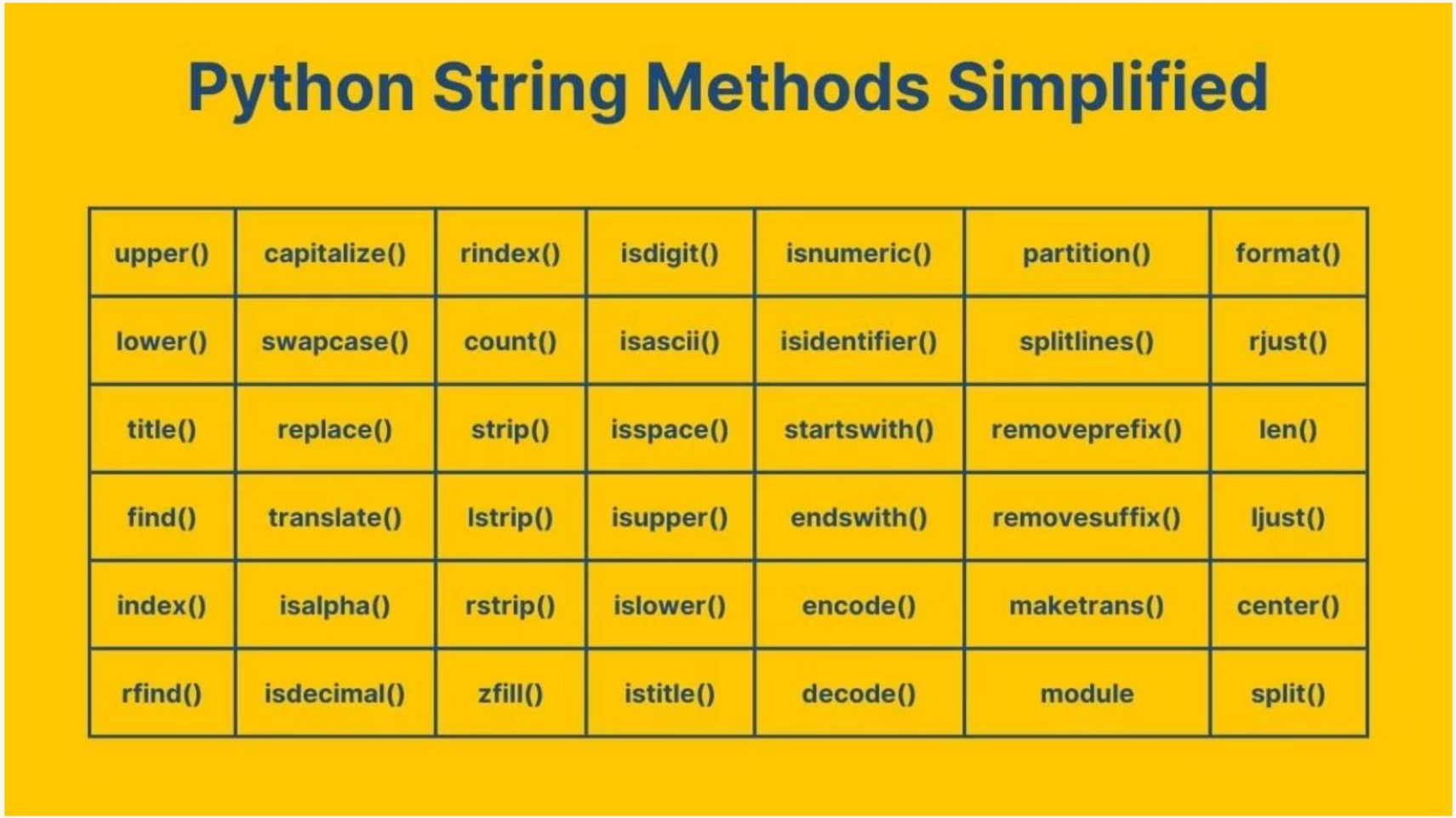


Velkommen til åttende gruppetime!!

