



Seminartime(+lab) uke 3 – Gr. 10

IN1000 – HØST 2021

Hvor langt har vi kommet?

NB: Endringer kan komme	Legges ut ca.	Frist
1.innlevering (3 poeng)	25. august kl 13.00	31. august kl 23.59
2. innlevering (5 poeng):	1. september kl 13.00	7. september kl 23.59
3.innlevering (5 poeng):	08. september kl 13.00	14. september kl 23.59
4.innlevering (5 poeng):	15. september kl 13.00	21. september kl 23.59
5.innlevering (5 poeng):		
6. innlevering (6 poeng):		

Uke	Tema
1	Innføring i python – deres første program! <ul style="list-style-type: none">- Variabler- Printe ut i terminal- Feilmeldinger- Input fra bruker- Beslutninger: If-setninger & conditions
2	Dypere forståelse fra uke 1 <ul style="list-style-type: none">- Datatyper- Operasjoner på datatyper- Evaluering av uttrykk- Boolske variabler- Sammenlikningsoperatorer- Funksjoner & prosedyrer
3	Samlinger <ul style="list-style-type: none">- Lister (list)- Mengder (set)- Ordbøker (dictionary)

Plan for i dag

- Gjennomgang av pensum uke 3
 - Lister
 - Mengder (set)
 - Ordbøker (dictionaries)
- Egen jobbing



KOLLEKSJONER/SAMLINGER

Hva brukes det til?

- Variabler = lagre 1 ting
- Samlinger = lagre flere ting → mer ryddig & du kan gjøre den samme prosedyren på *alle* tingene
- Ulike typer samlinger har ulike egenskaper
- De dere skal kunne er:
 - Lister
 - Mengder
 - Ordbøker

Lister

- Alle elementer(ting) har en plass (indeks) i listen → aksessere dem med indeksen
- Har en rekkefølge → kaller det for «ordered»
- KAN inneholde duplikater (flere like verdier)
- KAN bli endret → Kaller det for «mutable»
- Tre måter å opprette en liste på:

minListe = [2, 4, 4, 6]	minListe = [0] * 4	minListe = [] (tom liste)																
<table border="1" data-bbox="152 933 838 1089"> <tr> <td>2</td> <td>4</td> <td>4</td> <td>6</td> </tr> <tr> <td>Indeks 0</td> <td>Indeks 1</td> <td>Indeks 2</td> <td>Indeks 3</td> </tr> </table> <p data-bbox="104 1153 774 1253">Vet hvor lang lista skal være + vet hvilke verdier som skal være i lista.</p>	2	4	4	6	Indeks 0	Indeks 1	Indeks 2	Indeks 3	<table border="1" data-bbox="963 933 1648 1089"> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Indeks 0</td> <td>Indeks 1</td> <td>Indeks 2</td> <td>Indeks 3</td> </tr> </table> <p data-bbox="907 1153 1643 1253">Vet hvor lang lista skal være, men ikke hva som skal være i den.</p>	0	0	0	0	Indeks 0	Indeks 1	Indeks 2	Indeks 3	<div data-bbox="2048 933 2099 1089" style="border: 1px solid black; width: 20px; height: 100px; margin: 0 auto;"></div> <p data-bbox="1709 1153 1895 1196">Tom liste.</p>
2	4	4	6															
Indeks 0	Indeks 1	Indeks 2	Indeks 3															
0	0	0	0															
Indeks 0	Indeks 1	Indeks 2	Indeks 3															

Metoder på lister

(som dere bør huske/kjenne til)

	LISTE = []
Legge til	<p><code>liste.append("Sol")</code> → Legger til nytt element bakerst i lista</p> <p><code>liste[<index>] = "Sol"</code> → Tilordner nytt element på denne indeksen/endrer verdi</p> <p><code>sammen = liste1 + liste 2</code> → konkatinerer 2 lister</p>
Hente ut	<p><code>x = liste[<index>]</code> → Henter elementet på denne indeksen og lagrer det i variabelen x</p> <p><code>ny_liste = liste[<index start>:<index slutt>]</code> → ny liste av en annen liste (kopi av liste)</p>
Fjerne	<p><code>liste.pop()</code> → Fjerner siste element i lista</p> <p><code>liste.pop(<index>)</code> → Fjerner element på indeksen</p> <p><code>liste.remove(<element>)</code> → Fjerner selve elementet</p>
Gå gjennom	<p><code>if x in liste:</code></p> <p>Argument → For hvert element i lista, gjør noe</p>

Cheat sheet for Python lists

Construction	Meaning
<code>a = []</code>	initialize an empty list
<code>a = [1, 4.4, 'run.py']</code>	initialize a list
<code>a.append(elem)</code>	add elem object to the end
<code>a + [1,3]</code>	add two lists
<code>a.insert(i, e)</code>	insert element e before index i
<code>a[3]</code>	index a list element
<code>a[-1]</code>	get last list element
<code>a[1:3]</code>	slice: return sublist (here: index 1, 2)
<code>del a[3]</code>	delete an element (index 3)
<code>a.remove(e)</code>	remove an element with value e
<code>a.index('run.py')</code>	find index corresponding to an element's value
<code>'value' in a</code>	test if a value is contained in the list
<code>a.count(v)</code>	count how many elements have the value v
<code>len(a)</code>	number of elements in list a
<code>min(a)</code>	the smallest element in a
<code>max(a)</code>	the largest element in a
<code>sum(a)</code>	add all elements in a
<code>sorted(a)</code>	return sorted version of list a
<code>reversed(a)</code>	return reversed view version of list a
<code>b[3][0][2]</code>	nested list indexing

Oppgave 1

Hvor mange elementer er det i listene?

liste = [0]

liste1 = [1, 2, 3]

liste2 = [3, 3, 5, 7]

liste3 = ["A", "BC", "D", "E", "F"]

liste4 = ["mange elementer i denne listen"]

liste5 = []

Oppgave 2

1. Du skal på reise. Lag 3 lister som heter «klær», «toalettsaker» og «div» der du legger til ting du trenger
2. Istedenfor å ha 3 separate beholdere, vil du heller ha en stor beholder (koffert) der du kan putte disse tre i. Lag en slik liste der du putter de tre listene i.
3. Hvordan ser utskriften ut når du printer «print(koffert)»?
4. Hvordan kan du printe ut *kun* «klær»?
5. Hvordan kan du få tak i tannpasta fra «toalettsaker»?
6. Hvordan kan du legge til «selfiestang» i «div»?

