



Samling 1

Programmering

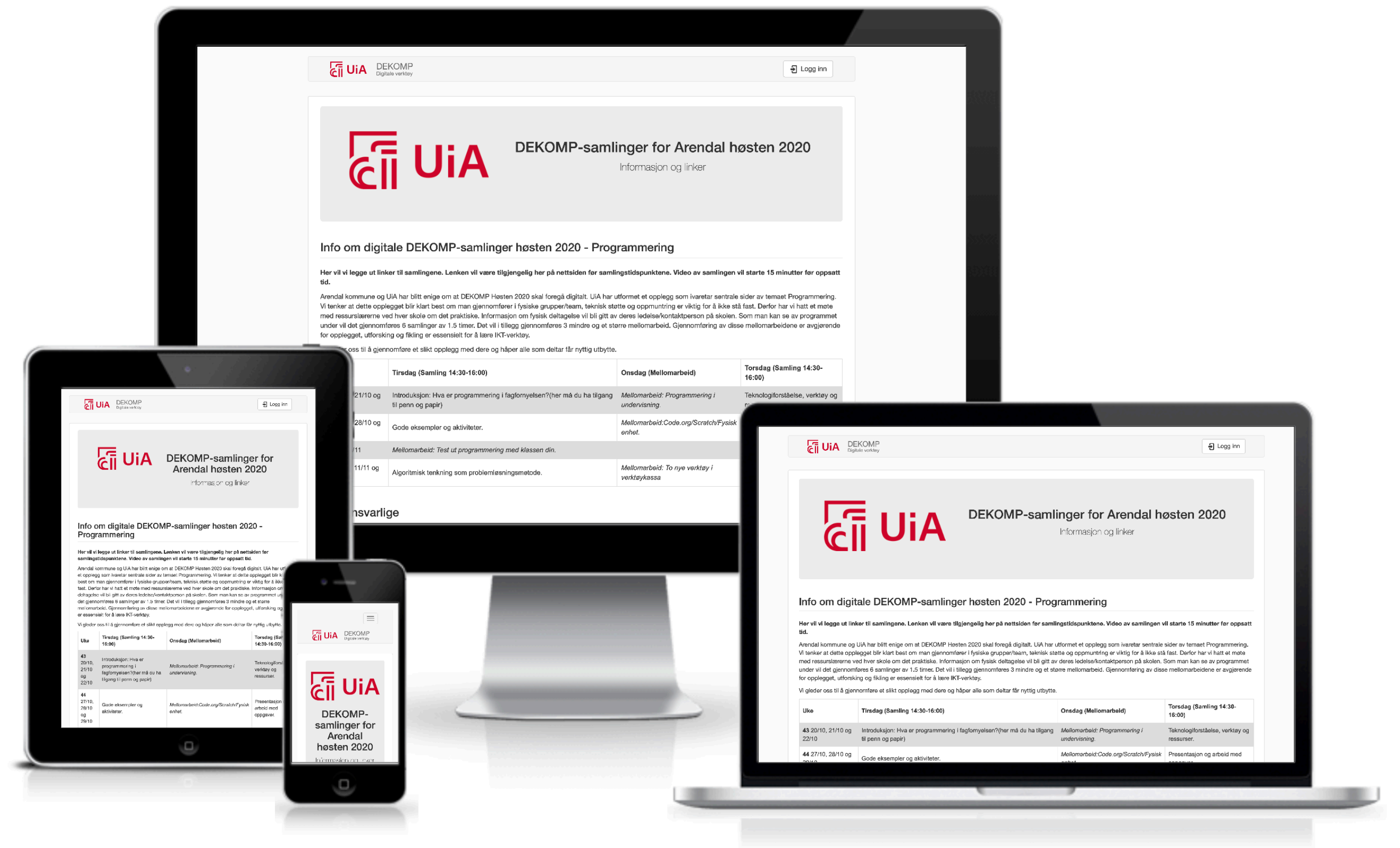
Desentralisert lærerutdanning
Arendal

Universitetet i Agder

Hva skal skje i perioden

Infoside:

bit.ly/arendaldekomp



Oversikt over innholdet

- Uke 43: Innføring til programmering, «Analog koding», tankesettet
- Uke 44: «Digital koding»
- Uke 45: Mellomarbeidsuke, Test ut med klassen din
- Uke 46: Digital kompetanse, inkorporere tankesettet, teknologiforståelse

Første uke (43)

Uke 43: Innføring i programmering, «Analog koding»; tankesettet

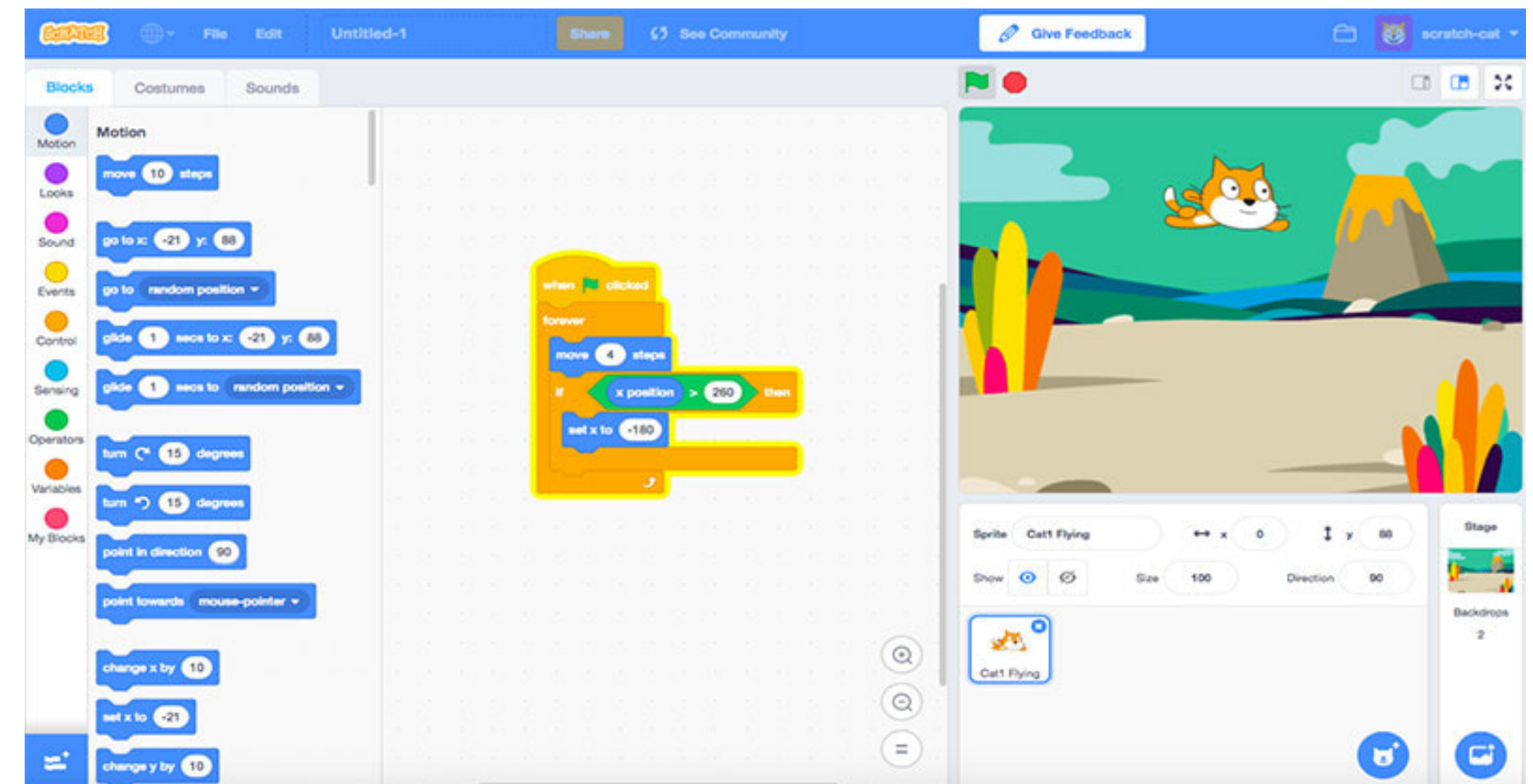
- Tirsdag 20. oktober – Kickoff
- Onsdag 21. oktober – Mellomarbeid
- Torsdag 22. oktober – Praktisk samling



