

Velkommen til gruppetime i IN1000



4. september 2020
Jessie Yue Guan

Planen for i dag

- Kahoot! :D
- Primitiver versus kolleksjoner
- Lister
- Mengder
- Ordbøker
- Prosedyrer?

Primitiver versus kolleksjoner

- Hittil har vi for det meste bare jobbet med "primitives" (primitive variabeltyper/datatyper)
- Disse kan bare inneholde/lagre en verdi om gangen
- Men nå skal vi begynne å se litt på "collections" (kolleksjoner)
- Disse kan inneholde/lagre flere verdier om gangen

Hvorfor lagre flere verdier sammen?

- `alder1 = 23`

- `alder2 = 19`

- `alder3 = 27`

- Dette blir veldig kronglete etter hvert...

- `aldre = [23, 19, 27]`

- `#...eller...`

- `aldre = set([23, 19, 27])`

- Dette blir mye mer ryddig og organisert

Lister

- En samling av verdier som har en rekkefølge og som kan inneholde duplikater
- For eksempel:
 - **minListe = ["C", "A", "C", "B"]**

- Rekkefølgen i denne lista blir da:
 - "C" på indeks 0
 - "A" på indeks 1
 - "C" på indeks 2
 - "B" på indeks 3
- Duplikater i denne lista blir da:
 - "C" som forekommer 2 ganger

Hvordan lage en liste

- Metode 1:
 - **minListe = [4, 2, 6, 5, 5]**
 - Vi vet hvor lang lista skal være og vi vet hvilke verdier den skal inneholde
-
- Metode 2:
 - **minListe = [0] * 5**
 - Vi vet hvor lang lista skal være men ikke hvor mange verdier den skal inneholde
-
- Metode 3:
 - **minListe = []**
 - Vi vet ikke hvor lang lista skal være og heller ikke hvor mange verdier den skal inneholde

Hvordan legge til et element i en liste

- Utvidelse av en tom liste:

- `minListe = []`
- `minListe.append(12.34)`
- `print(minListe)`
- #Vil printe `[12.34]`

- Utvidelse av en ikke-tom liste:

- `minListe = [2, 4, 6]`
- `minListe.append(8)`
- `print(minListe)`
- #Vil printe `[2, 4, 6, 8]`

Hvordan aksessere eller printe en verdi i en liste

- La oss si at vi har følgende kode:
 - `spillere = ["Ola", "Ida", "Per", "Hanne"]`

- Aksessering for å lagre en verdi i en variabel:
 - `tur = spillere[1] #Holder styr på hvem sin tur det er`

- Aksessering for å printe en verdi til terminalen:
 - `print("Det er nå " + spillere[1] + " sin tur")`

Hvordan endre en verdi i en liste

- Vi har en liste som ser slik ut:
 - `aldre = [25, 20, 32]`
- La oss si at en av personene fyller bursdag
- Da skriver vi bare følgende lengre nede i koden:
 - `aldre[2] = 33`
- Vi sier at verdien på indeks 2 i lista som heter `aldre` skal settes til 33

Hvordan slette en verdi i en liste

- Hvordan slette basert på indeks:

- `minListe = [1, 3, 5, 7, 9]`
- `del minListe[2]`
- `print(minListe)`
- #Vil printe `[1, 3, 7, 9]`

- Hvordan slette basert på verdi:

- `minListe = [1, 3, 5, 7, 9]`
- `minListe.remove(7)`
- `print(minListe)`
- #Vil printe `[1, 3, 5, 9]`

Hvordan konkatenerer lister

- To variabler:

- `liste1 = [1, 2, 3]`

- `liste2 = [4, 5, 6]`

- `sammen = liste1 + liste2`

- En variabel og en verdi:

- `liste = [1, 2, 3]`

- `sammen = liste + [4, 5, 6]`

- To verdier:

- `sammen = [1, 2, 3] + [4, 5, 6]`

- En verdi og en variabel:

