

## Bonusoppgaver: Variabler og regning

### Oppgave 1 *Jordkloden*



I denne oppgaven skal vi øve på å bruke Python som kalkulator, ved å regne litt på jordkloden. Husk at formelen for volumet av en kule er

$$V = \frac{4}{3}\pi R^3.$$

- a) Jordkloden er tilnærmet en perfekt kule, og har en radius på 6371 km. Lag et kort program som først definerer en variabel `radius`, og deretter regner ut en variabel `volum`. Skriv til slutt ut svaret til brukeren med `print()`-funksjonen. La svaret være i  $\text{km}^3$ .
- b) Endre programmet ditt så svaret istedet skrives ut i antall liter ( $1 \text{ km}^3$  er det samme som 1 000 000 000 000 L).
- c) Den totale massen til jordkloden er omtrent  $M = 5.972 \cdot 10^{24} \text{ kg}$ . Regn ut hvor mange kg hver liter av jordkloden veier i gjennomsnitt. Virker svaret ditt rimelig?

### Løsning oppgave 1 *Jordkloden*

a)

```

1 radiuskm = 6371
2 volumkm = (4/3)*3.14*(radiuskm**3)
3 print("Volumet til jorden er", volumkm, "
    kubikkilometer.")

```

b)

```

1 radiusdm = 6371 * 10**4
2 volumdm = (4/3)*3.14*(radiusdm**3)
3 print("Volumet til jorden er", volumdm, "liter.")

```

c)

```

1 massejord = 5.972 * 10**24
2 vektperliter = massejord/volumdm
3 print("I gjennomsnitt veier hver liter av jorda",
    vektperliter, "kg.")

```

## Oppgave 2 *Help!*

Sacha er misfornøyd med at `print`-funksjonen lager en ny linje for hvert funksjonskall. Hun har lyst til å se i dokumentasjonen om det er mulig å bruke `print` uten dette. I denne oppgaven skal du bruke `help` funksjonen i Python til å lese dokumentasjonen til `print`-funksjonen

- Bruk `help(print)` til å skrive dokumentasjonen til `print` ut til terminalen.
- Hvilke 4 “Optional keyword arguments” står listet i dokumentasjonen til `print`?
- Hva står som beskrivelsen til end argumentet. Hva tror du det betyr?
- Hva vil output for følgende program bli? (Skriv ned svaret først. Kjør programmet for å sjekke.)

```
1 print("sofa")
2 print("pute")
```

e) Hva med følgende program ? (Skriv ned svaret først. Kjør programmet for å sjekke)

```
1 print("sofa", end=" ")
2 print("pute", end=" ")
```

f) Modifiser kun end-argumentene i programmet over for å få følgende output:

```
sofa-pute!
```

## Løsning oppgave 2 *Help!*

a)

```
1 help(print)
```

b)

```
Optional keyword arguments:
file:  a file-like object (stream); defaults
       to the current sys.stdout.
sep:   string inserted between values, default
       a space.
end:   string appended after the last value,
       default a newline.
flush: whether to forcibly flush the stream.
```

c)

```
end:    string appended after the last value,  
        default a newline.
```

Dette betyr at end-argumentet bestemmer hva som skal være på slutten av beskjedene som skrives ut til brukeren. Som standard er dette et linjeskift. Husk: `print`-funksjonen lager et nytt linjeskift hver gang den blir brukt. Men vi kan også sette den til å være noe annet.

d)

```
sofa  
pute
```

e)

```
sofapute
```

f)

```
1 print("sofa", end="-")  
2 print("pute", end="!")
```

### Oppgave 3 *len-funksjonen*

Koden under bruker en funksjon `len`, som du kanskje ikke har sett før. I denne oppgaven skal vi undersøke hva denne funksjonen gjør.

```

1 bokstav = "y"
2 print(len(bokstav))
3
4 mat = "pizza"
5 print(len(mat))
6
7 topping = "ananas"
8 print(len(topping))

```

- a) Skriv av og kjør koden. Hva får du ut?
- b) Basert på det du fikk ut i oppgave a), hva tror du `len`-funksjonen gjør?
- c) Skriv `help(len)` nederst i programmet og kjør koden. Hva får du ut?  
Basert på det, hva tror du `len` gjør?
- d) Hva tror du du får om du skriver `print(len("sopp"))`?

