

# Lær kidsa programmering med

```
kode { timen_
```

Hva er programmering?

Hvorfor skal barna lære det?

...og hvordan?

# Et program er en oppskrift som sier hva datamaskinen skal gjøre

## Vafler med rømme

ca. 6–7 plater

2	egg
1 dl	vann
$\frac{1}{4}$ ts	salt
$1\frac{1}{2}$ dl	hvetemel
1 ts	bakepulver
3 dl	seterrømme

Visp sammen egg, vann, salt og hvetemel blandet med bakepulver til en glatt røre. Bland rømmen i til slutt.

Varm vaffeljernet. Smør med litt smør eller margarin før røren til den første platen has i. Stek vaflene gylne. Server dem helt nystekte eller legg dem på rist.

Server vaflene med syltetøy.

(\* Start dagen \*)

**PROGRAM** Morgen(Barn)

Stå opp.

Ta på klær.

**GJENTA**

Spis brødslike

**HELT TIL** Du er mett.

Puss tenner.

Ta på jakke og sko.

**HVIS** Det er hverdag

Gå til skolen

**ELLERS**

Begynn å leke.

**FERDIG.**

# Dataprogrammer er skrevet av mennesker i et *programmeringsspråk* som maskinen forstår

Med programmeringsspråket  
kan vi bestemme hva  
datamaskinen skal gjøre.

Mange programmeringsspråk er litt  
vanskelige å lære...

...men idag finnes det mange gode  
programmeringsspråk for barn!

```
/**  
 * Simple HelloButton() method.  
 * @version 1.0  
 * @author john doe <doe.j@example.com>  
 */  
HelloButton()  
{  
    JButton hello = new JButton( "Hello, wor  
    hello.addActionListener( new HelloBtnList  
  
    // use the JFrame type until support for t  
    // new component is finished  
    JFrame frame = new JFrame( "Hello Button"  
    Container pane = frame.getContentPane();  
    pane.add( hello );  
    frame.pack();  
  
    frame.show(); // display the fra
```

*Del av program skrevet i  
programmeringsspråket  
Java*

# Programmering er en del av informatikk, en fagdisiplin på linje med matematikk!

Informatikk er viktig, og mer enn programmering:

- programmeringsspråk, algoritmer og datamaskiner
- representasjon og strukturering av data
- datakommunikasjon og -styring
- abstraksjon, modellering, design av systemer
- store sammenhenger: samfunn, filosofi, etikk



# Det er mange gode grunner til å la skoleelever lære programmering

1. Elevene får trening i systematisk tenking, presisjon og kreativ problemløsning.
2. De blir mye bedre i stand til å forstå det digitale samfunnet: etikk, sikkerhet, muligheter..
3. Programmering er en nyttig ferdighet i seg selv.
4. Barna får bedre mulighet til å uttrykke seg med digitale medier.
5. Elevene erfarer å lage noe selv, og at det krever innsats hvis resultatet skal bli bra.
6. Elever som oppdager at de har talent og/eller interesse kan lett lære mer utenom skolen

...og det er gøy!



# Kodetimen tar elevene til første trinn på informatikk-stigen

- **Nivå 1 - Avmystifisering: For alle, også voksne!**
  - forstå konseptet programmering gjennom lek
  - lære at de kan gjøre det selv hjemme
- **Nivå 2 - Informatikk som allmenndannelse: Bør inn i skolen!**
  - lære om programmering og algoritmer
  - abstraksjon, modellering av virkeligheten
  - store sammenhenger: samfunn, filosofi, etikk
- **Nivå 3 - Programmering som håndverk: Valgfag i ungdomsskole/VGS**
  - lære fullverdige programmeringsspråk
  - forstå digital representasjon av informasjon
  - annen avansert bruk av IT: animasjon, simulering, digital fabrikasjon
- **Nivå 4 - Avansert egenaktivitet: For de som er interessert**
  - utvikle avanserte programmer og systemer



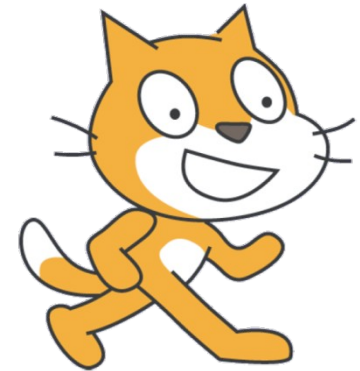
# Det finnes nå mye lærestoff på norsk om programmering på grunnskolenivå

- Samlingen av undervisningsopplegg for Kodetimen er unik i den Norske språkverdenen
- Bevegelsen "Lær kidsa koding" har på mindre enn ett år mangedoblet mengden av aktuelt lærestoff på norsk!
- Mye av stoffet er nytt, eller nylig oversatt.
- Viktig å få tilbakemeldinger fra praktisk bruk i Norge!