- `liste = [4, 5, 6]`

- `sammen = [1, 2, 3] + liste`

Hvordan sjekke om en verdi finnes i en liste

- `byer = ["Oslo", "Bergen", "Tromsø"]`

- `print("Trondheim" in byer)`

- `#Vil printe ut "False"`

- `print("Oslo" in byer)`

- `#Vil printe ut "True"`

- `if "Bergen" in byer:`

- `print("Bergen er registrert!")`

- `else:`

- `print("Bergen er ikke registrert!")`

- `if "Trondheim" not in byer:`

- `byer.append("Trondheim")`

Hvordan sjekke hvor ofte en verdi forekommer i en liste

- `tall = [5, 7, 2, 5, 6, 7, 5]`
- `print(tall.count(5))`
- `#Vil printe 3`

- `print(tall.count(6))`
- `#Vil printe 1`

Hvordan sortere en liste

- `tall = [5, 1, 4, 2, 3]`
- `tall.sort()`
- `print(tall)`
- `#Vil printe [1, 2, 3, 4, 5]`

- `bokstaver = ["a", "s", "d", "w"]`
- `bokstaver.sort()`
- `print(bokstaver)`
- `#Vil printe ["a", "d", "s", "w"]`

Mengder

- En samling av verdier som ikke har en rekkefølge og som ikke kan inneholde duplikater

- For eksempel:

- `minMengde = set([26374, 93853, 48375, 72836, 58273])`

- Ingen rekkefølge betyr at...

- `print(minMengde[3])`

- ... resulterer i en feilmelding

- Ingen duplikater betyr at...

- `liste = set([1, 1, 1, 5, 5, 9])`

- `print(len(set))` #Printer 3

- `print(liste)` #Printer [1, 5, 9]

Hvordan lage en mengde

- Lage en tom mengde:
 - `minMengde = set()`
 - Blir da $\{\}$
- Lage en ikke-tom mengde:
 - `minMengde = {1, 1, 3, 2, 2, 1}`
 - #Blir da $\{1, 3, 2\}$
- Eller enda bedre:
 - `minMengde = set([1, 1, 3, 2, 2, 1])`
 - #Blir da $\{1, 3, 2\}$

Hvordan legge til et element i en mengde

- Legge til en verdi:
 - `minMengde.add(4)`
 - #Blir da {1, 3, 2, 4}
- Legge til flere verdier:
 - `minMengde.update([4, 4, 6, 4, 6, 5])`
 - #Blir da {1, 3, 2, 4, 6, 5}

Ordbøker og rekkefølge

- Slår opp en innholdsverdi basert på en nøkkelverdi
- En samling av verdier som ikke har en rekkefølge og som kan inneholde duplikate innholdsverdier men ikke duplikate nøkkelverdier
- For eksempel:
 - `minOrdbok = {"Ola Nordmann" : 12345678, "Kari Nordmann" : 87654321}`

- Oppskriften:
 - `ordbok = {`
 - `nøkkelverdi1 : innholdsverdi1,`
 - `nøkkelverdi2 : innholdsverdi2,`
 - `nøkkelverdi3 : innholdsverdi3`
 - `}`
- Både nøkkelverdier og innholdsverdier kan bestå av en hvilken som helst variabeltype
- Ingen rekkefølge betyr at...
 - `print(minOrdbok[1])`
- ... resulterer i en feilmelding

Ordbøker og duplikater

- `matteordbok = {`
 - `"sum" : "resultatet av addisjon",`
 - `"pi" : 3.14,`
 - `3 : "det andre primtallet",`
 - `3 : "det tredje trekantallet",`
 - `"totalsum" : "resultatet av addisjon",`
 - `3 : "det andre oddetallet"`
 - `}`

 - `print(matteordbok)`
- `{'sum': 'resultatet av addisjon',`
 - `'pi': 3.14,`
 - `3: 'det andre oddetallet',`
 - `'totalsum': 'resultatet av addisjon'}`

