-Boten mens den kjører.
- Del klassen inn i grupper på minst to og to.** La dem lage samme kode på roboten som dere laget felles. Snakk om hvorfor det er viktig å lage koden ferdig først og at man må nullstille koden etterpå. Elevene kan så jobbe med å bli kjent med roboten. Bruk kodebrikkene ved siden av, slik at koden alltid legges i sin helhet på kodebrettet og med visuelle brikker før roboten programmeres og kjøres:
 1. Hvor langt fremover går Bluebot når du trykker en gang? Bruk et målebånd med taktile/tydelige kjennetegn.
 2. Hvor mange ganger må du trykke på venstre eller høyre for å snu helt rundt?
 3. Lag en kode med tre instruksjoner.
 4. Hvor mange ganger må du trykke for å komme til veggen nærmest deg?
 5. Lag en kode slik at roboten går i en firkant.
- Ta frem og beskriv plansjene.** La eleven som har synsnedsettelse bli kjent med kartet ved å kjenne på det håndholdte kartet.
- Gi elevene oppgaver til de taktile kartene:**
 1. La de gå rundt på hele kartet, i ytterkant, på tvers osv.
 2. Gi konkrete oppgaver der de skal komme til en spesifikk rute på kartet
 3. Gi dem mulighet til å lage oppgaver selv.

Oppsummering

Snakk om:

- Hva lærte du om bruk av roboter?
- Har du noen tips å dele til de andre i klassen?
- Hvordan gir de taktile/visuelle kodebrikkene oss oversikt over hele koden?
- Hvorfor er det viktig å lage hele koden før vi lar roboten bevege seg?

Blue-Bot og Bee-Bot - robot



Hvordan virker den?

Bee-Bot og Blue-Bot er frittstående roboter som kan programmeres ved hjelp av taktile knapper. Bee-Bot og Blue-Bot har ulike farger, men den viktigste forskjellen er at Blue-Bot er utstyrt med Bluetooth-tilkobling. Du trykker på knappene i den rekkefølgen du ønsker for å lage en sekvens med instruksjoner (forover, bakover, høyre, venstre). Siden Blue-Bot har støtte for Bluetooth, kan den også programmeres ved hjelp av en app eller tilbehøret Tactile Reader.

Tactile Reader gir en litt annen tilnærming der du kan legge fysiske brikker i 10 separate spor. Det finnes ulike brikker som har de samme instruksjonene som du finner på roboten (forover, bakover, høyre og venstre). Inntil tre Tactile Reader kan seriekobles for å lage lengre kodesekvenser. Det finnes dessuten pakker med ekstra brikker, såkalte Extension Tiles. Disse pakkene inneholder brikker med andre instruksjoner.

En annen type tilbehør er underlag (matter) som er utstyrt med rutemønster der rutene kan inneholde tekst eller bilder. Det finnes flere typer i salg, men du kan lage like gode underlag selv. Les mer om dette under delen om [underlag til roboter](#).

Underlagene brukes til å lage utfordrende og spennende oppgaver. På [tts- international.com](https://tts-international.com) kan du lese mer om utstyr og tilbehør.

Tilrettelegging for elever med synshemming

Knappene som brukes til å programmere roboten har ulike fargekoder, og er oversiktlige å bruke for elever med nedsatt syn. De syv knappene ligger plassert slik at elevene kan forstå logikken uten at de nødvendigvis forstår den taktile merkingen.

Så Blue-Bot kan du spille inn lyder for hver knapp. Roboten gir da lydtilbakemelding både når du koder den og når den forflytter seg/utfører koden. Da kan roboten for eksempel si «framover» når den flytter seg forover eller «bakover» når den flytter seg bakover. Lyd gir en ekstra dimensjon til opplevelsen for elever med synsnedsettelse. Å slå enheten og lyden på og av er enkelt ved hjelp av to skyvebrytere på undersiden.

Når roboten er i bevegelse, kan punktskriftbrukere legge en hånd på den, og eventuelt en hånd på et referansepunkt. Da kan hun følge roboten underveis selv om hun ikke ser den. Med utfordrende oppgaver kan Bee-Bot og Blue-Bot brukes av elever med synsnedsettelse helt opp

til ungdomsskolen som en introduksjon til koding.

Tactile Reader



Tactile Reader egner seg godt for elever med synsnedsettelse så fremt den tilpasses noe. Du kan forbedre merkingen ved å klistre på lapper med større symboler og bedre kontrast. Det er også mulig å merke brikkene med punktskrift.



Tactile Reader har to knapper som brukes til henholdsvis Bluetooth-pairing og overføring av kode til BlueBot. Disse krever ingen spesiell merking, men du må vite hvilken knapp som gjør hva.



Blue-bot-appen

Blue-Bot-appen kan brukes sammen med Blue-Bot, men den kan også brukes uten å koble den til roboten. Appen har enkel tilpasning for farger på bakgrunn og knapper. Mulighetene er svært begrensede og gjelder bare for Windows.

Appen fungerer greit ved relativt lav forstørring, men man mister fort oversikten hvis man har behov for mye forstørring. Du kan aktivere Windows forstørrelsesprogram i Windows eller Zoom på iPad. Du kan også prøve Windows fargefiltre hvis det er behov for å endre kontraster i appen. Men tilpasning av fargene i appen kan være

utfordrende siden appen bruker mange ulike fargekombinasjoner. Windows høykontrast fungerer ikke. Det er viktig at tilpasninger prøves ut med den enkelte elev.

Når du har laget en kodesekvens i appen, kan du få Blue-Bot til å utføre koden. Da kan du følge roboten på skjermen og se at roboten beveger seg der. Samtidig kan den fysiske Blue-Bot-roboten utføre den samme koden. Å følge roboten på skjermen er vanskelig for elever med synsnedsettelse.

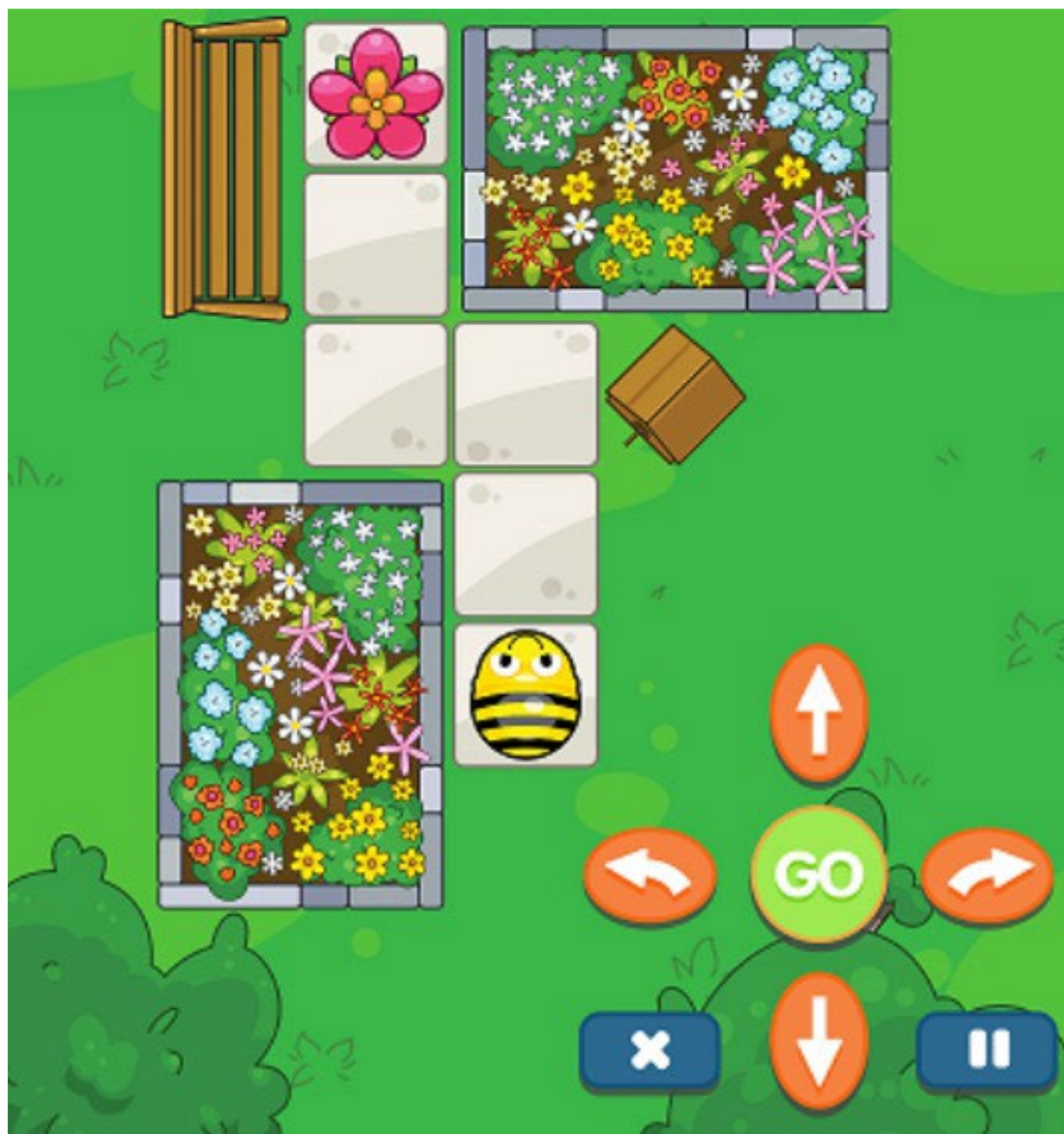
Appen er ikke tilgjengelig for elever som bruker skjermleser.

Blue's blocks app

Blue's blocks er en app for Blue-Bot hvor man skal sette sammen blokker for å styre Blue-Boten. Denne er hakket mer avansert enn blue-bot appen. Appen har egen funksjon for å forstørre rutenettet og blokkene man har satt i rutenettet. Man får ikke forstørret blokkene man skal velge ut ifra, som befinner seg på venstre side. For å forstørre disse må man aktivere forstørrelse på smarttelefon eller nettbrettet. Det vil imidlertid fort gjøre at man mister oversikten.



Bee-Bot-appen



Hensikten med Bee-Bot appen er å lære seg funksjonene til Bee-Boten. Her kan man styre en digital Bee-Bot ut ifra et styringspanel med tilsvarende knapper som Bee- Boten har på ryggen. Det er mulig å forstørre med forstøringsprogrammet på smarttelefon eller nettbrett, men forstørring gjør at man fort mister oversikt. Det kan derfor være en fordel å bruke selve Bee-Boten i stedet for appen for barn som er sterkt svaksynte. Appen er ikke tilgjengelig for blinde.

Tips og ideer

- Spill inn lyd til knappene på Blue-Bot-en. Husk å slå på høyttaleren.
- Ved bruk av Tactile Reader: snakk om at koder vanligvis skrives i leseretningen under hverandre. Bruk gjerne Statpeds kodebrikker som supplement.
- Vurder behovet for Windows Forstørrelsesprogram og/eller Fargefiltre i Windows (**Win+U**) i samarbeid med elev.
- Vurder behovet for Zoom kombinert med Zoomfiltre i iOS (Innstillinger > Tilgjengelighet > Zoom) i samarbeid med elev.

- [Tilpasninger i Windows for personer med nedsatt syn.](#)
- [Tilpasningsmuligheter på iPad for elever med nedsatt syn.](#)
- Elever med alvorlig synssvekkelse som leser visuelt og elever som bruker punktskrift vil ikke klare å bruke appen. Bruk i stedet roboten frittstående eller med Tactile Reader.

Forslag til undervisningsopplegg

Oppgaver med Blue-Bot

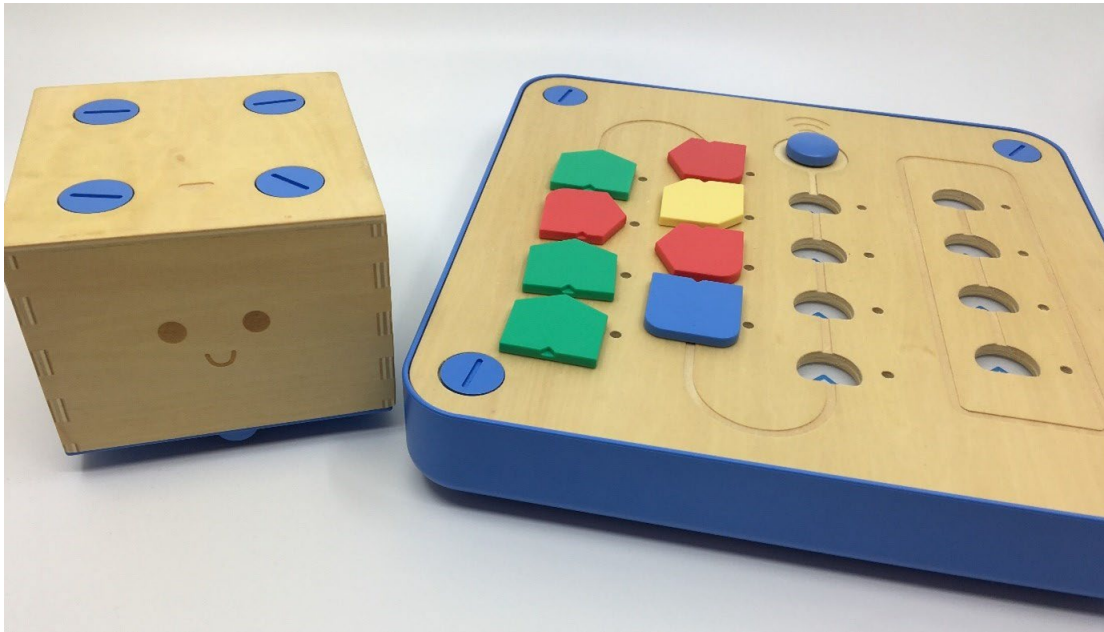
1. Hvor langt fremover går Blue-Bot når du trykker en gang? Bruk et målebånd.
2. Hvor mange ganger må du trykke på venstre eller høyre for å snu helt rundt?
3. Lag en kode med tre instruksjoner.
4. Hvor mange ganger må du trykke for å komme til veggen nærmest deg?
5. Lag en kode slik at roboten går i en firkant. Kan du lage kode både for kvadrat og rektangel?
6. Finn deg et underlag og lag en kode der roboten kjører innom alle rutene.

Oppgaver til et underlag med dyr

Forberedelser: Lage et underlag med bilder av en bondegård, norske kjæledyr, husdyr og rovdyr.

1. Du er på bondegården, lag en kode for å gå til et dyr som begynner på bokstaven K.
2. Hva sier hunden? Lag en kode for å gå til hunden.
3. Hvilket dyr får vi ull fra? Lag en kode for å komme deg til dette dyret.
4. Du skal ut på tur. Lag kode for turen. Hvilke dyr ser du på turen?
5. Lag en kode for å gå til et av rovdyrene. Lag deretter en kode for å gå til et byttedyr.
6. Lag en kode for å gå til katten. I koden skal du også legge inn en løkke. Fortell litt om katten når du kommer dit.
7. Lag en rute til et dyr du selv bestemmer. Hvorfor valgte du dette dyret? Hvilke dyr passerer du for å komme dit?

