

## Oblig 2 – Programmering 1

### TEORI:

- Teorioppgave 1 – *Forklar hva en if-test/if-statement er med egne ord. Beskriv også sammenhengen mellom if, elif og else.*

Svar: En if-statement er en måte vi kan utføre ting på i vår kode basert på spesifikke betingelser. Vi setter betingelser for noe som enten blir true eller false og får deretter kjørt forskjellig kode avhengig av hva utfallet er. Vi kan sette opp en betingelse, altså *if*, for eksempel *x* er et positivt tall så ønsker vi å printe ut «positiv». Omvendt, hvis *x* er et negativt tall ønsker vi å printe ut «negativ». Da kjører forskjellig kode basert på oppfylt(e) kriterium eller ikke oppfylte kriterium.

Sammenhengen mellom *if*, *elif* og *else* er at vi ved hjelp av disse kan ta i bruk flere ulike betingelser som vår kode skal ta stilling til for å se hva den skal kjøre. Vi starter som vanlig med *if* og sjekker om den betingelsen er true. Er den det så kjører koden vi har spesifisert skal kjøre. Men hvis den betingelsen er false så kan vi sette opp annen kode som skal kjøre dersom dette skjer. Her kommer *elif* (*else if*) inn i bildet. Hvis den første betingelsen (*if*) er false så går programmet videre og sjekker om den neste spesifiserte betingelsen er true. Hvis denne er det så kjøres koden som er spesifisert under *elif*.

Så kommer vi eventuelt til *else*. Dette blir en default-kode som skal kjøre dersom både betingelse 1 og 2, altså *if* og *elif*, er false. Da går programmet videre og kjører det vi har spesifisert under *else*.

Samlet sett så er if-statements noe som kan gjøre vår kode mer fleksibel og funksjonell med tanke på mulighetene til å sjekke for veldig mange parametere.

- Teorioppgave 2 – *Forklar hva lister er i python og hva de kan brukes til.*

Svar: Lister er en måte vi kan organisere og samle dataen vi har. Hvis/når vi jobber med større datamengder er det fint å kunne samle denne dataen et sted, da kan en liste være smart. Fra listen kan vi så hente ut data, enten hele listen eller kun de elementene vi ønsker. Da bruker vi elementenes indeksplassing i listen. Vi kan også modifisere listen. Legge til, slette, endre, sortere, for eksempel. Dette gjør listene dynamiske. Det er også fint å bruke

lister og loops sammen, til eksempel for å iterere over alle elementer i en liste. Listene kan inneholde f.eks. tall, strings, eller andre lister.

- Teorioppgave 3 – *Løkker. Forklar hva løkker er og gi noen eksempler på tilfeller man kan ønske å benytte dem*

Svar: Loops(løkker) er en måte vi kan kjøre kode på gjennom flere omganger – «loope» gjennom. Vi kan ved hjelp av løkker gå gjennom datatyper vi har i python, f.eks. lister. Dette kalles å iterere gjennom en liste. Dette gjør vi ofte med en for-loop. Vi kan ved hjelp av denne løkken utføre forskjellige ting etter hva vi ønsker. Vi kan sjekke hvor mange elementer det er av noe i en liste ved hjelp av en løkke, vi kan gå gjennom et dataset og f.eks. printe ut utvalgte verdier.

While-loopen vil kjøre så lenge det som er satt som betingelsen evaluerer til «true». Vi kan f.eks. ha et matematisk regnestykke som skal evaluere til en maks-verdi, og når denne verdien oppnås stopper koden å kjøre. Basert på dette kan vi så utføre en ønsket handling for hver iterasjon som regnestykket evaluerer til «true».

Løkkene har egenskapene til å automatisere og effektivisere det vi ønsker å utføre, de kan være med på å hjelpe oss skrive mindre kode.