

# Programmering i Python

	<b>27 oktober</b>	Grunder i Python
	<b>10 november</b>	Slingor, moduler
➡	<b>17 november</b>	Funktioner, läsa data
	<b>1 december</b>	?
	<b>8 december</b>	?

# Idag

- Repetition
- Genomgång uppgift 4 från förra gången
- Lite mer om listor
- Funktioner
- Läs/Skriv till en fil på datorn
- Hämta data från internet (om vi hinner)
- Lös uppgifter individuellt

# Repetition

## For-loopar över listor

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

## For-loopar över range

```
for i in range(10):  
    # Gör något 10 gånger
```

## While-loopar

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

## Moduler: Random och Time

```
random.randrange(10)  
time.sleep(1)
```

# Genomgång uppgift 4

Sten, Sax, Påse mellan datorn och användaren.

1. Vi sätter datorns poäng till 0 och spelarens poäng till 0
2. Vi låter datorn välja ett slumpstal mellan 1 och 3
3. Vi låter användaren välja ett tal mellan 1 och 3
4. Vi jämför användarens och datorns tal och ser vem som vunnit
5. Vi lägger till poäng till den som vann, och printar vem som vann
6. Gör om punkt 1 till 4 tre gånger, sen presenteras vinnaren

# Genomgång uppgift 4

```
from random import randint
alternatives = ["Sten", "Sax", "Påse"]

computer_guess = alternatives[ random.randrange(3) ]
user_guess = input("Sten, Sax eller Påse?")
if user_guess == computer_guess:
    print("Oavgjort! Ni valde båda " + computer_guess)
elif user_guess == "Sten":
    if computer_guess == "Sax":
        print("Du vinner med " + user_guess + " mot " + computer_guess)
    else:
        print("Du förlorar med " + user_guess + " mot " + computer_guess)
elif user_guess == "Sax":
    if computer_guess == "Påse":
        print("Du vinner med " + user_guess + " mot " + computer_guess)
    else:
        print("Du förlorar med " + user_guess + " mot " + computer_guess)
elif user_guess == "Påse":
    if computer_guess == "Sten":
        print("Du vinner med " + user_guess + " mot " + computer_guess)
    else:
        print("Du förlorar med " + user_guess + " mot " + computer_guess)
```

# Lite mer om listor

## 2 viktiga metoder för listor: append, remove

`list.append(värde)` → Läger till ett värde i listan

`list.remove(värde)` → Tar bort ett värde ur listan

### Exempel:

```
list = [1, 2, 3]
```

```
list.append(4)
```

```
list.append(5)
```

```
list.remove(1)
```

```
print(list)
```

# Lite mer om listor

**Låt säga att vi har en lista:**

```
list1 = [-10, -8, -5, 1, 2, 3, 4, 5, 6, 7, 8]  
pos_list = []
```

**Och vi vill ta bort alla tal i listan som är negativa**

```
for element in list1:  
    if element > 0:  
        pos_list.append(element) # ny lista med alla positiva element  
print(pos_list)
```

# Funktioner

## Definition funktion i Python:

- Tar in ett eller flera argument (värden)
- Gör något med dessa
- Kan ge tillbaka (returnera) något
- När man kallar på en funktion skriver man `funktionsnamn(argument)`

## 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



# 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="Kalle"  
age1=15  
print("Hej " + name1 + ", " + str(age1) + " år")
```

```
name2="Erik"  
age2=25  
print("Hej " + name2 + ", " + str(age2) + " år")
```

```
name3="Nadja"  
age3=22  
print("Hej " + name3 + ", " + str(age3) + " år")
```

## Med funktion

```
def greeting(name, age):  
    print("Hej " + name + ", " + str(age) + " år")  
  
greeting("Kalle", 15)  
greeting("Erik", 25)  
greeting("Nadja", 22)
```

# Funktioner

## Gör din egen funktion sum i Python:

```
def my_sum(list):  
    result = 0  
    for number in list:  
        result += number  
  
    return result  
  
print(sum([1, 2, 3, 4]))  
print(my_sum([1, 2, 3, 4]))
```

# Funktioner

I första kurstillfället så gjorde vi ett program som hittade max-värdet av tre tal. Låt oss göra om det till en funktion istället!

```
def max_of_three(a, b, c):  
    if a >= b and a >= c:  
        Return a  
    elif b >= a and b >= c:  
        Return b  
    else:  
        return c
```

```
print(max_of_three(1, 2, 3))  
=> "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():
    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")  
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")  
f.write("Detta lägger till denna text i en fil")  
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 cmd (visar på tavlan)

```
python -m pip install --user pip
```

```
python -m pip install --user matplotlib
```

2. Ladda ned textfilen från github, spara i mapp på datorn
3. Skapa en Pythonfil (plot.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, filen ska kunna fyllas på och inte nollställas
3. Gör en funktion som tar in en sträng och returnerar strängen baklänges. Du får inte använda metoden `reverse()`.
4. Skapa en funktion som kontrollerar om en sträng är ett palindrom (samma baklänges som framlänges. Tips: `string.reverse()` )
5. Skapa en funktion som tar in en lista med siffror och returnerar den största siffran
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