

Listor

Programmering 1

ht 23

Innehåll

Repetition

Skapa lista

Index

Ändra ett element

Lägg till element

Ta bort element

List slicing

[start:slut]

[start:slut:steg]

Sträng till lista

Tecken för tecken

Ord

Listor till strängar

Köer och Stackar

En kö

En stack

Övningar

Repetition

Skapa lista

```
1 min_lista = [5,3,0,8,1]
2 ny_lista = ["Knatte", "Fnatte", "Tjatte"]
```

Repetition

Index

```
1 ny_lista = ["Knatte", "Fnatte", "Tjatte"]
2 print(ny_lista[0])
3 print(ny_lista[1])
4 print(ny_lista[2])
```

Repetition

Index

```
1 ny_lista = ["Knatte", "Fnatte", "Tjatte"]
2 print(ny_lista[0])
3 print(ny_lista[1])
4 print(ny_lista[2])
```

```
1 Knatte
2 Fnatte
3 Tjatte
```

Repetition

Index

```
1 ny_lista = ["Knatte", "Fnatte", "Tjatte"]
2 print(ny_lista[-1])
3 print(ny_lista[-2])
4 print(ny_lista[-3])
```

Repetition

Index

```
1 ny_lista = ["Knatte", "Fnatte", "Tjatte"]
2 print(ny_lista[-1])
3 print(ny_lista[-2])
4 print(ny_lista[-3])
```

```
1 Tjatte
2 Fnatte
3 Knatte
```

Repetition

Ändra ett element

```
1 min_lista = [5,3,0,8,1]
2 min_lista[2] = 9
3 min_lista[3] = 'a'
4 print(min_lista)
```


Repetition

Ändra ett element

```
1 min_lista = [5,3,0,8,1]
2 min_lista[2] = 9
3 min_lista[3] = 'a'
4 print(min_lista)
```

```
1 [5,3,9,'a',1]
```

Repetition

Lägg till element

```
1 en_lista = ["Frodo", "Sam"]
2 en_lista.append("Merry")
3 en_lista.insert(1, "Pippin")
4 print(en_lista)
```

Repetition

Lägg till element

```
1 en_lista = ["Frodo", "Sam"]  
2 en_lista.append("Merry")  
3 en_lista.insert(1,"Pippin")  
4 print(en_lista)
```

```
1 ["Frodo", "Pippin", "Sam", "Merry"]
```

Repetition

Ta bort element

```
1 en_lista = ["Frodo", "Pippin", "Sam", "Merry"]
2 en_lista.pop(1)
3 en_lista.pop()
4 en_lista.remove("Sam")
5 print(en_lista)
```

Repetition

Ta bort element

```
1 en_lista = ["Frodo", "Pippin", "Sam", "Merry"]
2 en_lista.pop(1)
3 en_lista.pop()
4 en_lista.remove("Sam")
5 print(en_lista)
```

```
1 ["Frodo"]
```

Innehåll

Repetition

Skapa lista

Index

Ändra ett element

Lägg till element

Ta bort element

List slicing

[start:slut]

[start:slut:steg]

Sträng till lista

Tecken för tecken

Ord

Listor till strängar

Köer och Stackar

En kö

En stack

Övningar

List slicing

[start:slut]

```
1 en_lista = ["Frodo", "Pippin", "Sam", "Merry"]  
2 print(en_lista[1:3])
```

```
1 ["Pippin", "Sam"]
```

List slicing

[start:slut]

```
1 en_lista = ["Frodo", "Pippin", "Sam", "Merry"]  
2 print(en_lista[1:3])
```

```
1 ["Pippin", "Sam"]
```

```
1 print(en_lista[1:])
```

```
1 ["Pippin", "Sam", "Merry"]
```


List slicing

[start:slut]

```
1 en_lista = ["Frodo", "Pippin", "Sam", "Merry"]
```

```
2 print(en_lista[1:3])
```

```
1 ["Pippin", "Sam"]
```

```
1 print(en_lista[1:])
```

```
1 ["Pippin", "Sam", "Merry"]
```

```
1 print(en_lista[:2])
```

```
1 ["Frodo", "Pippin"]
```

List slicing

[start:slut:steg]

```
1 cool_lista = [0,1,2,3,4,5,6,7,8,9]
2 print(cool_lista[::2])
```

```
1 [0,2,4,6,8]
```

List slicing

[start:slut:steg]

```
1 cool_lista = [0,1,2,3,4,5,6,7,8,9]
2 print(cool_lista[::2])
```

```
1 [0,2,4,6,8]
```

```
1 print(cool_lista[::-1])
```

```
1 [9,8,7,6,5,4,3,2,1,0]
```

