

UNDERVISNINGSSOPPLEGG – PROGRAMMERING OG GEOMETRI

Reiar Kravik
Universitetet i Sørøst-Norge

GEOMETRI OG PROGRAMMERING

Tema: Geometriske figurer

Foreslått klassetrinn: 8-10.trinn

Tidsbruk: 2 timer

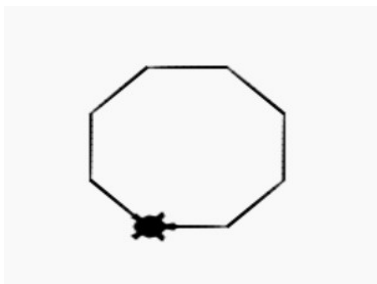
Når kan du bruke dette?

- Repetisjon av geometri
- Oppstart med programmering i skolen

Vedlegg 1: Koder til lærer

Vedlegg 2: Elevoppgaver –
Geometriske figurer

Vedlegg 3: Kommandoer til
elevne.



TIL LÆREREN.

Forarbeid for læreren:

- Kunnskap om programmeringsspråket python
- Kjenne til pakken "turtle" til python
- Skrive ut ark med kommanderer til elevne
- Skrive ut oppgavearket «Elevoppgaver – Geometriske figurer» til elevne.
- Sende ut oppgavearket «Elevoppgaver – Geometriske figurer» til elevne.

Bakgrunn til elevne

- I utgangspunktet anbefales en introduksjon til programmering gjennom et opplegg i analog programmering. Dette for å forstå grunnkonseptene i programmering.

Hvordan gjennomfører jeg dette undervisningsopplegget?

1. Elevne kan sitte sammen eller hver for seg med sin egen Ipad/data.
2. Elevne åpner linken <https://trinket.io/python>
3. Lærer viser hvordan vi legger til "turtle" pakken i Trinket. Elevne gjør dette samtidig som læreren. Det er viktig med en liten innføring i hvor vi skriver kode (venstre felt), hvordan vi får kjørt kode (run), hvor vi får resultatet av å kjøre koden og hvordan vi laster ned koden. Veiledning til læreren ligger under vedlegg 1.
4. Lærer bruker litt tid på å gå igjennom hvilke kommandoer som er tilgjengelige, forward(), backward(), left(), right(). Kommandoene de skal bruke er gitt under og må deles ut til elevne.
5. Elevne får litt tid teste ut hva som skjer med ulike tall i parentesene. Man får da også praktisk testet hvordan programmering fungerer, programmet gjør eksakt det man ber det om.

6. Lærer viser hvordan man skriver en forklarende tekst på linjen før kommandoer. Ved å bruke # blir ikke teksten med i programmet.
7. Lærer viser de så oppgavearket «Elevoppgaver – Geometriske figurer», og gjennomgår oppgave i plenum.
8. Forslag til hvordan elevene kan lagre svarene: De limer inn et skjermbilde i oppgavearket under riktig oppgave, lagrer dette, og deler dette med lærer på slutten av timen.
9. Elevene jobber så videre med utdelt oppgaveark.

Ulike variasjoner av aktiviteten

- Man tilpasser nivået til nivået i klassen.
- De svakeste elevene kan lage enkle figurer og allikevel få frem figurer og diskutere disse.
- Elever med kunnskap om programmering kan gå til et høyere nivå og det vil være greit å differensiere oppover.
- Ved noen tilpasninger kan aktiviteten brukes på mellomtrinnet, da må blant annet elevene ha et forhold til ulike geometriske figurer og vinkler.

TANKER RUNDT OPPLEGGET

Tanken er at elevene skal reflektere rundt geometriske figurer i en annen sammenheng enn vanlig undervisning. Ved hjelp av denne typer programmering må elevene bli bevisst på en del kjennetegn ved de ulike geometriske figurene, som gradetall til vinklene og forhold mellom lengder på sidene til figurene. Programmering skal inn i læreplanene, og bruk av "turtle" ses på som en enkel måte å forstå hvordan programmering fungerer. Det å se og oppleve at man må være nøyaktig når man skriver kode og se at man får direkte ut den man spør om skal være med på å legge grunnlaget for videre arbeid med programmering. Programmering er også med på å skape logisk tenkning og forhåpentligvis refleksjon. I tillegg jobber elevene med digital kompetanse.