Strings

- Strings er en liste, der elementene er bokstavene/mellomrommene
- MEN python strings kan ikke endres på → kaller dette «immutable»
- Hvis man vil endre på en string (altså selve listen), må man lagre den nye strengen i en ny variabel (kopi) → Å endre på en string ≠ tilordne ny string til variabel

Syntaks	Mening/returnerer
<code>``ko`` in <string></code>	True hvis <string> inneholder ``ko``
<code><string>.find(``i``)</code>	Indeksen til første forekomst av ``i``
<code><string>.split()</code>	Deler opp <string> på mellomrom, returnerer en liste med hvert ord
<code><string>.replace(``chips``, ``salad``)</code>	Bytter alle forekomster av ``chips`` med ``salad``
<code><string>.upper()</code>	Endres til store bokstaver
<code><string>.lower()</code>	Endres til små bokstaver
<code><string>.strip()</code>	Fjerner tomrom foran/bak

Mengder (set)

- Likner på lister, men *uten* rekkefølge → unordered
- Ingen duplikater → sørger for at alt er unikt
- Brukes ofte til å se på delmengder eller «intersects» → intersect som i venn-diagrammer

Opprette tom mengde:	<code>nyMengde = set()</code>
En ikke-tom mengde:	<code>nyMengde = {1, 2, 2, 3, 4, 4,}</code> → printer {1, 2, 3, 4}

	MENGDE = ()
Legge til	<code>mengde.add("Ann")</code> <code>mengde.update(5,6,7,8)</code>
Fjerne	<code>mengde.remove(<element>)</code> → feilmld hvis element ikke finnes <code>mengde.discard(<element>)</code> → ikke feilmld hvis element ikke finnes <code>mengde.pop()</code> → fjerner siste element lagt inn <code>mengde.clear()</code> → fjerner siste element lagt inn
Mer	<code>len(mengde)</code> → returnerer antall elementer <code>mengde1.union(mengde2)</code>

Oppgave 1

mengder/set	lister
navn = {"Kari", "Per", "Ola"} navn2 = {"Per", "Ola", "Kari"}	navn = ["Kari", "Per", "Ola"] navn2 = ["Per", "Ola", "Kari"]
if navn == navn2 :.....?	if navn == navn2 :.....?

- Siden det ikke er noe rekkefølge (eller duplikater) i mengder, vil disse ansees som *like*
- Lister har en rekkefølge/indeksering. Siden rekkefølgen i disse to listene er ulike, regnes de som *ikke like*

Ordbøker (dictionaries)

- Slik som lister kan det endres på → mutable
- Bruker nøkler istedenfor indekser → Ikke lenger en innebygd rekkefølge
- Kan aksessere verdiene med nøkler istedenfor indekser

Tom ordbok:	<code>nyOrdbok = {}</code>
En ikke-tom ordbok:	<code>minOrdbok = { "Norge" : "Oslo", "Danmark" : "København", "Sverige" : "Stockholm" }</code>

	ORDBOK = {}
Legge til	<code>ordbok[<nøkkel>] = "innhold"</code>
Hente ut	<code>innhold = ordbok[<nøkkel>]</code>
Fjerne	<code>ordbok.pop(<nøkkel>)</code> <code>ordbok.popitem()</code> → fjerner siste element satt inn
Mer	<code>if x in ordbok:</code> "gjør noe" → hvis nøkkel finnes i ordboka, gjør noe

Dictionary cheat sheet

Construction	Meaning
<code>a = {}</code>	initialize an empty dictionary
<code>a = {'point': (0,0.1), 'value': 7}</code>	initialize a dictionary
<code>a = dict(point=(2,7), value=3)</code>	initialize a dictionary w/string keys
<code>a.update(b)</code>	add key-value pairs from b in a
<code>a.update(key1=value1, key2=value2)</code>	add key-value pairs in a
<code>a['hide'] = True</code>	add new key-value pair to a
<code>a['point']</code>	get value corresponding to key point
<code>for key in a:</code>	loop over keys in unknown order
<code>for key in sorted(a):</code>	loop over keys in alphabetic order
<code>'value' in a</code>	True if string value is a key in a
<code>del a['point']</code>	delete a key-value pair from a
<code>list(a.keys())</code>	list of keys
<code>list(a.values())</code>	list of values
<code>len(a)</code>	number of key-value pairs in a
<code>isinstance(a, dict)</code>	is True if a is a dictionary

Oppgave 1

1. Definer en ordbok/dictionary som heter brukere, hvor du legger inn følgende par av nøkler og verdier (brukernavn som nøkler og navn som verdi).
2. karsi har registrert seg med feil navn, mellomnavnet hennes mangler, endre navnet hennes til “Kari Marie Sirisen”.
3. Register deg selv som bruker med et brukernavn og navn.
4. Skriv ut hanjo sitt navn.
5. Fjern Hanne Johansen fra ordboken