# Keywords

* **And** = kan brukes i en if, løkke osv, brukes for å si at flere forskjellige ting må være oppfylt
* **Del** = brukes til å slette ting (del vairabel)
* **from** = brukes når vi skal importere noe – from *pakke* import *modul*
* **not** = brukes til å si at ikke dette er sant(not true = false)
* **while** = brukes for å lage en løkke (sammen med en boolean verdi)
* **as =** gir et referanse navn til noe annet
* **elif** = else if statement, brukes hvis en if ikke blir oppfylt og vi vil sjekke om noe annet blir oppfylt
* **global** = brukes til å gjøre en lokal variabel global, så den kan brukes utenfor en funksjon
* or = brukes for eksempel i en if statement hvor vi kan si at noe må være oppfylt *eller*(or) en annen
* **with =** lager ressursen, utfører koden også lukker ressursen
* **assert =** brukes til å teste om en condition er true og gi feilmelding hvis ikke
* **else** = brukes hvis ikke noen av if statementene blir oppfylt
* **if** = brukes for å sjekke om noe blir oppfylt
* **pass =** sier at vi kan hoppe videre/ut av løkken
* **yield =** som return, men funksjonen vil returnere en generator(en iterasjon som bare kan bli brukt en gang)
* **break =** avslutter fra en while løkke
* **except** = brukes sammen med en try, vi broker den for å kunne fortsette etter en feil(som vi vet kan inntreffe)
* **import** = brukes for å importere en modul
* **print** = brukes for å printe ut tekst til brukeren
* **class** = klasse er en oppskrift/abstrakt som kan gjenbrukes mange ganger som objekter
* **exec** = utfører noe
* **in =** brukes ofte sammen med lister/for løkker, for eksempel sier vi: *for lokalvariabel i(in) liste*
* **raise =** brukt for å bruke egen produserte feil/meldinger(raiser egne feil)
* **continue =** fortsetter den neste løkke gjenomgangen
* **finally =** brukes for å renske opp eller kjøre ferdig, uansett om feil skjer
* **is =** brukes til å sjekke etter identitet (sammenligner om minne addresser er like)
* **return** = returnerer noe fra forekesempel en funksjon
* **def** = brukes til å definer en metode/funksjon
* **for** = brukes til å lage en for løkke, som da går gjennom alle elementene I en liste eller alle verdiene I noe vi setter(10 – går gjennom 1,2,3 osv)
* **lambda =** anonym funksjon(returner alltid noe) g = lambda x: x\*\*2
* **try** = brukes for å prøve å utføre en kode, som vi vet kan gå feil, brukes ofte sammen med except

# Data types

* **True** – var = False
* **False** - var = True
* **None** - var = None
* **Strings** – var = «This is a string»
* **Numbers** –var = 1
* **Floats** -
* **lists** – var = [1,1,2,4,5]

# String Escape Sequences

* \\ = \
* \' = '
* \" = double quote
* \a = ASCII Bell (BEL)
* \b = ASCII Backspace (BS)
* \f = ASCII Formfeed (FF)
* \n = ny linje
* \r = ASCII Carriage Return (CR)
* \t = innrykk
* \v = ASCII Vertical Tab (VT)

# Operators

* + = plusser
* - = minus
* \* = ganger
* \*\* = eksponensiell
* / = dele
* // = floor division
* % = modules
* < = mindre enn
* > = større enn
* <= = mindre enn eller er lik
* >= = større enn eller er lik
* == = er lik
* != = ikke er lik
* <> =
* ( ) =
* [ ] =
* { } =
* @ =
* , =
* := avslutter definisjon syntax/if/løkke osv
* .
* = = setter en variabel til å være det som kommer etter
* ;
* += = plusser på det som kommer etter på variabelen
* -= = trekekr ifra det som kommer etter på variabelen
* \*= = ganger med det som kommer etter på variabelen
* /= = trekker ifra det som kommer etter på variabelen
* //
* %=
* \*\*=