

Listor

Programmering 1

2024/25

Innehåll

Intro

Skapa en lista

Datatyper

Indexering

Hitta element

Börjar på 0

Lägg till & ta bort

Ändra på ett värde

`append()`

`insert()`

`remove()`

`pop()`

Längden på en lista

Sammanfattning

Övningar

Listor

Ganska ofta vill man hantera flera objekt samtidigt. Till exempel alla domarresultat från en simhoppstävling, eller namnen på alla i en familj. Då kan man använda sig utav listor.

```
1 poäng = [7.56, 5.3, 9.34, 8.2] # Poäng från domarna
2
3 namn = ['Dwalin', 'Balin', 'Kili', 'Fili', 'Dori', 'Nori', 'Ori',
          'Oin', 'Gloin', 'Bifur', 'Bofur', 'Bombur', 'Thorin']
```

Listor

```
1 lista = [] # En tom lista
2 namn = ['Dwalin', 'Balin', 'Kili', 'Fili', 'Dori', 'Nori', 'Ori',
          'Oin', 'Gloin', 'Bifur', 'Bofur', 'Bombur', 'Thorin']
```

En lista känns igen på hakparanteserna, `[]`. Dessa markerar var listan börjar och slutar. Varje element separeras med ett komma. Så `'Dwalin'` är ett element och `'Balin'` är ett annat.

Listor

En lista kan innehålla objekt av olika datatyper.

```
1 supercoollista = ['Snövit', 7, 'T-rex', 3.54, False]
```

En lista kan till och med innehålla flera listor:

```
1 coolare_lista = [['Snövit', 'Bambi', 'Fantasia'], 1, 7, ['Dwalin',  
    'Balin']]
```

Nu innehåller `coolare_lista` de fyra elementen:

```
1 ['Snövit', 'Bambi', 'Fantasia']  
2 1  
3 7  
4 ['Dwalin', 'Balin']
```

Innehåll

Intro

Skapa en lista

Datatyper

Indexering

Hitta element

Börjar på 0

Lägg till & ta bort

Ändra på ett värde

append()

insert()

remove()

pop()

Längden på en lista

Sammanfattning

Övningar

Indexering

Vill man komma åt enskilda objekt i en lista skriver man listans namn, åtföljt av hakparanteser med indexplatsen emellan.

```
1 hober = ['Frodo', 'Sam', 'Merry', 'Pippin']  
2  
3 print(hober[2])
```

Indexering

För att utnyttja datorns minne till max börjar man att räkna på 0. (Här är en argumenterande text om detta: Why numbering should start at zero, prof. dr. Edsger W. Dijkstra)

```
1 platser = [0,1,2,3,4,5,6] # Sju lång
2
3 istari = ['Saruman', 'Gandalf', 'Radagast']
4 #           Plats 0       Plats 1       Plats 2
5 print(istari[0])
6 -> 'Saruman'
7 print(istari[1])
8 -> 'Gandalf'
9 print(istari[2])
10 -> 'Radagast'
```


Indexering

Man kan också komma åt element i listan genom att räkna baklänges:

```
1 istari = ['Saruman', 'Gandalf', 'Radagast']
2 #           Plats 0      Plats 1      Plats 2
3 #           Plats -3     Plats -2     Plats -1
4 print(istari[-1]) # Första elementet bakifrån
5 -> 'Radagast'
6 print(istari[-2]) # Andra elementet bakifrån
7 -> 'Gandalf'
8 print(istari[-3]) # Tredje elementet bakifrån
9 -> 'Saruman'
```

Innehåll

Intro

Skapa en lista

Datatyper

Indexering

Hitta element

Börjar på 0

Lägg till & ta bort

Ändra på ett värde

append()

insert()

remove()

pop()

Längden på en lista

Sammanfattning

Övningar

Listor

Ändra ett värde

Om man vill ändra på en specifik plats i listan behöver man veta dess position.

```
1 tal = [2,4,8,89]
2 tal[3] = 10 # Plats 3 innehöll tidigare 89
3 print(tal)
4 -> 2 4 8 10
```

`append()`

Vill man lägga till ett element i en lista gör man det lättast genom att använda kommandot `listan.append(sak)`

```
1 tal = [2,4,8,89]
2 tal.append(3)
3 print(tal)
4 -> [2,4,8,89,3] # En trea har lagts till
5 tal.append('Hej')
6 print(tal)
7 -> [2,4,8,89,3,'Hej']
```

Med `append` hamnar det nya elementet alltid sist.

insert()

Vill man inte att det nya objektet ska hamna sist, utan man vill att det ska hamna på en specifik position kan man använda sig utav `listan.insert(index, objekt)`

```
1 tal = [2,4,8,89]
2 tal.insert(3, 'Hej') # Läger till 'Hej' på position 3
3 print(tal)
4 -> [2,4,8,'Hej',89]
```

