

KLIPP OG LIM KODEØVELSE

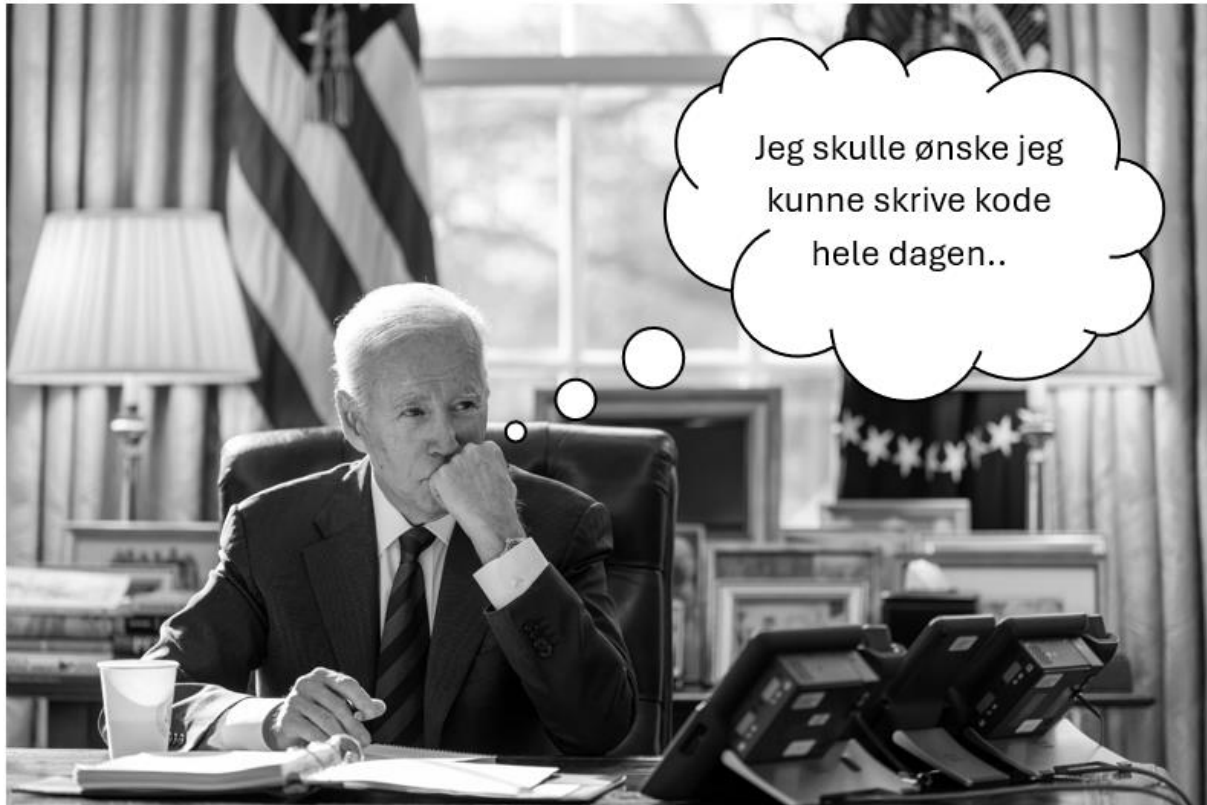
Kodeøvelse til Oppdrag: Mars – Andøya Mission Control



Kort om aktiviteten

Hva trenger elevene for å løse kodeoppgaven i Oppdrag: Mars?

I denne kodeøvelsen er det fire aktiviteter som vedlegg som lar elevene trene litt på generelle programmeringskunnskaper, men også få en liten forsmak på kodeverktøyet som brukes i Oppdrag: Mars.



Innhold

Kort om aktiviteten	2
Aktivitet 1 – Trekk en linje	4
Aktivitet 2 – Finn feil	5
Aktivitet 3 – Bestem hvor marsrover skal kjøre.....	6
Aktivitet 4 – Gjør vitenskapelige målinger.....	7
Lærerveiledning.....	8
Aktivitetene	9
Læringsmål	10
Kilder	11
Lisensiering:	11

Aktivitet 1 – Trekk en linje

Trekk en linje mellom Python kode i førte kolonne og riktig forklaring i høyre kolonne

```
def summere(a, b):
    return a + b
```

En funksjon som ser på variabel "b" og skriver den ut til skjerm helt til den er 9.

```
If a >= 3:
    print(a)
```

Kode som fyller en liste med tallene 1-10.

```
b = 0
while b < 10:
    print(b)
    b = b + 1
```

En sjekk om en variabel som inneholder verdien «mars» finnes i listen over planeter.

```
from time import sleep
```

En funksjon ved navn "summere" som tar inn en variabel "a" og "b" og sender tilbake summen av variablene.

```
a = []
for j in range(1, 11, 1):
    a.append(j)
```

Lengden til en definert liste skrives ut til skjerm.

```
klode = "mars"
planet = ["jorden", "mars", "venus"]
if klode in planet:
    print("fant en planet")
```

Kode som skriver ut verdien til en variabel "a" dersom den er større eller lik 3.

```
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(len(b))
```

Fra et bibliotek som heter "time" hentes ut kunnskapen om en funksjon som heter "sleep"

Aktivitet 2 – Finn feil



Se på koden under og bruk dine feilsøkingsevner. Det betyr at du må finne ut hva som er feil og forklare det med ord i rutene under.

