IN1000 Obligatorisk innlevering 3

Frist for innlevering: 12. september kl 23:59 **Sist endret:** 6.9.22

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 ordbøker). Du skal vite at samlingene representeres av objekter som tilbyr tjenester i programmet. 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. Beregn summen og produktet av (multipliser) tallene i den første lista, og legg disse to tallene i en ny liste. Lag enda en liste ved å slå sammen den opprinnelige tall-listen med listen som inneholder sum og produkt, og skriv ut denne. Fjern deretter de to siste elementene fra lista, og skriv den ut på nytt.

Synes du denne oppgaven var vanskelig? Se Trix-oppgave 3.01, 3.03, 3.12
Synes du denne oppgaven var enkel? Se Trix-oppgave 3.10

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 3.07, 3.04
Synes du denne oppgaven var enkel? Se Trix-oppgave 3.13

Oppgave 3: 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 alder på kjøperen.

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. Test at prosedyren fungerer når du skriver inn at brukeren er 15, 31 og 63 år gamle. Er det noe galt med denne prosedyren? Skriv svaret som en kommentar.

Synes du denne oppgaven var vanskelig? Se Trix-oppgave 1.16, 2.03, 2.09
Synes du denne oppgaven var enkel? Se Trix-oppgave 3.14, 3.15

Oppgave 4: Matplan

Filnavn: *matplan.py*

I denne oppgaven skal vi se for oss at hver beboer på et gitt sykehjem spiser tre måltider på en gitt dag.

1. Lag en ordbok hvor hver nøkkelverdi er navnet til en beboer, og innholdsverdien er en liste med tre måltider. Måltidene skal være henholdsvis frokost, lunsj og middag. For eksempel spiser Kari Nordmann brød til frokost, egg til lunsj og pølser til middag.

2. Lag så en prosedyre som skriver ut navnene til alle beboerne og spør brukeren om å skrive navnet til en beboer i terminalen. Deretter skal prosedyren skrive ut matplanen til den beboeren. Skriv ut en melding til brukeren hvis beboeren ikke er registrert. Husk å kalle prosedyren.

3. Hvilken type samling (liste, mengde, ordbok) ville du brukt for å representere hver av de følgende eksemplene på data? Skriv litt om hvorfor, eventuelt med forbehold eller presiseringer.
    a. Brukernavn på alle IN1000 studentene
    b. Brukernavn og antall poeng på innlevering 3 for alle studentene på IN1000
    c. Alle vinnere i Lotto siste år (kun navn)
    d. All mat noen gjester i et selskap er allergisk mot (for å planlegge menyen)

Synes du denne oppgaven var vanskelig? Se Trix-oppgave 3.02, 3.11
Synes du denne oppgaven var enkel? Se Trix-oppgave 3.13

Oppgave 5: Egen oppgave

1. Skriv oppgavetekst til en oppgave som handler om lister eller ordbøker. Et forslag kan være å skrive en quiz som i forrige uke, men ved å bruke lister. Alternativt et program som sjekker alle allergier (for eksempel for deltakere ved påmelding til et arrangement), og skriver ut en oversikt til slutt over hva slags mat som bør unngås.

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 din maskin. Test dette før du leverer!

- Kun *.py*-filene skal leveres inn.

- Spørsmålene til innleveringen skal besvares i kommentarfeltet.

- 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. Kommenter på følgende spørsmål i kommentarfeltet i Devilry. Spørsmålene **skal** besvares.
    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?
        iv. Hva vil du ha tilbakemelding på?

2. Logg inn på Devilry.

3. Lever alle *.py*-filene, og husk å svare på spørsmålene i kommentarfeltet.

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

5. Dersom du er sikker på at du har levert for siste gang, og ikke har noe imot å få oppgaven rettet før fristen er ute kan du også skrive “KLAR FOR RETTING” i kommentarfeltet.

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