IN1000 Innlevering 3

Frist for innlevering: 18. februar 2025, kl. 23:59 **Sist endret:** 10.02.2025

Introduksjon

Innleveringen består av 5 oppgaver. 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 Trix-oppgaver først (se lenker under hver oppgave).

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, slik at det blir tydelig hva du har tenkt. Andre viktige krav til innleveringen og beskrivelse av hvordan du leverer inn finner du nederst.

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 det tredje elementet i lista.

2. Lag en ny, tom liste. Be 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 så en tredje 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.

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

Oppgave 2: Ordbok

Filnavn: *ordbok.py*

1. Lag en ordbok (dictionary) som skal inneholde butikkvarer og tilhørende pris. Du skal bruke varenavn som nøkkelverdier og varepriser (flyttall) som innholdsverdier. Opprett ordboka med følgende varer og tilhørende pris:
   melk - kr 14.90,
   brød - kr 24.90,
   yoghurt - 12.90,
   pizza - 39.90.

   Skriv ut innholdet i ordboken med en enkel print-setning.

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

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

Oppgave 3: Billettpris

Filnavn: *billettpris.py*

Nå skal du lage et system som beregner billettpris avhengig av kjøperens alder.

1. Lag en prosedyre, som du velger navnet på 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 reflektere følgende (helt presist slik det er beskrevet):
   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 gammel. Er det noe galt med denne prosedyren? Skriv svaret som en

kommentar i billettpris.py.

Syntes du denne oppgaven var vanskelig? Se Trix-oppgave 1.16, 2.03, 2.09 Syntes 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 beboere på et gitt sykehjem spiser tre måltider på en gitt dag.

1. Lag en ordbok med fem beboere. Hver nøkkelverdi er navnet til en beboer, og innholdsverdien er en liste med tre måltider. Måltidene er 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 studiepoeng dette semesteret for hver student 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)

Syntes du denne oppgaven var vanskelig? Se Trix-oppgave 3.02, 3.11
Syntes 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 og kommentarer som forklarer hvordan du har tenkt underveis.
- Programmet skal inneholde gode utskriftssetninger som gjør det enkelt for bruker å forstå.

Hvordan levere oppgaven

1. Svar på spørsmålene under i kommentarfeltet i Devilry. De **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(r) 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, og husk å svare på spørsmålene i kommentarfeltet.

4. Husk å trykke på *Add delivery or question*. Sjekk deretter at innleveringen din er komplett. Du kan levere så mange ganger du vil frem til fristen. Det er ikke mulig å slette innleveringer i Devilry. *Alle* filer må være med hver gang du leverer, og filene må ha samme navn som oppgitt i oppgaveteksten. Retteren din vil gi tilbakemelding på kun den siste innleveringen, og derfor er det viktig at alle filer er med i siste innlevering.

5. Innleveringen gir deg mulighet for en konstruktiv tilbakemelding på hva du har fått til, i tillegg til forbedringer/misforståelser og om du er omtrent i rute med faget. De fleste vil ha behov for å løse mange flere oppgaver enn de du finner i denne innleveringen. Det får du anledning til i gruppetimene og i Trix. Du finner Trix-oppgaver for denne uken her. Læreboken inneholder også mange

oppgaver (noen går litt utenfor vårt pensum). Du finner dessuten mye på nett - og ikke minst kan du finne på egne programmer og utvidelser til oppgaver.