

IN1000 - Seminaroppgaver til uke 1

Nyttig å vite:

[PEP 8](#) -- Style Guide for Python Code. Dersom det som undervises i IN1000 ikke samstemmer med PEP, så skal dere følge den stilen som undervises og ikke PEP.

Oppgave 1

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

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

Løsningsforslag:

Programmet skriver ut følgende:

```
Din alder er 30
```

*Grunnen til at det er 30, og ikke 15, er fordi vi endrer verdien til variabelen **alder** etter at vi har skrevet ut til terminalen. Merk også at vi får et mellomrom. Det mellomrommet får vi fordi når vi skriver ut flere ting med `print()`, vil hvert komma bli et mellomrom.*

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:

```
variabel1 = 4
variabel2 = 5
```

```
#Vi kan enten mellomlagre summen...
```

```
summen = variabel1 + variabel2
print(summen)
```

```
#...eller så kan vi regne ut og skrive ut på samme programlinje:
```

```
print(variabel1 + variabel2)
```

Merk da at først blir summen av variabel1 og variabel2 evaluert, og så blir den verdien skrevet ut.

OBS! Ikke bruk ordet *sum* som variabelnavn. Det er et reservert ord i python, som *if*, *print* osv.

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:

Den første linja skriver ut:

39 + 3

Den andre linja skriver ut:

42

Grunnen til dette har med typer i Python å gjøre. Alt som er mellom enkle eller doble anførselstegn regnes av Python som strenger. Alt mellom disse, som i for eksempel "1 + 2", vil da bare ses som tekst. I den andre linja er det to tall med en operator (pluss) i midten.

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:

$a = 4 + 5$

#her får a verdien 9

```
a = 6 - 2
```

```
#her får a verdien 9
```

```
a = 4 * 2
```

```
#her får a verdien 9
```

```
a = 11/2
```

```
#her får a verdien 9
```

```
a = 11//2
```

```
#her får a verdien 9
```

I tillegg så vi på *modulo* i timen:

```
a = 11 % 2
```

```
#her får vi verdien 1
```

```
// brukes for heltallsdivisjon, og % er modulo.
```

En måte å visualisere det på: Resultatet av heltallsdivisjon er tallet før komma, og modulo er resten, før man trekker ned 0.

resultat av heltallsdivisjon //

$$11:2 = 5,5$$
$$\begin{array}{r} 10 \\ \hline 10 \\ \hline 0 \\ \hline 0 \end{array}$$

modulo (rest) %

Hvis vi må holde oss innenfor heltall, det vil si at vi bare tillater tall som -2, -1, 0, 1, 2, 3 osv, ikke tall som 4,5 eller π , kan vi ikke gjøre noe med resten. Det er denne resten, som altså er resten etter heltallsdivisjon, som er modulo.

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:

Det som skrives ut er:

Tall 1: 120 og tall 2: 3 blir til sammen: 1203

Mellom de ulike elementene, separert av komma, blir det skrevet ut ett mellomrom. Det blir to mellomrom etter "tall1: " og "og tall 2: ", fordi det er et mellomrom der fra før av.

Siden begge variablene er strenger, blir verdiene konkatenerert, og vi får strengen "1203".

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, "er", alder, "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:

Når $a = 4$ og $b = 6$, skrives følgende ut:

4 er større enn 6

Når $a = 4$ og $b = 4$, skrives følgende ut:

4 er større enn 4

Grunnen til at dette skjer, er fordi siden den første testen ($a > b$) blir usann, fordi 4 ikke er større enn 4, så går testen til *else*. Det er ikke tatt hensyn til at to tall kan være like.

Oppgave 8

Diskuter forskjellen på disse to:

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

Hvis a = 4, og b = 3

4 er større enn 3	4 er større enn 3 4 er lik 3
-------------------	---------------------------------

Hvis a = 5, og b = 7

7 er større enn 5	7 er større enn 5
-------------------	-------------------

Hvis a = 3, og b = 3

3 er lik 3	3 er lik 3
------------	------------

Grunnen til at det blir slik, er at til høyre er det egentlig to tester. Når vi har flere if-tester, vil de ikke være avhengig av hverandre. Ved å bruke *elif* vil testen kun utføres hvis den if-en eller elif-en som kommer *før* *ikke* er sann. Hvis vi har en ny *if* vil den utføres uavhengig av testene som kommer *før*.