

# INF1001 - Obligatorisk innlevering 4

Frist: 26.9.16 12:00

## Introduksjon

Du får her fem oppgaver du skal løse. Oppgave 1 teller to poeng og er en repetisjonsoppgave. De fire andre oppgavene teller ett poeng hver, disse er det ikke forventet at dere kan løse før etter forelesningen mandag 19.9. Oppgaven skal leveres på devilry, hvor hver oppgave skal leveres som en egen python-fil. I en oppgave med flere deloppgaver skal du kun levere en fil. Merk her at du skal levere to filer for oppgave 1, én for a-oppgaven og én for b-oppgaven. Merk at det kun er ett forsøk på poenggivende obligatoriske innleveringer. Hvert program skal starte med en kommentar som forteller hva programmet skal gjøre. Kommenter koden videre underveis mens du jobber med den og lever kode med kommentarer. Les krav til innlevering lengre ned i dokumentet før du starter arbeidet med oppgavene.

## Oppgave 1: Repetisjon

Merk at vi her ønsker at du løser hver oppgave i samme fil slik at hver deloppgave utvider det tidligere programmet. Legg også merke til at oppgave a og b skal leveres i to forskjellige filer!

a)

**Filnavn:** *repetisjon1.py*

1. Definer en prosedyre som heter *multiplisere*. Prosedyren skal hente inn to forskjellige flyttall fra brukerinput, den skal så multiplisere de to tallene med hverandre og så skrive ut resultatet.
2. Definer en prosedyre som heter *dividere*. Prosedyren skal hente inn to forskjellige flyttall fra brukerinput, den skal så dividere det første tallet på det andre tallet. Hvis det andre tallet som blir gitt er 0 så skal du skrive ut en egendefinert feilmelding istedenfor å dividere. Skriv så ut resultatet (såfremt det andre tallet ikke er 0!).

3. Lag en løkke. I løkken skal programmet ta i mot en streng fra brukerinnt. Hvis brukeren skriver inn "m" skal du kalle prosedyren *multiplisere*, hvis brukeren skriver inn "d" skal du kalle prosedyren *dividere* og hvis brukeren skriver inn "s" skal vi gå ut av løkken. Sørg for at programmet gir meningsfulle utskrifter så brukeren forstår hvordan programmet fungerer.

b)

**Filnavn:** *repetisjon2.py*

1. Definer en liste som inneholder verdiene: 43, 12, 10, -34, 94, 22, 34, 99, 10, 54.
2. Bruk en while-løkke til å gå igjennom listen og regn ut produktet av alle elementene i listen (multipliser alle elementene i listen). Skriv ut resultatet.

Synes du denne oppgaven var vanskelig? [Se oppgaver for uke 3.](#)  
Synes du denne oppgaven var lett? [Se utfordringsoppgave 4.08.](#)

## Oppgave 2: For-løkke

**Filnavn:** *for.py*

Denne oppgaven baserer seg på pensum som kommer på forelesningen mandag 19.9. Merk at vi her ønsker at du løser hver oppgave i samme fil slik at hver deloppgave utvider det tidligere programmet.

1. Definer en liste som inneholder følgende verdier: 6, -4, 7, -2, 8, -3, 9, -11.
2. Bruk en for-løkke til å iterere igjennom alle verdiene i listen og finn den minste verdien. Skriv ut den minste verdien. Gjør dette *uten* å bruke Python sin innebygde min-funksjon.

Synes du denne oppgaven var vanskelig? [Se øvingsoppgave 4.01.](#)  
Synes du denne oppgaven var lett? [Se utfordringsoppgave 4.06.](#)

## Oppgave 3: Funksjoner

**Filnavn:** *funksjoner.py*

Denne oppgaven baserer seg på pensum som kommer på forelesningen mandag 19.9. Merk at vi her ønsker at du løser hver oppgave i samme fil slik at hver deloppgave utvider det tidligere programmet.