Cubetto - robot

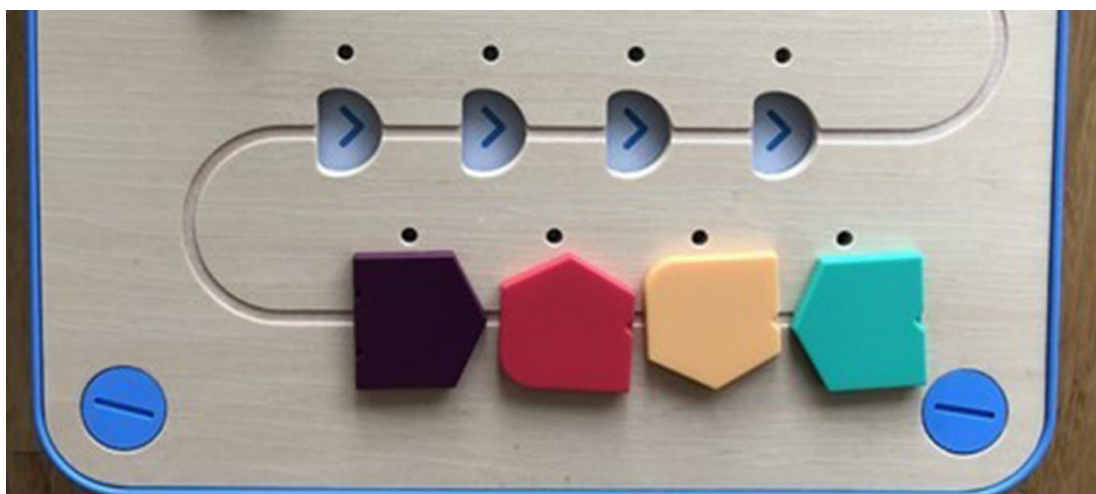


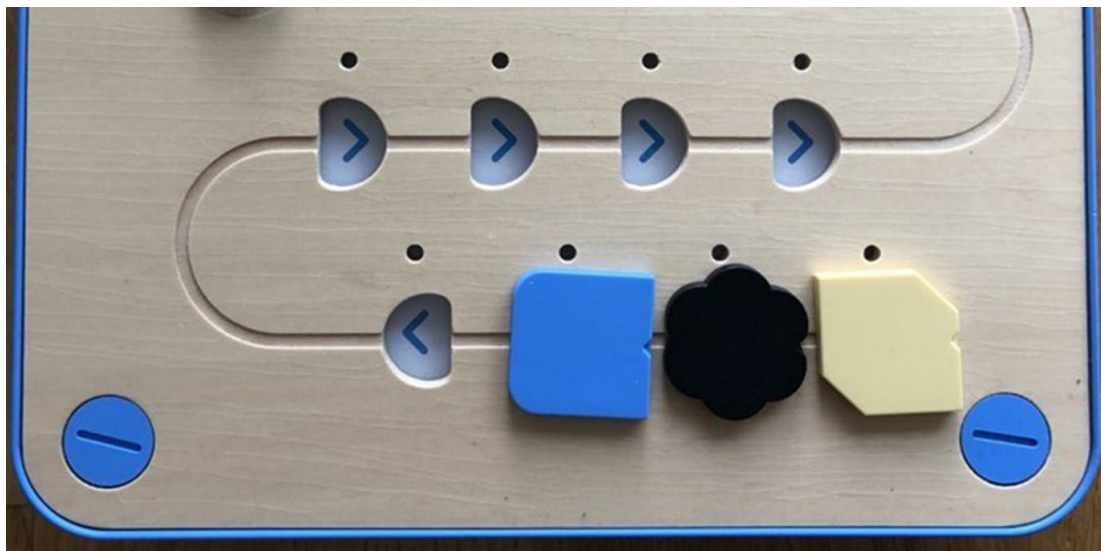
Hvordan virker den?

Cubetto er en kubeformet robot i tre som kodes ved hjelp av et fysisk kodebrett og kodebrikker.

Cubetto er svært stillegående og beveger seg 15 cm av gangen. Den har et taktilt ansikt i retningen den beveger seg, i tillegg til en taktil pil på toppen av roboten. Roboten går på batteri med god kapasitet. Den skrur av og på med skyveknapper som er lett tilgjengelig, på både brett og robot. Når koden er lagt, startes roboten ved hjelp av en stor knapp midt på brettet.

Det fysiske kodebrettet kobles til roboten med Bluetooth. Det er enkelt å legge brikkene ned i brettet, og de kan kun legges på en måte. Hver brikke har et hakk i bakkant, som symboliserer hvordan de skal legges på brettet. I tillegg er det, på alle retningsbrikkene, en pil den veien roboten vil gå. Unntaket her er bakover, som er symbolisert med to hakk i bakkant.





Nettside: <http://www.primotoys.com/>.

Tilrettelegging for elever med synshemming

For elever med nedsatt syn vil det være nyttig at roboten markeres med god kontrast. Et eksempel er å feste farget papp på toppen av roboten, i farge og form som tilsvarer de fysiske kodebrikkene:



Merket for fremover kan også få taktil forsterkning i form av at det lages en pil med flytende lim som blir gjennomsiktig (eks. skolelim) på toppen. Når det er satt på papp vil denne nemlig fjerne den taktile pilen som er i roboten her. Ved å gjøre denne markeringen vil det være tydeligere hvilken vei som er frem. Man kan også velge å bruke svart tusj for å forsterke øyne og munn med god kontrast.

Cubetto er testet på elever med synsnedsettelse fra 4 år og til 6.klasse. Den kan, på lik linje med

Blue-Bot og Bee-Bot, fungere som en god introduksjon til koding.

Det følger med et underlag i stoff. For elever som har nedsatt syn anbefaler vi heller å lage egne matter med god kontrast og taktile kjennetegn. Se nærmere beskrivelse under delen om [underlag til roboter](#).

Tips og ideer

Cubettos kode er laget som en slange. Det gjør progresjonen fra dette verktøyet og over på mer avanserte kodeverktøy som tekstkoding osv. litt mindre intuitiv. En bør derfor snakke med eleven om at koden vanligvis skrives i leseretning og under hverandre, eventuelt bruke Statpeds fysiske kodebrikker som et supplement til verktøyet.

Forslag til undervisningsopplegg



Oppgaver til underlag med tall

Forberedelse: Lag et underlag med tall fra 1 til 12. Ha to terninger klar.

1. **Kast to terninger, summer tallet. Lag en kode for å gå til dette tallet.**
2. **Tenk på et tall under 100. Finn de to tallene man må gange sammen for å få dette tallet. Lag en kode for å gå til disse to tallene.**
3. **Lag et regnestykke som inneholder tre siffer som til sammen blir 18 når du legger dem sammen. Lag kode for en rute der du går innom disse tre tallene.**
4. **Lag tre regneoppgaver for Cubetto, der ingen tall (heller ikke summen) blir høyere enn det største tallet på underlaget. Lag koden for å gå innom tallene i gitt rekkefølge.**
5. **Lag regnestykker for hverandre.**

MatataLab - robot

Hvordan virker den?

MatataLab er en koderobot som kan programmeres på mange måter. Kodesettet (grunnsettet) består av et kontrolltårn, en robot, kodebrikker, et landskapsbrett og diverse landskapsfigurer. Brikkene legges på et brett foran et kontrolltårn. I tårnet er det et kamera som leser av koden og sender den trådløst til roboten.



Kodebrikkene har taktile merker som angir hvilken vei de skal ligge, og passer bare i denne posisjonen på kodebrettet. Kodebrikker som har ulike verdier, har også et hjul med taktile kjennetegn som er enkelt å dreie på. Koderoboten har et ansikt slik at du kan se hvilken vei som er fremover for den. Øynene har lys, og kan kjønnnes taktilt.



Det følger med mye ekstrautstyr til Matatalab. Den kan kodes med notebrikker, slik at man kan lage musikk. Man kan også bruke en sensor som gjør at Matatalab kan kodes på flere måter. Den kan da styres som Cubetto og Blue-Bot ved fysiske knapper oppå selve roboten. Knappene er taktile og enkle i bruk.



Man kan også kjøpe et taktilt kart som tilbehør. Kartet er vist på bildet over. Dette er enkelt å sette sammen, og man kan sette inn historier og bilder som følger med, eller lage egne. Kartet gir en god oversikt for elever med synssvekkelse og gir mulighet for både god kontrast og taktile kjennetegn.

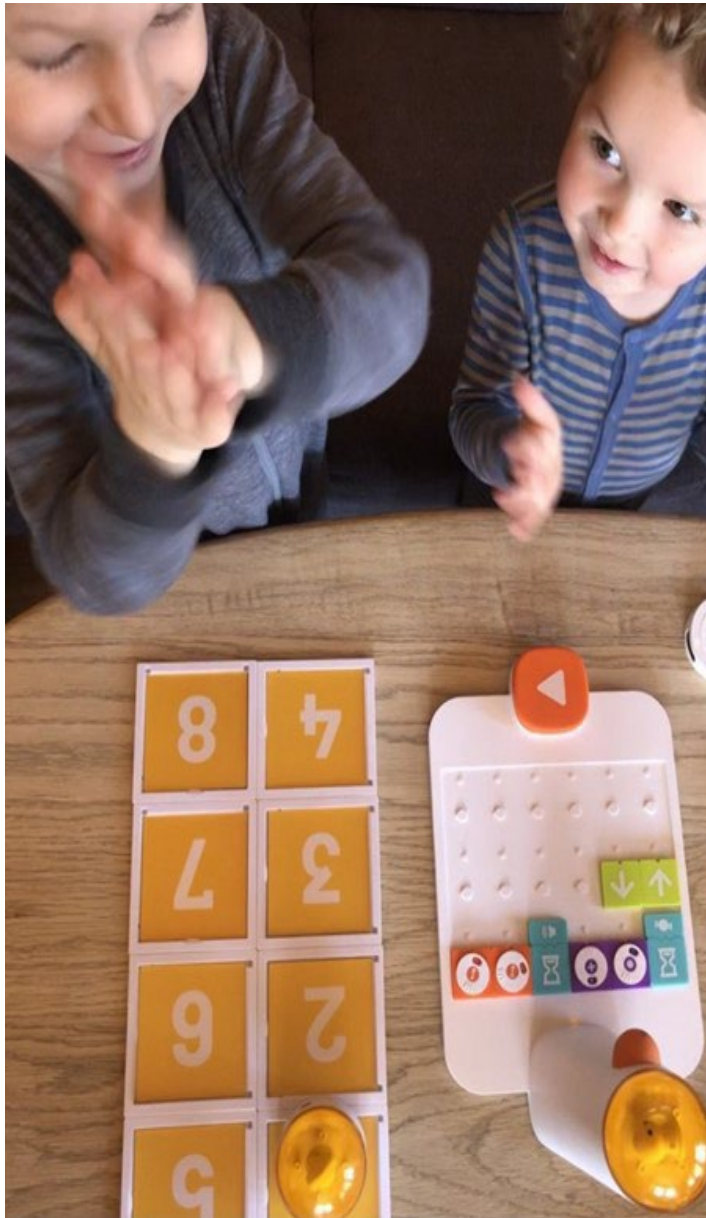
Det finnes en egen app for Matalab-roboten, som heter MatataCode. Her kan man sette sammen blokker for å styre Matatab-roboten. Det er mulig å forstørre med forstørrelse på smarttelefon og nettbrett. Dette gjør at man fort mister oversikten, den fungerer derfor dårlig for barn/elever med en synssvekkelse.

Tilrettelegging for elever med synshemming

Mange av kodebrikkene har dårlig kontrast for elever med nedsatt syn. Statped forsøkt å

tilrettelegge brikkene på ulike måter. På ulike brikker har vi tegnet på de hvite pilene med sort tusj, markert dem taktilt med lim, og merket dem med

punktskrift. Dette har gjort brikkene enklere å bruke for elever med nedsatt syn, men vi har også opplevd at kontrolltårnet kan få vansker med å lese av koden. Vi tror det kommer av at vi har endret utseende på brikkene, og det da kan bli gjensinn i det taktile limet og punktmerkingen. Det kan være mulig å punktmerke brikkene på undersiden.

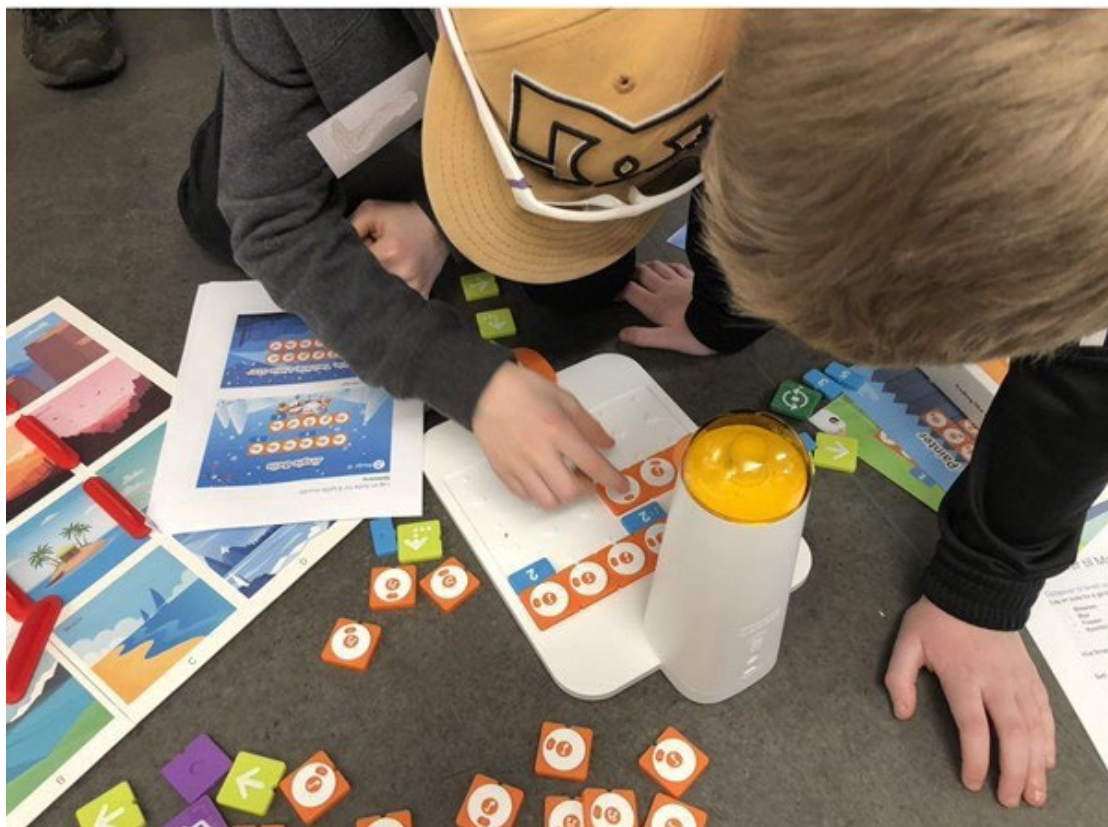


Batteritiden på Matata-roboten har opplevdes som noe kort, men holder fint til en undervisningsøkt.

Sensoren kan styres med taktile knapper. Det er ikke satt en fast avstand for hvor mye Matata-roboten beveger seg når man holder knappene inne. Dette er en stor ulempe for elever med synssvekkelse. Når man lager et program med sensoren går den 10 cm, og dette er derfor kun et problem i begynneropplæringen når man styrer roboten med sensoren direkte.