Andre uke (44)

Uke 44: «Digital koding»

- Tirsdag 27. oktober –
Innføring i «digital koding»
- Onsdag 28. oktober –
Mellomarbeid
- Torsdag 29. oktober –
Praktisk samling



Tredje uke (45)

Uke 45: Mellomarbeidsuke, Test ut med klassen din

- Dokumenter arbeidet ditt underveis



Fjerde uke (46)

Uke 46: Digital kompetanse, inkorporere tankesettet, teknologiforståelse

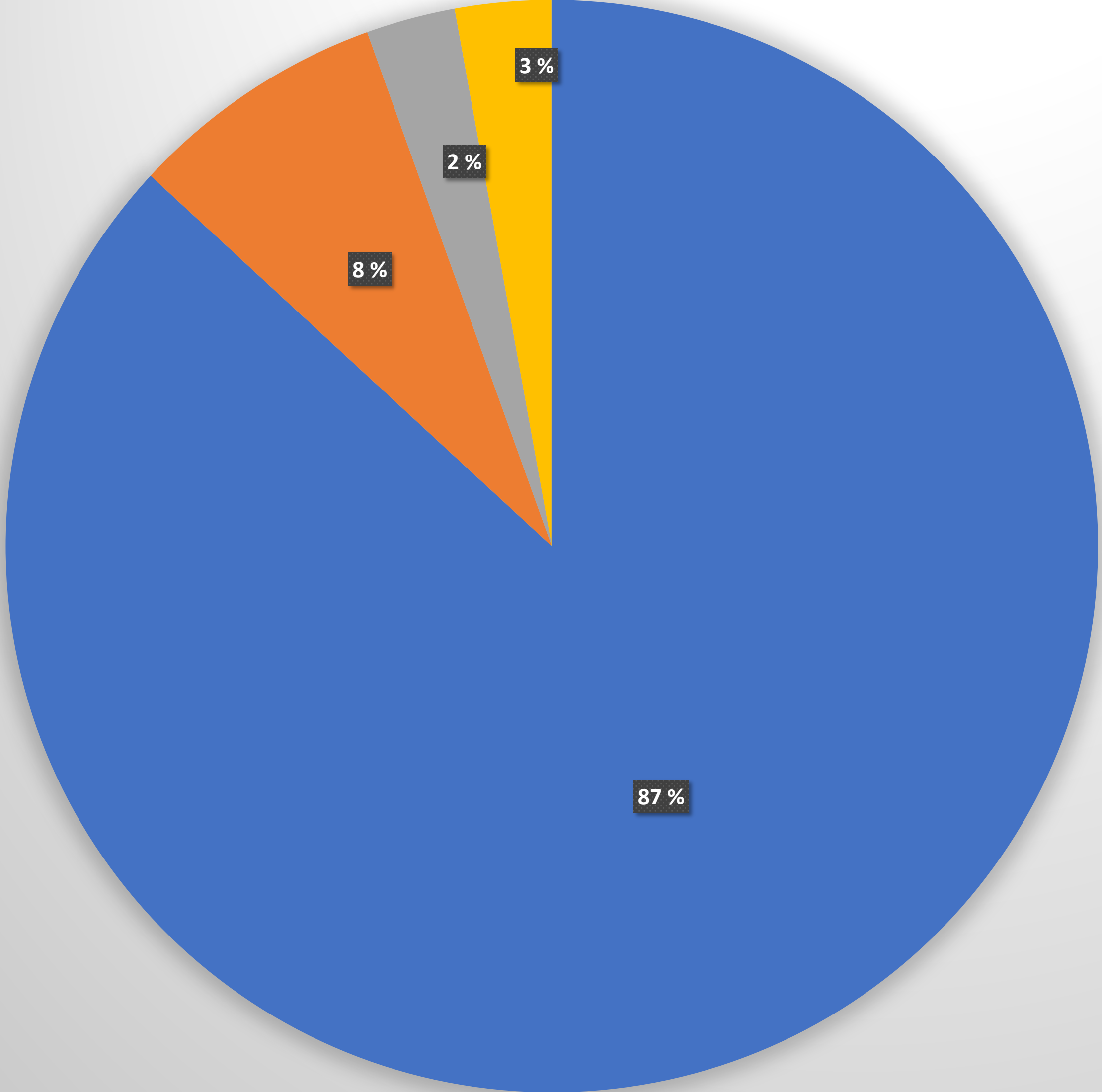
- Tirsdag 10. november – Algoritmisk tenkning som problemløsningsmetode
- Onsdag 11. november – Mellomarbeid
- Torsdag 12. november – IKT i fagfornyelsen



