

# IN1000 Obligatorisk innlevering 1

**Frist for innlevering:** 28.08.23, kl. 23:59

**Sist endret:** 22.08.23

## Introduksjon

Velkommen til første obligatoriske oppgave i IN1000! Før du går i gang med å løse oppgaven bør du ha tatt forkurs, deltatt på ukens seminartime og jobbet deg gjennom relevante Trix-oppgaver. Om du står fast kan du se nærmere på (og eventuelt få hjelp med) Trix-oppgavene det lenkes til under hver oppgave.

Det er tre (fire) deloppgaver som skal løses. For å bestå obligen må oppgave 0-2 være besvart og koden må kjøre uten feil. Les gjennom hver oppgave før du begynner å programmere, og forsøk gjerne å løse oppgavene på papir først!

For hvert program du skriver skal du legge ved en kommentar i toppen av filen som forklarer hva programmet gjør. Videre forventes det at du kommenterer koden underveis, så det blir tydelig hva du har tenkt. Dersom oppgaven sier du skal endre programmet trenger du kun å levere den endrede løsningen. Andre viktige krav til innleveringen og beskrivelse av hvordan du leverer finner du nederst i dette dokumentet.

**NB:** Hvis du enda ikke vet hvordan du bruker en teksteditor eller kjører et program i terminalen er det viktig at du går gjennom forkurset i informatikk som du finner [her](#).

## Læringsmål

Målet for disse oppgavene er at du skal ha kommet i gang med programmeringen. Du skal dessuten vise at du kan lagre variabler med verdier og gjøre beslutninger ved hjelp av *if*-setninger.

## Oppgave 0: Filsystem og kjøring av Python-programmer

Ta bilde av skjermen din som viser navigering i filsystemet i terminalen.

**Filnavn:** *filssystem.jpg*

1. Åpne terminalen din og start i roten av filsystemet.
2. Finn frem til IN1000-mappen din ved hjelp av kommandoene `cd`, `ls` og `pwd`.
3. Bruk `ls`-kommandoen i terminalen til å skrive ut innholdet i IN1000-mappen din.
4. Ta skjermbilde av terminalvinduet ditt som viser hva du har gjort, og lever dette.

Synes du denne oppgaven var vanskelig? Ta en titt på eksemplene fra forkurset.

## Oppgave 1: Utskrift og innlesing med variabler

**Filnavn:** *variabler.py*

1. Lag en fil ved navn *variabler.py*.
2. Skriv et program i denne filen som skriver ut "Hei Student!" til terminalen.
3. Endre programmet slik at du ber brukeren om å oppgi et navn i form av en tekststreng ved hjelp av funksjonen *input()*, og lagre denne verdien i en variabel *navn*. Skriv så ut "Hei " og variabelen *navn*.
4. Utvid programmet ditt med to variabler. Du kan velge variabelnavn selv, men gi begge variablene heltallsverdier. Skriv ut variablene på hver sin linje.
5. Beregn differansen av de to variablene (den første minus den andre) og legg resultatet inn i en ny variabel. Skriv ut "Differanse:" etterfulgt av denne tredje variabelen.
6. Be bruker om å oppgi et nytt navn, og legg svaret i en ny variabel. Lag nok en variabel ved navn *sammen*, og gi den verdien av det første navnet etterfulgt av det andre navnet. Skriv ut *sammen* på en ny linje.
7. Du skal nå endre verdien av variabelen *sammen*. Dette skal du gjøre ved å slå sammen de to navnene som i forrige deloppgave, men denne gangen skal du legge til "og" med et mellomrom på hver side mellom dem. For eksempel: Dersom *sammen* først hadde verdien "OlaKari" skal den nå ha verdien "Ola og Kari".

**Viktig:** Du skal utvide programmet ditt ved å endre verdien av variabelen *sammen* på en ny linje, ikke endre linjen der du først definerte variabelen.