Musikkbrikkene fungerer godt som de er, for elever med synssvekkelse. Som vist i undervisningsopplegget under, kan disse brukes for enkel innlæring av punktskriftnoter.

Sensorens ekstrabrikker som gir mulighet for lyd- og bevegelsestilbakemelding, er enkle å bruke for elever med synssvekkelse.



Tips og ideer

Matatalab har mye annet ekstrautstyr som ikke er tatt med her, da det ikke er så relevant for denne elevgruppa.

Forslag til undervisningsopplegg

Hvordan bruke musikk med Matatalab?



- 1. Musikkdelen er fullt tilgjengelig for blinde og de med nedsatt syn. Musikkdelen har**

musikkbrikker med et dreibart hjul fra 1-7 som viser C-dur-skalaen i enstrøken og tostrøken oktav. Se tabell. Ved å følge ulike oppskrifter eller sanger (se [Noen enkle sanger.pdf](#)) kan elevene lage gitte sanger. Man kan også komponere sine egne sanger.

2. Man kan også blande inn punktnotasjon der hvert tall representerer en note. Punktnotene kan være fjerdedel, halvnote, punktert halvnote og helnote (se [Instruks punkt.pdf](#)). Her kan man også skrive sanger med punktnoter (se [Gubba Noah og Fola fola blakken.pdf](#)). Man kan velge om man eventuelt vil markere brikkene ytterligere.
3. Det finnes også rytmebrikker som spiller ulike sanger og rytmer. Her kunne man tilrettelagt ved bl.a. å markere brikkene med punktnoter, eller nummerert de og hatt et ark med punktnoter ved siden av.

Posisjon på brikkehjul	Note
1	C
2	D
3	E
4	F
5	G
6	A
7	H (B)

Underlag til roboter

Hvordan virker de?

Underlag med rutemønster er veldig godt egnet til bruk sammen med roboter som Bee-Bot, Blue-Bot og Cubetto. Rutene kan inneholde tekst eller bilder. Underlagene gir deg mulighet til å lage spennende oppgaver der elevene skal programmere roboten til å forflytte seg i rutemønsteret. Det finnes underlag å få kjøpt, men du kan lage like gode underlag selv. Underlagene vi beskriver her, kan brukes med alle roboter som har skrittlengde lik 15 cm. Det vil si at når roboten beveger seg ett skritt, vil den flytte seg fra én rute til den neste.





Tilrettelegging for elever med synsnedsettelse

Når du bruker underlag eller matter, kan det være vanskelig for elever med synsnedsettelse å forstå sammenhengen. En typisk oppgave kan være at roboten skal programmeres til å flytte seg fra et sted til et annet på underlaget. Da må eleven planlegge en vei som de deretter må programmere roboten til å utføre. Tilrettelegging er derfor avgjørende.



Du lager ditt eget underlag slik: Bruk en forholdsvis stor papplate, eller en del av en voksdug. En voksdug vil gjøre det enklere å frakte/pakke bort underlaget, men pass på at limet/tapen er sterk nok til å ikke løsne når den rulles sammen.

Lag et rutemønster (15 x 15 cm.) og fyll rutene med tall, bokstaver, tekst, bilder eller lignende. Fyll gjerne rutene helt slik at innholdet blir så stort som mulig. Sørg også for at innholdet har god kontrast. For at elever med alvorlig synssvekkelse som leser visuelt eller elever som bruker punktskrift skal forstå ruteinndelingen, går det an å legge en tynn strek med lim på grensen mellom rutene. Da kan elevene kjenne hvordan rutemønsteret er inndelt. Pass på at ikke streken bli så tykk at roboten ikke klarer å bevege seg fra en rute til en annen. Merk hver rute i punktskrift hvis det er behov for det. Du bør også vurdere å lage et kart i mindre format med punktskrift som gir oversikt over underlaget.



Husk:

- Bruk underlag med ruter som er 15 x 15 cm.
- Sørg for at innholdet i hver rute er stort nok og med god kontrast.
- Merk rutenettet taktilt (for eksempel med en strek med lim, eller tape med tråd under).
- Merk rutene i punktskrift.
- Lag et taktilt oversiktskart over matta.

micro:bit - mikrokontroller



Hvordan virker den?

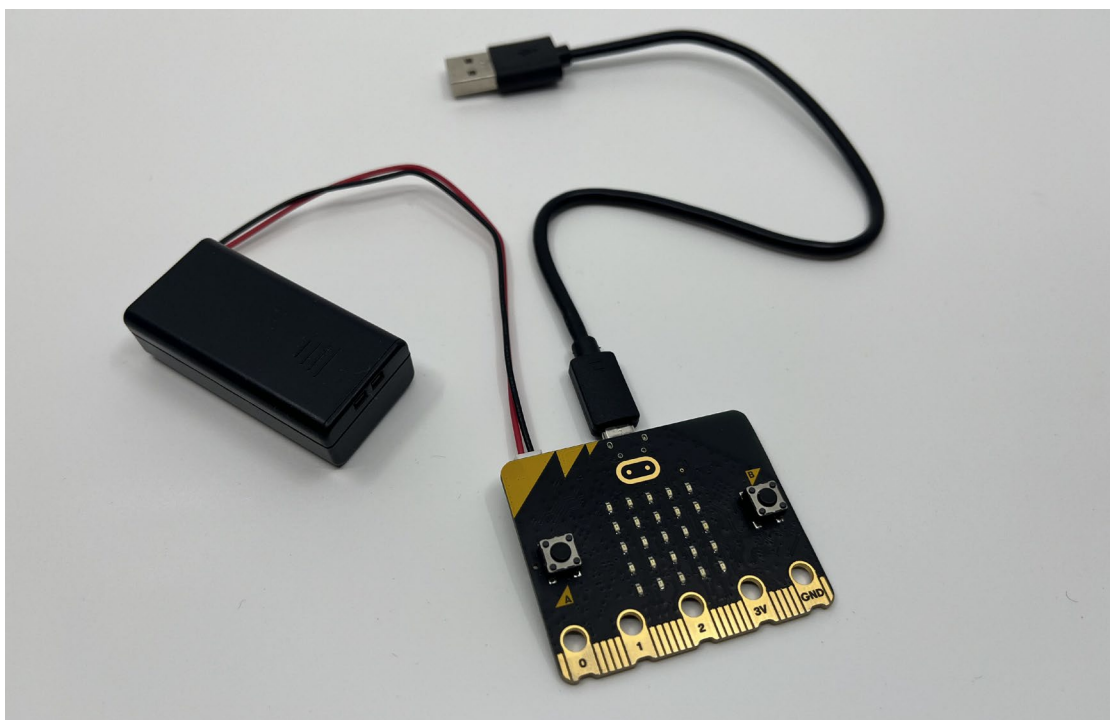
micro:bit er en mikrokontroller som du kan programmere ved hjelp av datamaskin, mobiltelefon eller nettbrett. Det er utviklet spesielle apper og webbaserte løsninger for å programmere micro:bit.

Du kan godt bruke micro:bit alene, men det går også an å koble til forskjellig slags tilbehør. micro:bit er svært utbredt i skolen, både på mellomtrinnet, i ungdomsskolen og på videregående.

Det finnes i hovedsak to løsninger for å programmere micro:bit. Den vanligste er Microsoft MakeCode som både kan kjøres i nettleseren og som en app. Den andre er Python Editor som bare kan kjøres i nettleseren. I tillegg finnes en rekke andre løsninger.

Nettside: <https://www.microbit.org/no/>

Tilrettelegging for elever med blindhet, svaksynthet og hjernerelaterte synsvansker

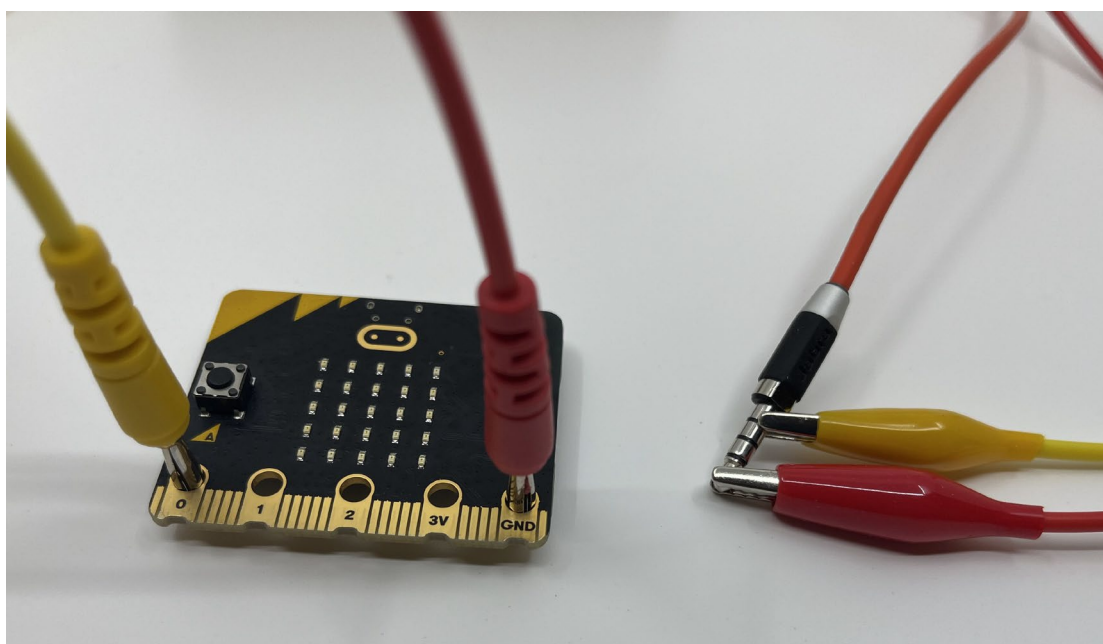


microbit startpakke med USB-ledning og batteri.

micro:bit kan fint brukes av elever med blindhet, svaksynthet og hjernerelaterte synsvansker. Men husk at den må brukes til noe som gir mening for denne elevgruppen. Skjermen (LED-matrisen) på micro:bit kan være vanskelig å se for mange av dem. I stedet kan det være bedre å bruke micro:bit med høyttaler. Da kan du lage programmer som skaper musikk, lydeffekter og tekst-til-tale.

Selve micro:bit-enheten er enkel å bruke og krever ingen tilrettelegging. Den har to knapper på forsiden (A og B) som er taktile og lette å finne. Det samme gjelder for reset-knappen på baksiden. På den nye micro:bit (versjon 2) er det mulig å bruke micro:bit-logoen på forsiden som en knapp. Den er ikke taktil og kan dermed være litt vanskeligere å finne.

Elever med synsnedsettelse vil klare å koble til USB-ledningen, batteriene og forskjellig tilbehør. Det kan være utfordrende å koble til høyttalere. Langs den ene siden av micro:bit ligger en rekke med tilkoblingspunkter. Hvis du skal bruke høyttalere, må du koble ledninger til punktene merket henholdsvis 0 og GND. Du kan bruke ledninger med krokodilleklemmer eller bananplugg. Tilkoblingen må gjøres slik at ledningene ikke kommer i kontakt med tilkoblingspunktene ved siden av. Det enkleste og sikreste er å bruke bananplugger som du kan stikke inn i hullene.



Tilkobling av høyttalere til microbit med bananplugg og krokodilleklemmer.

Det kan være utfordrende å koble utstyr til micro:bit. Hvis du skal bruke høyttalere eller hodetelefoner, må du koble ledninger til punktene merket henholdsvis 0 og GND på micro:bit-enheten. Du kan bruke ledninger med krokodilleklemmer eller bananplugg. Tilkoblingen må gjøres slik at ledningene ikke kommer i kontakt med tilkoblingspunktene ved siden av. Det enkleste og sikreste er å bruke bananplugg som stikkes inn i hullene på micro:bit.

micro:bit med Musicbox.



I stedet for å koble til høyttalere ved hjelp av ledninger, går det an å bruke et tilbehør som kalles [4tronix Music Box](#). Det er en enhet med høyttaler. micro:bit versjon 2 har også en egen høyttaler bygget inn. Men verken høyttaleren på Music Box eller micro:bit er av særlig god kvalitet. De fungerer ok til musikk og lydeffekter, men blir for dårlige til tekst-til-tale.

Husk:

- Unngå å bare bruke skjermen. Bruk musikk, lydeffekter og syntetisk tale.
- Tilkobling av høyttalere gjøres enklest med bananpluggen.
- Bruk alternativt 4tronix Music Box eller micro:bit V2.

Programmere micro:bit

De to viktigste appene som brukes til å programmere micro:bit kalles MakeCode og Python Editor. MakeCode har mulighet for både blokkbasert og tekstbasert programmering. Python Editor kan derimot bare brukes til tekstbasert programmering. Det finnes også en rekke andre løsninger.

Video: [MakeCode: Tilpasninger for elever med synsnedsettelse](#)

Blokkeditoren i MakeCode kan brukes av enkelte elever med nedsatt syn, men det er viktig at dette prøves ut med eleven først. MakeCode har mulighet for forstørrelse i deler av programvinduet, men funksjonen er utilstrekkelig. Det er bedre å bruke forstørrelsesmulighetene i operativsystemet. Hvis eleven trenger mye forstørrelse, blir det utfordrende å holde oversikten. Det kan være vanskelig å oppfatte hva som skjer andre steder i programvinduet, for eksempel hvis du får melding om at det er feil i programmet ditt. MakeCode har også en høykontrastfunksjon, men den er heller ikke imponerende. Det kan være bedre å bruke kontrastmulighetene som finnes i operativsystemet.

Video: [micro:bit Python Editor: Tilpasninger for elever med synsnedsettelse](#)

Elever med nedsatt syn som ikke klarer å bruke MakeCodes blokkeditor, anbefales i stedet å programmere micro:bit med en tekstbasert editor. Selv om MakeCode også har en teksteditor, anbefaler vi heller å bruke Python Editor eller Visual Studio Code. Men det er alltid viktig å prøve

ut løsningen med eleven først. Python Editor har utilstrekkelige muligheter for forstørring og mangler kontrastinnstillinger helt. Visual Studio Code er derimot langt bedre på både forstørring, kontrast og forenkling.

Visual Studio Code må installeres på elevens datamaskin, mens MakeCode og Python Editor kan kjøres i nettleseren.

Verken MakeCodes tekstbaserte editor eller Python Editor fungerer godt nok med skjermleser. For elever som er blinde, anbefaler vi derfor å programmere micro:bit med Python i Visual Studio Code. Se [Statped's "Komme i gang-veileder" for programmering av micro:bit med Visual Studio Code](#).

Husk:

- Enkelte elever med nedsatt syn vil klare å bruke MakeCodes blokkeditor, men det må prøves ut først.
- Elever med nedsatt syn som ikke klarer å bruke MakeCodes blokkeditor, anbefales heller å bruke Python Editor. Må prøves ut med eleven først. Visual Studio Code er et alternativ.
- Vurder om det er behov for å bruke MakeCode og Python Editor med Windows Forstørrelsesprogram, Kontrasttemaer eller Fargefiltre (Win+U).
- Vurder om det er behov for å bruke MakeCode og Python Editor med Zoom kombinert med Zoomfiltre i iOS (Innstillinger > Tilgjengelighet > Zoom).
- Elever som bruker skjermleser anbefales å bruke Visual Studio Code med NVDA, JAWS eller VoiceOver.