<pre>a = "jorden" b = 137 c = a + b print(c)</pre>	<pre>liste = [3, 4, 12, 7, 9] tall = liste[5] print(tall)</pre>	<pre>if a == b: a + 1 elif a != b: b + 1</pre>
--	---	--

Forklar hva som ikke er riktig i kodesnuttene over med egne ord:

--	--	--

Aktivitet 3 – Bestem hvor marsrover skal kjøre

I rutenettet til høyre på 5x5 ruter skal du forflytte robotfartøyet eller rover fra ruten merket A til ruten merket B ved hjelp av kode. Underveis skal du aktivere algoritmen til Marsgeolog (M) på et bestemt sted. Du skal også kjøre innom et vitenskapelig interessant område merket (V) og ta et bilde.

				B
	M			
			V	
A				

Klipp ut kodelistene under og lim i kodelisten under eller skriv inn med penn/blyant. De kan bare brukes en gang hver. Mulige parametere til kodelistene er på høyresiden:

RETNING ()	Ø, N, S, V
RETNING ()	Ø, N, S, V
RETNING ()	Ø, N, S, V
RETNING ()	Ø, N, S, V
RETNING ()	Ø, N, S, V
START	Ruten du starter på
AKTIVERE SEGMENT ()	M, F, T, I
KAMERA ()	Ta bilde
STEG ()	1 – 5
STEG ()	1 – 5
STEG ()	1 – 5
STEG ()	1 – 5
STEG ()	1 – 5
STEG ()	1 – 5

Kodeliste:

START

Aktivitet 4 – Gjør vitenskapelige målinger

Som en programmerer av roveren må du gi den nøye instruksjoner, slik at den kan gjøre sine oppgaver uten at noen styrer den. Det kan bety at roveren selv må kunne ta beslutninger basert på omgivelser og situasjoner. Du skal nå ta i bruk to av roverens instrumenter. Ett kamera som skal brukes når roveren står i ro og en radar som skal brukes når roveren kjører. Kamera skal ta tre bilder med ett minutt forsinkelse og lagre bildene til disk.

Klipp ut kodelistene under og lim i kodelisten eller skriv inn med penn/blyant. De kan bare brukes en gang hver. Mulige parametere til kodelistene er på høyresiden:

IF () == ()	Rover, kamera, radar, kjører, aktiv
END IF	
IF () != ()	Rover, kamera, radar, kjører, aktiv
END IF	
IF () != ()	Rover, kamera, radar, kjører, aktiv
END IF	
IF () != ()	Rover, kamera, radar, kjører, aktiv
END IF	
LØKKE (), ()	Antall gjentakelser, antall linjer i løkken
KAMERA ()	Ta bilde, bruk radar
KAMERA ()	Ta bilde, bruk radar
AKTIVERE ()	Kamera, radar
AKTIVERE ()	Kamera, radar
FORSINKELSE ()	1 s, 1 m, 1 t
LAGRE ()	Til disk, til minne

Kodeliste:

Lærerveiledning

Denne kodeøvelsen kan brukes helt fritt og slik du selv ønsker det. Du kan gi elever en eller alle fire oppgavene.

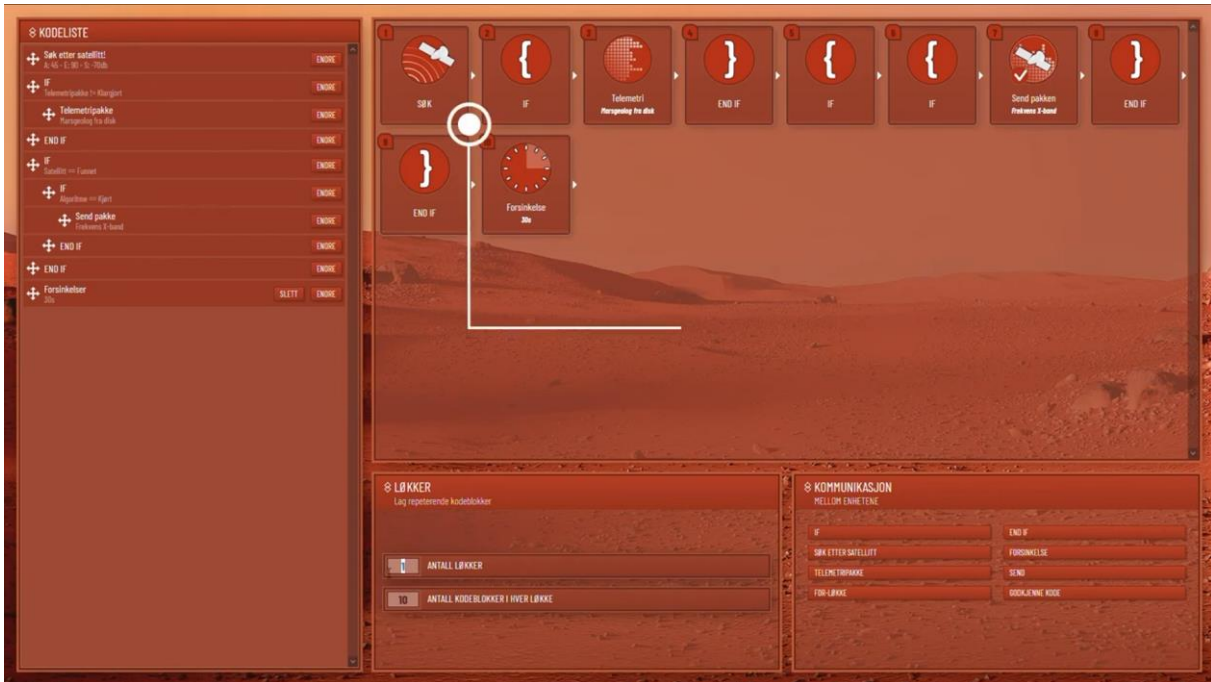
Formålet er å la elever som skal delta i Oppdrag: Mars øve litt på forhånd.

