# If-tester

Vanligvis kjøres alle kodelinjer i et program i den orden de er skrevet i. Med if-tester kan vi bestemme hvilke kode som skal kjøres og hvilke ikke, og på den måte påvirke hva programmet skal gjøre.

|  |
| --- |
|  |

Vi lager en test i Python med kontrollordet «if», etterfulgt av et <kriterium> og et

kolon. Så kommer kode som skal utføres hvis utsagnet evalueres til True (er sant). Dette kan kombinere med kontrollordene elif, som også må etterfulges av et <kriterium> og avsluttes med kolon. Til slutt kan vi bruke else som kjøres hvis ingen av kriterium inntreffes.

Kontrollordene if, elif og else skal ikke være innrykket, men resten av koden skal være innrykket ett nivå.

Et kriterium er et utsagn som vi kan uttrykke med variabler og logiske operatorer og som kan være sann eller usant.

|  |
| --- |
| Et bilde som inneholder tekst  Automatisk generert beskrivelse |

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Man kan teste dette i konsollvindu (heter «Shell» i Thonny).   
Prøv å skrive forskjellige kriterier og sjekk om de er «True» (sann)   
eller «False» (usant)

Eksempel på bruk av if-tester og kriterium

|  |
| --- |
|  |

I eksempelet over blir output «30 er større enn 10», siden verdien av variabelen

*a* er større enn verdien av variabelen *n*. Da blir utsagnet a > n tolket til True,

og linje 4 blir utført.

Thonny sørger i utgangspunktet for riktig innrykk automatisk hvis du husker å skrive kolon i linje over, men det kan være nødvendig å rette opp manuelt. Det kan være mer enn en linje i kode som skal utføres under kontrollordene. I så tilfelle må alle linjer være innrykket.

# Oppgaver

1. Lag et program som ber brukeren om to tall og sjekker om tallene er like. Programmet skal skrive resultatet til skjerm.
2. Skriv et program som ber brukeren om et tall og så svarer om det er et oddetall eller partall.
3. Lag et program som ber brukeren om to tall og sjekker om tallene har forskjellig fortegn. Programmet skal skrive resultatet til skjerm
4. Lag et program som ber brukeren om to tall og sjekker hvilke tall er større og skriv resultatet til skjerm.
5. Lag et program som tar inn sidelengdene til en trekant.​ Programmet skal sjekke om trekanten er likesidet, likebent og/eller rettvinklet

# Løkker – kodegjentagelse

Datamaskinens fortrinn kommer tydelig til syne når vi ser hvordan vi kan implementere

kodegjentakelse. Er det en ting datamaskiner og programmering er flink til er å gjøre ting om og om igjen.

Vi skal se på to måter å programmere repeterende kode, for-løkker og while-løkker.

|  |
| --- |
|  |

# while-løkke

Syntaksen for en while-løkke er kontrollordet «while», etterfulgt av en logisk test, altså et utsagn som blir evaluert til True eller False, og et kolon. Instruksjonene i while-løkka blir utført så lenge utsagnet blir evaluert til True. Alle instruksjoner i løkka skal rykkes inn ett nivå.

|  |
| --- |
|  |

While løkke trenger som oftest en variable med startverdi, som obligatorisk må oppdateres inn i kode som kjøres i while-løkke (linjer 1 og 6). Hvis du ikke oppdaterer variabel vil du havne i en «evig-løkke» (bare teste det 😊)

Du kan bruke whileløkker mange ting. F.eks kan vi bruke det til å lage og skrive ut verditabell til en funksjon: vi regner funksjonsverdien for en x-verdi og skriver det i en tabell gjentatte ganger. Se eksempel nede:

Et bilde som inneholder tekst

Automatisk generert beskrivelse

# for-løkke

Når vi på forhånd ikke vet hvor mange ganger en løkke skal kjøre, må vi bruke

while-løkke. Når vi derimot vet hvor mange ganger koden skal gjentas, bruker

vi en for-løkke. Syntaksen er kontrollordet «for» etterfulgt av en fritt valgt variabel,

kontrollordet «in» og til slutt et objekt som kan itereres over, for eksempel

en mengde, og et kolon. Alle instruksjoner i løkka skal rykkes inn ett nivå.

|  |
| --- |
|  |

Vi kan stor sett gjør det samme med en for-løkke og en while-løkke, men avhengig av problemet vi prøver å løse vil noen ganger lønne seg å bruke den ene fram for den andre.

## range()

Vi skriver for-løkker på følgende måte:

|  |
| --- |
|  |

Det er funksjon range() som er ansvarlig for å skape en sekvens med verdien til variabel i for-løkke, Hvis du utelater første argument i range, antar Python at første argument er 0. Dersom du legger til et tredje argument, angir dette steglengden i intervallet.

Et bilde som inneholder tekst

Automatisk generert beskrivelseEt bilde som inneholder bord

Automatisk generert beskrivelse

Indeksen trenger ikke brukes inni løkka. Da brukes range til å bestemme hvor mange ganger løkka skal utføres.

Eksempel med verditabell blir enklere med for-løkke enn med while-løkke:

Et bilde som inneholder tekst

Automatisk generert beskrivelse

For-løkker kan også brukes med andre objekter enn de vi lager med range.

Noen datatyper, som strenger og lister, kan man iterere over direkte.

|  |  |
| --- | --- |
|  |  |