Gruppe 8

Uke 4: Løkker

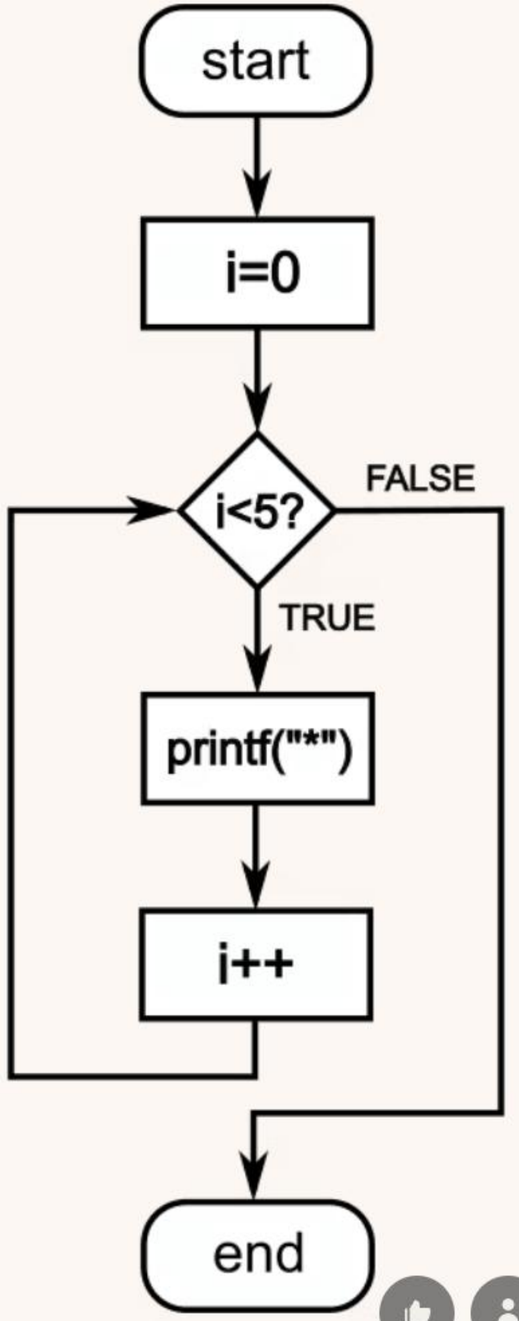


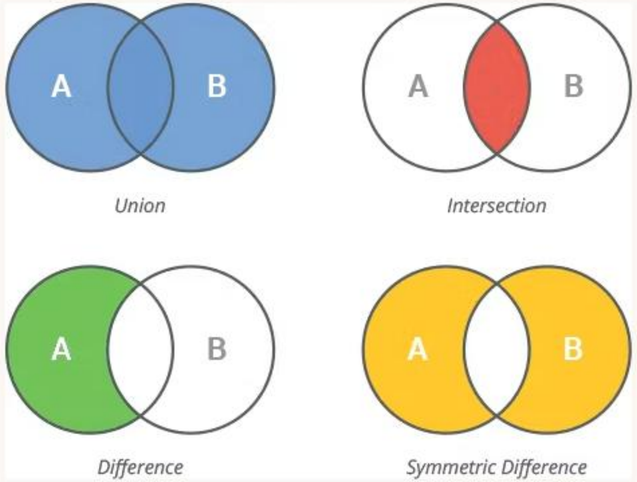
Litt generelt om obliger og strenger i Python

Diskuter: Må en løkke alltid ha en "if" inni seg?

Diskuter om dere kan lage nyttige løkker uten betingelser inni løkken, og om en løkke med "if" alltid er mer nyttig enn en uten

Diskuter: Må en løkke alltid gjøre det samme for hvert element den går gjennom?





Diskuter: Må man alltid gå gjennom hele samlingen når man bruker en løkke?

Diskuter: Kan det være lurt å bruke en løkke til å lage eller bygge opp en ny samling?

Diskuter: Hvis to løkker (som en for og en while) gir samme resultat, er det da noen grunn til å foretrekke den ene fremfor den andre?

Oppsummering fra gruppetime 1 (uke 4)

- **Iterasjon**: Gjenta kode, mulig for hvert element i en samling (lister, mengder, strenger, ordbøker)
- **While-løkke**: Brukes når en **betingelse** styrer varigheten (unngå "while True")
- **For-løkke**: Brukes når du på forhånd vet **antall runder** (eks. gå gjennom en liste)
- **range(start, stop, step)**
  - **start**: Fra og med
  - **stop**: Til (men ikke med)
  - **step**: økning/minking
- **Feiltyper**
  - ***Syntaks***: Manglende kolon eller feil innrykk
  - ***Logikk***: Løkken kjører feil antall ganger
  - ***Kjøretid***: Evige løkker som krasjer programmet

Oppgaver

Også lov å jobbe med oblig 4, trix eller forelesning fra forrige uke

Oppgave 1

Lag et program som teller og skriver ut antallet ulike bokstaver som finnes i et ord.

For eksempel har ordet “heisann” seks ulike bokstaver og “hei” tre ulike bokstaver.