Hva skal skje i samlingen

1. Om evaluering
2. Fang grisen - aktivitet
3. Om temaet
4. Algoritmisk tenkning i fagfornyelsen
5. Pause
6. Om programmering i fagfornyelsen
7. Film om analog koding
8. Presentasjon av analog aktivitet
9. Aktivitet: Presise verbale instruksjoner
10. Info om mellomarbeid
11. Undersøkelse

Arendal: Onedrive og Padlet



- Ja, dette kan jeg
- Nei, men dette finner jeg ut av selv
- Nei, det trenger jeg en digital ressurs til
- Nei, det trenger jeg å prate med en ressursperson om

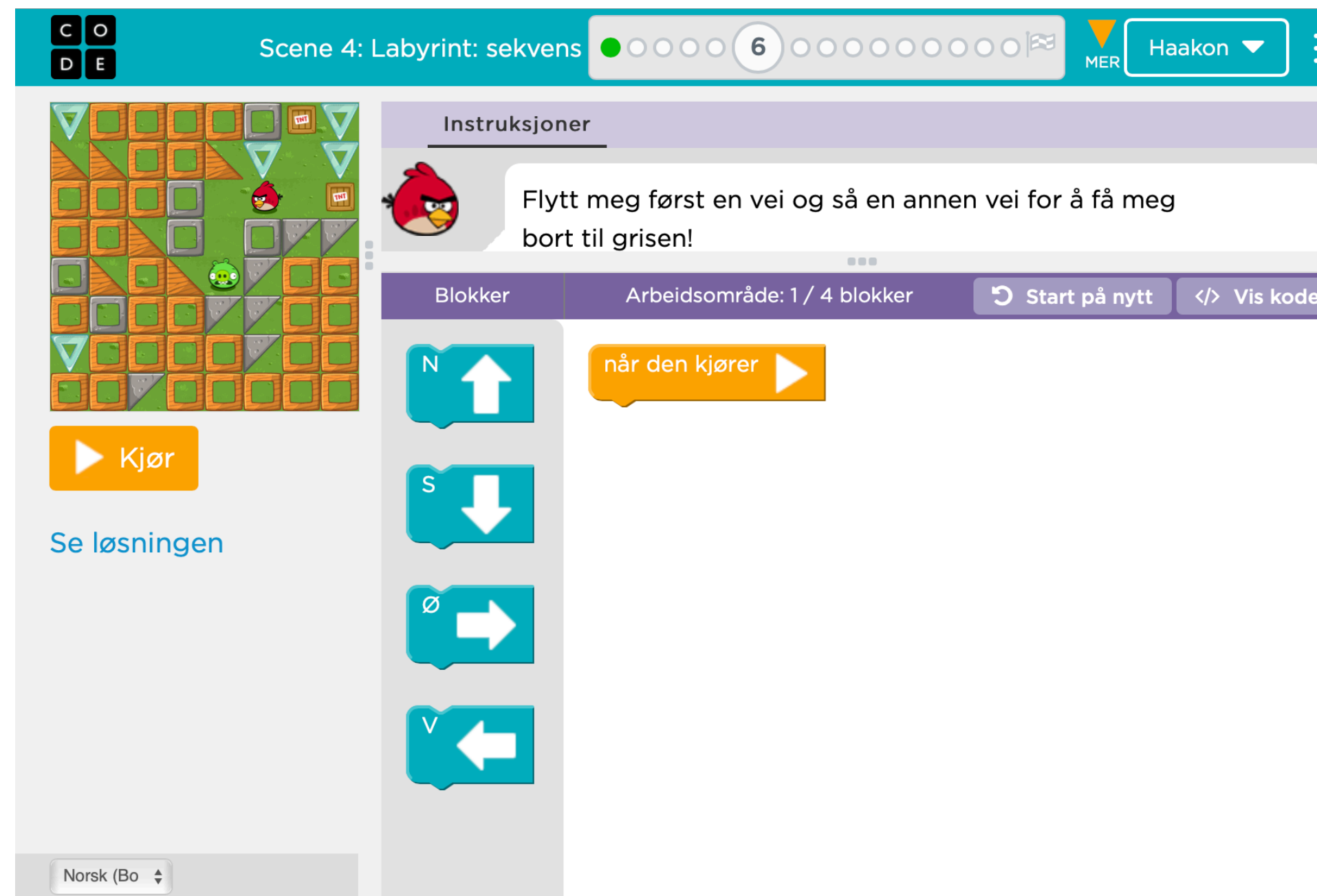
Fra tekstsvar:

- Konkret, brukervennlig og nyttig gjennomgang. Applaus!!
- Litt for lang økt, med litt for lite innhold.
- Det gikk litt fort.
- Vi var flere som diskuterte og savnet konkrete eksempler på hva vi kan bruke det til i undervisning.
- Godt forklart, samtidig som man fikk drodle litt på egenhånd.
- Kunne godt vært "hjemmelekse" også



Klarer du å ta grisen?

Åpne nettadressen ->
bit.ly/tagrisen



Om temaet

Innledning - hvorfor
temaet/programmering/teknologiforståelse

Dette har vi holdt på med frem til nå



Padlet

Office 365

Skjermopptak

Programmering



Regjeringen vil:

- Gi skolen et digitalt løft gjennom (...) vekt på utvikling av teknologiforståelse, digital dømmekraft og skaperkraft.
- Sikre at elever får kunnskap om og forståelse for teknologi, algoritmisk tenkning og koding i flere fag, allerede tidlig i skoleløpet.
- Gi innovasjon og skaperkraft en god plass i fremtidens skole.

“Everybody in this country should learn how to program a computer... because it teaches you how to think.”

