

Programmering i Python

	F1	Grunder i Python
	F2	Loopar, moduler
→	F3	Funktioner, läsa data, dictionaries
	F4	Dictionaries, API:er/requests
	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 argument (värden) som läggs inom parenteser
- Gör något med värdena/argumenten
- 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 add(a, b):  
    # Denna funktion returnerar a + b  
    return a + b
```

Funktionen returnerar a+b, men gör ingenting annat

Den kommer inte att köras av python förrän vi skriver

`add()` någon annanstans i koden

I parenteserna behöver vi också ha två 'argument' som används av funktionen.

Ex. `add(10, 20)`

En till funktion

```
def betalar_fullpris(age):  
    if age > 17 && age < 65:  
        return True  
    else:  
        return False
```

Funktionen returnerar True eller False, beroende på om personen ska betala fullpris för exempelvis en biljett

```
betalar_fullpris(20) => True
```

```
betalar_fullpris(12) => False
```

```
betalar_fullpris(66) => False
```

Fråga

Vad stämmer om denna funktion?

```
def funktion(a):  
    vokaler = "aeiouyåäö"  
    if a in vokaler:  
        return True  
    else:  
        return False
```

1. Den returnerar alltid True
2. Den returnerar alltid False
3. Den tar in ett argument
4. Den returnerar True om a är en vokal
5. Den returnerar True om a är en vokal

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

Vi har en textfil innehållandes t.ex.

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

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

Öppna filer

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

Öppna filer med Python - "r"

Öppna en textfil och gå igenom varje ord:

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

Skriva till filer med Python - “w”

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 - “a”

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()
```

Fråga

Hur lägger vi till text i slutet av filen "betyg.txt", och skapar en ny rad?

1. `open("betyg.txt", "r").write("Betyg: A\n")`
2. `open("betyg.txt", "a").write("Betyg: A\n")`
3. `open("betyg.txt", "r").write("Betyg: A")`
4. `open("betyg", "a").write("Betyg: A\n")`

Installera externa moduler i python

- Allt finns inte i Python, men nästan alla moduler man kan tänka sig går att ladda ned och installera som tillägg
- Python har ett system för att installera externa moduler. Det heter pip.
- För att installera paket med pip så brukar man skriva in ett kommando i datorns kommandotolk
 - cmd/powershell på windows, terminal på Mac OS
- Vi ska installera en modul som används för att skapa diagram/grafer
 - matplotlib

Installera matplotlib

Enklaste sättet:

1. Gå in på kursens hemsida och ladda ned filen **install-matplotlib.py**
2. Öppna och kör filen i IDLE

Om det inte fungerar så testas vi en annan lösning...

Installera matplotlib om det första sättet ej fungerar

Annat sätt:

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

Mac:

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

Windows:

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

Plotta arbetslöshet i en graf

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

1. Ladda ned textfilen `arbetsloshet.txt` från github-sidan, spara i en ny mapp på datorn
2. Skapa en Pythonfil (`plotta.py`) i samma mapp som textfilen med hjälp av IDLE
3. 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 som argument och skriver det till en ny rad i en fil (namnlista.txt). Funktionen behöver inte returnera något. Kalla sedan på funktioner ett antal gånger.
4. Gör en **funktion** som tar in en bokstav som argument och returnerar True om bokstaven är en konsonant och annars False. Testa funktionen med olika värden.
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 en textsträng och returnerar strängen i rövarspråket (hej => hohejoj, erik => erorikok) och testa den med några strängar.