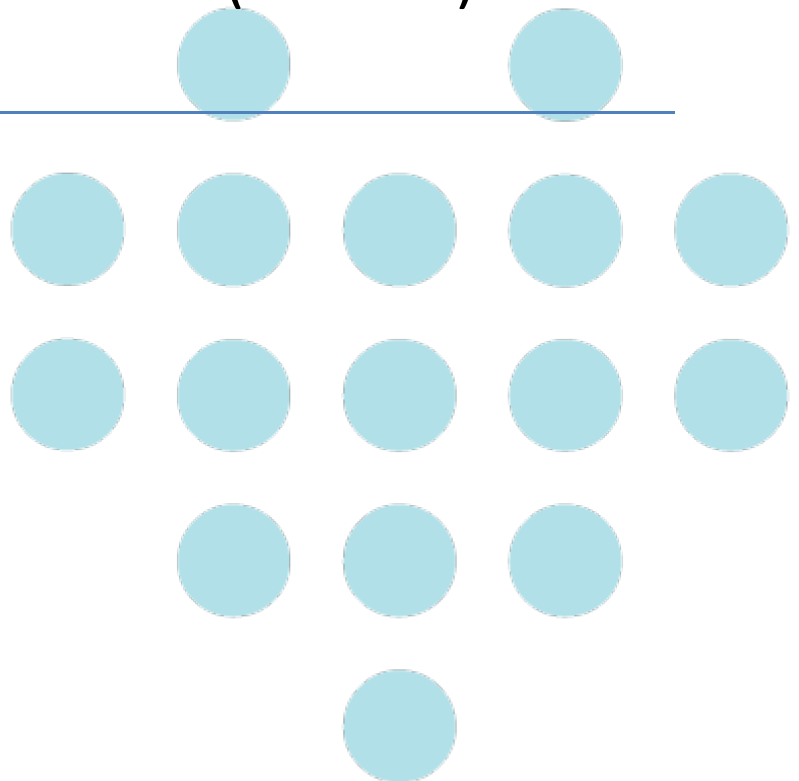




Før super:bit-oppgøret (120 min)

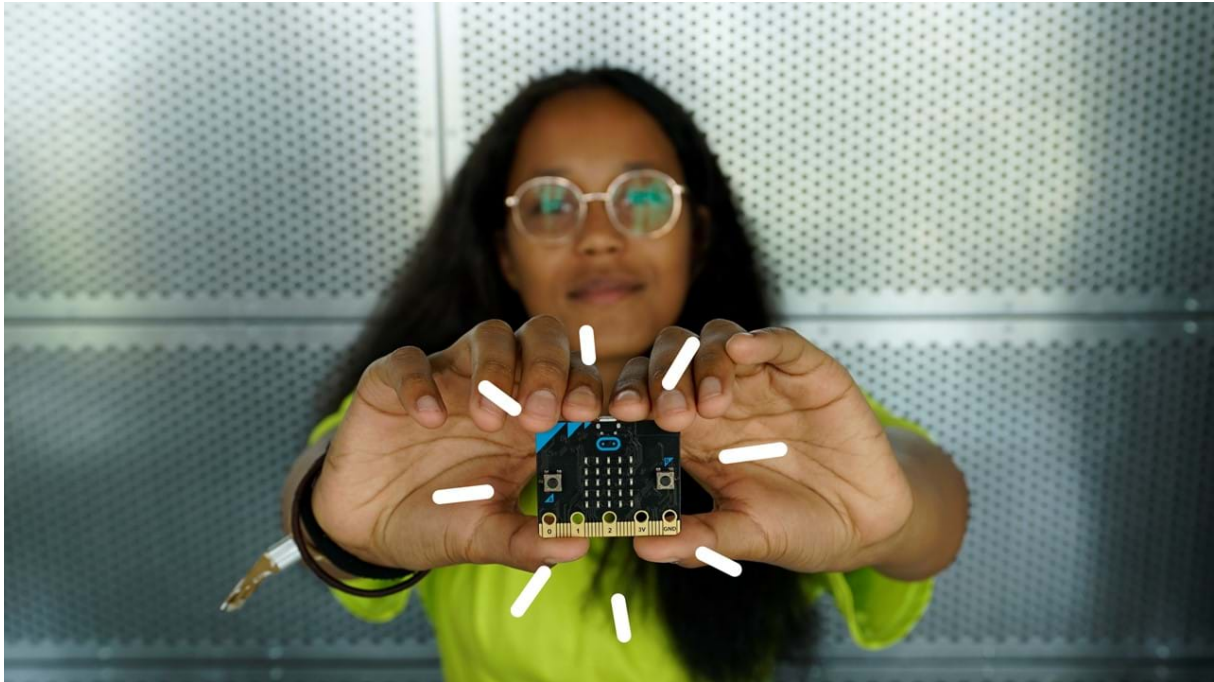
Lærerveiledning forarbeid (6. trinn)