- Steve Jobs

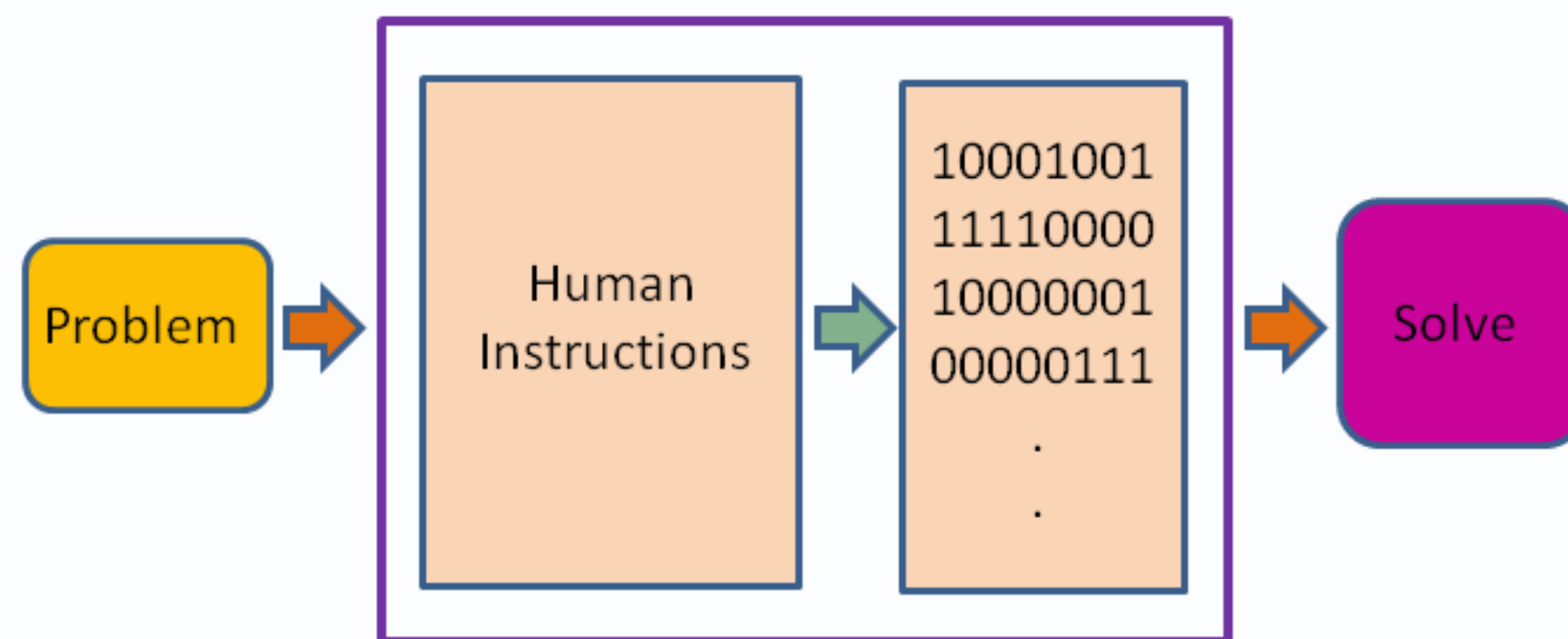
Hva er programmering



Udirs definisjon:

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. Programmering omfatter også arbeidet med å teste programmet, finne feil og rette dem. Programmering er vanligvis et ledd i en større prosess som kan kalles *systemutvikling*. Selve prosessen med å skrive programmet kalles i IT-sjargong ofte for *koding*. Det inneholder kartlegging av brukerens behov og design av en overordnet struktur for det som skal lages. Det handler om å lage programkode, det vil si et sett med regler og uttrykk for å styre digitale enheter. I dette inngår prosessen fra å identifisere problemer og utforme mulige løsninger, til å lage kode som kan forstås av en datamaskin, systematisk feilsøke og forbedre denne koden, og dokumentere løsningen på en forståelig måte. Det omfatter alle nivåer fra å forutse og analysere hva et program skal gjøre, til å kjenne igjen mønstre, eksperimentere og evaluere mulige løsninger, og samarbeide med andre.

"Med programmering gjelder det å kunne formulere et problem og finne løsningen ved hjelp av et dataprogram."



Hentet fra:

<https://www.uv.uio.no/forskning/satsinger/fiks/om/aktuelt/aktuelle-saker/2018/programmering-er-mer-enn-koding%21.html>

**Er koding og programmering
det samme?**

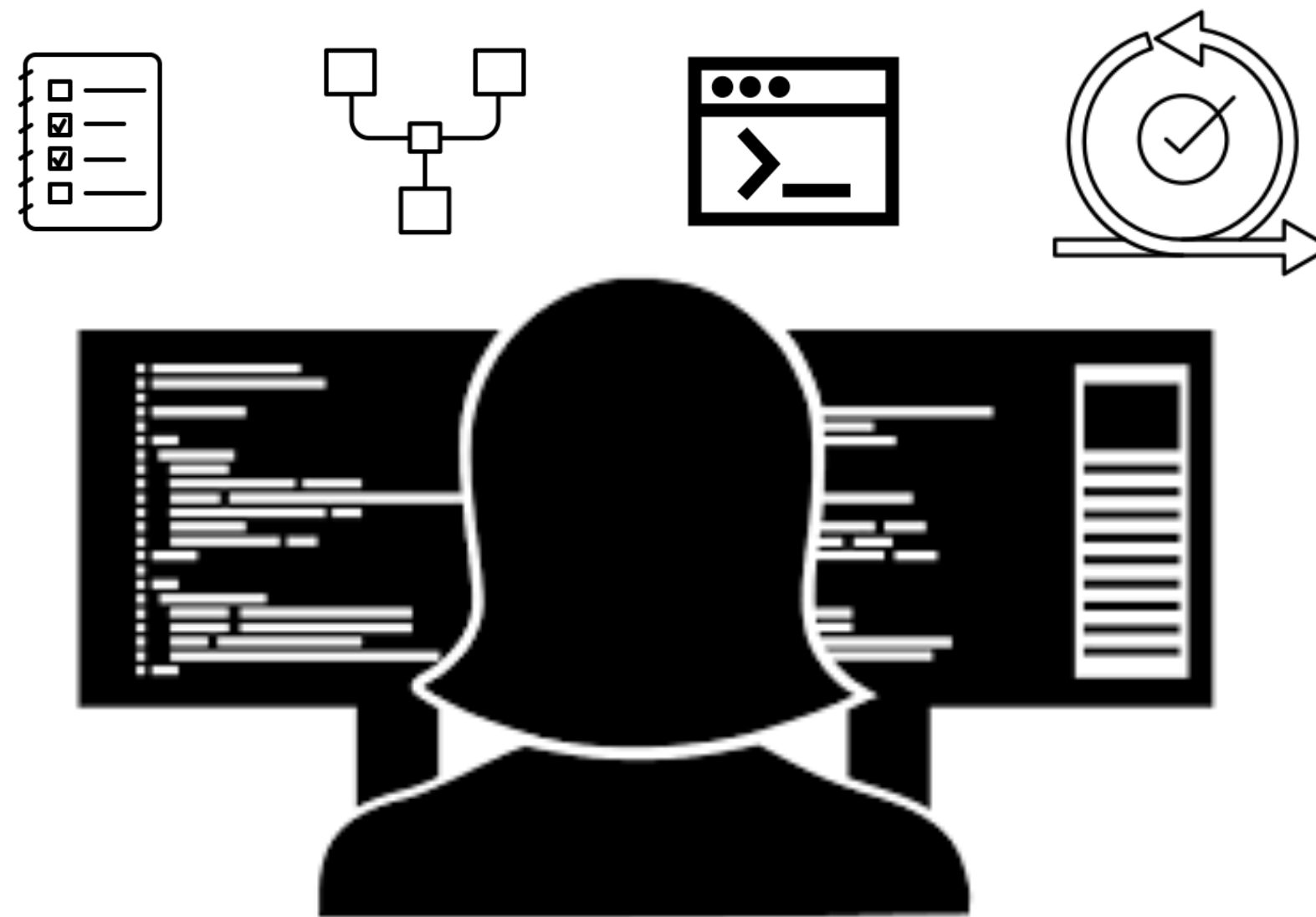
"Å lære å kode handler om å lære seg det språket som skal til for å kommunisere med datamaskinen."