VEDLEGG 1 – KODER TIL LÆRER

Hvordan opprette, laste ned, og lagre dokumenter på Trinket:

- Man oppretter filer ved å bruke + tegnet i Trinket. Det er viktig at filene lagres med navn.py
- Man kan laste opp filer om ved å bruke ikonet som har en flate og en pil opp.
- Man kan dele filer ved å bruke deletegnet (tre rundinger med sterk mellom)

Eksempler på figurer elevene kan lage:

- Trekanter (likesidet, likebeint og rettvinklet)
- Sirkel (dette gjøres ved å gå litt fremover, så snu noen grader, litt fremover, snu noen grader, osv, helt til du er tilbake til startpunktet)
- Kube
- Parallelogram
- Trapes
- Rettvinklet prisme
- Rombe

Eksempel på kode til et kvadrat til lærer

```
#vi importerer skilpaddepakken i python
from turtle import *
#vi får frem skilpadden i figuren
shape("turtle")
#Skilpadden går fremover 150 steg
forward(150)
#skilpadden snur 90 grader mot venstre
left(90)
#Skilpadden går fremover 150 steg
forward(150)
#skilpadden snur 90 grader mot venstre
left(90)
#Skilpadden går fremover 150 steg
forward(150)
#skilpadden snur 90 grader mot venstre
left(90)
#Skilpadden går fremover 150 steg
forward(150)
```

-

Eksempel på kode til likesidet trekant til lærer

```
#vi importerer skilpaddepakken i python
from turtle import *
#vi får frem skilpadden i figuren
shape("turtle")
#Skilpadden går fremover 150 steg
forward(150)
#skilpadden snur 120 grader mot venstre
left(120)
#Skilpadden går fremover 150 steg
forward(150)
#skilpadden snur 120 grader mot venstre
left(120)
#Skilpadden går fremover 150 steg
forward(150)
```

ELEVOPPGAVE – GEOMETRISKE FIGURER

Du skal ta et skjermbilde av alle figurene du lager i Trinket. Skjermbilde skal inneholde både koden din, og figuren du lager. Skjermbildet limes inn i dette dokumentet under den oppgaven du jobber med. Dokumentet skal leveres i lærer i slutten av timen.

Navn:

1. Lag et kvadrat ved hjelp av kommandoene `forward()` og `left()`
Husk å skrive en forklaring til kommandoene du gjør, slik at en annen elev kan forstå kommandoene dine uten at du forklarer muntlig. Dette gjelder alle oppgavene.
Skjermbilde:
2. Lag et rektangel:
Skjermbilde:
3. Lag en trekant.
Skjermbilde:
4. Kan du noen andre typer trekanter enn den du har laget i oppgaven over? Se om du kan lage disse.
Skjermdump:

5. List opp flere geometriske figurer du kjenner til her:

Geometriske figurer:

Kan du lage noen av disse i Trinket?

Skjermdump:

6. Kan du nå lage et kvadrat igjen, og så speile dette?
Hint: Kanskje kommandoen `penup()` kan være til hjelp?
Skjermdump:

7. Velg deg 4 andre kommandoer fra arket «Kommandoer til elevene». Test ut disse.

VEDLEGG 3 – KOMMANDOER TIL ELEVENE

Legger til datapakken slik at vi får fram en skilpadde. Dette må alltid stå øverst i koden vår når vi jobber med skilpadde.	<code>from turtle import *</code>
Viser skilpadden i vinduet når du kjører programmet.	<code>shape("turtle")</code>
Bevege skilpadden fremover. Angi antall piksler skilpadden skal gå fremover i parentes.	<code>forward()</code>
Bevege skilpadden bakover. Angi antall piksler skilpadden skal gå bakover i parentes.	<code>backward()</code>
Gå til venstre. Angir grader skilpadden skal "snu" seg	<code>left()</code>
Gå til høyre. Angir grader skilpadden skal "snu" seg til høyre.	<code>right()</code>
Løfter pennen/markøren, trenger ikke innhold i parentes. Bruk forward-kommando for å gå frem så mye du vil uten at det blir "strek". Avsluttes med pendown.	<code>penup()</code>
Setter ned pennen/markøren etter å ha brukt "penup". Trenger ikke innhold i parentes.	<code>pendown()</code>
Reset, fjerner figuren. Trenger ikke innhold i parentes.	<code>reset()</code>
Angir en posisjon du vil gå til på skjermen. Startpunkt er (0,0).	<code>goto()</code>
Man kan angi at vi vil fylle en figur vi lager. Da angir vi <code>begin_fill()</code> før vi lager figuren og avslutter med <code>end_fill()</code>	<code>begin_fill()</code>
Avslutter kommandoen som fyller en figur.	<code>end_fill()</code>
Tegner en sirkel. Vi angir størrelse med verdi i parentes.	<code>circle()</code>
Angir farge på skilpadda. (215, 100, 170) er rosa.	<code>fillcolor()</code>
Vi kan angi hvilken retning skilpadda skal ha hodet. Angi grader i parentes.	<code>setheading()</code>
Angir tykkelse på penn.	<code>pensize()</code>
Lager en punktum. Angi størrelse i parentes	<code>dot()</code>
Gjør skilpadden usynlig.	<code>hideturtle()</code>
Skriver tekst der skilpadden befinner seg. Eks <code>write("test", move=False, align='left', font=('Arial', 22, 'normal'))</code>	<code>write()</code>
Sett farge på pennene. Eksempel kan være <code>pencolor("pink")</code> .	<code>pencolor()</code>