Versjon: August 2019



Innhold

Om super:bit	3
Hovedområder og kompetansemål fra læreplanene 2020.....	4
Forarbeid	4
1 Telle 1 – 2 – 3	5
2 Tegn etter instruksjoner	6
3 Kompisprogrammering	8
Begrepsforklaringer	10



Om super:bit

super:bit er et skoleprogram som er utviklet av vitensentrene i Norge i forbindelse med Udir sin Nasjonale satsning: «Den teknologiske skolesekken.» Ett av målene til Den teknologiske skolesekken er å gi elever kunnskap om og forståelse for teknologi, algoritmisk tenkning og programmering

Etter å ha deltatt på vårt opplegg skal vi i tillegg til å gi elevene kunnskap skal vi ha gitt lærerne innsikt/evne og vilje til å undervise innen programmering/koding fremover. Lærerne skal også ha lyst til å delta på mer formell kursing og annet innen emnet senere. Ønsket er at vårt opplegg skal fungere både som en inspirasjon for lærerne og fjerning av «frykt for koding/programmering».

Programmet er variert og involverende, og elevene jobber aktivt og utforskende. Det legges stor vekt på mestring og trivsel, og elevene lærer gjennom opplevelse, aktivitet, samarbeid og innspill.

Det beste er at elever og lærere er godt forberedt før undervisningsprogrammet, og vi oppfordrer lærerne til aktivt å ta del i opplegget sammen med elevene.

Hovedområder og kompetansemål fra læreplanene 2020

Matematikk

Utforsking og problemløsning

Utforsking i matematikk handlar om at elevane leiter etter mønster, finn samanhengar og diskutere seg fram til ei felles forståing.

Algoritmisk tenking er viktig i prosessen med å utvikle strategiar og framgangsmåtar for å løyse problem. Problemløsning i matematikk handlar om at elevane utviklar ein løysingsmetode på eit problem dei ikkje kjenner frå før.

Kompetansemål etter 6.trinn

Mål for opplæringa er at eleven skal kunne bruke variablar, lykkjer, vilkår og funksjonar i programmering til å utforske geometriske figurar og mønster.

Naturfag

Teknologi

Elevene skal forstå, skape og bruke teknologi, inkludert programmering og modellering, i arbeid med naturfag. Gjennom å bruke og skape teknologi kan elevene kombinere erfaring og faglig kunnskap med å tenke kreativt og nyskapende.

Verdier og prinsipper

Naturfaget skal være en arena for å arbeide praktisk og utforskende og gi rom for undring, nysgjerrighet og fasinasjon. Dette legger til rette for at elevene kan utvikle skaperglede, engasjement og evne til nytenkning.

Kompetansemål etter 7.trinn

Mål for opplæringa er at eleven skal kunne;

- skille mellom observasjoner og slutninger, organisere data, bruke årsaks-virkningsargumenter, trekke slutninger, vurdere noen feilkilder og presentere funn og hvordan en har kommet fram til disse
- utforske, forstå og lage teknologiske systemer som består av deler som virker sammen
- reflektere over hvordan teknologi kan løse og skape utfordringer

Forarbeid

Før programmet skal gjennomføres bør elevene ha utført enkelte aktiviteter og ha kjennskap til en del begreper knyttet til skoleprogrammet. Begrepsforklaringer ligger helt til slutt i dokumentet.

Begreper

Algoritmer, kode, kommando, logikk, løkker, problemløsning, program, programmering, sekvens og symbol.

