Samlinger og objekter

Uke 3, time 2.

Gruppe 6

Frøya Lydersen

skagel@uio.no

Contents

Contents

Noen vanlige metoder 2
..på strenger/strings 3
..på lister/lists 4
..på ordbøker/dictionaries 5
..på mengder/sets 6
Nøstede samlinger 7
Aksessere verdier i nøstede samlinger 9
Konvertere mellom typer samlinger 10
Utfordringer på menti 12
Parprogrammering / programmering 13
–Exit ticket– 15

Noen vanlige metoder

..på strenger/strings

https://www.w3schools.com/python/python_ref_string.asp

```
1  x = " hei på deg"
2  print("0", x)
3  print("1", x.strip())  # fjerner mellomrom og linjeskifte rundt tekst
4  print("2", x.upper())  # gjør alt til stor tekst
5  print("3", x.lower())  # gjør alt til liten tekst
6  print("4", x.strip().capitalize())  # capitalize gjør første bokstav i hvert
   ord stor
7  print("5", x.count("e"))  # teller tilfeller av substreng
8  # splitter listen for hver plass symbolet du ga den er, og gir en liste med
   alle delene:
9  print("6", x.split(" "))
10 print("7", x.split("p"))
11 print("8", "å" in x)  # returnerer True hvis substrengen "å" er i strengen i
   x.
12 print("9", x.strip().split(" ")[0].upper())
```

..på lister/lists

https://www.w3schools.com/python/python_ref_list.asp

```
1    x = ["x", "t", "l"]
2    x.append("z")  # sett inn "z" på slutten (dette gror listen)
3    print(x)
4    # sett inn "g" på indeks 0, og flytt resten opp for å gjøre plass:
5    x.insert(0, "g")
6    print(x)
7    print(x.index("z"))  # indeks av den første verdien "z"
8    # fjern og gi tilbake verdien på øverste indeks (dette krymper listen):
9    print(x.pop())
10   # fjern og gi tilbake verdien på indeks 2 (dette krymper listen):
11   print(x.pop(2))
```

..på ordbøker/dictionaries

https://www.w3schools.com/python/python_ref_dictionary.asp

```
1 x = {"a":1, "b":2, "c":3}
2 print(x.pop("a")) # fjerner nøkkel-verdi paret for nøkkelen "a"
3 print(x.values()) # returnerer en liste med alle verdiene i ordboka
4 print(x.keys()) # returnerer en liste med alle nøklene i ordboka
5 # Merk at teknisk sett er nøkler også "verdier" i den forstand at de er
  verdier i python. Men når vi snakker om ordbøker og sier "nøkkel-verdi"
  par, eller snakker om "verdiene i en ordbok", snakker vi om de verdiene
  som nøklene assosieres med.
```

..på mengder/sets

https://www.w3schools.com/python/python_ref_set.asp

```
1  x = {"a","b","c"}
2  z = {"a","b"}
3  y = {"c","d","e"}
4  y.add("x") # legger verdien "x" til mengden y
5  y.remove("x") # fjerner verdien "x" i mengden y
6
7  # Mengdeoperasjonene kan godt googles
8  print(x.union(y)) # alt fra begge mengdene
9  print(x.intersection(y)) # det som er i både x og y
10 print(x.difference(y)) # det som er i x men ikke y
11 print(z.issubset(x)) # sjekker om z ikke har noe som x ikke har
12
13 y.pop() # fjerner et tilfeldig element fra y
```

Nøstede samlinger

Nøstede samlinger

- Lister er i seg selv verdier og kan lagres i andre lister.
- Vi kan faktisk lagre ordbøker, mengder, og lister, i lister.
- Merk at vi ikke kan lagre lister, ordbøker, eller mengder, i ordbøker eller mengder.

```
1 x = []
2
```

Nøstede samlinger

- Lister er i seg selv verdier og kan lagres i andre lister.
- Vi kan faktisk lagre ordbøker, mengder, og lister, i lister.
- Merk at vi ikke kan lagre lister, ordbøker, eller mengder, i ordbøker eller mengder.

```
1 x = []
2 y = [  1,  2,  3,  4 ]
```

- Lister er i seg selv verdier og kan lagres i andre lister.
- Vi kan faktisk lagre ordbøker, mengder, og lister, i lister.
- Merk at vi ikke kan lagre lister, ordbøker, eller mengder, i ordbøker eller mengder.

```
1 x = []
2 y = [  1,  2,  3,  4 ]
3 z = [ [], [], [], [] ]
```

Nøstede samlinger

- Lister er i seg selv verdier og kan lagres i andre lister.
- Vi kan faktisk lagre ordbøker, mengder, og lister, i lister.
- Merk at vi ikke kan lagre lister, ordbøker, eller mengder, i ordbøker eller mengder.

```
1 x = []
2 y = [  1,  2,  3,  4 ]
3 z = [ [], [], [], [] ]
```

```
1 a = [[1,2,3],[4,5,6],[7,8,9]]
2 a = [
3       [1,2,3],
4       [4,5,6],
5       [7,8,9]
6 ]
```

Aksessere verdier i nøstede samlinger


```
1    a = [[[1,2],[3,4]],[[5,6],[7,8]]]
2    b = a[0]
3    print(a[0])
4
5
```

Aksessere verdier i nøstede samlinger


```
1    a = [[[1,2],[3,4]],[[5,6],[7,8]]]
2    b = a[0]
3    print(a[0])
4    print(b[0])
5
```

Aksessere verdier i nøstede samlinger


```
1    a = [[[1,2],[3,4]],[[5,6],[7,8]]]
2    b = a[0]
3    print(a[0])
4    print(b[0])
5    print(a[0][0])
```

Aksessere verdier i nøstede samlinger


```
1    a = [[[1,2],[3,4]],[[5,6],[7,8]]]
2    b = a[0]
3    print(a[0])
4    print(b[0])
5    print(a[0][0])
6    print(a[0][0][0])
```

Konvertere mellom typer samlinger

Konvertere mellom typer samlinger

```
1    a = [1,3,2,1,4,3,2,1]
2    print(set(a))
3    print(list(set(a)))
```

Utfordringer på menti

Parprogrammering / programmering

Parprogrammering / programmering

Oppgaver for uke 3 time 2 på gruppe 6 sin side :v

–Exit ticket–