Hentet fra:

<https://www.uv.uio.no/forskning/satsinger/fiks/om/aktuelt/aktuelle-saker/2018/programmering-er-mer-enn-koding%21.html>

"Programmering er derimot noe langt mer. Det involverer blant annet også tankeprosessene som går forut for kodingen."



Hentet fra:

<https://www.uv.uio.no/forskning/satsinger/fiks/om/aktuelt/aktuelle-saker/2018/programmering-er-mer-enn-koding%21.html>

Algoritmisk tenkning i fagfornyelsen

Den algoritmiske tenkeren

Nøkkelbegrep

1 **Logikk**
Analysere og forutse

2 **Algoritmer**
Regler og steg-for-steg

3 **Dekomposisjon**
Bryte ned i mindre deler

4 **Mønstre**
Finne og bruke likheter

5 **Abstraksjon**
Fjerne unødvendige detaljer

6 **Evaluering**
Gjøre vurderinger

Arbeidsmåter

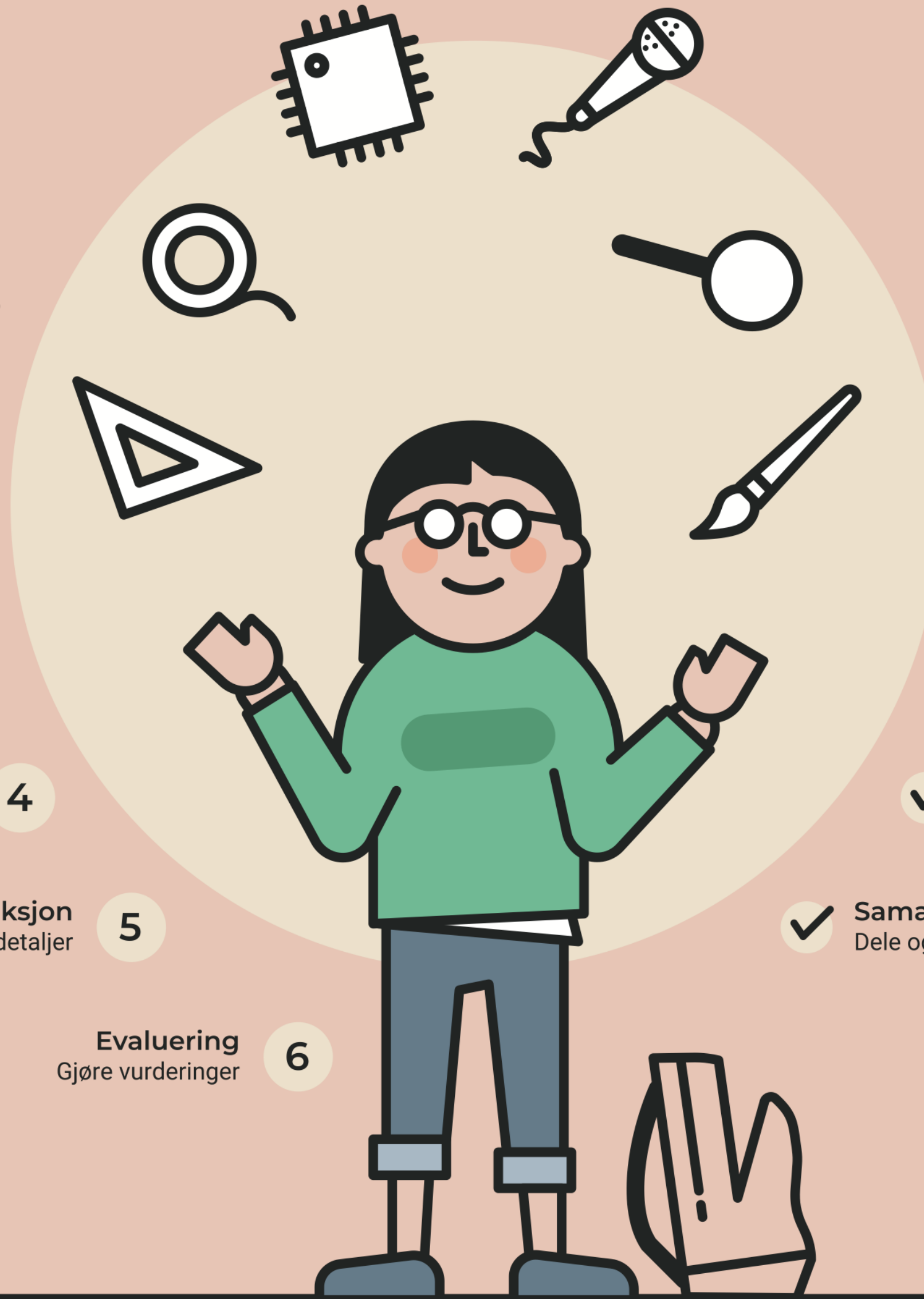
✓ **Fikle**
Utforske og eksperimentere

✓ **Skape**
Designe og lage

✓ **Feilsøke**
Oppdage og rette feil

✓ **Holde ut**
Fortsette og prøve igjen

✓ **Samarbeide**
Dele og jobbe sammen

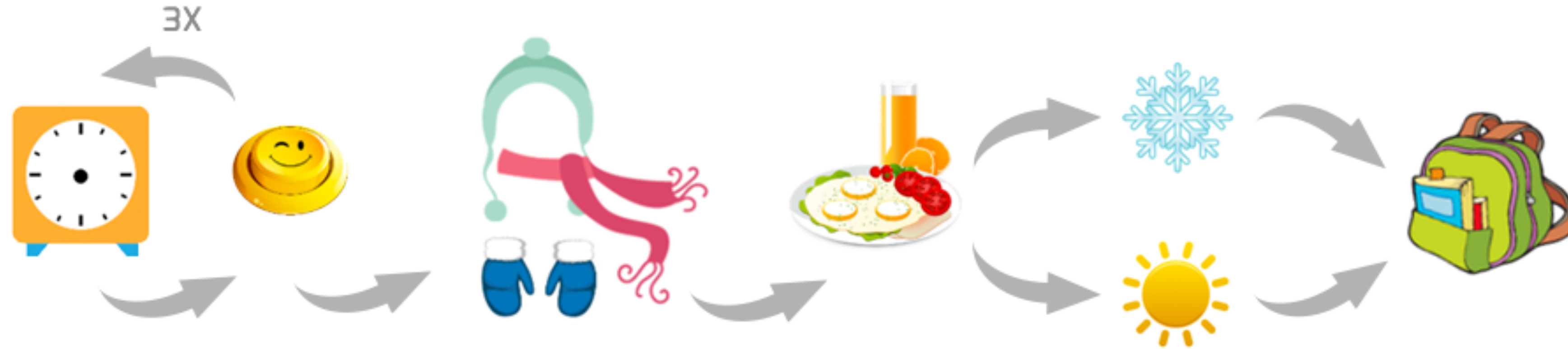


Hva er en algoritme?

Udirs definisjon:

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.

