IN1000

UKE 6 - GRUPPE 1

MIAU@IFI.UIO.NO

Skop

- Den delen av programmet du kan aksessere/få tilgang på variabel
- Skiller mellom globale og lokale variabeler

```python
def funksjon():
    x = 5 # lokal variabel

funksjon()
print(x)
```

```python
x = 5 # global variabel

def funksjon():
    print(x)

funksjon()
```

Skop

Vi bruker helst lokale variabler

```
tall1 = 23
tall2 = 24

def størst_av_to():
    if tall1 > tall2:
        return tall1

    return tall2
```

```
def størst_av_to(tall1, tall2):
    if tall1 > tall2:
        return tall1

    return tall2

print(størst_av_to(23, 34))
```

Skop

Hvorfor fungerer ikke dette?

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

def summer():
    print(f'Sum: {a + b}')

variabler()
summer()
```



Oppgave 1

Tenk at dere har fått utdelt en fil med gruppeoppmøte i IN1000. Filnavn: oppmøte_in1000.txt. Skriv et program som ved hjelp av en funksjon leser inn filen og skriver ut gruppene som har hatt 20 eller flere oppmøtte studenter i uke 11. Funksjonen har filnavn som parameter.

Første linje i filen består av gruppenummer: gruppe_1, gruppe_2, gruppe_3, o.s.v., til og med gruppe_8. Gruppene er atskilte med ; (semikolon). På linjene nedenfor står antall oppmøtte i hver gruppe per uke. Antallene er atskilte med ; (semikolon). Andre linje i fila har oppmøteantall for uke 1, tredje linje i fila har oppmøte for uke 2, o.s.v., til og med uke 12.

Her er et eksempel på hvordan de to første linjene i oppmøte_in1000.txt kan se ut:
gruppe_1; gruppe_2; gruppe_3; gruppe_ 4; gruppe_5; gruppe_6; gruppe_7; gruppe_8
10; 19; 25; 18; 23; 11; 13; 17

Oppgave 2

Du ønsker hjelp til å gruppere sammen personer som har felles interesse.

Skriv en funksjon, **lag_interessegrupper(personers_interesse)**, som tar som argument en ordbok hvor nøklene er strenger som er navn på personer, mens hver verdi er en streng som sier hvilken interesse personen har.

Funksjonen skal returnere en ordbok, hvor hver nøkkel er en interesse (streng) og hver verdi er en liste av personer som var oppført med denne interessen i ordboken personers_interesse. Alle interessene som finnes i ordboken personers_interesse skal være med i den returnerte ordboken.

Altså skal for eksempel kallet lag_interessegrupper({"Per":"Mat", "Palle":"Film", "Espen":"Mat"}) evaluere til ordboken {"Mat":["Per","Espen"], "Film":["Palle"]}

Utvid programmet, slik at det lager en fil med interesser og antall personer som har hver interesse. Hver linje skal bestå av interesse og antall, atskilte med komma. Filen skal hete *interesseantall.txt*.

Oppgave 3

Et land har følgende regler for reisekarantene:

- Dersom en person er vaksinert, trenger personen ikke å gå i karantene (0 dager i karantene)
- Dersom en uvaksinert person har vært i et land som har fargekode rød eller oransje, skal personen 10 dager i karantene.
- Dersom en uvaksinert person har vært i et land som har fargekode grønn, skal personen 3 dager i karantene.

Skriv en funksjon, **karantene(vaksinert, farge)**, som returner antall dager (heltall) en person må i karantene i henhold til reglene ovenfor. Parameteren vaksinert er enten True eller False (boolsk verdi) og parameteren farge er enten "rød", "oransje" eller "grønn" (streng).

Altså skal for eksempel kallet karantene(True, "roed") returnere 0, mens kallet karantene(False, "oransje") skal returnere 10.

Utvid programmet i a), slik at det leser inn en fil ved navn *karantene.txt*. Hver linje i filen består av vaksinasjonsstatusen til en person (True eller False) og fargen på landet de har vært i (rød, oransje eller grønn). Vaksinasjonsstatus og farge er atskilte med komma. Videre skal programmet summere og skrive ut antall karantenedager fra henholdsvis røde, oransje og grønne land (antall for hver farge separat samt antall karantenedager totalt).