 - Legg merke til at...
 - Nøkkelverdien 3 og tilhørende innholdsverdi printes kun én gang
 - Nøkkelverdien 3 og tilhørende innholdsverdi printes når de forekommer først i ordboka
 - Nøkkelverdien 3 inneholder kun den siste innholdsverdien
 - Innholdsverdien "resultatet av addisjon" printes to ganger

Hvordan lage en ordbok

- Hvis du ikke vet hva ordboka skal inneholde:

- `minOrdbok = {}`

- Hvis du vet hva ordboka skal inneholde:

- `minOrdbok = {`

- `"Norge" : "Oslo",`

- `"Danmark" : "København",`

- `"Sverige" : "Stockholm"`

- `}`

Hvordan legge til et verdipar i en ordbok

- `stoersteLand = {`
 - `"Kina" : 3,`
 - `"Russland" : 1,`
 - `"Canada" : 2`
 - `osv...`
- `}`

- `stoersteLand["Norge"] = 68`
- `print(stoersteLand)`

- `{"Kina" : 3,`
- `"Russland" : 1,`
- `"Canada" : 2,`
- `osv...`
- `"Norge" : 68}`

Hvordan aksessere eller printe en innholdsverdi i en ordbok

- Aksessering for å lagre en innholdsverdi i en variabel:
 - `mittLandsStoerrelse = stoersteLand["Norge"]`
 - `#Variabelen blir da satt til 68`

- Aksessering for å printe en innholdsverdi til terminalen:
 - `print(stoersteLand["Canada"])`
 - `#Vil printe 2`

Hvordan endre eller slette en innholdsverdi i en ordbok

- Endring av innholdsverdi:
 - `stoersteLand["Norge"] = 1`
 - `print(stoersteLand["Norge"])`
 - #Vil nå printe 1 i stedet for 68

- Sletting av innholdsverdi:
 - `del stoersteLand["Norge"]`
 - `print(stoersteLand["Norge"])`
 - #Vil nå resultere i en `KeyError`

Subrutiner: Prosedyrer/funksjoner/metoder

- En prosedyre kan ta imot input...
- **def personlig_hilsen(navn):**
- **print("Hei " + navn + "! :D")**
- **personlig_hilsen("Espen Askeladd")**
- (Da blir den mer fleksibel, og kan brukes igjen, og med små variasjoner)

- Eller så kan den ikke ta imot noen input...
- **def hilsen():**
- **print("Hei Espen Askeladd! :D")**
- **hilsen()**
- (Da blir den mindre fleksibel, og kan brukes igjen, men uten små variasjoner)

Subrutiner: Prosedyrer/funksjoner/metoder

- En prosedyre kan returnere output...

- **def multipliser():**

- **return 2 * 2**

- **print(multipliser())**

- (Da blir den kalt for en funksjoner, den blir mer fleksibel, og kan brukes av andre prosedyrer/funksjoner)

- Eller så kan den ikke returnere noen output...

- **def multipliser():**

- **print(2 * 2)**

- **multipliser()**

- (Da blir den kalt for en prosedyre, den blir mindre fleksibel, og kan ikke brukes av andre prosedyrer/funksjoner)

Subrutiner: Prosedyrer/funksjoner/metoder

- En prosedyre kan ha ingen input
- Den kan ha en input
- Den kan ha flere input

- Det spiller ingen rolle, dette velger du selv

- Men den kan kun ha output i form av **print()**

- En funksjon kan ha ingen input
- Den kan ha en input
- Den kan ha flere input

- Det spiller ingen rolle, dette velger du selv

- Men den kan kun ha output i form av **return**

- Kort huskeregel: Input spiller ingen rolle, men output spiller en rolle for terminologien
- En metode er en prosedyre eller funksjon som tilhører en klasse/et objekt