1 Telle 1 – 2 – 3

Målet med øvelsen;

er å lære elevene at symboler brukes som kommandoer når man programmerer og at det er noe som heter «løkker». Løkker er en ting som gjentar seg ved START og stopper ikke før riktig kommando blir gitt. I dette tilfellet STOPP.

I tillegg kan man lære litt om variabler ved at tallet tre kan bli byttet ut med enten knips, plystre eller snu seg rundt.

Utstyr:



Tid: 5 - 10 min

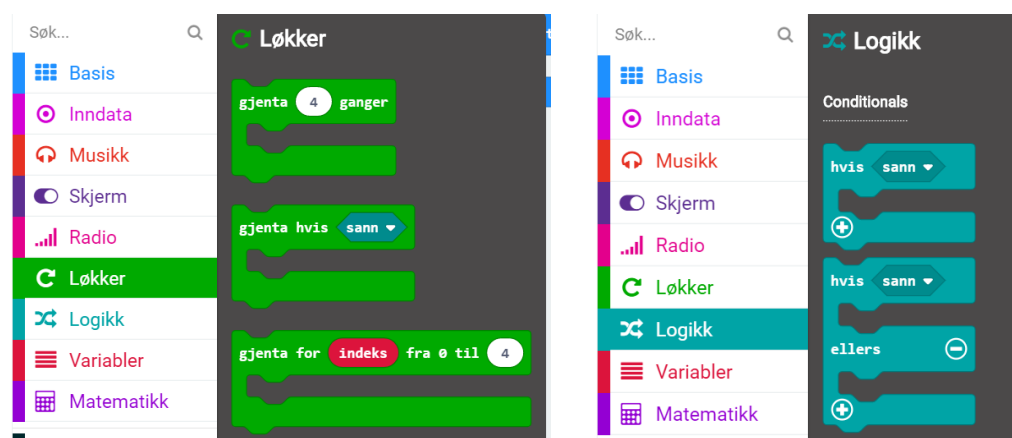
Gjennomføring:

To og to elever går sammen. De skal telle til tre annenhver gang. (Nr 1 sier 1, nr 2 sier 2, nr 1 sier 3, nr 2 sier 1 osv) Læreren starter og stopper hver enkelt øvelse ved å si et klart og tydelig START og STOPP.

Etter at elevene har fått til og forstått den første aktiviteten, stoppes den og elevene får beskjed om at nå skal man klappe istedenfor å si 1. Etterhvert byttes 2 ut med tramp og til slutt skal 3 byttes ut med knips. Hvis det viser seg at elevene ikke kan knipse, så kan de plystre, hvis det ikke kan plystre, så kan de heller snurre rundt en gang.

Samtale i etterkant:

Her kan man snakke om at i programmering er det noe som heter «løkker». Det er noe skjer fra man starter programmet til man stopper programmet. Her startes det ved kommandoen START, og avsluttes når man hører STOPP. Man gjør det samme hele tiden i en løkke. I tillegg finnes det noe innen programmering som heter logikk. Her kan man utføre forskjellige ting avhengig av hvilke betingelser som er lagt inn i programmet.



2 Tegn etter instruksjoner

Målet med øvelsen:

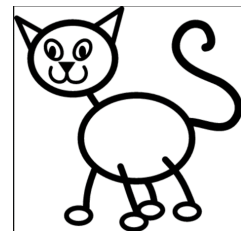
Lære at det er veldig viktig å gi og ta imot korte presise forklaringer. **Det er også viktig å vite at denne øvelsen er lagt opp til litt «tvetydige» instruksjoner med vilje.** Det er for at elevene skal kunne bli i tvil om hva instruksjonen egentlig er og hva de egentlig skal tegne. Slik at de selv opplever at det er viktig med korte presise instruksjoner.

