

Oppgaver

I denne seksjonen finner du oppgaver som hører til dag 1 av Kodeskolens kræsjkurs i programmering. Tema for første dag er *introduksjon til Python, variabler, betingelser og løkker*. Dersom du står fast er det bare å spørre. I tillegg anbefaler vi å lese i kompendiet hvis det er noen temaer du synes er spesielt vanskelige.

God koding!

Variabler og regning

Oppgave 1 *Printing*

- a) Lag et program som skriver ut teksten «Hei, verden!», til skjermen.
- b) Lag et program der du først lagrer navnet ditt i en variabel, og så få programmet ditt til å skrive ut en hilsen direkte til deg.
- c) Hvis du bruker kommandoen `len()` på variabelen din får du ut antall bokstaver i navnet ditt. Endre programmet ditt så det også skriver ut denne informasjonen.
- d) Du kan bruke funksjonen `input()` til å stille brukeren et spørsmål. Bruk dette til å lage et program som spør brukeren om navnet deres, og deretter skriver ut en beskjed som bruker navnet de har oppgitt. Skriv også ut hvor mange bokstaver det er i navnet til vedkommende.

Oppgave 2 *Kvadrattall*

Kvadrattall er heltall som er blitt kvadrert, altså ganget med seg selv, eller opphøyet i annen. I Python kan du regne ut kvadratet av et tall n enten ved å skrive `n*n` eller `n**2`.

- a) Skriv et program som spør brukeren om et tall, og deretter skriver ut kvadratet av tallet brukeren ga.
Pass på at når vi spør om et tall må vi gjøre om svaret til fra brukeren til

en tallvariabel ved å skrive `int` på følgende måte: `int(input())`. Dette er fordi «int» står for «integer», som er engelsk for heltall.

- b) Bruk programmet ditt og prøve-feile metoden til å finne det minste tallet som har et kvadrat på over 1000.

Oppgave 3 *Finn fire feil!*

Her følger det fire programmer som har blitt skrevet feil. Finn feilen i hver programsnitt. Du kan godt kjøre programmet inn på din egen maskin, og kjøre det, da kan kanskje feilmeldingen hjelpe deg å skjønne hva som er galt.

Når du tror du skjønner hva som er galt, rett feilen på din egen maskin, og kjør programmet for å sjekke at det fungerer som det skal.

a)

```
1 print(Hei , Verden!)
```

b)

```
1 Print("Hei , Verden!")
```

c)

```
1 person = input("Hva heter du?")
2 print("Hei på deg, {navn}")
```

d)

```
1 pi = 3,14
2 radius = 4
3 areal = radius*pi**2
```

Betingelser

Oppgave 4 *Finn størst tall*

I denne oppgaven skal vi øve på **if**-tester ved å finne det største av to tall.

- a) Skriv et program som ber brukeren om å skrive inn to tall og lagre dem i to variabler, **tall1** og **tall2**. Gjør om tallene til flyttall ved hjelp av **float**.
- b) Utvid programmet ditt ved hjelp av **if** slik at dersom **tall2** er større enn **tall1**, skriver du ut en beskjed om at **tall1** er størst.
- c) Utvid programmet ditt videre ved hjelp av **elif**, slik at dersom **tall2** er større enn **tall1**, får brukeren beskjed om at **tall2** er størst.
- d) Fullfør til slutt koden ved å legge til en **else**-blokk, slik at dersom tallene er like, vil brukeren få beskjed om det.

Oppgave 5 *Absoluttverdi*

Et reelt tall består av et fortegn og en tallverdi, kalt *absoluttverdi*. Når vi finner absoluttverdien til et tall «fjerner vi fortegnet». Det betyr at absoluttverdien til et tall alltid er positiv. Absoluttverdien til et tall a skrives $|a|$ og er definert som:

$$|a| = \begin{cases} a, & \text{hvis } a \geq 0 \\ -a, & \text{hvis } a < 0 \end{cases} \quad (1)$$

- a) Lag et program som ber brukeren om et tall ved hjelp av **input** og **float**, og skriver ut absoluttverdien av tallet
- b) Lag et program som ber brukeren om to tall og skriver ut hvilket av tallene som har høyest absoluttverdi

Oppgave 6 *Vinkeltyper*

Vi kan dele vinkler inn i tre forskjellige typer:

1. En vinkel som er mindre enn 90° kalles en *spiss* vinkel.
2. En vinkel som er større enn 90° kalles en *stump* eller *butt* vinkel.
3. En vinkel som er akkurat 90° kalles en *rett* vinkel

Be brukeren om å gi en vinkel og bruk **if**, **elif** og **else** til å fortelle brukeren om vinkelen er *spiss*, *stump* eller *rett*

For-løkker

Oppgave 7 *For-løkker*

Bruk en **for**-løkke til å gjennomføre disse oppgavene.

- a) Print meldingen **'Hei, verden'** 5 ganger.
- b) Print alle tallene fra 1 til 101.
- c) Modifiser programmet slik at det også printer ut alle kvadrattallene (x^2) mellom 1 og 10000 (100^2). Husk at du kan opphøye tallet `tall` i andre med kommandoen `tall**2`.

Oppgave 8 *Fakultet*

Si at du har 5 forskjellige små skulpturer du skal sette opp på rekke. Hvor mange ulike måter kan du gjøre dette på? For den første har du 5 valg, deretter har du 4 valg, så 3, og så videre. Dermed blir antall kombinasjoner til

$$5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120.$$

I matematikken skriver vi dette mer kompakt som $5!$, som vi kaller *fakultet*. Å rangere n ting kan altså gjøres på $n!$ måter, som blir

$$n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1.$$

For små tall er det enkelt å regne ut fakultet for hånd, men dersom n begynner å bli litt større vil dette ta fryktelig lang tid.

- a) Du skal nå skrive et program som kan regne ut faktoriell for oss. Du kan enten prøve å gjøre dette helt selv, eller følge vår «steg-for-steg»-instrukser under.
- Spør brukeren om hva n er, lagre svaret i en variabel. Husk å konvertere svaret med `int(input())`.
 - Lag en variabel `faktoriell`, som du gir verdien 1.
 - Lag en for-løkke som går fra 1 til n .
 - For hver iterasjon av løkka, gang `faktoriell`-variabelen din med tallene du løkker over.
 - Skriv ut det endelige svaret så brukeren kan se det.
- b) Test at programmet ditt gir $5! = 120$.
- c) En vanlig kortstokk har 52 unike kort. Hvor mange mulige rekkefølger kan vi få om vi stokker en kortstokk?
- d) I noen kortspill legger man til 2 jokere i kortstokken, slik at det nå er 54 kort i kortstokken. Hvor mange måter blir det nå å stokke kortstokken på?

Oppgave 9 *For-løkker for hånd*

For hver løkke, gå igjennom for hånd og forutsi hva som skrives ut. Etterpå kan du kjøre løkkene og se hva om du hadde rett:

a)

```
1 for tall in range(1, 5):
2     print(tall)
```

b)

```
1 sum = 0
2 for tall in range(1, 5):
3     sum += tall
4     print(sum)
```

c)

```
1 produkt = 0
2 for tall in range(1, 5):
3     produkt *= tall
4     print(produkt)
```

d)

```
1 produkt = 1
2 for tall in range(1, 5):
3     produkt *= tall
4     print(produkt)
```

e)

```
1 x = 1
2 for _ in range(10):
3     x *= 2
4     print(x)
```