Forslag til undervisningsopplegg

- [Tilrettelagte oppgaver for micro:bit med Python](#)
- [super:bit-oppgaget: Tilrettelagte oppgaver for micro:bit med Python i Visual Studio Code for Windows](#)

Tips og ideer

- [Komme i gang med micro:bit og Python i Visual Studio Code for Windows](#)
- [Visual Studio Code for elever med blindhet, svaksynthet og hjernerelaterte synsvansker](#)
- [Tilpasningsmuligheter i Windows for elever med nedsatt syn](#)
- [Tilpasningsmuligheter på iPad for elever med nedsatt syn](#)

super:bit - kodesett



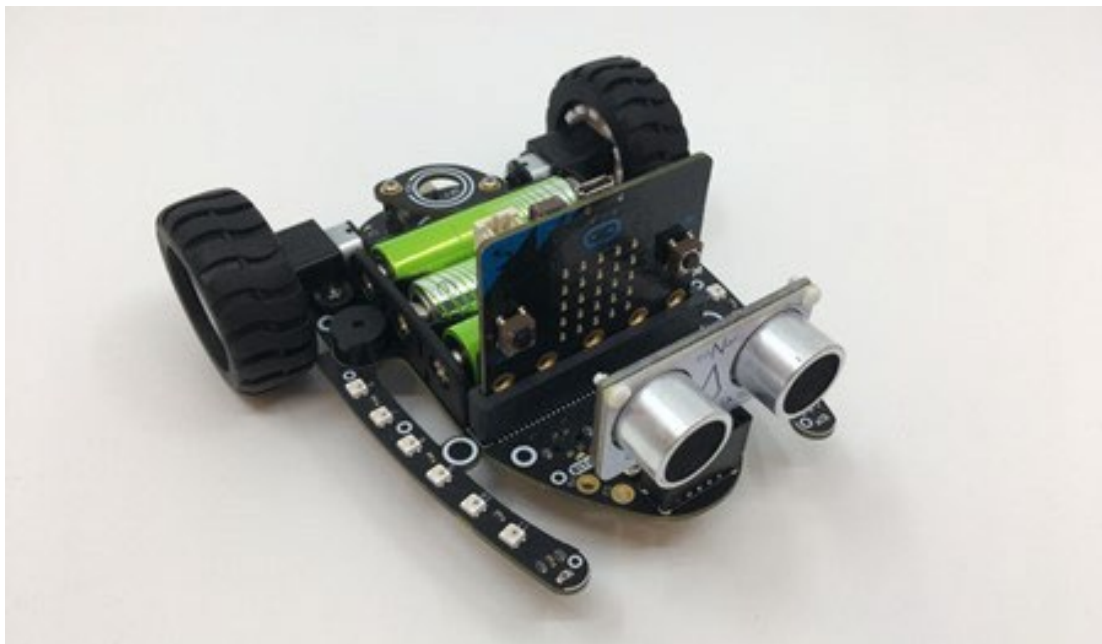
Hvordan virker det?

super:bit er et kodesett som distribueres til skolene for 6. klasse i forbindelse med kurs fra Vitensentrene. Settene inneholder micro:bit (lenke til micro:bit), roboten Bit:bot XL, super:bit-matta og diverse annet tilbehør. super:bit-matta er stor (234 x 114 cm) og har et påtrykt bymiljø med bygninger, parker, veier, m.m. Noen av oppgavene i super:bit-prosjektet går ut på at elevene skal programmere roboten til å kjøre i dette bymiljøet. Matta har flere veier eller kjørebaner; to svingete baner og to meterlange, rette baner ytterst på hver side.

Nettsider: superbit.no, shop.4tronix.co.uk, pspressreklame.no.

Tilrettelegging for elever med synshemming

Roboten Bit:bot XL



Roboten er enkel å betjene. Den har en taktil av/på-bryter på baksiden og et spor der du setter inn micro:bit-enheten. Hvis avstandssensoren (ultralydsensoren) skal brukes, settes den inn i sporet helt foran på roboten.

Når roboten kjører, kan den være utfordrende å følge underveis, spesielt hvis den skal kjøre rundt på super:bit-matta. Selv om du bruker lav hastighet, kan det likevel være vanskelig å følge den. Men husk at for lav hastighet kan gjøre at roboten kjører ujevnt eller ikke i det hele tatt, så du må gjøre en avveining. Roboten har en liten høyttaler (buzzer) som kan programmeres til å lage lyd. Den kan gi elevene en rikere opplevelse når roboten kjører, for eksempel ved at den lager et pipesignal ved start og stopp, eller underveis når den endrer retning.

NB! Husk at mens selve micro:bit-kortet er ganske robust, kan vi dessverre ikke si det samme om Bit:Bot XL:

- Kjør roboten på gulvet, ikke på bordet. Den tåler ikke å falle ned.
- Hvis micro:bit står i roboten mens du overfører et program, bør roboten slås av først.
- Ikke vri på hjulene, det kan skade de to elektromotorene.

super:bit-matta

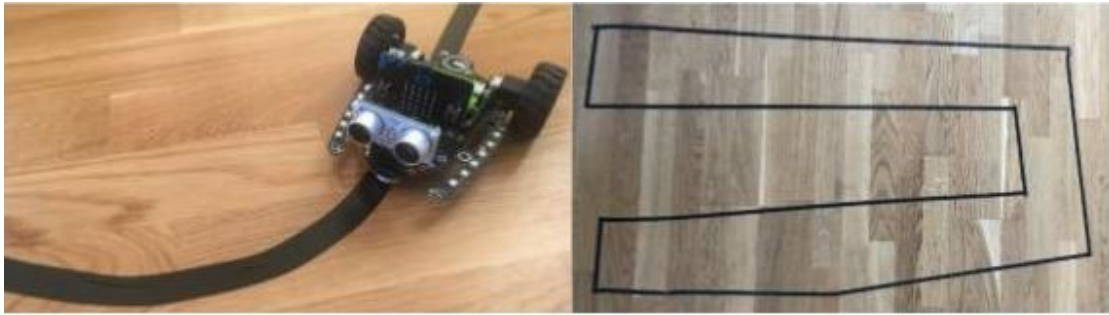
For mange elever med synsnedsettelse, vil ikke super:bit-matta gi noe mening hvis den ikke tilrettelegges. Du kan lage konkrete bygninger med bokser av for eksempel tre eller papp, fotgjengerfeltene kan merkes med tape som har litt struktur, start- og stoppunkt for kjørebane kan merkes taktilt og gjerne også i punktskrift. Husk at

bygningene du lager må være festet til underlaget da de lett flytter seg når en kjenner på dem. Kjørebane kan merkes med to streker med lim slik at det går an å føle hvor veien går. Disse strekene må danne ytterkantene av kjørebane, slik at de ikke forstyrrer roboten. De svingete kjørebane (start til mål) har ganske god kontrast.

Men de meterlange kjørebane Hva er meterkjørebane. ytterst på hver side av matta, har dårlig kontrast som bør forbedres med kontrastfarger. Her også kan man merke ytterkantene av kjørebane med limstreker.

Du bør også lage et kart (en kopi av super:bit-matta i mindre format) som elevene kan bruke for å få oversikt over matta. Det kan for eksempel lages i A4-format. Kartet må være taktilt med

punktskrift, med de viktigste elementene i byen merket med lim, og eventuelt andre kjennetegn. Det er også mulig å lage en 3D-modell av kartet.



Hvis super:bit-matta blir for vanskelig å bruke, er det et alternativ å kjøre roboten fritt på gulvet. Da kan du lage et mer avgrenset miljø der elever med synsnedsettelse lettere kan ha oversikt. Bruk den sorte elektriker-tapen som følger med klassesettet til dette. Linjesensorene på roboten kan programmeres til å følge tapen eller til å kjøre mellom to parallelle tapestreker.

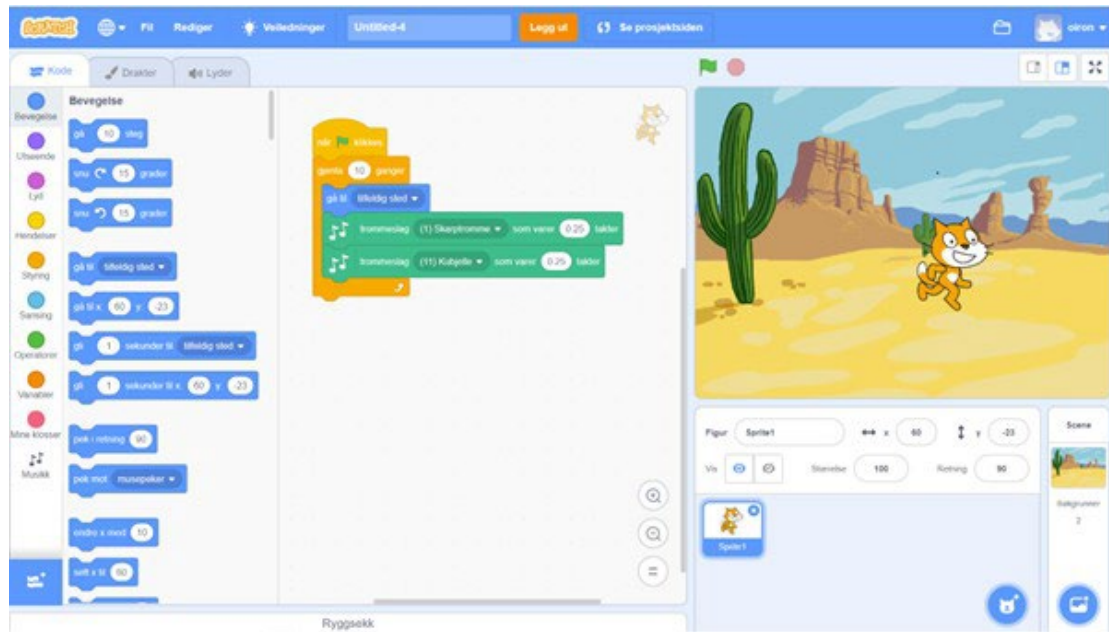
Husk:

- Ikke bruk for høy (eller for lav) hastighet på Bit:bot XL-roboten. Prøv deg fram med utgangspunkt i for eksempel 30% hastighet.
- Bruk gjerne lyd på roboten.
- Tilrettelegg super:bit-matta.
- Lag et taktilt oversiktskart over matta. Vurder å lage en 3D-modell.
- Vurder å kjøre roboten i et mer avgrenset miljø utenfor matta.

Forslag til undervisningsopplegg

- [super:bit-oppdraget. Tilrettelagte oppgaver for micro:bit og Bit:bot \(Python\)](#)

Scratch web-basert



Hvordan virker det?

Scratch er et blokkbasert programmeringsspråk. Det kan brukes til å utvikle animasjoner, spill og historier som du deretter kan dele med andre.

Scratch-appen fungerer på den måten at du drar blokker fra en meny over til et kodeområde. Hver blokk representerer en spesiell instruksjon, og på den måten bygger du opp en programkode.

Du kan også bruke Scratch til å programmere micro:bit, Lego WeDo og Lego Mindstorms, og til å bruke musikk og tekst-til-tale. Scratch er tilgjengelig både som skrivebordsapp og som nettside.

Nettside: scratch.mit.edu.

Tilrettelegging for elever med synshemming

Scratch er utfordrende å bruke for elever med synsnedsettelse. Spesielt gjelder det hvis du skal lage animasjoner eller spill, der det vil foregå mye bevegelse i skjermbildet. Det er dessuten vanskelig å slippe blokkene på riktig sted, særlig når blokk-elementene ligger inne i hverandre. Det er kun elever med moderat synssvekkelse som leser visuelt som vil kunne ha mulighet til å mestre Scratch.

I appen kan blokkene i koden forstørres ved hjelp av forstørrelsesknappene nederst til høyre i kodeområdet. Det er kun det som ligger i kodeområdet som blir forstørret. For å forstørre hele programvinduet, kan du aktivere Windows forstørrelsesprogram og elevene behov for å justere farger eller kontrast, kan du prøve Windows fargefiltre. Windows Høykontrast har ingen virkning på skrivebordsappen, men kan brukes med nettsiden.

På iPadOS vil du på samme måte kunne bruke zoom for forstørrelse, eventuelt kombinert med zoomfiltre for fargeinvertering og kontrastforsterkning. Eventuelle tilpasninger må prøves ut med den enkelte elev på forhånd.

Når du skal kjøre programmet du har laget, kan det lønne seg å vise scenen i fullskjerm, slik at

figuren blir større. Scenen kan også forstørres i nettleseren.

MERK: Scratch er ikke egnet for elever med alvorlig synssvekkelse som leser visuelt eller for elever som bruker punktskrift.

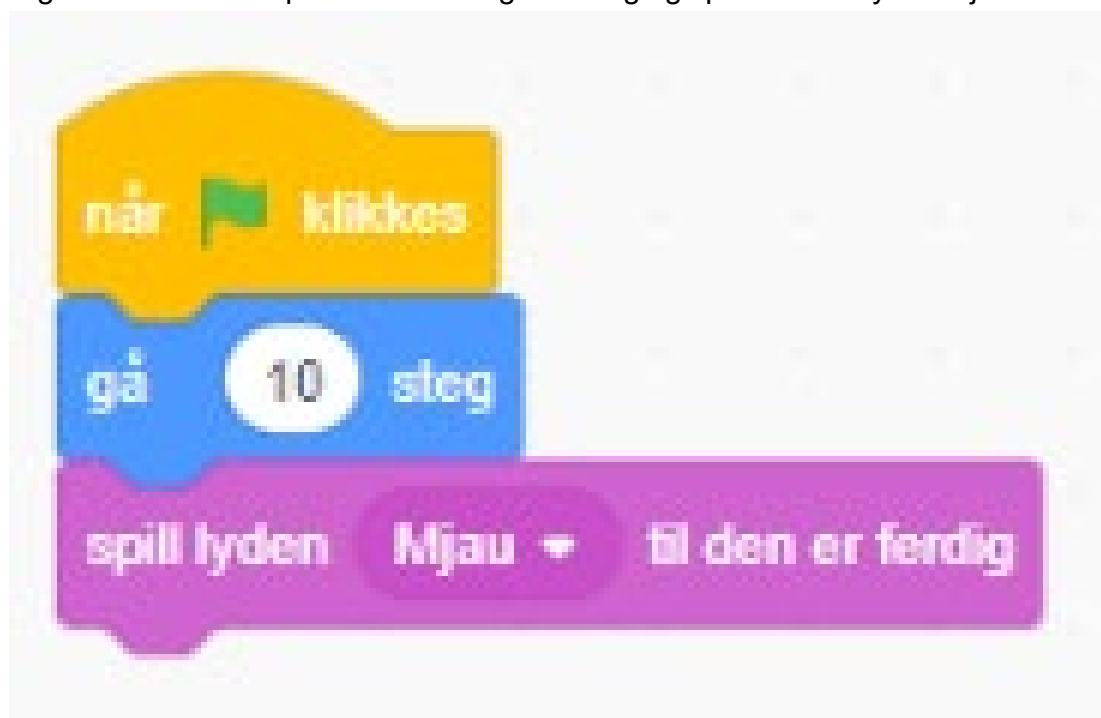
Tips og ideer

- Elever med moderat synssvekkelse som leser visuelt kan klare å bruke Scratch, men det må først prøves ut i samarbeid med eleven.
- Vurder behovet for Windows forstørrelsesprogram og/eller fargefiltre i Windows (Win+U) i samarbeid med elev.
- Vurder behovet for zoom kombinert med zoomfiltre i iOS (Innstillinger > Tilgjengelighet > Zoom) i samarbeid med elev.
- Kjør animasjoner og lignede i fullskjerm. Scratch kan ikke brukes for elever med alvorlig synssvekkelse som leser visuelt eller for elever som bruker punktskrift.
- [Tilpasningsmuligheter i Windows for elever med nedsatt syn.](#)
- [Tilpasningsmuligheter på iPad for elever med nedsatt syn.](#)

Forslag til undervisningsopplegg

Lag en kode som gjør at katten Scratch går 10 steg fremover, og avslutter med en lyd. Koden kan se slik ut:

Figur 1: Kodeeksempel fra Scratch: gå 10 steg og spill deretter lyden Mjau.