Programmet skal fungere for ethvert ord. Test derfor programmet med ulike ord.
Diskuter hva dere trenger i programmet og hvordan dere vil gå frem for å lage det før dere programmerer.

Oppgave 2

Dere skal fordele gjester langs den ene siden av et langt bord. På forhånd har dere delt gjestene inn i to grupper - introverte og ekstroverte - og ønsker å fordele disse på annenhver plass langs bordet. Det er alltid like mange introverte og ekstroverte gjester, men totalt antall gjester er vilkårlig. Den endelige bordplasseringen med annenhver introverte og ekstroverte gjest er en liste.

Programmet skal avslutte med å skrive ut en ryddig oversikt som viser alle par av introverte og ekstroverte gjester. Husk at en gjest kan inngå i flere par; hvis for eksempel Per som er introvert er plassert mellom ekstroverte Lisa og ekstroverte Sigrid, inngår Per i to par.

Siden totalt antall gjester er vilkårlig, test programmet med forskjellig antall gjester.

Oppgave 3

En butikk har en liste med varelD-er (strenger). Lista inneholder alltid flere duplikater (like verdier) for én eller flere varenummer. Dere er ansatt for å hjelpe butikken med varetelling.

Butikken ønsker seg et program som tar varelD-er fra lista og lager en samling. Samlingen skal vise antallet av hver vare som erstatning for lista med duplikater. De ønsker dessuten en alfabetisk oversikt over varelD-er og antall fra samlingen skrevet ut i terminalen. Hint: bruk operatorene >, <, <=, >= og == for å sammenligne strenger.

Programmet skal fungere for enhver liste med varelD-er (strenger) med flere duplikater. Test derfor programmet med ulike lister.

Oppgave 4 (Utfordring)

Lag et program som fjerner alle nuller (hvis det finnes noen) i starten og slutten av en liste. Nuller som både har andre tall foran og bak seg i lista, skal ikke fjernes. Til slutt skal en liste som oppfyller de nevnte kravene skrives ut. **Eksempler**:

- For listene [0, 0, 0, 0, 0] og [0] skal [] (tom liste) skrives ut.

- For [5] skal programmet skrive ut [5], og for listene [0, 0, 1, 2, 3] og [1, 2, 3, 0, 0] skal [1, 2, 3] skrives ut.

- For [1, 0, 2, 0, 3] skal programmet skrive ut [1, 0, 2, 0, 3] og for [1, 2, 3, 4] skal [1, 2, 3, 4] skrives ut. For [0, 0, 1, 2, 0, 3, 0, 0, 4, 0] skal programmet skrive ut [1, 2, 0, 3, 0, 0,4].

Programmet skal fungere for enhver liste. Test derfor programmet med listene nevnt ovenfor, i tillegg til andre lister.

Oppgave 5 (Utfordring)

Lag et program som finner og fjerner alle like tall som står rett etter hverandre i en liste, slik at det maksimalt står ett eksemplar av hvert tall sammenhengende. Rekkefølgen på tallene skal ellers beholdes. Eksempler:

- For listen [1, 1, 2, 2, 2, 3] skal [1, 2, 3] skrives ut.
- For [5, 5, 5, 5] skal [5] skrives ut.
- For [1, 2, 3, 4] skal [1, 2, 3, 4] skrives ut.
- For [1, 1, 2, 1, 1, 3, 3, 2] skal [1, 2, 1, 3, 2] skrives ut.
- For [] skal [] (tom liste) skrives ut.

Programmet skal fungere for enhver liste med tall. Test derfor programmet med listene nevnt ovenfor, i tillegg til andre lister.

Oppgave 6 (Utfordring)

Lag et program som finner det lengste sammenhengende området med like tall i en liste. Programmet skal skrive ut både tallet som forekommer sammenhengende, og hvor mange ganger tallet står etter hverandre. Dersom flere tall har like lang sammenhengende rekke, holder det å skrive ut ett av dem. Eksempler:

- For listen [1, 1, 2, 2, 2, 3] skal programmet skrive ut at 2 står 3 ganger sammenhengende.

- For [5, 5, 5, 5] skal programmet skrive ut at 5 står 4 ganger sammenhengende.

- For [1, 2, 3, 4] skal programmet skrive ut at ett av tallene står 1 gang sammenhengende.

- For [1, 1, 2, 2, 2, 1, 1, 1] skal programmet skrive ut at 1 står 3 ganger sammenhengende.

Ferdig?

Var du borte forrige uke på torsdag? Se gjennom gruppetimens slides

Har du ikke fått med deg forelesningen fra onsdag forrige uke? Finn den på emnesiden

Gjør ferdig oblig 4 (+ frivillige :D)

Trix