# Eksempel: Scratch - et programmerings-språk som er lett å lære og morsomt å bruke

- Scratch er laget for å være lett å lære
- Utviklet ved MIT-universitetet i USA
- Programmer bygges opp med klosser
- Det er lett å lage animasjoner og lyd
- Scratch er oversatt til norsk
- Scratch er helt gratis og brukes i hele verden
- Scratch har et eget nettsamfunn
- Du kan legge programmene ut på nett så andre kan prøve dem



SCRATCH



Lær Kidsa Koding  
kidsakoder.no



# Undervisningsopplegg for Kodetimen

```
kode { timen_
```

Her oppdaterer vi undervisningsoppleggene etterhvert som de blir klare. Kom og sjekk ofte! Om det er noe du mener bør være med, eller har andre kommentarer: kontakt [Line Moseng](#).

De første oppgavene ("Etter Aldersgrupper" ) er spesielt utvalgt og laget eller oversatt av oss. Under finner du en seksjon med alle oppgavene hos [code.org](#).

## Etter aldersgrupper

Øvelse uten datamaskin: [Mine robotvenner](#)

3. trinn og oppover: [Scratch innføring v/Torbjørn Skauli](#)

4. trinn og oppover: [Visuell programmering på norsk hos Code.org](#)

5. trinn og oppover: [Spillprogrammering med KoduGameLab](#)

# Bruk de kompetente elevene som hjelpere!



- De flinkeste elevene kan hjelpe dem som strever
- Eldre elever kan brukes som hjelpelærere (og rollemodeller!)
- Pilotforsøk på Årvoll har vist at ungdomsskoleelever kan kjøre kurs i Scratch på SFO
  - kjøreplan for dette opplegget er tilgjengelig



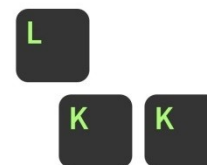
# La elevene få lyst til å fortsette utenom skolen. Ta tilbake noe av dataspill-tiden!

- På SFO - kanskje med ungdommer som instruktører?
- Kodeklubb utenom skolen - finn en frivillig forelder!
- Uorganisert fritidsaktivitet, helst mellom venner som hjelper hverandre (og ideelt sett i kontakt med en voksen mentor)
- Introduser konseptet programmering til foreldre og andre voksne!



# Det er viktig å tilby elevene ideer til ting de kan programmere

- La dem starte med et skjelett av et program som de skal gjøre ferdig
- Elevene kan visualisere en historie eller noe de har lært. Gi dem konkrete ideer.
- La dem få mulighet til å lage presentasjoner i f.eks. Scratch i stedet for Powerpoint
- Lag et fint og visuelt program til semesteravslutningen
- Hele klassen kan lage hvert sitt bursdagskort-program når noen har bursdag
- Foreslå enkle spill: gjett et tall, ...



# Lær elevene at de kan gå videre: de kan bli flinke og lage veldig kule ting

- Lær tekstbasert programmering  
f.eks. Python eller Java (via Greenfoot eller Processing)
- Kombiner koding med å lære elektronikk:  
f.eks. med Arduino eller Raspberry Pi
- Lær animasjon i 3D:  
SketchUp, Blender, Maya, 3D studio, Unity, UDK, ...
- Programmering for robotikk og modellfly/helikopter
- Lage apper for telefon/brett
- Finne på kreative installasjoner eller kunstneriske programmer
- Løse praktiske problemer og oppgaver



# Det er mange grunner til at læreren bør ha god kompetanse - akkurat som i norskfaget!

Kodetimen har opplegg som ikke krever kompetanse hos læreren.

Vær likevel oppmerksom på ting som krever kompetanse hos læreren:

- Finne metoder til å løse en programmeringsoppgave
- Strukturere et stort program
- Foreslå ny/endret funksjonalitet i et program
- Vurdere realismen i elevenes prosjektidéer
- Finne og rette feil i programmer
- Vurdere elevenes arbeid



Ferdig.



Lær Kidsa Koding  
kidsakoder.no