

# Prøve i Python-programmering

---

**Varighet:** 180 minutter

**Tillatte hjelpemidler:** Ingen

---

## Del 1: Korte kodesnutter (10 oppgaver)

### Oppgave 1

Hva vil følgende kode skrive ut? Forklar hvorfor.

```
x = 5
y = 10
print(x + y)
```

---

### Oppgave 2

Finn feilen i følgende kode og rett den:

```
for i in range(5)
    print(i)
```

---

### Oppgave 3

Hva er verdien av `result` etter at følgende kode er kjørt?

```
a = 10
b = 3
result = a % b
```

---

### Oppgave 4

Fullfør koden slik at den skriver ut tallene fra 1 til 10:

```
for i in range(    ):
    print(i)
```

---

## Oppgave 5

Forklar hva følgende kode gjør:

```
navn = ["Anna", "Bernt", "Cecilie"]
for n in navn:
    print("Hei, " + n + "!")
```

---

## Oppgave 6

Hva er output av følgende kode? Forklar:

```
tall = [1, 2, 3]
tall.append(4)
print(tall)
```

---

## Oppgave 7

Finn og rett feilen i koden:

```
if x = 10:
    print("x er lik 10")
```

---

## Oppgave 8

Hva blir resultatet av følgende kode?

```
liste = [1, 2, 3, 4]
print(liste[4])
```

---

## Oppgave 9

Hva vil følgende kode skrive ut? Forklar hvorfor.

```
liste = [0, 1, 2, 3, 4]
print(liste[2] + liste[-1])
```

---

## Oppgave 10

Avgjør hva som blir skrevet ut.

```
person = {"navn": "Ola", "alder": 25}
print(person["alder"])
print(person.keys())
print(person.values())
print(person[list(person.keys())[-1]])
```

Her er Del 2 av prøven:

## Del 2: Programmeringsoppgaver (6 oppgaver)

### Oppgave 1

Fullfør det følgende programmet som ber brukeren om et tall, og skriver ut om tallet er positivt, negativt eller null.

```
# Be brukeren om å taste inn et tall
tall = # Din kode her

# Sjekk om tallet er positivt, negativt eller null
if # Din kode her:
    print("Tallet er positivt")
elif # Din kode her:
    print("Tallet er negativt")
else:
    print("Tallet er null")
```

### Oppgave 2

Fullfør programmet nedenfor slik at det skriver ut alle partall fra 0 til og med 20.

```
# Bruk en løkke til å iterere gjennom tallene
for i in range( ):
    # Sjekk om tallet er et partall
    if # Din kode her:
        print(i)
```

### Oppgave 3

Fullfør programmet som tar en liste med tall og beregner summen av tallene i listen.

```
tall_liste = [3, 5, 7, 9, 11]

# Initialiser en variabel for summen
summen = 0

# Bruk en løkke til å legge til hvert tall i summen
for tall in tall_liste:
    # Din kode her

print("Summen av tallene er:", summen)
```

---

## Oppgave 4

Skriv et program som ber brukeren om et ord, og deretter skriver ut hver bokstav i ordet på en egen linje.

**Eksempel på kjøring:**

```
Skriv inn et ord: Python
P
y
t
h
o
n
```

---

## Oppgave 5

Lag et program som fungerer som en enkel ordbok. Programmet skal:

- Ha en predefinert dictionary med minst tre engelske ord som nøkler og deres norske oversettelser som verdier.
- Be brukeren om å skrive inn et engelsk ord.
- Skrive ut den norske oversettelsen hvis ordet finnes i ordboken.
- Hvis ordet ikke finnes, skrive ut en melding om at ordet ikke er i ordboken.

**Eksempel på kjøring:**

```
Skriv inn et engelsk ord: apple
Den norske oversettelsen er: eple
```

---

## Oppgave 6

Skriv et program som teller antall forekomster av hver bokstav i en gitt tekst. Programmet skal:

- Be brukeren om å skrive inn en tekst.
- Bruke en dictionary til å telle antall forekomster av hver bokstav.
- Skrive ut resultatet.

#### Eksempel på kjøring:

```
Skriv inn en tekst: hei
h: 1
e: 1
i: 1
```

---

## Del 3: Større programmeringsoppgave

---

### Oppgaven: Høstepler på Eplegården

Du skal lage et program for en eplegård som trenger hjelp til å administrere eplehøsten sin. Programmet skal hjelpe gården med å holde oversikt over innhøstede epler, sortere dem etter størrelse, beregne total vekt, og mer. Oppgaven er delt opp i fem deler, og du skal fullføre hver del i rekkefølge.

#### Del 1: Registrere Epler

Lag en liste som heter `epler` for å lagre informasjon om hvert eple som blir plukket. Hvert eple skal representeres som en dictionary med følgende nøkkelverdipar:

- `vekt`: vekt i gram (float)
- `diameter`: diameter i centimeter (float)
- `sort`: navnet på eplesorten (string)

Skriv en funksjon `registrer_eple()` som:

- Ber brukeren om å taste inn vekt, diameter og sort for et eple.
- Oppretter en dictionary med denne informasjonen.
- Legger dictionaryen til i listen `epler`.

---

#### Del 2: Vise Oversikt over Epler

Lag en funksjon `vis_oversikt()` som:

- Går gjennom listen `epler` og skriver ut informasjon om hvert eple i et lesbart format.
- Viser totalt antall epler som er registrert.

#### Eksempel på output:

```
Eple 1: Sort: Rød Aroma, Vekt: 150.0g, Diameter: 7.5cm
Eple 2: Sort: Gravenstein, Vekt: 180.0g, Diameter: 8.0cm
Totalt antall epler: 2
```

---

### Del 3: Beregne Total Vekt

Utvid programmet med følgende funksjoner:

- `beregn_total_vekt()`: Beregner og returnerer den totale vekten av alle eplene.

Skriv ut resultatene på en brukervennlig måte.

#### Eksempel på output:

```
Total vekt av alle epler: 330.0 gram
Gjennomsnittlig diameter: 7.75 cm
```

---

### Del 4: Sortere Epler etter Vekt

Legg til en funksjon `tunge_epler()` som:

- Teller opp antall epler som er tyngre enn 200 g, og skriver ut antallet.

---

### Del 5: Finne Mest Vanlige Eplesort

Lag en funksjon `finn_mest_vanlige_sort()` som:

- Bruker en dictionary til å telle antall forekomster av hver eplesort.
- Finner og returnerer den eplesorten som forekommer mest.
- Skriver ut hvilken sort som er mest vanlig, og hvor mange ganger den forekommer.

#### Eksempel på output:

```
Den mest vanlige eplesorten er 'Gravenstein' med 5 forekomster.
```

---

### Ekstra utfordring (valgfri):

- La brukeren registrere så mange epler de ønsker (oppgave 1). Avslutt registreringen når brukeren ikke ønsker å legge inn flere epler. **Tips:** Du kan bruke en while-løkke som fortsetter å spørre brukeren om de vil legge til flere epler.
- Lag en meny som lar brukeren velge mellom de ulike funksjonene (registrere epler, vise oversikt, beregne total vekt osv.).

---

**Lykke til!**