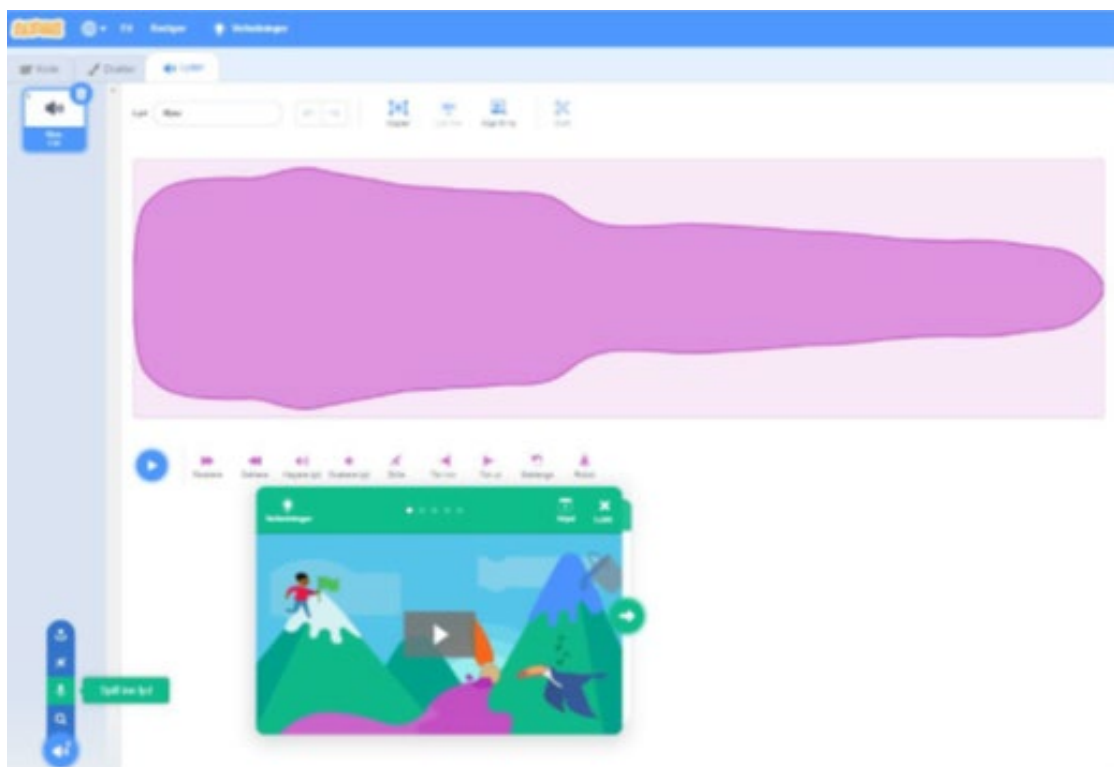


Utvid med en løkke som gjør at Scratch gjentar dette 4 ganger. Koden kan se slik ut:



Figur 2: Kodeeksempel fra Scratch: gjenta 4 ganger: gå 10 steg og spill deretter lyd Mjau. e

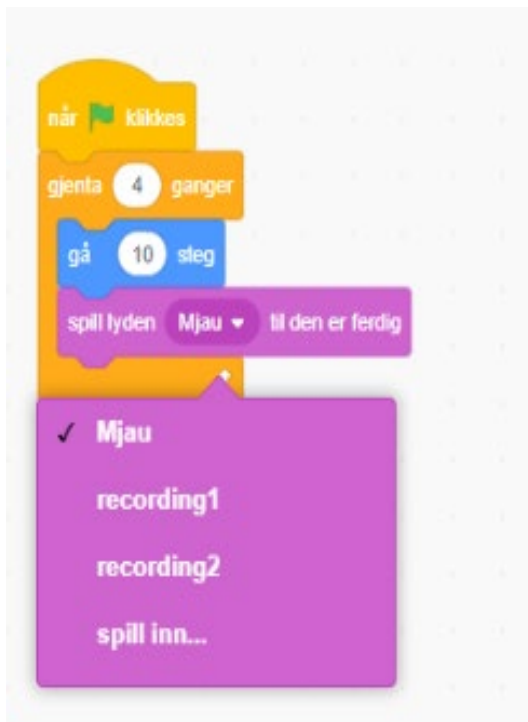
Spill inn en egen lyd, og legg den til i koden.



Figur 3: Skjermdump Scratch.

Lyder kan du legge til og spille inn selv under fanemarket "Lyder". Nederst til venstre trykker du på knappen med høyttalersymbol, og velger mikrofonikonet.

Lyden bytter du i kodeblokken:



Figur 4: Samme kodeeksempel fra Scratch. Viser hvordan du kan endre lyden.

Code Jumper - kodeverktøy

Code Jumper er et kodeverktøy som er utviklet for å lære elever med alvorlig synssvekkelse og blindhet om programmering. Code Jumper består av fysiske deler. Delene kalles hub, poder og plugger og kan settes sammen på ulike måter. Når du har satt dem sammen blir dette til en fysisk kode, med lydtilbakemelding som også kan avleses på pc. På PC-en vil koden vises på skjermen, eller leselisten.

Huben kobles trådløst til en PC/Chromebook eller Android-enhet, og programmet lastes enkelt ned fra Microsoft store. Tilkoblingen virker sømløs og rask, og programmet er enkelt å sette seg inn i.

Det er sammensetningen av podene som bestemmer hvordan koden ser ut:

Video: [Codejumper introduksjon](#)

Code Jumper kan benyttes alene, i mindre grupper og i klasserommet og er et verktøy som gir god mulighet for inkluderende læringsfellesskap. Introduksjonen bør for de fleste foregå i mindre grupper eller være lærerstyrt.

Code Jumper består av følgende deler:

- Hub: Hjernen og høyttaler. Den største enheten. Play pod: Velger lyder. To knapper. Blå.
- Pause pod: Setter pause. 1 knapp. Rød. Loop pod: Antall repetisjoner. 1 knapp. Gul.
- Selection pod: Lager vilkår (if/else). 1 knapp. Grønn.

Hvordan virker den?



For å starte koding med Code Jumper, anbefales det å følge opplæringsoppleggene som er lenket til under "tips og ideer". Det er noen flere ting det kan være lurt å huske på:

Code Jumper er ikke mulig å bruke uten appen.

- Når skjermleser er på, kommer all pc-lyd også gjennom hub-en.
- Programmet forsvinner i app'en når du kobler ut pod'ene. Det anbefales derfor å skrive ned koden i et Word-dokument eller lignende for å beholde den til senere læring. Du kan bruke [Statpeds kodebrikker](#) til det.
- Lydene kan endres i appen, men ikke antall og annet. Antall og annet kan kun endres fysisk. Det vil si at man i begynnelsen bør stille inn lydoppsettet for elevene.
- Når man skal legge inn egne lydfiler støttes kun mp3 og wav-filer.
- Looper lages i rekkefølge med klokka.

Tilrettelegging for elever med synshemming

Her trengs det ingen tilrettelegging. Elevene kan fra ung alder bli kjent med verktøyet på egne premisser og ved hjelp av utforskning. Det er et intuitivt verktøy som passer

for hele elevgruppa. Pedagogen kan gradvis innføre mulighetene som ligger verktøyet; endre hastighet, endre lyd, lage egne lyder og sette til poder som skaper mer utfordring og variasjon i koden.

Video: [Codejumper](#)

For å gjøre programmene enkle i bruk, anbefales det å bruke en utgang, tread, og da helst tread 1 på venstre side av hub'en. Det gjør det enklere å forstå hvordan tekstprogrammering er bygd opp. I økter der man skal lage musikk, er det mer naturlig å bruke flere treads, for å få flere instrumenter til å gi lyd samtidig.

Tips og ideer

- På codejumper.com finner du lenke til et MEC-kurs som er en god introduksjon til kurset. Vi anbefaler lærere og barnehagelærere å sette seg inn i dette på forhånd.
- Det er også utviklet mange veiledende videoer om de ulike pod'ene.
- I samarbeid med APH er det utviklet flere undervisningsforslag. Oppleggene har både aktiviteter med fysisk koding og med Code Jumper. De er enkle å sette seg inn i og har tydelige mål som kan sees i sammenheng med fagfornyelsen.
- Se instruksjonsvideoer under, laget av APH — American Printing House for the Blind. Du finner en video om hver av de ulike podene.

Forslag til undervisningsopplegg

Code Jumper kan benyttes fra tidlig barnehagealder. Her ser dere en gutt på tre år som selv finner frem til dyrellyder og hastigheten på lyden. Dette kan gjøres i små grupper i barnehagen. Barna kan så få gjenta koden før de lager den. Begynn med dyrellyder eller andre kategorier som er kjent for barna. De andre podene kan gradvis innføres når barna er klare for de

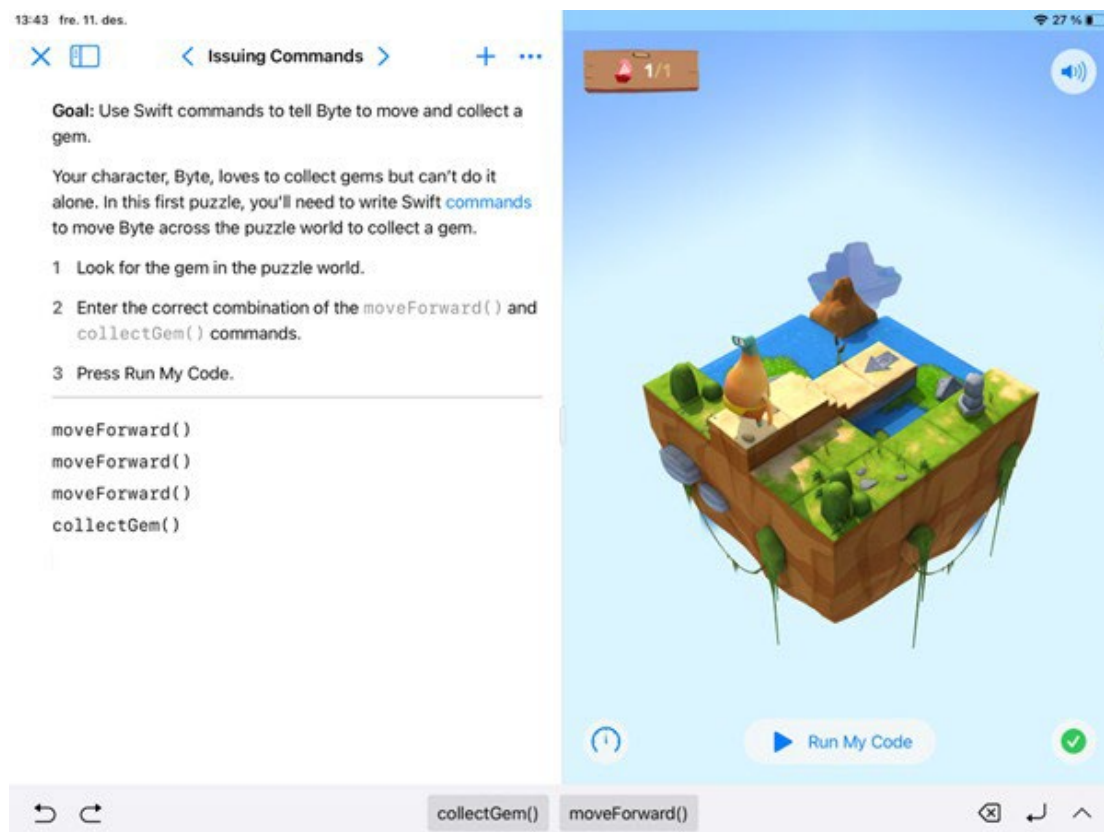
Om eleven alt er kjent med koding, kan Code Jumper brukes for å utvikle denne forståelsen.

Eleven kan lage musikkstykker, historier med mer og bli utfordret til å sette inn løkker, pauser, betingelser og bestemme antall/konstanter.

Det finnes ganske mange oppgaver som er utviklet av produsenten. Du finner dem her:

- [Code Jumper Lessons](#)

Swift Playgrounds - app for programmering



Hvordan virker den?

Swift er et programmeringsspråk som er utviklet av Apple. For å gjøre det enkelt og attraktivt å lære Swift, har Apple også utviklet Swift Playgrounds. Swift Playgrounds er et undervisningsopplegg og spill i en og samme app. Den kan kjøres på både iPad og Mac.

Det finnes mange såkalte Playgrounds som kan lastes ned og brukes med appen. Vi anbefaler at du bruker undervisningsoppleggene Learn to Code 1 og 2. De er beregnet på elever som ikke har noen erfaring med programmering fra før. Learn to Code 1 og 2 lastes ned og tas i bruk i selve appen.

Når du bruker Learn to Code 1 og 2, vil skjermbildet i Swift Playgrounds være todelt. Den ene delen viser såkalte «puzzle worlds» som er representert ved et tredimensjonalt rutemønster. Figuren Byte programmeres til å bevege seg i rutemønstret for å løse oppgaver. Den andre delen av skjermbildet består av tekst. Her presenteres selve oppgaven du skal løse, og her skriver du programkoden som bestemmer hva Byte skal gjøre. Learn to Code 1 og 2 inneholder mange «puzzle worlds» med ulik vanskelighetsgrad.

Swift Playgrounds kan også brukes til å programmere micro:bit, LEGO Mindstorms EV3, Skoog, Osmo og Sphero. Men dette er andre playgrounds som ikke inngår i undervisningsoppleggene Learn to Code.

Nettsted: <http://www.apple.com/swift/playgrounds>.