List slicing

[start:slut:steg]

```
1 cool_lista = [0,1,2,3,4,5,6,7,8,9]
2 print(cool_lista[::2])
```

```
1 [0,2,4,6,8]
```

```
1 print(cool_lista[::-1])
```

```
1 [9,8,7,6,5,4,3,2,1,0]
```

```
1 print(cool_lista[1:6:3])
```

```
1 [1,4]
```

Innehåll

Repetition

Skapa lista

Index

Ändra ett element

Lägg till element

Ta bort element

List slicing

[start:slut]

[start:slut:steg]

Sträng till lista

Tecken för tecken

Ord

Listor till strängar

Köer och Stackar

En kö

En stack

Övningar

Strängar till listor

Konvertering

```
1 min_text = "Upp till kamp"  
2 min_lista = list(min_text)  
3 print(min_lista)
```

Strängar till listor

Konvertering

```
1 min_text = "Upp till kamp"  
2 min_lista = list(min_text)  
3 print(min_lista)
```

```
1 ['U', 'p', 'p', ' ', 't', 'i', 'l', 'l', ' ', 'k', 'a', 'm', 'p']
```

Strängar till listor

Bryt upp i ord

```
1 min_text = "Upp till kamp"  
2 min_lista = min_text.split()  
3 print(min_lista)
```


Strängar till listor

Bryt upp i ord

```
1 min_text = "Upp till kamp"  
2 min_lista = min_text.split()  
3 print(min_lista)
```

```
1 ['Upp', 'till', 'kamp']
```

Med `.split(x)` kan man stycka upp strängen varje gång `x` dyker upp.

Listor till strängar

```
1 min_lista = ["U", "p", "p", " ", "n", "e", "r"]
2 min_text = "".join(min_lista)
3 print(min_text)
```

Listor till strängar

```
1 min_lista = ["U", "p", "p", " ", "n", "e", "r"]
2 min_text = "".join(min_lista)
3 print(min_text)
```

```
1 "Upp ner"
```

Innehåll

Repetition

Skapa lista

Index

Ändra ett element

Lägg till element

Ta bort element

List slicing

[start:slut]

[start:slut:steg]

Sträng till lista

Tecken för tecken

Ord

Listor till strängar

Köer och Stackar

En kö

En stack

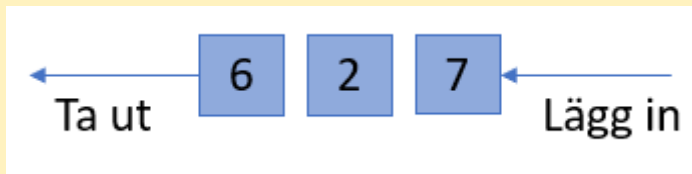
Övningar

Köer och Stackar

I programmering är struktur viktigt, två grundläggande sätt att strukturera data är med

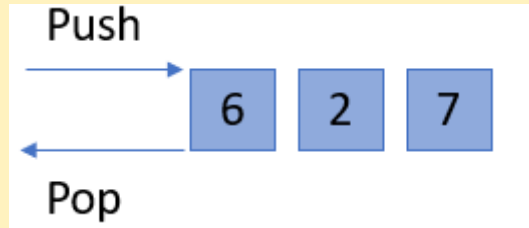
- ▶ en kö
- ▶ en stack.

Kö



En kö är en ordning där "först in först ut gäller" (FIFO, *first in first out*) och fungerar som en kö i en affär. Då kan man lägga till element längst bak och plocka ut element längst fram

Stack



En stack är en ordning där "sist in först ut gäller" (LIFO, *last in first out*) och fungerar som en tallrikshög. Man lägger på nya tallrikar överst, och plockar alltid översta tallriken först.

Vi kan enkelt simulera en stack i Python med listor om vi bara använder

`.append(x)` och `.pop()`

Innehåll

Repetition

Skapa lista

Index

Ändra ett element

Lägg till element

Ta bort element

List slicing

[start:slut]

[start:slut:steg]

Sträng till lista

Tecken för tecken

Ord

Listor till strängar

Köer och Stackar

En kö

En stack

Övningar

Övningar

- ▶ Ladda ner filen `listor3.py`
 1. Dela upp `texten` i ord
 2. Hur många ord är det i `texten`?
 3. Räkna hur många ord som är exakt tre tecken långa.
 4. Dela nu istället upp `texten` i tecken. Hur många bokstäver finns det? Vilken bokstav är vanligast?
 5. Kryptera `texten` med ett Caesarchiffer.