1. Lag en funksjon *addisjon* som tar i mot to tall, plusser disse to tallene sammen og returnerer summen av disse to tallene. Test funksjonen ved bruk av følgende setning og la setningen være en del av koden du leverer:

```
assert addisjon(4, 2) == 6
```

2. Lag en funksjon *subtraksjon* som tar i mot to tall og returnerer differansen mellom disse to tallene. Test funksjonen ved bruk av følgende setning og la setningen være en del av koden du leverer:

```
assert subtraksjon(4, 2) == 2
```

3. Lag en funksjon *divisjon* som tar i mot to tall og returnerer det første tallet delt på det andre tallet. Test programmet ved bruk av følgende setning og la setningen være en del av koden du leverer:

```
assert round(divisjon(9.9, 3), 1) == 3.3
```

4. Lag en funksjon *tommer\_til\_cm* som tar i mot antall tommer og returnerer hvor mange centimeter det er. For å regne om fra tommer til centimeter behøver du kun å gange antall tommer med 2.54. Test programmet ved bruk av følgende setning og la setningen være en del av koden du leverer:

```
assert round(tommer_til_cm(12), 2) == 30.48
```

Synes du denne oppgaven var vanskelig? [Se øvingsoppgave 4.03.](#)

Synes du denne oppgaven var lett? [Se utfordringsoppgave 4.07.](#)

## Oppgave 4: Temperatur

**Filnavn:** *temperatur.py*

Denne oppgaven baserer seg på pensum som kommer på forelesningen mandag 19.9. Merk at vi her ønsker at du løser hver oppgave i samme fil slik at hver deloppgave utvider det tidligere programmet.

1. Vi har en fil ([temperatur.txt](#)) som inneholder 12 linjer hvor hver linje er gjennomsnittstemperaturen for en måned i et år. Første linje tilsvare januar, andre linje tilsvare februar, osv. Vi har rundet av temperaturene til heltall.
2. Les inn temperaturene en etter en og putt verdiene inn i en liste.
3. Skriv en funksjon som tar i mot en liste og returnerer gjennomsnittet av verdiene i listen ved hjelp av en for-løkke. Kall denne funksjonen med listen av temperaturer og skriv ut resultatet.

Synes du denne oppgaven var vanskelig? [Se øvingsoppgave 4.05.](#)

Synes du denne oppgaven var lett? [Se utfordringsoppgave 4.09.](#)

## Oppgave 5: Egen oppgave 3

**Filnavn:** *min\_oppgave3.py*

Denne oppgaven baserer seg på pensum som kommer på forelesningen mandag 19.9.

1. Skriv oppgavetekst til en oppgave som handler om innlesing fra fil og funksjoner med returverdier. Skriv oppgaveteksten som kommentarer i Python-filen.
2. Løs oppgaven!

Du skal levere både oppgaveteksten og besvarelsen.

## Krav til innleveringen

1. Oppgaven må kunne kjøres på IFI sine maskiner.
2. Kun .py-filene og README.txt skal leveres inn.
3. Koden skal inneholde gode kommentarer som forklarer hva programmet gjør.
4. Programmet skal inneholde gode utskriftssetninger som gjør det enkelt for bruker å forstå.

## Fremgangsmåte for innlevering i INF1001

1. Lage en fil som heter README.txt. Følgende spørsmål skal være besvart i filen:
  - Hvordan synes du innleveringen var? Hva var enkelt og hva var vanskelig?
  - Hvor lang tid (ca) brukte du på innleveringen?
  - Var det noen oppgaver du ikke fikk til? Hvis ja:
    - Hvilke(n) oppgave er det som ikke fungerer i innleveringen?
    - Hvorfor tror du at oppgaven ikke fungerer?
    - Hva ville du gjort for å få oppgaven til å fungere hvis du hadde mer tid?
2. Logg inn på Devilry.
3. Lever alle .py-filene samt README.txt i *samme innlevering*.
4. Husk å trykke lever og sjekk deretter at innleveringen din er komplett.

Den obligatoriske innleveringen er minimum av hva du bør ha programmert i løpet av en uke. Du finner flere oppgaver for denne uken her.