Det er ikke et krav, men litt forkunnskaper om kodekonsepter som funksjoner, variabler, løkker og vilkår er fint å ha.

Selve kodeverktøyet som brukes i oppdraget er spesialutviklet, men kan minne om blokkprogrammering. Mange elever tar kodeutfordringen på sparket og klarer det helt utmerket, men vi ser at det likevel kan være ønskelig å få en liten introduksjon til måten programmeringen utføres på.



Se også video som viser verktøyet i bruk på <https://www.youtube.com/watch?v=c-4VkeggCw>



Aktivitetene

I denne kodeøvelsen er det fire aktiviteter:

Aktivitet 1 går ut på å trekke linjer mellom programkode og forklaring. Klarer elevene å finne forklaringen som passer? Her er riktig forklaring for hver kodesnutt:

```
def summere(a, b):
    return a + b
```

En funksjon ved navn "summere" som tar inn en variabel "a" og "b" og sender tilbake summen av variablene.

```
b = 0
while b < 10:
    print(b)
    b = b + 1
```

En funksjon som ser på variabel "b" og skriver den ut til skjerm helt til den er 9.

```
If a >= 3:
    print(a)
```

Kode som skriver ut verdien til en variabel "a" dersom den er større eller lik 3.

```
from time import sleep
```

Fra et bibliotek som heter "time" hentes ut kunnskapen om en funksjon som heter "sleep"

```
a = []
for j in range(1, 11, 1):
    a.append(j)
```

Kode som fyller en liste med tallene 1-10.

```
klode = "mars"
planet = ["jorden", "mars", "venus"]
if klode in planet:
    print("fant en planet")
```

En sjekk om en variabel som inneholder verdien «mars» finnes i listen over planeter.

```
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(len(b))
```

Lengden til en definert liste skrives ut til skjerm.

I **aktivitet 2** er det feil med kode og oppgaven går ut på å forklare hva som er feil (feilsøke/debugge). I begge tilfellene er det brukt Pythonkode. Debugging er en stor del av programmering og fint å øve seg på.

Aktivitet 3 og **4** er laget for å ligne på kodeverktøyet i oppdraget. Her skal elevene sette sammen en algoritme som skal gjøre bestemte ting. De kan klippe ut papirstrimler og sette sammen eller skrive direkte på arket med penn/blyant. Livet er fullt av valg, det er flere måter å løse en utfordring på.

Tips til aktivitet 3 og 4:

- RETNING setter orienteringen til roveren, og STEG forteller hvor mange ruter roveren skal bevege seg fremover.
- IF må alltid avsluttes med END IF.
- Innrykket kode: Kode som skal skje som en del av en løkke eller vilkårsjekk må rykkes inn.

Læringsmål

- Få generell forståelse for oppbygging av enkel programkode
- Bli kjent med kodeverktøyet til Oppdrag: Mars

Kilder

Bildene med snakke- og tenkebobler er alle merket public domain.

Bilde side 2: President Joe Biden den 13. februar 2023. Kilde: Offisiell fotograf for Det hvite hus, Adam Schultz, <https://www.flickr.com/photos/whitehouse/52734536306/>

Bilde side 5: Martin Luther King, 28. august 1963. Kilde: ukjent. Public domain, <https://snl.no/borgerrettsbevegelsen>

Bilde 1 side 8: Ukjent par som sitter utendørs. Kilde: ukjent, <https://www.pickpik.com/man-sitting-couple-glass-field-cute-57838>

Bilde 2 side 8: Et skjermbilde som viser verktøyet for programmering som inngår i Oppdrag: Mars. Kilde: Andøya Space Education.

Lisensiering:

Dette verket er lisensiert under en [Creative Commons Navngivelse-IkkeKommersiell-IngenBearbeidelse 4.0 Internasjonal lisens \(CC BY-NC-ND 4.0\)](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Du kan dele dette materialet så lenge du krediterer oss, ikke bruker det kommersielt, og ikke endrer det.