Elevene kan trenge ganske mye veiledning for å komme i gang med denne øvelsen. Dette er også en øvelse som dere gjerne kan gjenta flere ganger. På den måten utvikler elevene ferdigheten i å gi presise instruksjoner

Utstyr:

Papir og blyant
Tegneinstruksjoner

Tid: 20 - 25 min



Gjennomføring:

Hele klassen skal være «datamaskiner» og tegne etter lærerens instruksjoner. Læreren eller en elev leser opp instruksjonen under, og alle skal tegne etter den. Oppleseren kan ikke forklare instruksjonen på en ny måte, men han/hun kan gjenta den hvis det var noen som ikke fikk den med seg eller trengte å høre den på nytt. NB! Det kan være lurt å gi beskjed om at elevene ikke får lov til å tegne før hele punktet er lest opp.

Alternative øvelser:

- Elevene lager sine egne tegninger med instruksjoner og prøver ut på hverandre
- Figurer med bruk av passer, linjal og gradskive kan også lages

Instruksjoner:

1. Tegn en vannrett strek litt nede på arket.
2. Tegn en lang og en kort, loddrett strek til venstre; de skal røre den vannrette streken på tuppen.
3. På den lange streken, begynn øverst og tegn en pilspiss. Gjør det samme på den korte streken. Pilspissene skal peke oppover.
4. Tegn en ny pilspiss litt lengre ned på den lange streken. Gjør det samme på den korte streken.
5. Gjenta pilspissen på begge strekene to ganger til.
6. Gjenta en gang til på den lange streken.

7. Tegn et trapes ved siden av den korte streken. Bunnen av trapeset er på den vannrette streken.
8. Lag en likebeint trekant på høyre side av trapeset. Trekanten skal være inni trapeset.
9. Ved siden av trapeset: tegn fem sirkler som ligger på streken.
10. Tegn tre dråper over de frem sirklene. De tre dråpene skal ha ulik størrelse og røre sirklene.
11. Tegn en loddrett strek inne i trekanten
12. Over trapeset: Tegn en motsatt C
13. Start i enden av C-en og tegn en større motsatt C, som ender opp i den andre enden av den første.

Tegningene er ferdig 😊



Samtale i etterkant:

Her kan man snakke om hvor nøye man må være når man instruerer/programmerer en datamaskin. Det er viktig å tenke på å gi instruksjonene i riktig rekkefølge. De må være presise og inneholde alt som datamaskinen trenger å vite for å utføre sitt oppdrag.

I tillegg kan man også samtale om at det er forskjeller mellom mennesker og datamaskiner. Mennesker tolker ting forskjellig og at din kamerat ikke tolker din instruksjon slik som du hadde tenkt deg.

3 Kompisprogrammering

Målet med øvelsen:

Her skal elevene lage en algoritme med hjelp av piler. Oppgaven er kompleks og må deles opp i mindre ledd/sekvenser for å løse et problem.

Utstyr: 11 stk A4-ark med sifferne/tallene fra 0 til 10
8 stk programmeringskort
En liten «programmeringsbane»
En spillebrikke

Tid: 35 – 45 min

Gjennomføring:

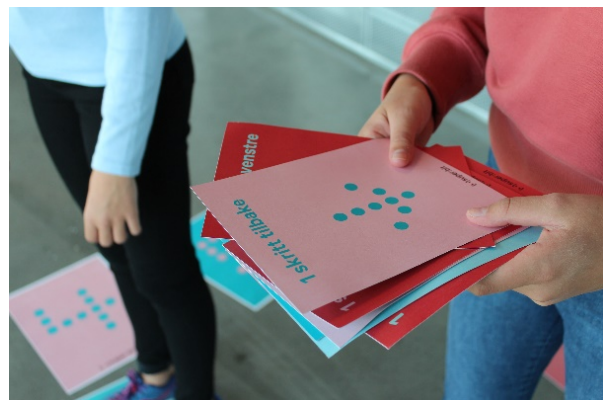
1. Del elevene inn i grupper (3-5 i hver gruppe.) Alle elevene skal programmere, men en av elevene er utstedt til å være robot.
2. Bygg opp banen med de nummererte kortene på gulvet slik bildet viser.
3. Gruppene skal også ha hver sin bunke med programmeringskort, spillebrikke og den lille «programmeringsbanen» på et eget A4 ark.
4. Gruppene skal sammen prøve å lage en programmeringskode og programmere en av elevene i gruppen til å gå fra 0 til 10. De skal sortere kommandokortene i den rekkefølgen de vil ha dem for så å kunne gi instruksjonen til sin robot. De får kun bruke hvert kort én gang.
5. Hver gruppe får planlegge, teste og lage sitt program, en sekvens av kommandoer, på den lille «programmeringsbanen» før det skal vises frem for klassen. Alle elevene «kjører» sine roboter. De som har programmert skal lese opp sekvensen og roboten skal følge instruksjonene. Start på 0.
6. Alle gruppene får kjøre sin robot for klassen.
7. Klarte alle å løse oppgaven? Fant de forskjellige løsninger eller hadde alle samme løsning?

