

Programmering i Python

	F1	Grunder i Python
	F2	Loopar, moduler
→	F3	Funktioner, läsa/skriva filer
	F4	Dictionaries, läsa data från internet
	F5	?

Idag

- Repetition
- Genomgång uppgifter från förra gången
- Funktioner
- Läsa/Skriva till en fil på datorn
- Lösa uppgifter

Repetition

For-loopar listor

```
for element in list:  
    # Gör något med varje element i listan
```

For-loopar range

```
for i in range(10):  
    # Gör något 10 gånger, i är siffran vi är på
```

While-loopar

```
counter = 0  
while counter < 10:  
    counter = counter + 1
```

Moduler: Random och Time

```
import random  
import time  
random.randint(0, 10)  
time.sleep(1)
```

Genomgång sten sax påse

Sten, Sax, Påse mellan datorn och användaren.

Funktioner

Definition funktion i Python:

- Tar in 0 eller flera argument (värden) som läggs inom parenteserna
- Gör något med dessa
- Kan ge tillbaka (returnera) något
- När man kallar på en funktion skriver man `funktionsnamn(argument)`. Ex. `len("hej")`

Exempel på inbyggda funktioner i Python:

`max([1, 5, 10, 4])` : en funktion som tar in en lista med siffror och returnerar det högsta värdet

`sum([1, 2, 3, 4])` : en funktion som tar in en lista med siffror och returnerar summan av dessa

`print("Hej!")` : en funktion som tar in en sträng och skriver ut den till användaren

`len("Hej")` : en funktion som tar in en sträng och returnerar längden på strängen

Vår egen funktion

För att göra en funktion i Python skriver vi:

```
def fun():  
    # Denna funktion returnerar en siffra  
    return 1337
```

Funktionen returnerar en siffra, 1337, men gör ingenting annat

Den kommer inte att köras av python förrän vi skriver

`fun()` någon annanstans i koden

I parenteserna kan vi också välja att ha ett eller flera 'argument' som används av funktionen

En mer meningsfull funktion

Vi gör en funktion som faktiskt returnerar något meningsfullt

```
def birthyear(age):  
    return 2020 - age
```

Funktionen returnerar födelseåret för någon med åldern 'age'

```
birthyear(20) => 2000
```

```
birthyear(23) => 1997
```

```
birthyear(63) => 1957
```

Fråga

Vad stämmer om denna funktion?

```
def funktion(a, b, c):  
    return a == b and a == c
```

1. Den returnerar alltid True
2. Den returnerar alltid False
3. Den tar in 3 argument
4. Den returnerar True om a, b och c är samma värde
5. Den returnerar a, b och c

Funktioner

Varför skulle man göra sin egen funktion?

- Om det är något kodstycke man utför flera gånger (men med olika värden)
- Kan göra koden mer lättläst och delar upp den på ett bra sätt

Utan funktion

```
name1="Pierre"  
age1=15  
print("Hej", name1, "som är", age1, "år")
```

```
name2="Erik"  
age2=25  
print("Hej", name2, "som är", age2, "år")
```

```
name3="Nadja"  
age3=22  
print("Hej", name3, "som är", age3, "år")
```

Med funktion

```
def greeting(name, age):  
    print("Hej", name, "som är", age, "år")  
  
greeting("Pierre", 15)  
greeting("Erik", 25)  
greeting("Nadja", 22)
```

Funktioner

Gör din egen funktion sum i Python:

```
def my_sum(lista):  
    result = 0  
    for number in lista: # gå igenom varje värde i listan  
        result += number # plussa ihop värdena ett i taget, spara i resultat  
    return result  
  
print(sum([1, 2, 3, 4]))  
print(my_sum([1, 2, 3, 4]))
```

Funktioner

En funktion som tar tre tal och returnerar det största

```
def max_of_three(a, b, c):  
    if a >= b and a >= c: # om a är störst  
        return a  
    elif b >= a and b >= c: # om b är störst  
        return b  
    else: # annars, c är störst  
        return c  
  
print(max_of_three(-11, 200, 3))
```

Öppna filer med Python

Låt säga att vi har en textfil innehållandes t.ex.

- alla elever på en skola och deras betyg
- en textfil med årets banktransaktioner

Denna information kan vi använda oss av i ett program, men då måste vi öppna den filen.

Öppna filer

```
f = open("fil.txt", "r")
for line in f.readlines():
    print(line)
f.close()
```

```
f = open("fil.txt", "r")
for line in f.readlines():
    for word in line.split():
        print(word)
f.close()
```

Skriva till filer med Python

Skapa en fil och skriv till den

```
f = open("nyttfilnamn.txt", "w") # 'w' betyder write
f.write("Detta skrivs i en fil, och skriver över det som stod innan")
f.close()
```

Lägga till text till filer med Python

Skapa en fil och skriv till den

```
f = open("nyttfilnamn.txt", "a") # 'a' betyder append (lägg till)
f.write("\nDetta lägger till denna text i en fil på ny rad")
f.close()
```

Plotta arbetslöshet i en graf

Vi har en textfil med årtal och arbetslöshet och vill plotta det i ett diagram

1. Installera paketet *matplotlib* med er terminal (terminal i mac, powershell/cmd i windows)

Mac:

```
python3 -m pip install --user pip  
python3 -m pip install --user matplotlib
```

Windows:

```
py -m pip install --user pip  
py -m pip install --user matplotlib
```

2. Ladda ned textfilen från github, spara i ny mapp på datorn
3. Skapa en Pythonfil (plotta.py) i samma mapp som textfilen
4. Sedan skriver vi koden tillsammans

Uppgifter:

1. Gör ett **program** som ber användaren om ett filnamn och sedan skapar den filen (bestäm själv om du vill skriva något i filen)
2. Gör ett **program** som ber användaren om namn, efternamn och ålder och skriver det till en ny rad i en fil (namnlista.txt), filen ska kunna fyllas på och inte nollställas
3. Skapa en **funktion** som tar in förnamn, efternamn och ålder och skriver det till en ny rad i en fil (namnlista.txt). Funktionen behöver inte returna något.
4. Gör en **funktion** som tar in en sträng och returnerar strängen baklänges.
Tips: Gör en for-loop som läser strängen baklänges och bygger upp en ny sträng, eller googla på andra möjliga lösningar. Annars, fråga mig!
5. Skapa en **funktion** som kontrollerar om en sträng är ett palindrom (samma baklänges som framlänges). Den ska returnera True eller False.
6. Gör en **funktion** som tar in ett namn (sträng) och returnerar strängen i rövarspråket (hej => hohejoj, erik => erorikok) och testa den med några strängar.