Tilrettelegging for elever med synshemming

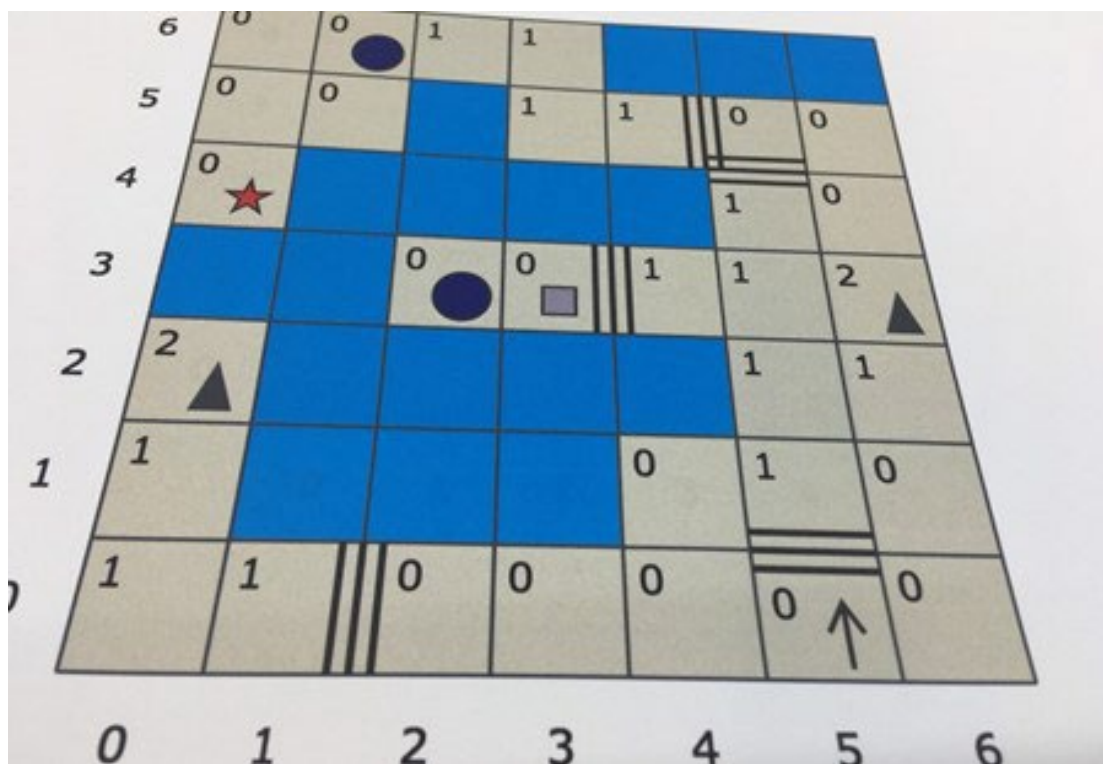
Video: [Swift Playgrounds: Tilpasninger for elever med synsnedsettelse](#)

Vi forutsetter her at du bruker undervisningsoppleggene Learn to Code 1 og 2.

Det lønner seg å bruke appen slik at du har kodeområdet og rutemønsteret ved siden av hverandre, ikke over hverandre. På Mac er dette standard, men på iPad bør du bruke enheten liggende.

For elever som bruker tekst-til-tale, kan det være nødvendig å slå av lyden i appen. Appen bruker både bakgrunnsmusikk og lydeffekter, og dette kan være forstyrrende. Bakgrunnsmusikken har ingen direkte sammenheng med det som skjer i appen, men det har lydeffektene. Bakgrunnsmusikk og lydeffekter kan slås av uavhengig av hverandre.

Swift Playgrounds er utviklet for å fungere sammen med VoiceOver, Zoom og Større skriftstørrelse. Dessuten kan invertering eller filtre fungere. Invertering vil kun fungere for den tekstlige delen av skjermbildet. Men det går an å slå inverteringen av og på avhengig av om du skal skrive kode eller følge med når Byte skal utføre programkoden du har laget.



Mange elever med synsnedsettelse vil ha problemer med å se de ulike «puzzle worlds» i appen. De består av mange farger og kan inneholde mange detaljer. Men det er utarbeidet et støttmateriell for Learn to Code som vil være helt avgjørende for mange elever i denne gruppa. Støttmaterialet finnes i formatene svart skrift i kontrast, i punktskrift, og for svellepapir. Her vil du kunne utforske de ulike «puzzle worlds» i detalj før du går løs på dem i appen.

På iPad er det en linje nederst i appen med kodeforslag (Code completion). For å løse oppgavene i Learn to Code kan du bruke forslagene som til enhver tid dukker opp. På iPad er det en fordel å bruke appen med skjermtastaturet skjult. Hvis skjermtastaturet er åpent, vil forslagene fra Code completion få dårligere kontrast. På Mac vises kodeforslagene litt annerledes.

Elever som bruker VoiceOver bør bytte til engelsk talesyntese siden appen er på engelsk. Elevene bør også være fortrolige med Rotoren i VoiceOver. Bruk av VoiceOver med Swift Playgrounds er ganske godt forklart i en serie med videoer fra Hadley (se under Tips og ideer).

- Bruk støttmateriell i ønsket format (kontrast, punktskrift og svellepapir). Bruk appen i liggende

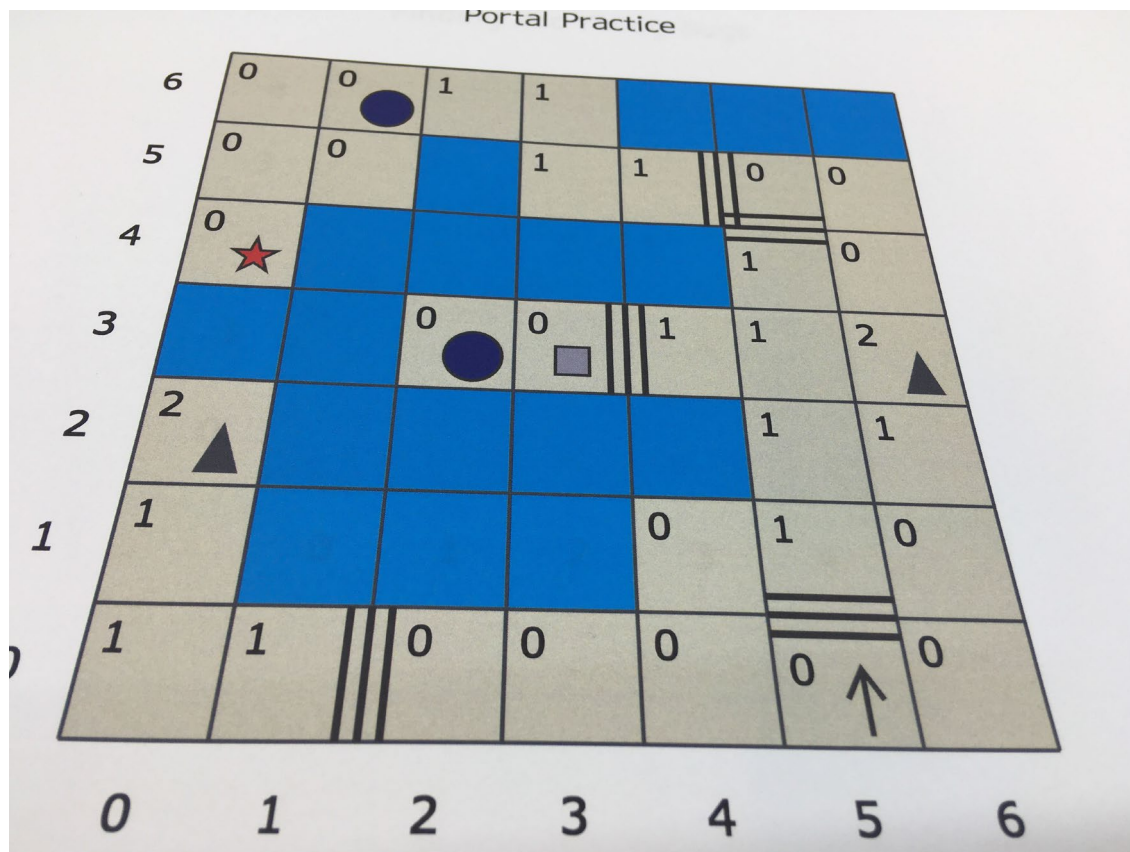
format (landscape).

- Slå av bakgrunnsmusikk og/eller lydeffekter ved behov. Bruk appen med skjermtastaturet skjult (iPad).
- Bytt språk i VoiceOver slik at det stemmer overens med språket i appen (engelsk).
- Vurder behovet for å bruke appen med Zoom i iOS (Innstillinger > Tilgjengelighet > Zoom) eller Zoom i macOS (Systemvalg > Tilgjengelighet > Zoom).
- Vurder behovet for å bruke appen med Zoomfiltre i iOS (Innstillinger > Tilgjengelighet > Zoom) eller med invertering og fargefiltre i macOS (Systemvalg
 - Tilgjengelighet > Skjerm > Skjerm og Systemvalg > Tilgjengelighet > Skjerm > Fargefiltre).
- Vurder behovet for å bruke appen med større tekst i iOS (Innstillinger > Tilgjengelighet > Skjerm og tekststørrelse > Større tekst > Større skriftstørrelser).

Tips og ideer

- [Swift Playgrounds Tactile Puzzle Worlds](#) Støttmateriell som kan skrives ut i punkt, kontrastfarger og på svellepapir. Materiellet i punktskrift er laget for ulike ViewPlus punktskrivere (Columbia, Delta, Elite, Premier og Max). Versjonene for kontrast og svellepapir er i A3-format eller tilsvarende.
- [VoiceOver med Swift Playgrounds](#) (Videoer fra Hadley. NB Krever pålogging.)
- [iPad for personer med nedsatt syn](#)

Forslag til undervisningsopplegg



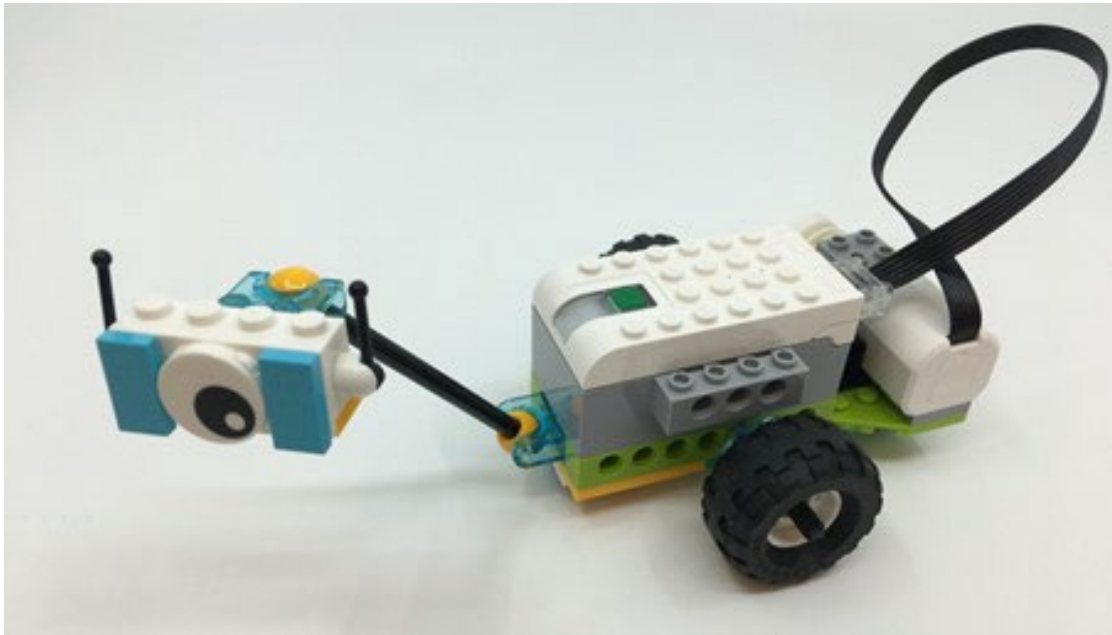
Det finnes et undervisningsopplegg for Swift Playgrounds som er tilrettelagt for elever med synsnedsettelse. Det består av undervisningsoppleggene Learn to Code 1 og 2. De er tilrettelagt i punktskrift, i kontrast, og for svellepapir. Det finnes også en Teachers Guide. Brukere av VoiceOver anbefales videoene fra Hadley som gir en god innføring i hvordan skjermleseren kan brukes med Learn to Code 1 og 2.

Undervisningsoppleggene Learn to Code 1 & 2 lastes ned i Swift Playgrounds-appen.

- [Swift Playgrounds Tactile Puzzle Worlds](#) (punkt, kontrast, svellepapir)
- [Learn to Code 1 & 2 Teacher Guide](#) (epub)
- [VoiceOver med Swift Playgrounds](#) (fra Hadley)

NB! Dokumenter i epub-format kan bare leses med bestemte apper. Mange av dem er gratis. Du kan for eksempel installere et epub-tillegg i nettleseren. For Edge: Innstillinger > Utvidelser, søk etter «epub reader». Installer valgt utvidelse.

Lego WeDo - robot



Hvordan virker det?

Lego WeDo 2.0 er et Lego-sett som lar deg bygge Lego-figurer som deretter kan programmeres. Det er et enklere sett enn for eksempel Lego Spike og Lego Mindstorms.

Tilrettelegging for elever med synshemming

WeDo-klossene

Selv om WeDo-settet inneholder nesten 300 klosser, vil du klare deg med langt færre for å bygge mange av figurene. Figurene Milo, for eksempel, kan bygges med under 50 klosser. Likevel kan byggingen være utfordrende for elever med synsnedsettelse. Men mange av disse elevene vil likevel klare å bygge disse figurene, i hvert fall deler av dem. Blinde elever er avhengig av hjelp.

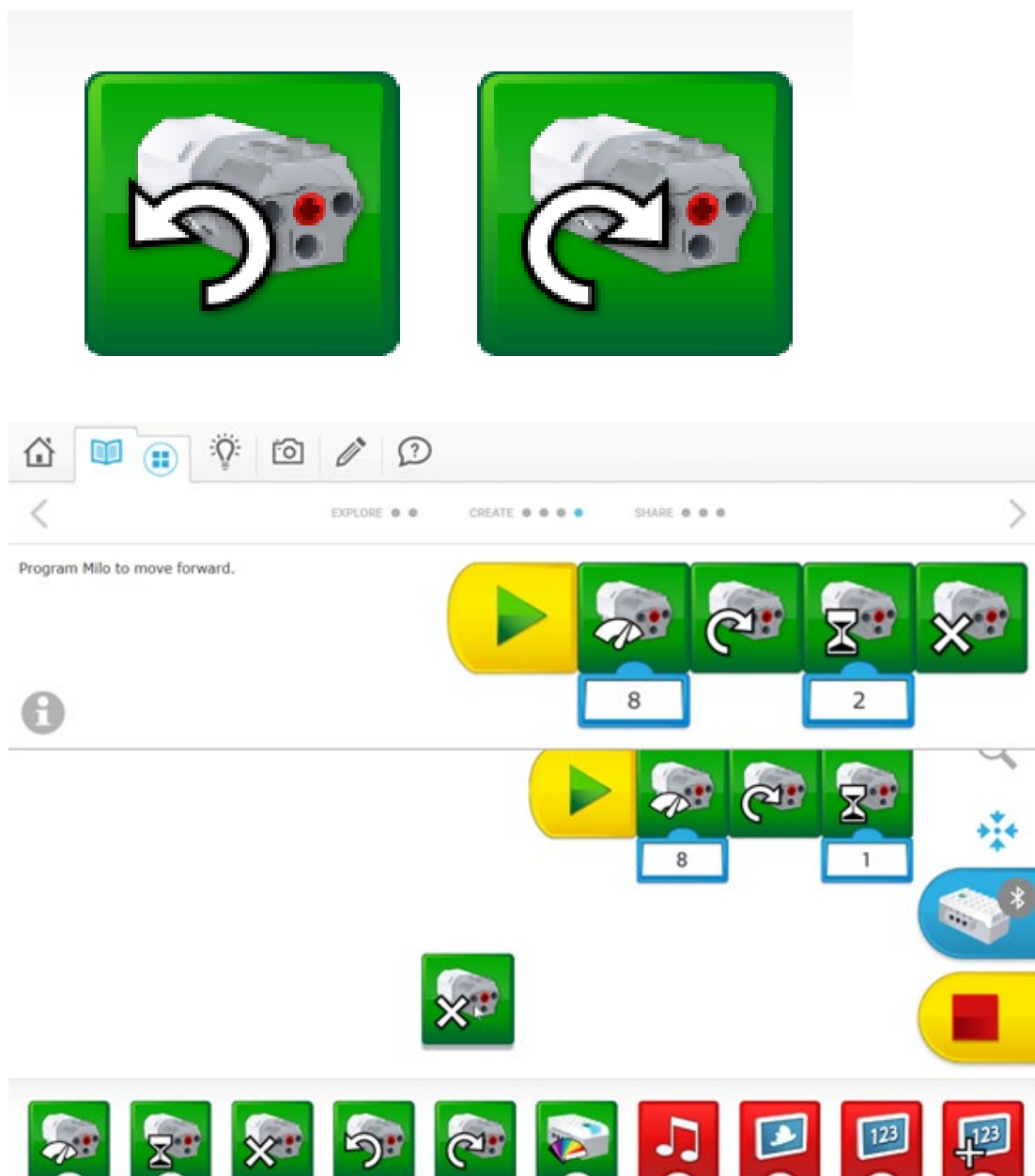
Husk:

- Mange elever med synsnedsettelse vil trenge hjelp til å bygge figurene.
- Det kan lønne seg å skrive ut instruksjonene på papir i litt stort format:

<http://education.lego.com/en-us/product-resources/wedo-2/downloads/building-instructions>.