Regler:

1. «Roboten»/eleven skal ta seg fra posisjon «0» til sluttposisjon «10»
2. Hvert programmeringskort er kun lov å bruke en gang.
3. Man trenger ikke bruke alle programmeringskortene.

Programmeringskort:

- Gå ett skritt fremover
- Gå to skritt fremover
- Gå tre skritt fremover
- Gå fire skritt fremover
- Gå ett skritt bakover
- Snu 90 grader til venstre
- Snu 90 grader til høyre
- Snu helt om 180 grader



Øke vanskelighetsgraden:

Ta bort programmeringskortet «Gå et skritt frem» og la gruppene prøve å gjøre oppgaven på nytt.

Kilder:

<https://www.kodboken.se/start/kom-igang/lekar-och-ovningar/robotkompis-med-uppdrag>

Begrepsforklaringer

Algoritme: Med begrepet algoritme mener vi en oppskrift som forteller oss trinnvis hva som skal gjøres for at noe skal bli fullført - hva som må gjøres for å nå målet. Utfordringen med algoritmer er at de må skrives og følges nøyaktig - og i riktig rekkefølge - for at vi skal få det resultatet vi forventer. Å skrive i riktig rekkefølge er derfor helt essensielt, i tillegg til at man må ta med alle trinnene. En datamaskin følger de instruksjoner den får i den rekkefølge de gis. Et dataprogram er en algoritme, og et dataprogram består ofte av mange algoritmer, men ikke alle algoritmer er dataprogram.

Kode: En samling instruksjoner som beskriver en oppgave. Koden kan være et program eller en del av et program.

Kommando: Instruksjon til en datamaskin for å utføre **en** oppgave.

Logikk: betegnelse for foreteelser som er underlagt en viss orden i et sluttet system.

Løkker: (eng: loop) er en datastruktur som brukes veldig mye i programmering. Når vi programmerer, så ønsker vi ofte å gjenta identiske instruksjoner i programmet vårt, uten å måtte skrive mye kode om igjen. Da bruker vi løkker istedenfor, og spesifiserer hvor mange ganger instruksjonene skal gjentas. Spesifikasjonen kan være et gitt antall ganger (for-løkker), eller til en betingelse er oppfylt (while-løkker).

Problemløsning: Å løse en sammensatt oppgave som kanskje krever at man må identifisere flere enkeltoppgaver som må løses før man kan finne løsningen på hele oppgaven. Problemløsning krever gjerne en algoritmisk tilnærming, der man utvikler en helt egen løsningsmetode for problemet.

Program: Et program er en oppskrift som sier hva datamaskinen skal gjøre. Programmer skrives i et programmeringsspråk, for eksempel Scratch, Python eller Java, og gir en rekke instruksjoner som forteller hva en datamaskin skal gjøre. Instruksjonene er et sett bestemte kommandoer som settes sammen for å utføre en oppgave eller løse et problem. Et program må skrives i et programmeringsspråk.

Programmering: Programmering er å lage et program for datamaskinen. Begrepet programmering kan også omfatte prosessen med å strukturere oppgaven som skal løses, og dele den opp i mindre biter som til slutt kan løses ved hjelp av de funksjonene som finnes i et programmeringsspråk. Selve prosessen med å skrive programmet kalles ofte for *koding*. Det handler om å lage programkode, det vil si et sett med regler og uttrykk for å styre digitale enheter.

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

Symbol: Et symbol er en abstraksjon. Bak symbolet skjuler det seg gjerne en rekke instruksjoner. F.eks. kan et bilde av en hånd bety at man skal klappe i hendene.