Kom ihåg att index börjar räkna på 0.

remove()

Vill man ta bort ett element ur en lista så kan man använda sig utav kommandot

`remove :`

```
1  istari = ['Saruman', 'Gandalf', 'Radagast']
2  istari.remove('Saruman')
3  print(istari)
4  -> ['Gandalf', 'Radagast']
```

remove()

Om samma element förekommer flera gånger så tas enbart det första elementet bort:

```
1 listan = [1,2,75,6,7,75,6,7,75,6,7]
2 listan.remove(75)
3 print(listan)
4 -> [1,2,6,7,75,6,7,75,6,7]
5 namn = ['Knatte', 'Fnatte', 'Tjatte', 'Tjatte', 'Fnatte', 'Knatte']
6 namn.remove('Fnatte')
7 print(namn)
8 -> ['Knatte', 'Tjatte', 'Tjatte', 'Fnatte', 'Knatte']
```

Tänk på att man bara kan ta bort element som finns i listan.

Listor

remove()

Man kan försäkra sig om att bara ta bort element som finns i listan på följande vis:

```
1 lista = [en lista]
2 while 'hej' in lista:
3     lista.remove('hej')
```

Den här loopen håller på så länge elementet `'hej'` finns i `lista`.

Listor

pop()

Ibland är man mer intresserad av att ta bort element på en särskild plats. Då kan man använda sig utav `pop()`. Anger man inget argument i `pop()` så plockar man bort det sista elementet. Det som särskilt skiljer `pop()` från `remove()` är att `pop()` *returnerar* det tal som "poppas". Man kan alltså skriva `a = lista.pop()` så får `a` samma värde som sista elementet i listan som försvinner.

```
1 lista = ['Aragorn', 'Boromir', 'Legolas', 'Gimli']
2 lista.pop()
3 print(lista)
4 -> ['Aragorn', 'Boromir', 'Legolas']
```

Listor

pop(index)

Vill man plocka bort ett särskilt element kan man ange dess index.

```
1 lista = ['Aragorn', 'Boromir', 'Legolas', 'Gimli']
2 lista.pop(1)
3 print(lista)
4 -> ['Aragorn', 'Legolas', 'Gimli']
```

Innehåll

Intro

Skapa en lista

Datatyper

Indexering

Hitta element

Börjar på 0

Lägg till & ta bort

Ändra på ett värde

`append()`

`insert()`

`remove()`

`pop()`

Längden på en lista

Sammanfattning

Övningar

Längd på listan

Om man vill veta längden på en lista så kan man använda sig utav kommandot

```
len(listan)
```

```
1 lista = ["Frodo", "Sam", "Merry", "Pippin", "Gandalf", "Aragorn",  
           ", "Boromir", "Legolas", "Gimli"]  
2 print(len(lista))  
3 -> 9
```

Innehåll

Intro

Skapa en lista

Datatyper

Indexering

Hitta element

Börjar på 0

Lägg till & ta bort

Ändra på ett värde

`append()`

`insert()`

`remove()`

`pop()`

Längden på en lista

Sammanfattning

Övningar

Listor

Sammanfattning

Följande kommandon är inbyggda i listor:

Kommando	Effekt
<code>append(sak)</code>	Lägg till <code>sak</code> i slutet på listan
<code>insert(index, sak)</code>	Lägg till <code>sak</code> på plats <code>index</code>
<code>pop()</code>	Tar bort och returnerar det sista elementet
<code>pop(index)</code>	Tar bort och returnerar elementet på plats <code>index</code>
<code>remove(sak)</code>	Tar bort den första förekomsten av <code>sak</code> i listan.

Innehåll

Intro

Skapa en lista

Datatyper

Indexering

Hitta element

Börjar på 0

Lägg till & ta bort

Ändra på ett värde

`append()`

`insert()`

`remove()`

`pop()`

Längden på en lista

Sammanfattning

Övningar

Listor

Övningar

Utgå från filen `listor 1.py`

1. Lägg till `'Gimli'` i slutet på listan
2. Lägg till `'Merry'` och `'Pippin'` mellan `'Sam'` och `'Aragorn'`
3. Ta bort `'Boromir'` ur listan.
4. Vänd på listan utan att använda `reverse()`
5. Ta bort alla udda tal ur listan.
6. Dela upp listan i tre nya listor, en för varje datatyp (`int`, `float` och `str`).
7. Gör varje element i listan dubbelt så stort.
8. Räkna hur många gånger varje element förekommer i listan.
9. Sortera listan utan att använda `sort()`