Programmere WeDo

Lego har utviklet en egen app for WeDo 2.0. Den minner om appen som brukes til å programmere Lego Mindstorms. Disse programløsningene er veldig utfordrende for mange synshemmede, men kan muligens brukes av enkelte elever med moderat synssvekkelse. Det må i så fall prøves ut med den enkelte elev.



WeDo-appen tar deg først gjennom byggeprosessen av den figuren du har valgt å bruke. Deretter kommer du til selve programmeringen. Programmeringen foregår ved at du må velge blokker fra en lang meny nederst i skjermbildet. Blokkene dras inn i selve editor-området der du bygger opp programkoden din. Blokkene er i ulike farger og har ofte symboler som kan være ganske detaljerte.

Symbolene er vanskelige å se. Blokkene kan også være vanskelig å skille fra hverandre siden mange av dem ser nesten like ut. Moderat bruk av Windows Skjermforstørring kan hjelpe. Windows Høykontrast har ingen effekt på WeDo-appen, men Windows Fargefiltre kan muligens hjelpe for noen elever. Dette må først utprøves sammen med eleven. Appen er ikke mulig å bruke med skjermleser.

For elever med alvorlig synssvekkelse som leser visuelt og elever som bruker punktskrift, anbefales i stedet å programmere WeDo 2.0 med Python. Det betyr at du kan bruke Visual Studio Code, eller en tilsvarende kildekode-editor, til å skrive programkode. Det er utarbeidet en Komme i gang-veileder for dette (se under Tips og ideer).

Husk:

- I stedet for å følge byggeinstruksjonene i appen, kan det lønne seg å skrive dem ut på papir.

- Vurder om det er behov for å bruke Windows Forstørrelsesprogram eller Fargefiltre (**Win+U**).
- Vurder behovet for å bruke Zoom kombinert med Zoomfiltre i iOS (Innstillinger > Tilgjengelighet > Zoom).
- Elever som ikke ser godt nok til å bruke WeDo-appen, anbefales i stedet å bruke Python. Se Komme i gang-veileder.

Forslag til undervisningsopplegg

- WeDo-appen er et undervisningsopplegg i seg selv
- [Instruksjoner for bygging av WeDo-figurene \(PDF\)](#)
- [Lego WeDo oppgaver \(Python\)](#)

Tips og ideer

- [Komme i gang med Lego WeDo 2.0 i Python med Visual Studio Code](#)
- [Tilpasningsmuligheter i Windows for elever med nedsatt syn](#)
- [Tilpasningsmuligheter på iPad for elever med nedsatt syn](#)

Lego Mindstorms - robot



Hvordan virker det?

Lego Mindstorms EV3 er et Lego-sett som lar deg bygge og programmere roboter. Det er mer avansert enn for eksempel Lego Spike og Lego WeDo.

NB! Lego Mindstorms produseres ikke lenger.

Tilrettelegging for elever med synshemming

Mindstorms-settet inneholder veldig mange klosser. For mange elever med synshemming vil byggingen være svært utfordrende.

Husk:

- Mange elever med synshemming vil trenge hjelp til å bygge figurene.
- Det kan lønne seg å skrive ut instruksjonene på papir i litt stort format: <http://education.lego.com/en-us/product-resources/mindstorms-ev3/downloads/building-instructions>

Programmere Mindstorms

Mindstorms-appen minner om appen for Lego WeDo. Disse programløsningen veldig utfordrende for mange synshemmede, men kan muligens brukes av enkelte elever med moderat synssvekkelse. Det må i så fall prøves ut med den enkelte elev.

Appen tar deg først gjennom byggeprosessen av den figuren du har valgt å bruke. Deretter kommer du til selve programmeringen. Programmeringen foregår ved at du må velge blokker fra en lang meny nederst i skjermbildet. Blokkene dras inn i selve editor-området der du bygger opp programkoden din. Blokkene er i ulike farger og har ofte symboler som kan være ganske detaljerte. Symbolene er vanskelige å se.

Blokkene kan også være vanskelig å skille fra hverandre siden mange av dem ser nesten like ut. Moderat bruk av Windows Skjermforstørrelse kan hjelpe. Dette må først prøves ut sammen med eleven.

Appen er ikke mulig å bruke med skjermleser. For elever med alvorlig synssvekkelse som leser visuelt og elever som bruker punktskrift, anbefales i stedet å programmere Mindstorms med Python. Lego har beskrevet hvordan du kan komme i gang med å programmere Lego Mindstorms med Python i Visual Studio Code. Dessverre er Bluetooth-støtte i Python lite utviklet. Vi anbefaler i stedet å bruke USB-kabelen som følger med settet.

Husk:

- I stedet for å følge byggeinstruksjonene i appen, kan det lønne seg å skrive dem ut på papir.
- Vurder om det er behov for å bruke Windows Forstørrelsesprogram eller Fargefiltre (**Win+U**).
- Vurder behovet for å bruke Zoom kombinert med Zoomfiltre i iOS (Innstillinger > Tilgjengelighet > Zoom).
- Elever som ikke ser godt nok til å bruke appen, anbefales i stedet å bruke Python.

Forslag til undervisningsopplegg

- Appen er et undervisningsopplegg i seg selv
- [Lego Mindstorms EV3 byggeinstruksjoner \(PDF\)](#)

Tips og ideer

- [Lego Mindstorms EV3 med Python i Visual Studio Code](#)
- [Tilpasningsmuligheter i Windows for elever med nedsatt syn](#)
- [Tilpasningsmuligheter på iPad for elever med nedsatt syn](#)

Arduino - mikrokontroller



Hvordan virker den?

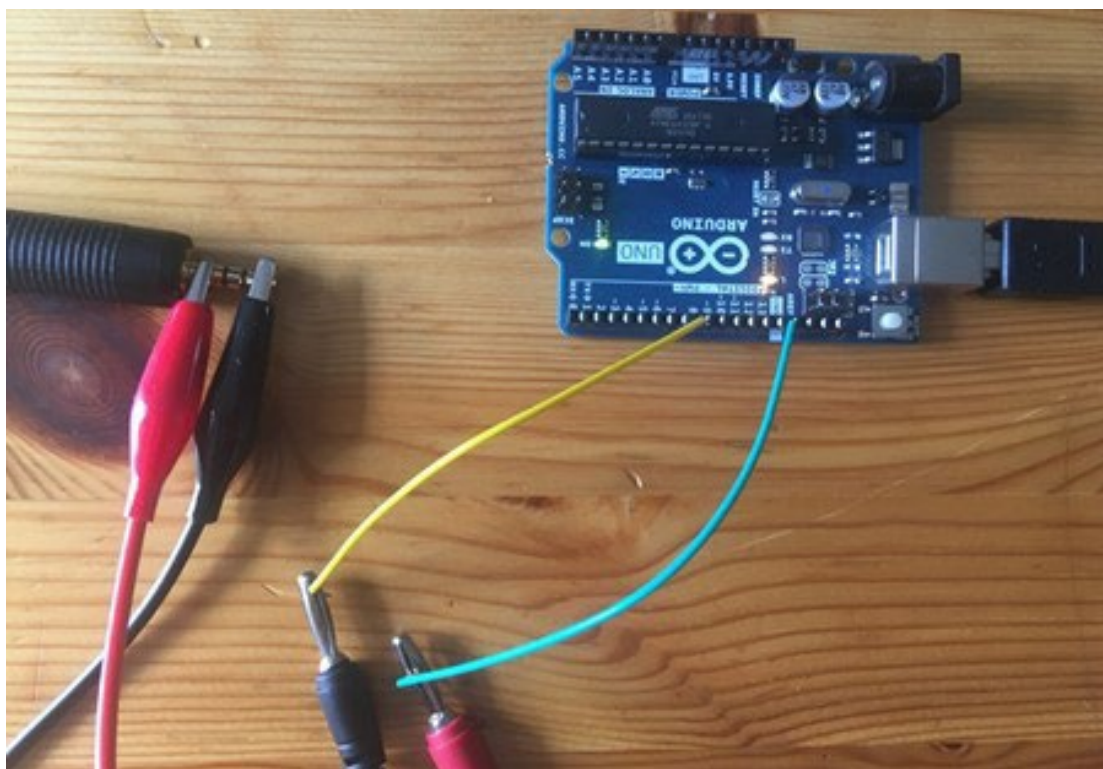
Arduino er en gruppe mikrokontrollere som kan programmeres med en datamaskin. Arduino består også av shields (kort som monteres direkte på mikrokontrollerne og som gir mer funksjonalitet) og av forskjellig tilbehør. Arduino mikrokontrollere kan kjøpes løse eller i sett som også inneholder tilbehør.

Arduino kan til en viss grad sammenlignes med micro:bit. Begge er mikrokontrollere som er utstyrt med strømtilkobling, USB-port, en reset-knapp, og som har en rekke tilkoblingspunkter. Men til forskjell fra micro:bit, kan ikke Arduino gi noen output på egen hånd, og den mangler også sensorer for å registrere bevegelse og temperatur. For å få gjort noe med Arduino, må du koble til utstyr.

Appen Arduino IDE brukes til å skrive programmer for Arduino på plattformene Windows, macOS og Linux. Arduino Web Editor er for deg som heller vil programmere Arduino i nettleseren.

Arduino programmeres med C++ som fremdeles er et av de mest brukte programmeringsspråkene. Det egner seg veldig godt til å programmere mikrokontrollere. Programmering av Arduino har en høyere terskel enn programmering av micro:bit.

Tilrettelegging for elever med synshemming



Elever med synsnedsettelse vil klare å gjøre seg kjent med strømtilkoblingen, USB- porten og reset-knappen uten videre. Det er ikke behov for merking. De aller fleste tilkoblingspunktene er allerede merket med navn eller nummer. Merkingen er gjort med liten skrift som er veldig vanskelig å lese, selv om kontrasten er god. En lupe eller lese-tv kan være til hjelp for å se små detaljer.



Tilkobling av ledninger og utstyr er svært utfordrende. Mange elever vil ha behov for hjelp til dette. For dem vil nok selve programmeringen av Arduino være det viktigste. Det finnes Arduino-utstyr som fokuserer mer på programmering enn på kobling, så det kan være verd å lete litt blant Arduino-utstyret som finnes på markedet.

Samtidig er jo litt av poenget med Arduino at du skal bygge og koble. Det finnes mange spennende prosjekter på nett hvis du søker etter «arduino» i kombinasjon med «visually impaired» eller «blind». Men selv om prosjektene kan være relevante, er de ikke nødvendigvis enkle å bygge.

Husk:

- Mange elever med synsnedsettelse vil trenge hjelp til å bygge og koble. Det er ikke sikkert kobling av ledninger gir så mye mening for de elevene som ser dårligst.
- Enkelte elever kan ha stor nytte av å bruke en lupe eller lese-tv når de skal bygge og koble utstyr til Arduino.

Programmere Arduino

Arduino IDE



For tiden (juni 2021) er gjeldende versjon Arduino IDE 1.8. Det finnes to versjoner, en som kan [lastes ned direkte fra arduino.cc](#) og en som er tilgjengelig fra Microsoft Store. Du bør velge den første. Det er i ferd med å komme en ny versjon 2.0 som foreløpig er en beta-versjon.

Versjon 1.8 har begrenset funksjonalitet for elever med synsnedsettelse. Den har en enkel zoom-funksjon, men det er stort sett det hele. Appen fungerer bra med skjermleserne NVDA og VoiceOver, men ikke med JAWS. [Java Access Bridge](#) kan muligens forbedre ytelsen med JAWS. Kryss av for Use accessibility features under Fil > Innstillinger hvis du bruker leselist.

Video: [Arduino IDE: Tilpasninger for elever med synsnedsettelse](#)

Versjon 2.0 har mye mer funksjonalitet som elever med synsnedsettelse kan dra nytte av. Den har bedre zoom-funksjon, flere fargeoppsett, forenklet oppsett, meget godt utvalg av hurtigtaster, og bedre navigering i programvinduet. Men den har noen utfordringer for elever som bruker skjermleser siden skjermleserstøtte ennå ikke er helt på plass. Men som nevnt er versjon 2.0 foreløpig beta. Arduino IDE 2.0 anbefales for elever som leser visuelt, men foreløpig ikke for elever som bruker skjermleser. De bør heller bruke Visual Studio Code.

Video: [Arduino med Visual Studio Code: Komme i gang](#)

Husk:

- Vi anbefaler ikke Arduino IDE fra Microsoft Store. [Last ned direkte fra arduino.cc](#).
- Elever som leser visuelt, kan ha stor nytte av tilgjengelighetsfunksjonene i Arduino IDE 2.0.
- Vurder om det er behov for å bruke Arduino IDE med Windows Forstørrelsesprogram eller Fargefiltre (Win+U).
- Elever som bruker skjermleser anbefales å bruke Visual Studio Code, fortrinnsvis med skjermleseren NVDA.

Arduino Web Editor

Arduino Web Editor er ikke å anbefale med skjermleser. Elever som leser visuelt kan godt bruke den, men Arduino IDE 2.0 har flere tilpasningsmuligheter.

Visual Studio Code

I utgangspunktet er det naturlig å velge Arduino IDE til å programmere Arduino mikrokontrollere. Men det kan likevel være en del å hente på å velge Visual Studio Code. Det er en svært utbredt app som kan brukes til å programmere nær sagt hva som helst. Den fungerer bedre med skjermleser. Med Visual Studio Code kan elevene bruke samme app til all tekstprogrammering de skal holde på med.

For å programmere Arduino med Visual Studio Code må du først installere Arduino IDE, deretter Visual Studio Code, og til slutt en Arduino-extension for Visual Studio Code.

Forslag til undervisningsopplegg

På [arduino.cc/education](#) finnes en del ressurser du kan dra nytte av. Det følger med prosjekter i de forskjellige Arduino-settene som er tilgjengelige. Disse kan være gode utgangspunkt for oppgaver. Men husk å bruke oppgaver som gir mening for elever som ikke ser.

[Arduinos Project Hub](#) inneholder massevis av spennende prosjekter. Du kan både søke og filtrere i oversikten.

