

# Oppgaver uke 1: Løsningsforslag

## Oppgave 1

Hva tror du følgende program skriver ut til terminalen? Diskuter med gruppen.

```
alder = 30
print("Din alder er", alder)
alder = 15
```

### Løsningsforslag:

Din alder er 30

Når `print()` kalles er verdien til variabelen `alder` 30. Programmet kjøres linje for linje, og derfor blir utskriften som den blir. Siden verdien til variabelen `alder` blir endret **etter** `print()`, vil ikke denne endringen vises.

## Oppgave 2

Skriv et program med penn og papir som gjør følgende:

1. Lag to variabler med verdiene 4 og 5.
2. Skriv ut summen av tallene.

### Løsningsforslag:

Her kan vi se for oss to løsninger:

Forslag 1:

```
tall1 = 4
tall2 = 5
print(tall1 + tall2)
```

Forslag 2:

```
tall1 = 4
tall2 = 5
sum = tall1 + tall2
print(sum)
```

Begge programmene gir samme utskrift:

9

Forslag 2 har den fordelen at man kan bruke variabelen `sum` senere også, mens i forslag 1 har vi ikke en egen variabel til å holde på summen. Husk at når vi skriver

```
sum = tall1 + tall2
```

så legges først `tall1` og `tall2` sammen, og så gis den verdien til `sum`.

## Oppgave 3

Hva tror du følgende program skriver ut til terminalen? Diskuter med gruppen. Skriv det som skrives ut som kommentarer i programkoden.

```
print("39 + 3")
print(39 + 3)
```

### Løsningsforslag:

Selve utskriften fra programmet blir:

```
"39 + 3"  
42
```

I oppgaven ble dere bedt om å skrive svaret som kommentarer i programkode, altså:

```
print("39 + 3")    #"39 + 3"  
print(39 + 3)     #42
```

I Python brukes # for å markere kommentarer. Man kan også bruke tre anførselstegn etter hverandre("""**kommentar**""") for å kommentere lengre avsnitt. Bruk disse for å gi informasjon til dere selv eller andre som skal lese koden deres. Det kan blant annet være nyttig for å huske hva dere holdt på med hvis dere skal se på koden seinere, eller for å forklare en løsning som ikke er lett å forstå med en gang.

## Oppgave 4

Hva blir variabelen a til i de ulike uttrykkene? Skriv som kommentar

1.  $a = 4 + 5$
2.  $a = 6 - 2$
3.  $a = 4 * 2$
4.  $a = 11 / 2$
5.  $a = 11 // 2$

### Løsningsforslag:

Her skal dere også svare med kommentarer, som om det skulle være kode:

```
a = 4 + 5 #gir 9  
a = 6 - 2 #gir 4  
a = 4 * 2 #gir 8  
a = 11 / 2 #gir 5.5  
a = 11 // 2 #gir 5
```

Dere kan gjøre alle de matematiske operasjonene dere er kjente med i Python. Det nyeste er kanskje heltallsdivisjon (//) og modulo (%). Dere kan lese mer om disse i seksjon 2.2 i Python-boka. + og - er som vanlig, \* er gange, / er dele, og \*\* er potens. Eksempel på potens:

```
a = 2**3 #gir 8
```

## Oppgave 5

Hva tror du følgende program skriver ut til terminalen? Diskuter med gruppen, og skriv ned den nøyaktige teksten.

```
tall1 = "120"  
tall2 = "3"  
tall3 = tall1 + tall2  
  
print("Tall 1: ", tall1, "og tall 2: ", tall2, "blir til sammen: ", tall3)
```

### Løsningsforslag:

### Utskrift:

```
Tall 1: 120 og tall 2: blir til sammen: 1203
```

Grunnen til at det er doble mellomrom, er fordi i oppgaven dere fikk utdelt var det mellomrom etter kolon i utskriften. Siden kommaer skrives ut som mellomrom i print() får vi noen doble mellomrom. Tenk derfor på når dere kan trenge mellomrom, og når dere ikke trenger det hvis dere har kommaer i print().

### Oppgave 6

Skriv et lite program som tar navn og alder til en person som input og deretter skriver ut "Hei! <navn> er <alder> år gammel!". Husk å gi tydelige beskjeder om hva som forventes som input av brukeren.

#### Løsningsforslag:

```
navn = input("Hva heter du?")
alder = input("Hvor gammel er du?")
print("Hei,", navn, ", du er", alder, "år gammel.")
```

Eksempelkjøring:

```
Hva heter du?Petter
Hvor gammel er du?28
Hei, Petter , du er 28 år gammel.
>>>
```

### Oppgave 7

Hva tror du skrives ut her?

*Utfordring: hva tror du skrives ut dersom  $a = 4$  og  $b = 4$ ?*

```
a = 4
b = 6

if a > b:
    print(a, "er større enn", b)
else :
    print(b, "er større enn", a)
```

#### Løsningsforslag:

Utskrift hvis  $a = 4$  og  $b = 6$ :

```
6 er større enn 4
```

Utskrift hvis  $a = 4$  og  $b = 4$ :

```
4 er større enn 4
```

Dette er altså «b er større enn a». Grunnen til dette er fordi programmet bare sjekker om a er mindre enn b, og siden det ikke er riktig, kjøres «else». Programmet kan forbedres ved å legge til en ekstra test:

```
a = 4
b = 6

if a > b:
    print(a, "er større enn", b)
elif b > a:
    print(b, "er større enn", a)
else:
    print(a, "er lik", b)
```

## Oppgave 8

Diskuter forskjellen på disse to:

<pre>if a &gt; b:     print(a, "er større enn", b) elif b &gt; a:     print(b, "er større enn", a) else :     print(a, "er lik", b)</pre>	<pre>if a &gt; b:     print(a, "er større enn", b) if b &gt; a:     print(b, "er større enn", a) else :     print(a, "er lik", b)</pre>
---	---

Tenk på; hva blir utskriften hvis...

1. a = 4, b = 3
2. a = 5, b = 7
3. a = 3, b = 3

### Løsningsforslag:

**Første alternativ** gir utskriften:

1. 4 er større enn 3
2. 7 er større enn 5
3. 3 er lik 3

**Andre alternativ** gir utskriften:

1. 4 er større enn 3     **OG**  
   4 er lik 3
2. 7 er større enn 5
3. 3 er lik 3

Grunnen til at det blir sånn er at med if – elif – else vil programmet sjekke både *if* og *elif* før den går videre til *else*. *Else* i dette programmet kjører derfor for alle tilfeller som ikke oppfyller `a > b` (etter *if*) eller `b > a` (etter *elif*).

I det andre programmet derimot er det if – if – else. To *if* etter hverandre vil bare gjøre at den første *if* tester, og så vil den begynne på den neste *if* **uavhengig** av hva resultatet av den første if-testen var.

*Else* i dette programmet vil derfor kjøre alle tilfeller som ikke oppfyller  $b > a$ , men bryr seg ikke om  $a > b$ , fordi den første if-testen ikke er knyttet til *else* her.