### Løsning oppgave 3 *len-funksjonen*

a)

```

1
5
6

```

b) Den sier hvor mange tegn det er i en streng (eller mer generelt, hvor mange element som er i en samling)

c)

```
Help on built-in function len in module builtins:

len(obj, /)
    Return the number of items in a container.
```

d) 4.

#### Oppgave 4 *Brøkgregning*

Nedenfor er kode som skriver om en uekte brøk  $\frac{5}{2}$  til et blandet tall og et desimaltall.

```
1 teller = 5
2 nevner = 2
3
4 hele = teller // nevner
5 rest = teller % nevner
6 desimaltall = teller/nevner
7
8 print(f"{teller}/{nevner} = {hele} + {rest}/{nevner} =
    {desimaltall}")
```

- a) Hva inneholder teller-variabelen? Hva skjer om du endrer den til 7?
- b) Hva betyr %-tegnet på linje 5?
- c) Hva må endres for å skrive om  $\frac{9}{4}$  til et blandet tall og et desimaltall?

#### Løsning oppgave 4 *Brøkgregning*

- a) teller-variabelen inneholder telleren til brøken vi vil skrive om.
- b) %-tegnet er *modulo*-operatoren. Den gir resten etter å ha delt to tall på hverandre.  $\frac{5}{2} = 2 + \frac{1}{2}$ , så  $5 \% 2 = 1$  siden vi har 1 i rest etter at vi har delt 5 på 2. Tilsvarende er  $7 \% 4 = 3$  siden vi har 3 i rest etter å ha delt 7 på 4 ( $\frac{7}{4} = 1 + \frac{3}{4}$ ).

c) Vi må endre teller-variabelen til 9 og nevner-variabelen til 4:

```
1 teller = 9
2 nevner = 4
3
4 hele = teller // nevner
5 rest = teller % nevner
6 desimaltall = teller/nevner
7
8 print(f"{teller}/{nevner} = {hele} + {rest}/{nevner} = {desimaltall}")
```

### Oppgave 5 *Printe tall*

Dere er tre venner som har spleiset på pizza. Pizzaen kostet 230,-. Koden under regner ut prisen hver av dere må betale

```
1 pris = 230
2 antall_venner = 3
3 pris_per_pers = pris / antall_venner
4 print(f"Dere må betale {pris_per_pers} hver.")
```

- a) Gå igjennom koden og forklar linje for linje hva den gjør. Hva vil bli skrevet ut?
- b) Kopier eller skriv av koden og kjør den. Hadde du rett i oppgave a)?
- c) Endre slik at det istedenfor å stå `{pris}` står `{pris:.2f}` og kjør koden. Hva blir endret sammenliknet med utskriften fra oppgave a)?
- d) Hva tror du 2-tallet i `{pris:.2f}` gjør?
- e) Endre koden slik at du printer ut i hele kroner (null desimaler nøyaktighet).

### Løsning oppgave 5 *Printe tall*

a) Dere måbetale 76.6666666666667 hver. skrives ut

- På linje 1 oppretter programmet en variabel, `pris`, som representerer prisen på pizzaen (230,-).
- På linje 2 opprettes en variabel, `antall_venner`, som representerer antall venner som spleiset på pizzaen.
- På linje 3 Regnes prisen hver person må betales ut.
- På linje 4 flettes variablene i en flettestreng som printes ut

b) Dere måbetale 76.6666666666667 hver. skrives ut

c) Dere måbetale 76.67 hver. skrives ut.

d) 2-tallet representerer hvor mange desimaler nøyaktighet vi ønsker i utskriften.

e)

```
1 pris = 230
2 antall_venner = 3
3 pris_per_pers = pris / antall_venner
4 print(f"Dere må betale {pris_per_pers:.0f} hver.")
```