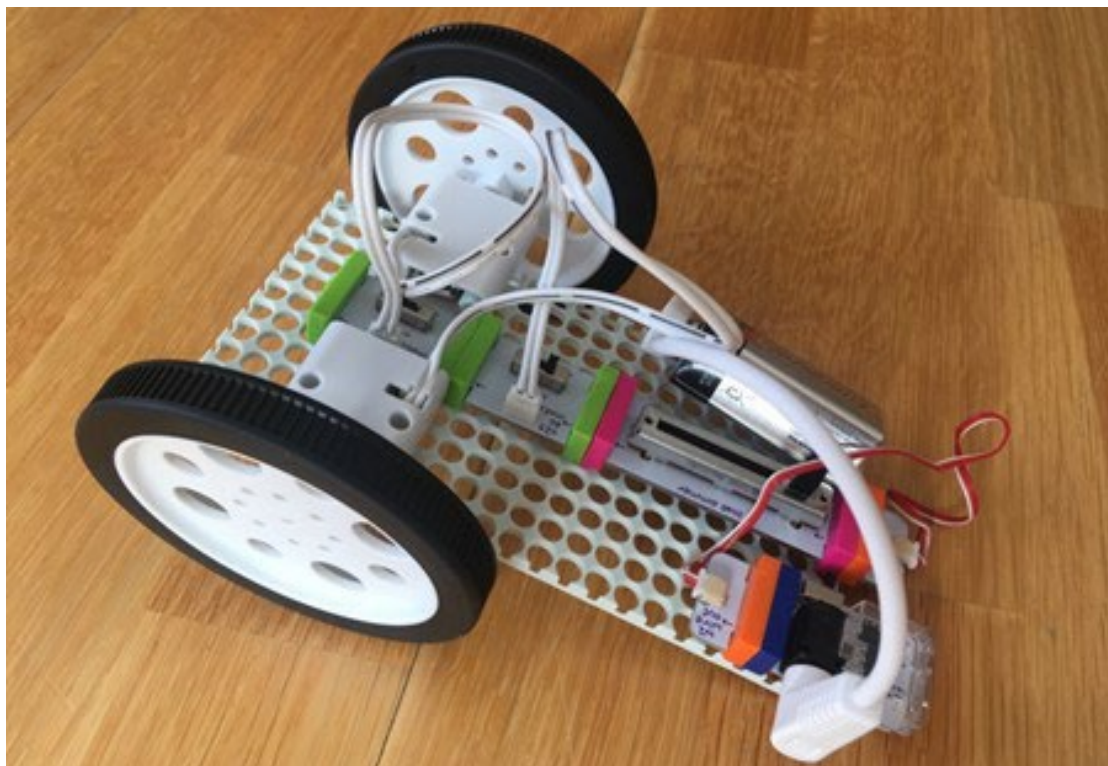
Det finnes også opplegg under [Remote Learning på arduino.cc](#). Her er det blant annet videoer. Under Fil > Examples i Arduino IDE-appen finnes dessuten mange eksempler.

[Kodeklubben har også noen få oppgaver](#).

Tips og ideer

- [Arduino-utstyr på markedet](#)
- [Tilpasningsmuligheter i Windows for elever med nedsatt syn](#)
- [Komme i gang med Arduino i Visual Studio Code Tilgjengelighetsfunksjoner og skjermleserstøtte i Arduino IDE](#)
- [Hvordan bruke Visual Studio Code med skjermleser](#)

LittleBits STEAM - elektronisk byggesett



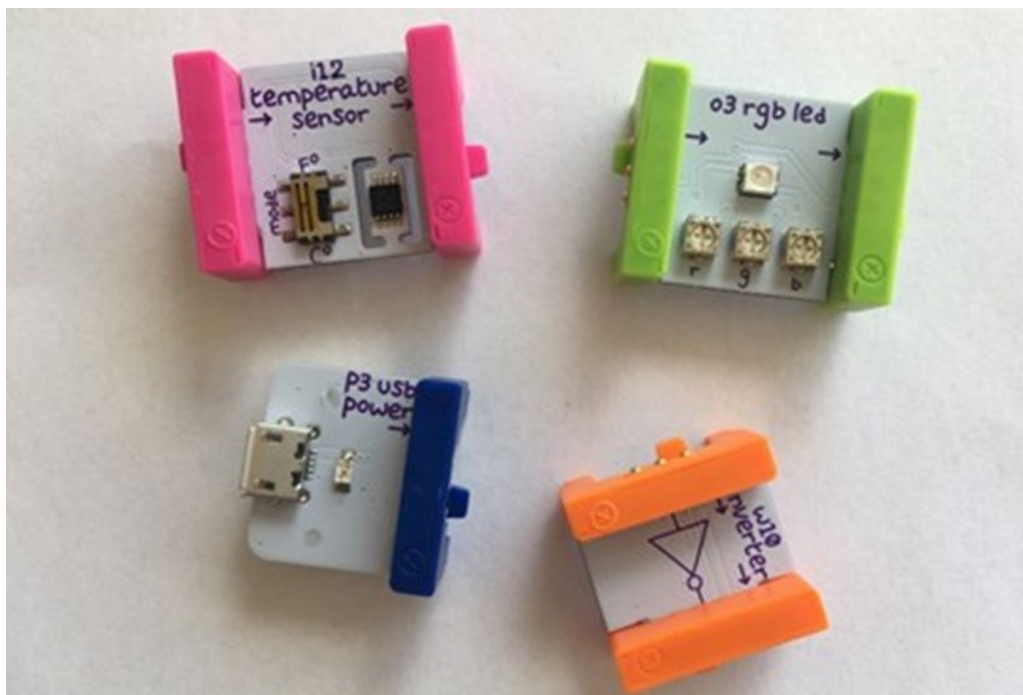
Hvordan virker det?

littleBits handler egentlig ikke så mye om programmering, men mer om elektronikk. Det består av mange forskjellige bits. En bit er en elektronisk komponent som utfører en bestemt oppgave. Du kan bygge kretser ved å sette sammen flere bit. I tillegg kan du koble på forskjellig tilbehør som for eksempel hjul og armer. Med littleBits kan du skape lys, lyd og bevegelse.

Selv om littleBits ikke handler så mye om programmering, dekker det likevel flere nøkkelbegreper og arbeidsmåter i den algoritmiske tenkeren.

Tilrettelegging for elever med synshemming

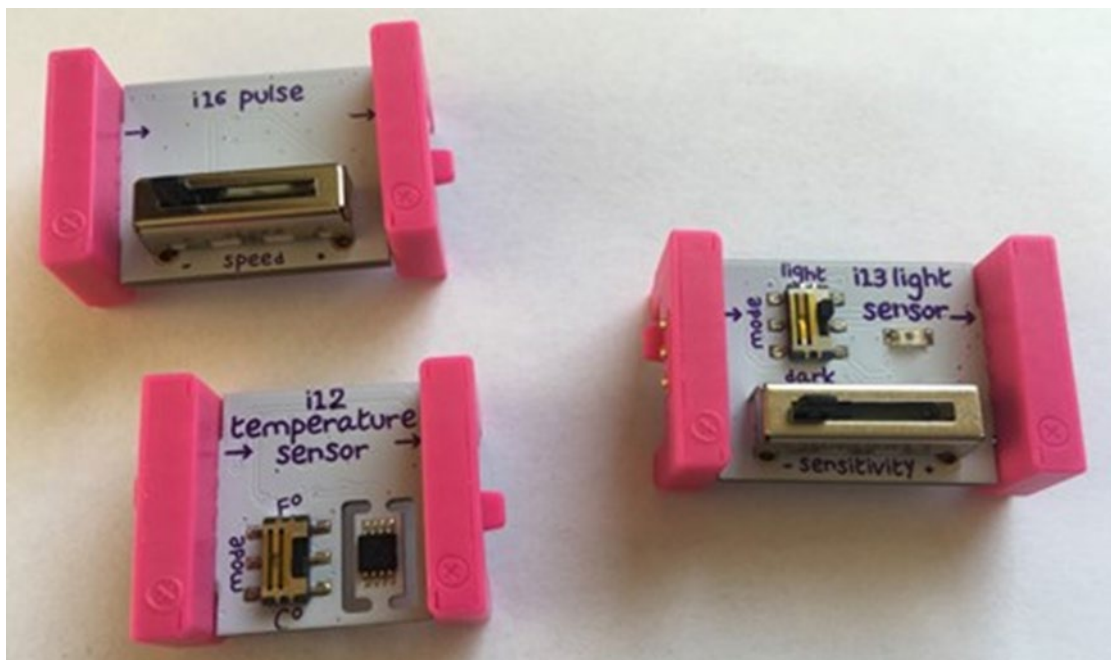
littleBits er veldig taktilt. Hver enkelt bit er taktil og det samme er også bryterne, skyvekontrollene og skruene som noen av bit-ene er utstyrt med. Bit-ene har ganske tydelige føtter som viser hva som er opp og ned.



Alle bit tilhører en bestemt kategori der blå betyr strømforsyning, rosa betyr input, grønn betyr output og oransje betyr kobling. Bortsett fra fargene, er det ikke mulig å vite hvilken kategori en bit tilhører. Med omtrent 20 forskjellige bits, blir det veldig vanskelig for blinde elever, og andre som ikke klarer å skille fargene, å vite hvilken kategori det er snakk om. For disse elevene kan det være en hjelp om du sorterer bit-ene i kategorier på forhånd, eller plukker fram de bit-ene som skal brukes i en bestemt oppgave.

På hver bit er det trykket på bit-navn. Kontrasten er god, men størrelsen på skriften er så liten at kun de færreste vil klare å lese den. Noen bit er ganske karakteristiske og vil være enklere å finne enn andre. Men mange elever vil nok trenge hjelp til å finne fram blant alle bit-ene. Men hver bit er unik, og det er derfor mulig å skille dem fra hverandre. Det er ikke plass nok til å merke i større skrift eller punktskrift.

Bit-ene settes sammen og danner en krets som skal utføre en bestemt oppgave. Hver bit har en pil som viser hvilken vei de skal kobles. Det starter alltid med en power-bit, altså en strømforsyning. Pilene er altfor små for de fleste i vår elevgruppe, men pilene er heller ikke så viktige. Hver bit har magneter som gjør at de bare passer én vei. Dette sikrer at hver bit kobles riktig og i neste omgang at kretsen blir riktig.



Noen av bit-ene har justeringsmuligheter. Justeringen gjøres med brytere, skyvekontroller eller skruer. Noen av bryterne og skruene er veldig små og svært utfordrende å bruke. Det er viktig for elevene å forstå hva justeringene gjør, men det kan hende de vil ha behov for hjelp til å utføre justeringene.

Å koble til noe av tilbehøret kan også være utfordrende siden tilbehøret ofte bare passer i en bestemt posisjon. Det er ofte trangt og ikke så enkelt å vite om man har tilbehøret i riktig posisjon eller ikke. Noe av tilbehøret skal dessuten festes med skruer og det vil også være krevende for mange. Her vil mange elever ha behov for hjelp.

Et nyttig tilbehør er monteringsbrettene. Etter at du har bygget en krets, kan den monteres på et monteringsbrett. Det gjøres ved å klemme bit-føttene ned i hullene på brettet. Det er ikke mulig å montere én og én bit på brettet. Det er litt vanskelig å få føttene ned i brettet på mange steder samtidig. Det er litt trangt og krever litt fikling. Her også kan mange elever ha behov for hjelp.

Husk:

- Vurder om det er behov for å sortere bit-ene etter kategori. Det gjør det enklere for elevene å finne den de leter etter. Det kan også være nødvendig at du hjelper elevene med å finne riktig bit.
- Mange elever vil trenge hjelp til å bruke justeringsmulighetene på enkelte av bit-ene.
- Mange elever vil også trenge hjelp til å koble til tilbehør og til å montere kretsene på monteringsbrettene.
- Bruk bits som er meningsfulle for elever med synsnedsettelse. Unngå LED-bit og Nummer-bit som bare gir visuell output. Bruk heller bits som lager lyd eller bevegelse. Men det må være bevegelse som elevene klarer å oppfatte og følge.

Forslag til undervisningsopplegg

Produsenten har et [ganske omfattende materiale for littleBits](#). Her finner du undervisningsopplegg, Teacher's Guide, Invention Guide (følger med settet), og mange andre ressurser.

Invention Guide inneholder også noen av undervisningsoppleggene i kortform.

Tips og ideer

littleBits leveres også i form av flere tilbehør, kit og såkalte expansion packs. Du kan også innføre programmering for littleBits, for eksempel ved å bruke en littleBit micro:bit Adapter.



Illustrasjon: Colourbox

I skolen brukes mange ulike digitale verktøy til programmering. Der finnes det både blokk- og tekstbaserte verktøy, og dessuten noen hybridløsninger som kombinerer blokk og tekst.

Men mange av verktøyene er vanskelige eller umulig å bruke for elever med nedsatt syn eller blindhet. Valg av digitalt verktøy er derfor svært viktig. Bruk et verktøy som har gode tilpasningsmuligheter eller som fungerer godt med hjelpemidlene eleven bruker.

I denne oversikten tar vi for oss digitale verktøy for tekstprogrammering. Disse kalles kildekode-editorer eller bare editorer.

Hva bør du velge?

For elever med relativt moderat synssvekkelse, kan mange av verktøyene egne seg. Mange editorer har tilpasningsmuligheter, for eksempel zoom-funksjon og fargetemaer, som enkelte elever med nedsatt syn kan ha nytte av. Men husk at det er viktig å prøve ut aktuelle verktøy sammen med eleven før du velger. Denne elevgruppa har svært individuelle behov og det er derfor vanskelig å si noe generelt som gjelder for alle.

For elever som bruker skjermleser, vil de fleste verktøyene ikke egne seg. Her må du velge blant de relativt få som faktisk fungerer.

Anbefalte kildekode-editorer for elever med moderat synssvekkelse

Dette gjelder for elever som jobber visuelt. De fleste verktøyene har tilpasningsmuligheter og kan fungere for enkelte elever, men nesten ingen av disse mulighetene imponerer. Derimot er det en editor som kan anbefales:

- **Visual Studio Code**

Anbefalte kildekode-editorer for elever med blindhet eller alvorlig synssvekkelse

Dette gjelder for elever som bruker skjermleser, enten alene eller i kombinasjon med skjermforstørring. Anbefalingene gjelder for Windows hvis ikke annet er nevnt:

- **Visual Studio Code (JAWS, NVDA)**
- **Spyder (JAWS, NVDA)**
- **Notepad++ (JAWS, NVDA, Supernova)**
- **PyDev (JAWS, Supernova)**
- **TextMate (macOS) (VoiceOver)**

Hvis du skal programmere Jupyter notebooks eller micro:bit, anbefaler vi Visual Studio Code:

- **Jupyter Notebooks in Visual Studio Code**
- **micro:bit med Python: Komme i gang**

Oversikt over kilde-kodeeditorer

Her finner du en kort oversikt over alle kilde-kodeeditorer vi har vurdert. Alle er testet på Windows-plattformen, bortsett fra når noe annet er angitt. En mer detaljert beskrivelse av ulike kildekode-editorer finner du på

<https://github.com/oivron/coding-without-seeing>.

- **IDLE**
- **JupyterLab Jupyter Notebook**
- **Microsoft MakeCode Mu**
- **Notepad++ PyCharm PyDev Python Editor repl.it**
- **Spyder**
- **TextMate (macOS)**
- **Thonny trinket.io**
- **Visual Studio Code**