IN1000 Obligatorisk innlevering 3

Frist for innlevering: 11.09. kl 12:00

Introduksjon

Innleveringen består av 5 oppgaver, der hver oppgave teller 1 poeng. Les gjennom hver oppgave før du begynner å programmere, og forsøk gjerne å løse oppgavene på papir først! Hvis du sitter fast på en oppgave bør du prøve å løse øvingsoppgavene i Trix (se lenke under hver oppgave) før du spør om hjelp.

Pass på at oppgavene du leverer ligger i riktig navngitte filer, som vist under oppgavetittelen. For hvert program du skriver skal du legge ved en kommentar i toppen av fila som forklarer hva programmet gjør. Videre forventes det at du kommenterer koden underveis så det blir tydelig hva du har tenkt. Andre viktige krav til innleveringen og beskrivelse av hvordan du leverer finner du nederst i dette dokumentet.

Læringsmål

Du skal ha forstått hvordan du kan holde styr på mange verdier ved hjelp av forskjellige samlinger (lister, mengder og mappinger). I tillegg skal du vise at du kan skrive litt mer komplekse programmer med kunnskapene du har tilegnet deg de siste ukene.

Oppgave 1: Lister

Filnavn: *lister.py*

1. Lag en liste fylt med 3 tall du velger selv. Legg deretter til et nytt tall på slutten av lista. Skriv ut det første og tredje elementet i lista.

2. Lag en ny, tom liste. Be deretter brukeren om å oppgi 4 navn, og legg disse inn i lista.

3. Bruk en if-sjekk for å se om brukeren har lagt inn navnet ditt i lista. Hvis brukeren har gjort det skal du skrive ut “Du husket meg!”. Hvis navnet ditt ikke finnes i lista skal du skrive ut “Glemte du meg?”.

4. Lag en ny liste ved å slå sammen de to listene du har laget til nå, og skriv ut den nye listen. Fjern deretter de to siste elementene fra lista, og skriv så ut listen på nytt.

Synes du denne oppgaven var vanskelig? Se Trix-oppgave 3.01.
Synes du denne oppgaven var enkel? Se Trix-oppgave 3.05.

Oppgave 2: Ordbok

Filnavn: *ordbok.py*

1. Lag en ordbok (dictionary) som skal inneholde varer i en butikk og pris. Du skal bruke varenavn som nøkkelverdier og varepriser (i form av flyttall) som innholdsverdier. Opprett ordboka med følgende varer med tilhørende pris: melk - kr 14.90, brød - kr

24.90, yoghurt - 12.90 og pizza - 39.90. Skriv ut innholdet i ordboken med en enkel print-setning.

2. Les inn to varenavn og priser fra brukeren og legg disse til i ordboken. Skriv ut ordboken på nytt.

Synes du denne oppgaven var vanskelig? Se Trix-oppgave 3.19.
Synes du denne oppgaven var enkel? Se Trix-oppgave 3.20.

Oppgave 3: Tegn en sirkel

Filnavn: *sirkel.py*

I denne oppgaven skal du importere en *modul* for å gjøre en enkel grafikkoppgave. Modulen du skal hente heter EzGraphics og er gratis som en del av pensumboken Python for Everyone. Følg disse stegene:

1. Les side 63-69 i Python for Everyone om hvordan du kan importere modulen og tegne figurer.

2. Last ned *ezgraphics.py* herfra.

3. Legg *ezgraphics.py* i samme mappe som *sirkel.py.*

4. I *sirkel.py*, importer *ezgraphics.py* og bruk informasjonen fra pensumboka for å tegne en rød sirkel.

Synes du denne oppgaven var vanskelig? Se Trix-oppgave 3.02.
Synes du denne oppgaven var enkel? Se Trix-oppgave 3.12.

Oppgave 4: Billettpris

Filnavn: *billettpris.py*

I denne oppgaven skal du lage et system som beregner billettpris avhengig av kjøperens alder.

1. Lag en prosedyre, velg navnet selv. Prosedyren skal inneholde en variabel *alder* og be brukeren om å skrive inn alderen sin.

2. Utvid prosedyren med en variabel *billettpris*, som du foreløpig gir tallverdien 0.

3. Videre skal prosedyren inneholde en if-else-sjekk. Den skal sjekke følgende:
    a. Hvis brukeren er under eller akkurat 17 år gammel skal billettprisen være 30 kroner (barnebillett).
    b. Ellers, hvis brukeren er over 17 år gammel, blir billettprisen 50 kroner.

    c. Ellers, hvis brukeren er 63 år eller eldre, blir billettprisen 35 kroner (pensjonistbillett).

4. Til slutt skal prosedyren på en ryddig måte skrive ut billettprisen til brukeren.

5. Kall prosedyren 4 ganger.

6. Hva er problemet med denne prosedyren? Skriv svaret som en kommentar under prosedyren.

Synes du denne oppgaven var vanskelig? Se Trix-oppgave 3.21.
Synes du denne oppgaven var enkel? Se Trix-oppgave 3.17.

Oppgave 5: Egen oppgave

1. Skriv oppgavetekst til en oppgave som handler om lister. Eller du kan bruke dette forslaget: Skriv et program som tar imot koordinater samt høyde og bredde fra bruker, legger disse etter hverandre i en liste og deretter bruker innholdet i listen til å tegne opp en form med EzGraphics.

2. Løs oppgaven! Du skal levere både oppgaveteksten og besvarelsen (skriv oppgaveteksten som kommentarer over løsningen din).

Krav til innlevering

- Oppgaven må kunne kjøres på IFI sine maskiner. Test dette før du leverer!

- Kun *.py*-filene (bortsett fra *ezgraphics.py*) og *README.txt* skal leveres inn.

- Koden skal inneholde gode kommentarer som forklarer hva programmet gjør.

- Programmet skal inneholde gode utskriftssetninger som gjør det enkelt for bruker å forstå.

Hvordan levere oppgaven

1. Lag en fil som heter README.txt. Følgende spørsmål **skal** være besvart i filen:
    a. Hvordan synes du innleveringen var? Hva var enkelt og hva var vanskelig?
    b. Hvor lang tid (ca) brukte du på innleveringen?
    Var det noen oppgaver du ikke fikk til? Hvis ja:
        i. Hvilke(n) oppave er det som ikke fungerer i innleveringen?
        ii. Hvorfor tror du at oppgaven ikke fungerer?
        iii. Hva ville du gjort for å få oppgaven til å fungere hvis du hadde mer tid?

2. Logg inn på Devilry.

3. Lever alle *.py*-filene samt *README.txt* i samme innlevering. **Ikke** lever den nedlastede modulen (*ezgraphics.py*)

4. Husk å trykke lever og sjekk deretter at innleveringen din er komplett.

5. Den obligatoriske innleveringen er minimum av hva du bør ha programmert i løpet av en uke. Du finner flere oppgaver for denne uken her.