

Teorioppgave 1:

en dictionary er en type liste som lagrer data som nøkkel-verdi-par, hvor nøkkelen “oppfører seg” som en “variabel”. Forskjellen mellom dette og vanlige lister er at vanlige liste bare har verdier. En fordel er at man kan lett “koble sammen” og assosiere ulike verdier sammen, f.eks. Et navn assosiert med et telefonnummer. En ulempe med liste er at den ikke kan håndtere like lett store databaser som krever tilgang til verdier som er relatert til hverandre.

Teorioppgave 2:

funksjoner er nyttige fordi man kan skrive en mengde kode i en funksjon og bruke funksjonen repeterende istedenfor å skrive den samme lange koden mange ganger og få en stor kode. Effektene av å bruke funksjoner er:

- * kortere, effektivisert kode

- * bedre lesbarhet gjennom å navngi funksjonen noe relevant

1B) Denne er hva jeg spurte GPT UiO:

Lag en dictionary med informasjon om en student:

```
student = {  
    "first name": "Ola",  
    "last name": "Nordmann",  
    "favourite course": "Programmering 1"  
}
```

1. Skriv ut studentens fullstendige navn (fornavn og etternavn).
2. Programmatisk endre studentens favorittkurs til å inkludere kursets emnekode: "ITF10219 Programmering 1"
3. Programmatisk legg til en alder for studenten i dictionaryen. Du kan selv velge hva alderen skal være.

Forskjellen mellom min og KI-ens løsning er at KI-en lagde en full-name-variabel med fornavn og etternavn og printet ut den og dette: `print("Oppdatert studentinfo:", student)`

ellers var det likt. KI-en klarte på første forsøk og virket som forventet.

2B) denne er hva jeg spurte GPT UiO:

Definer en funksjon som lager en fin utskrift med et tilfeldig generert tall mellom 0 og 100 (husk at du kan benytte `random.randrange()`). Funksjonen skal ikke ta noen parametere. Kall denne funksjonen noen ganger.

KI-en gjorde oppgaven ikke helt som forventet, men den klarte uansett. KI-en brukte ikke for-løkke og heller skrev `print_random_number()` 3 ganger og brukte ikke “return”.

3B) denne er hva jeg spurte GPT UiO:

Definer en funksjon som heter `print_list()`. Denne funksjonen skal ta i mot en liste som parameter, og printe ut hvert element i denne listen en etter en. Lag deretter kort liste med dine 3 favorittmatretter, og kall funksjonen din med denne listen som parameter.

KI-en klarte oppgaven som forventet og likt som min, den bare glemte "return".

4B) denne er hva jeg spurte GPT UiO:

Lag en funksjon for å regne ut volumet av et tredimensjonalt objekt. Vi lar ting være enkelt og forholder oss bare til enkle verdier for lengde, bredde og høyde. Volumet kan da beregnes med følgende formel: $\text{lengde} * \text{bredde} * \text{høyde}$. Du skal ta lengden, bredden og høyden som individuelle input-parametere for funksjonen, og returnere volumet. Kall funksjonen noen ganger med forskjellige verdier for lengde, bredde og høyde, og skriv ut resultatet av hver utregning.

KI-en klarte oppgaven delvis på en uforventet måte. Input-verdiene er hardkodet men ellers kjørbart.

5.1A, B, C) jeg skrev inn deloppgave A, B og C til GPT UiO

koden er kjørbart og virker. På a) måtte skrive `print(movies)` selv for å sjekke kjørbarheten. På b) brukte jeg `try-except`-kode for å få til default-ratingen fordi jeg valgte å ikke hardkode verdiene men heller brukte input. Samme med resten av koden, fordi jeg valgte input måtte jeg tilpasse koden for input

5.2A, B, C) jeg skrev inn deloppgave A, B og C til GPT UiO

koden er kjørbart og virker. på c) valgte KI å skrive en forenklet kode i return og brukte `sum`-koden. Ellers var det likt idé

5.3A, B, C) jeg skrev inn deloppgave A, B og C til GPT UiO

koden er kjørbart og virker. Kodene var mer eller mindre likt.