Et eksempel



```
when Alarm Rings occurs
repeat 3
  Hit Snooze Button
  wait until Alarm Rings
Turn Off Alarm
Get Up
Shower
Get Dressed
Eat breakfast
if weather = sunny then
  Walk to School
else
  Ride to School with Parent
```

Bilde hentet fra: <https://www.tynker.com/blog/articles/ideas-and-tips/how-to-explain-algorithms-to-kids/>

Den algoritmiske tenkeren

Nøkkelbegrep

1 **Logikk**
Analysere og forutse

2 **Algoritmer**
Regler og steg-for-steg

3 **Dekomposisjon**
Bryte ned i mindre deler

4 **Mønstre**
Finne og bruke likheter

5 **Abstraksjon**
Fjerne unødvendige detaljer

6 **Evaluering**
Gjøre vurderinger

Arbeidsmåter

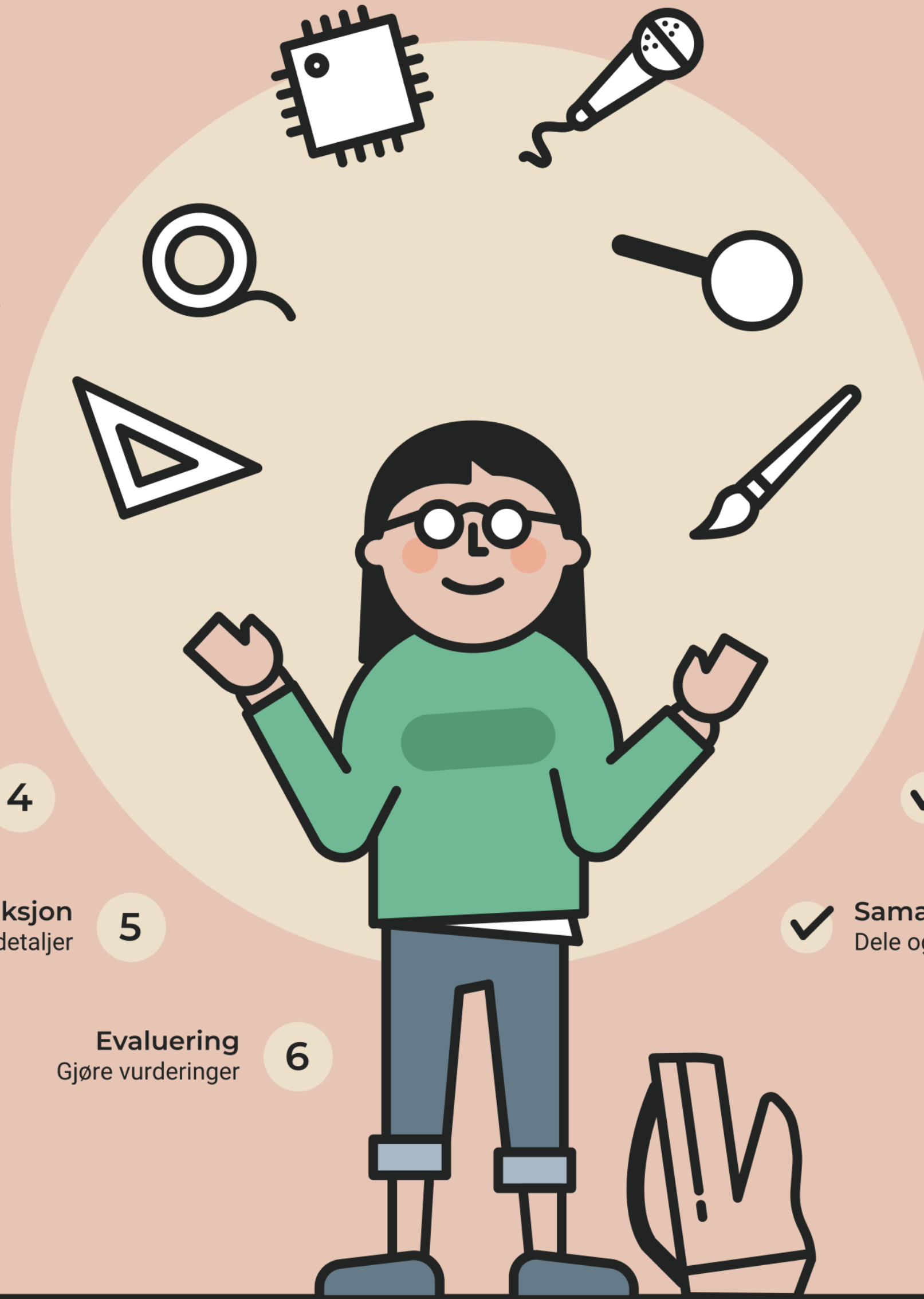
✓ **Fikle**
Utforske og eksperimentere

✓ **Skape**
Designe og lage

✓ **Feilsøke**
Oppdage og rette feil

✓ **Holde ut**
Fortsette og prøve igjen

✓ **Samarbeide**
Dele og jobbe sammen



Algoritmisk tenkning

- Problemløsningsmetode
- Systematisk
- Vurdering: teknologi vs. menneske
- *Algoritmisk tenkning er den norske oversettelsen av det engelske ordet computational thinking.*