Synes du denne oppgaven var vanskelig? Tren på Trix-oppgaver [1.01](#), [1.04](#), [1.06](#), [1.07](#) og [1.10](#)

Synes du denne oppgaven var enkel? Se Trix-oppgaver [1.05](#), [1.12](#) og [1.13](#)

## Oppgave 2: Beslutninger

Filnavn: *beslutninger.py*

1. Skriv et program som ber brukeren om å svare "ja" eller "nei" på om de kunne tenke seg en brus. Lagre svaret i en variabel.
2. Skriv en if-setning som gjør en beslutning ut fra hva brukeren har skrevet inn:
  - a. Hvis brukeren har svart "*ja*" skal programmet skrive ut "*Her har du en brus!*"
  - b. Hvis brukeren har svart "*nei*" skal setningen "*Den er grei.*" skrives ut.
  - c. Hvis brukeren har svart noe helt annet skal programmet skrive ut "*Det forstod jeg ikke helt.*"

Synes du denne oppgaven var vanskelig? Se Trix-oppgave [1.08](#) og [1.09](#)  
Synes du denne oppgaven var enkel? Se Trix-oppgave [1.13](#), [1.14](#) og [1.16](#)

## Oppgave 3: Problemløsning med beslutninger

Denne oppgaven kreves ikke for godkjenning, men det er sterkt anbefalt at du prøver å løse den.

Filnavn: *dato.py*

1. Skriv et program som ber om og leser inn to datoer, angitt med heltall for dag og måned (to variabler for hver dato). Eksempel 24 og 12 for 24. desember. Du kan anta at brukeren skriver inn en gyldig dato.
2. Skriv en if-setning som sjekker hvilken dato som kommer først i samme år:
  - a. Hvis den første datoen kommer først skal programmet ditt skrive ut *“Riktig rekkefølge!”*
  - b. Hvis den siste datoen som skrives inn er en tidligere dato, skal programmet skrive ut *“Feil rekkefølge!”*
  - c. Om datoene er like skrives *“Samme dato!”* ut.
3. Skriv et nytt program i filen *dato2.py* som gjør det samme som i punkt 2, men der brukeren skal skrive hver dato som ett heltall i stedet for to. Gi brukeren eksempel på hvordan datoen bør skrives for å sikre at sammenligningen i programmet ditt kan gjøres enklere enn i punkt 2.

Synes du denne oppgaven var vanskelig? Se Trix-oppgave [1.09](#) og [1.11](#)  
Synes du denne oppgaven var enkel? Se Trix-oppgave [1.14](#), [1.15](#), [1.17](#) og [1.18](#)

## Krav til innlevering

- Kun .jpg-filen og .py-filene skal leveres inn.
- Koden skal inneholde gode kommentarer som forklarer hva programmet gjør.
- Programmet skal inneholde gode utskriftsetninger som gjør det enkelt for bruker å forstå.

## Hvordan levere oppgaven

1. Du skal IKKE lage en .zip-fil
2. Du skal også fylle inn kommentarfeltet i devilry når du leverer oppgaven din.  
Følgende spørsmål **skal** være besvart:
  - a. Hvordan synes du innleveringen var? Hva var enkelt og hva var vanskelig?
  - b. Hvor lang tid (ca) brukte du på innleveringen?  
Var det noen oppgaver du ikke fikk til? Hvis ja:
    - i. For hvilke(n) oppgave fungerer ikke det innleverte programmet ditt ?
    - ii. Hvorfor tror du programmet ikke fungerer?

iii. Hva ville du gjort for å få programmet til å fungere hvis du hadde mer tid?

3. Logg inn på [Devilry](#).
4. Lever *.jpg*-filen og alle *.py*-filene i samme innlevering, samt fyll inn kommentarfeltet med svar på spørsmålene i punkt 2.
5. Husk å trykke **lever** og sjekk deretter at innleveringen din er komplett.
6. Den obligatoriske innleveringen med tilhørende Trix-oppgaver er minimum av hva du bør ha programmert i løpet av en uke. Du finner flere oppgaver for denne uken [her](#).