GRUPPE 7

Velkommen til
IN1000 - Uke 6

Læringsmål

- Begrense skopet til variabler og bruke parameteroverføring for å sende verdier dit de trengs i programmer.
- Repeter og forstå konseptene fra og med uke 1 til og med uke 5

Skop

- Alle variabler, funksjoner eller klasser opprettet inni en funksjon eller prosedyre er i et eget skop
- Funksjoner kan ikke anvende variabler som tilhører andre funksjoner sitt skop
- Alt som ikke er opprettet i en funksjon eller prosedyre er i det globale skop, og kan aksesseres av alt

```
x = 10

def funksjon():
    global x
    print(x)
    y = 20


funksjon()
print(y)
```

Unngå å bruke det globale skop

```python
tall = 8

def print_dobbel():
    global tall
    print(tall * 2)

print_dobbel()
```

```python
def dobbel(t):
    print(t * 2)

def hovedprogram():
    tall = 8
    dobbel(tall)

hovedprogram()
```

Er det noen temaer dere ønsker å gå repetere?

- Løkker
- Funksjoner
- Samlinger

```
def minus():
    a = 3
    b = 2


def summer():
    print("sum: ", a + b)


summer()
```

Oppgave 1

- Hvorfor fungerer ikke programmet?
- Endre programmet slik at det fungerer. Dere skal ikke bruke globale variabler

```python
def legg_til_bokstav(ordliste, bokstav):
    for ord in ordliste:
        ord += bokstav
    return ordliste


def hovedprogram():
    ordliste = ["Je", "de", "se"]
    ordliste = legg_til_bokstav(ordliste, "g")
    print(ordliste)


hovedprogram()
```

Oppgave 2

- Hvorfor endres ikke ordene i programmet? Diskuter
- Endre programmet slik at ordene endres. Ikke bruk globale variabler

```python
def hovedprogram():
    grense = -5
    grenseteller = 0


    while grenseteller <= 0:
        print("Har vi nådd grensen?", grenseteller == grense)
        grenseteller -= 1


hovedprogram()
```

Oppgave 3

- Hvorfor slutter programmet aldri å kjøre?
- Endre programmet slik at det stopper når grensen er nådd

```python
def bygg_samling(antall_rader, antall_kolonner):
    samling = []

    while antall_rader > 0:
        rad = []

        for kolonnenr in range(antall_kolonner):
            rad.append(kolonnenr)

        samling.append(rad)
        antall_rader -= 1

    return samling

def hovedprogram():
    samling = bygg_samling(7, 5)

hovedprogram()
```

Oppgave 4

- Hva slags samling er det programmet ovenfor bygger, og hvordan ser samlingen ut? Diskuter uten å kjøre programmet.
- Hva er oppgaven til while-løkka og hva er oppgaven til for-løkka?
- Utvid programmet med en funksjon, *finn_antall*, som tar inn den bygde samlingen og et tall. Funksjonen skal returnere antall ganger det oppgitte tallet finnes i samlingen. Hovedprogrammet skal skrive ut dette antallet.

```python
def hent_navn(personbok, personid):
    if personid in personbok:
        return personbok[personid]
    return None

def hovedprogram():
    personbok = {
        "01020300000": "Hjørdis Olsen",
        "02030400000": "Martin Halvorsen",
        "03040500000": "Kristin Andersen"
    }

    personid = "02030400000"
    navn = hent_navn(personbok, personid)
    startbokstav = "M"

    if navn:
        print("Fant person ved navn", navn + ".")
        if navn[0] == startbokstav:
            print("Navnet starter på", startbokstav + ".")
    else:
        print("Fant ingen person med id", personid + ".")

hovedprogram()
```

Oppgave 5

- Hva skrives ut her?
- Hva skrives ut hvis personid ikke finnes I personbok?
- Endre programmet, slik at det ikke lenger inkluderer nøstede beslutninger. Programmet skal likevel gjøre det samme etter endringen som før endringen.