Pause

Om programmering i fagfornyelsen

Eksempler fra planverket hvor digitalisering blir konkret satt opp som kompetanse

Musikk

Kompetansemål etter 10. trinn

- skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder

Kompetansemål etter 7. trinn

- bruke teknologi og digitale verktøy til å skape, øve inn og bearbeide musikk

Kompetansemål etter 4. trinn

- eksperimentere med rytmer, melodier og andre grunnelementer, sette sammen mønstre til komposisjoner, også ved bruk av digitale verktøy, og beskrive arbeidsprosesser og resultater

Kompetansemål etter 2. trinn

- leke med musikkens grunnelementer gjennom lyd og stemme, lage mønstre og sette sammen mønstrene til enkle improvisasjoner og komposisjoner, også med digitale verktøy

Kunst og håndverk

Kompetansemål etter 10. trinn

- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter

- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Kompetansemål etter 7. trinn

- bruke digitale verktøy til å planlegge og presentere prosesser og produkter

- analysere visuelle virkemidler i ulike medier og belyse et aktuelt tema gjennom foto eller infografikk

- bruke programmering til å skape interaktivitet og visuelle uttrykk

Kompetansemål etter 4. trinn

- bruke enkle komposisjonsprinsipper i fotografi og digitale verktøy

Naturfag

Kompetansemål etter 10. trinn

- bruke programmering til å utforske naturfaglige fenomener

- utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker

- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger

- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger

Kompetansemål etter 7. trinn

- designe og lage et produkt basert på brukerbehov

- utforske, lage og programmere teknologiske systemer som består av deler som virker sammen

Kompetansemål etter 4. trinn

- utforske teknologiske systemer som er satt sammen av ulike deler, og beskrive hvordan delene fungerer og virker sammen

- designe og lage et produkt basert på en kravspesifikasjon

Kompetansemål etter 2. trinn

- presentere funnene sine og beskrive hvordan eleven har kommet fram til dem

- presentere egne ideer til teknologiske oppfinnelser

Norsk

Kompetansemål etter 10. trinn

- utforske og vurdere hvordan digitale medier påvirker og endrer språk og kommunikasjon

- bruke kilder på en kritisk måte, markere sitater og vise til kilder på en etterrettelig måte i egne tekster

Kompetansemål etter 7. trinn

- orientere seg i faglige kilder på bibliotek og digitalt, vurdere hvor pålitelige kildene er, og vise til kilder i egne tekster

- reflektere etisk over hvordan eleven framstiller seg selv og andre i digitale medier

- presentere faglige emner muntlig med og uten digitale ressurser

- utforske og beskrive samspillet mellom skrift, bilder og andre uttrykksformer og lage egne sammensatte tekster

Kompetansemål etter 4. trinn

- holde muntlige presentasjoner med og uten digitale ressurser

Kompetansemål etter 2. trinn

- lese med sammenheng og forståelse på papir og digitalt og bruke enkle strategier for leseforståelse

- lage tekster som kombinerer skrift med bilder

Samfunnsfag

Kompetansemål etter 10. trinn

- bruke samfunnsfaglege metodar og digitale ressursar i egne undersøkingar, presentere funn ved bruk av digitale verktøy og drøfte kor gyldige og relevante funna er

- vurdere på kva måtar ulike kjelder gir informasjon om eit samfunnsfagleg tema, og reflektere over korleis algoritmar, einsretta kjelder eller mangel på kjelder kan prege forståinga vår

- utforske korleis teknologi har vore og framleis er ein endringsfaktor, og drøfte innverknaden teknologien har hatt og har på enkeltmenneske, samfunn og natur

- utforske og reflektere over eigne digitale spor og høvet til å få sletta spora og å verne om retten ein sjølv og andre har til privatliv, personvern og opphavsrett

- utforske ulike plattformer for digital samhandling og reflektere over korleis digital deltaking og samhandling påverkar forma på og innhaldet i samfunnsdebatten

Kompetansemål etter 7. trinn

- gjennomføre ei samfunnsfagleg undersøking og presentere resultata ved hjelp av eigna digitale verktøy

- presentere ei aktuell nyheitssak og reflektere over forskjellar mellom fakta, meiningar og kommersiell bodskap i mediebialetet

- utforske korleis menneske i fortida livnærte seg, og samtale om korleis sentrale endringar i livsgrunnlag og teknologi har påverka og påverkar demografi, levekår og busetjingsmønster

- reflektere over korleis ein sjølv og andre deltek i digital samhandling, og drøfte kva det vil seie å bruke dømmekraft sett i lys av reglar, normer og grenser

Kompetansemål etter 4. trinn

- samtale om reglar og normer for personvern, deling og beskyttelse av informasjon og om kva det vil seie å bruke dømmekraft i digital samhandling

- utforske og presentere samfunnsfaglege spørsmål, søkje etter informasjon i ulike kjelder og vurdere kor nyttig informasjonen er til å belyse spørsmåla

Kompetansemål etter 2. trinn

- samtale om moglegheiter og utfordringar ved digital samhandling

Matte

Kompetansemål etter 10. trinn

- utforske og samanlikne eigenskapar ved ulike funksjonar ved å bruke digitale verktøy

- bruke funksjonar i modellering og argumentere for framgangsmåtar og resultat

- modellere situasjonar knytte til reelle datasett, presentere resultata og argumentere for at modellane er gyldige

- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Kompetansemål etter 9. trinn

- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe ved å bruke programmering

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

Kompetansemål etter 8. trinn

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

Kompetansemål etter 7. trinn

- bruke programmering til å utforske data i tabellar og datasett

- lage og vurdere budsjett og rekneskap ved å bruke rekneark med cellereferansar og formlar

Kompetansemål etter 6. trinn

Kompetansemål etter 5. trinn

- lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer

Kompetansemål etter 4. trinn

- lage algoritmar og uttrykkje dei ved bruk av variablar, vilkår og lykkjer

Kompetansemål etter 3. trinn

- lage og følgje reglar og trinnvise instruksjonar i leik og spel knytte til koordinatsystemet

Kompetansemål etter 2. trinn

- lage og følgje reglar og trinnvise instruksjonar i leik og spel

Eksempler fra planverket hvor programmering blir konkret satt opp som kompetansemål

Musikk

Kompetansemål etter 10. trinn

- skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder

Kunst og håndverk

Kompetansemål etter 7. trinn

- bruke programmering til å skape interaktivitet og visuelle uttrykk

Naturfag

Kompetansemål etter 10. trinn

- bruke programmering til å utforske naturfaglige fenomener

Kompetansemål etter 7. trinn

- utforske, lage og programmere teknologiske systemer som består av deler som virker sammen

Matte

Kompetansemål etter 10. trinn

- utforske matematiske egenskaper og sammenhenger ved å bruke programmering

Kompetansemål etter 9. trinn

- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe ved å bruke programmering

Kompetansemål etter 8. trinn

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

Kompetansemål etter 7. trinn

- bruke programmering til å utforske data i tabellar og datasett

Kompetansemål etter 6. trinn

- bruke variablar, lykkjer, vilkår og funksjonar i programmering til å utforske geometriske figurar og mønster

Kompetansemål etter 5. trinn

- lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer

Eksempler på programmering i den nye læreplanen

Robot: algoritmisk tenkning på barnetrinnet

Her skal barna lære å:

- omgjøre virkelige handlinger til instruksjoner
- få praktiske erfaringer i å kode med symboler
- få erfaringer med behov for presisjon innen koding



Algoritmisk tenkning og koding motiverer til skriving

Oppgaven passer til:

Fag: Norsk

Trinn: 2. trinn

Tema: Fortelling

Mål for kodingen:

- kunne forstå retning, rekkefølger og algoritmer.
- kunne kode en figur.

Kompetansemål for norsk:

- Bruke bokstaver og eksperimenter med ord på tastatur.
- Uttrykke egne tekstopplevelser gjennom ord, tegninger, bilder, musikk og bevegelse.



IKT i praksis
Deling av digital pedagogisk praksis

Finns ressurser Læringsopplegg Blogg Apper og programmer Min side Logg inn

Du er her: [Hjem](#) > [Læringsopplegg](#) > Koding motiverer til skriving

Koding motiverer til skriving

Forfatter: [Linda Røberg](#) Sist endret: 16/08/2017 - 12:53
Organisasjon: [Alstad Barneskole](#) [Vis mer](#)

Kan man kode i faget norsk? Gjennom appen ScratchJr kan de yngste elevene programmere fortellinger på nettbrettet, til stor iver og glede.



ScratchJr

Innledning
Dette opplegget er knyttet til faget norsk på 2. trinn. Vi har brukt Appen ScratchJr som er en app som passer for de yngste elevene. Her kan elevene lage sine egne historier. Elevene velger bakgrunner, figurer, lyd og tekst. Ved hjelp av koding kan elevene bestemme hvordan figurene skal forflytte seg. Historien blir levende.

Slik gjør du
For elevene satte i gang med appen, arbeidet vi med algoritmer og

Lag et læringsopplegg
Har du gjennomført et læringsopplegg i skole eller barnehage som andre pedagoger kan ha interesse av? Del det gjerne på denne siden.

Del på facebook
Del på Twitter
Skriv ut

Relaterte opplegg

- [Programmering med Bee-Bot på 2. trinn: Del 2](#)
Kan elever på småskoletrinnet programmere? Ja det kan de! I dette innlegget får du et lite innblikk...
- [Lær koordinater med Scratch!](#)
I dette opplegget skal elevene lage et spill i Scratch der de utforsker koordinater.
- [Programmering med Bee-Bot på 2. trinn: Del 1](#)
Kan elever på småskoletrinnet programmere? Ja det kan de, og i dette innlegget får du et lite...



Eksempel på programmering i matematikk

- lage og følge regler og trinnvise instruksjonar i leik og spel (knytt til koordinatsystemet)
2. trinn (3. trinn)
- lage algoritmar med bruk av variablar, vilkår og lykkjer og programmere desse
5. trinn
- utforske korleis algoritmar kan skapast, testast og forbeترast ved hjelp av programmering
8. trinn
- bruke programmering til å utforske matematiske eigenskapar og samanhengar
9. trinn



<https://studio.code.org/s/artist/stage/1/puzzle/10>



Exact Instructions Challenge - THIS is why my kids hate me. | Josh Darnit:
https://www.youtube.com/watch?v=cDA3_5982h8

**Presentasjon av presise
verbale instruksjoner - aktivitet**

Presise verbale instruksjoner

- I denne aktiviteten skal du være datamaskinen. En datamaskin må ha presise instruksjoner. Rekkefølgen for instruksjonene er naturligvis avgjørende. Hva tror du skjer når instruksjonene ikke er presise nok?
- Nå skal du finne frem et ark og en blyant/penn og følge de verbale instruksjonene.



Lytt og tegn

- 2 min: Leser opp presise instruksjoner
- 1.Tegn en vannrett strek, litt nede på arket.
- 2.Tegn en lang og en kort loddrett strek til venstre på arket. De to loddrette strekene skal berøre den vannrette streken på tuppen.
- 3.På den lange loddrette streken, tegn en pilspiss øverst. Gjør det samme på den korte loddrette 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 inntil hverandre på streken.
- 10.Tegn tre dråper over de fem sirklene. De tre dråpene skal ha ulik størrelse og berøre 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.
- [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) udir.no

- **5 min:** Sammenlign tegningen din med gruppens tegninger. Hvordan kunne instruksjonen blitt tydeligere og mer presis? Diskuter med hverandre.

- **5 min:** Ta for deg spørsmålet under. Bruk 2 minutter på å diskutere dette med naboen. Bruk så 2 minutter på å dele dette med resten av gruppen.
- **Spørsmål:** Det er ikke bare når vi programmerer at vi trenger presise instruksjoner. Gi eksempler fra ulike fag der det er viktig å trene på å gi presise instruksjoner.

Info om mellomarbeid

Eksempler på (analoge, frakoblede) oppgaver

- [Mine robotvenner](#): introduserer elevene for prinsippene bak programmering, og illustrerer behovet for funksjoner.
- [Programmering med de yngste](#): bygg algoritmer og følg instruksjoner
- [Tegn i pixler](#): tegne bilder ved å bruke tall
- [Lage eggedosis](#): følg den eksakte instruksjonen (som en datamaskin)
- [Gi instruksjoner](#): lær om algoritmer og debugging
- [Flytt den, flytt den – lage algoritmer for å flytte en figur](#):
- [Bygg en robot \(oppgave 5\)](#): Flytt en karakter i en labyrint ved hjelp av algoritmer.
- [Kode navnet ditt i et armbånd](#): kode ditt eget navn med binært tallsystem.
- Du finner alle oppleggene her: dekomp.uia.no/doku.php?id=arendal:mellomarbeid_1

Undersøkelse

bit.ly/arendaldigital1



Vel hjem