

A still from the movie Toy Story showing Woody and Buzz Lightyear. Woody is on the left, looking slightly concerned. Buzz is on the right, looking excited and gesturing with his right hand. The background is a blurred indoor setting.

FUNCTIONS

FUNCTIONS EVERYWHERE

Definierar en funktion Funktionens namn Funktionens parameter

```
def shout(text):  
    result = text.upper()  
    print(result)
```

Kod i
funktionen

```
shout("Anton är bäst!")
```

Kör funktionen "shout"

Med argumentet
"Anton är bäst!"

Men, namngivning? Varför är det viktigt? Och varför är det svårt?

A

```
def shout(text):  
    result = text.upper()  
    print result  
  
shout("Anton är bäst!")
```

C

```
def hejsan(hoppsan):  
    tjosan = hoppsan.upper()  
    print tjosan  
  
hejsan("Anton är bäst!")
```

B

```
def a(b):  
    c = b.upper()  
    print c  
  
a("Anton är bäst!")
```

Moduler

Inbyggda funktioner i Python

Moduler – Funktioner för användning

- <https://docs.python.org/3/py-modindex.html>
- Många moduler som vi kan använda oss utav, t.ex.
 - math
- Där finns funktioner som t.ex.
 - ceil Avrundar ett tal uppåt
 - floor Avrundar ett ta nedåt
 - isnan Kontrollerar om ett tal **inte** är ett nummer
 - pi Returnerar pi

En annan modul

- Som ni kommer att använda er utav på inlämningsuppgiften är:
 - Random
- *Random*-modulen har t.ex. följande funktioner
 - *randint(a, b)* Genererar ett slumpstal mellan a & b (inkl, a & b)
 - *sample(population, k)* Väljer k antal värden från ett antal givna värden
 - *random()* Slumpar ett tal mellan 0,0 och 1,0

Att använda sig utav moduler

- Det är väldigt enkelt att använda sig utav dessa inbyggda moduler. Vill vi använda oss utav modulen "math" skriver vi:

```
# Importerar alla funktioner från modulen "math"
from math import *
print(floor(4.321))
print(ceil(4.321))
```



4.0
5.0

```
# Importerar funktionerna "floor" & "ceil" from modulen "math"
from math import floor, ceil
print(floor(4.321))
print(ceil(4.321))
```



4.0
5.0

```
# Importerar modulen "math"
import math
print(math.floor(4.321))
print(math.ceil(4.321))
```



4.0
5.0

Demo för moduler

Egna moduler

- Vi kan skapa egna moduler
 - I vårt fall genom en samling av funktioner
- Modulerna är ett eget dokument, med funktioner inuti sig.
 - Man anger sökväg + filnamn (utan ".py") för att inkludera modulen.

```
4-Songlist.py - C:\Users\TSANTI\Box Sync\HT 2015\DA354A - Introduktion till pro...  
File Edit Format Run Options Windows Help  
import let_it_be  
  
let_it_be.song()
```

```
let_it_be.py - C:\Users\TSANTI\Box Sync\HT 2015\DA354A - Introduktion till prog...  
File Edit Format Run Options Windows Help  
def song():  
    vers_1()  
    print ""  
    vers_2()  
    print ""  
    chours()  
    print ""  
    vers_3()  
    print ""  
    vers_4()  
    print ""  
    chours_2()  
    print ""  
    chours()  
    print ""  
    chours()  
    print ""  
    vers_5()  
    print ""  
    vers_6()  
    print ""  
    chours_2()  
    print ""  
    chours()  
  
def chours():  
    print "Let it be, let it be"  
    print "Let it be, let it be"  
    print "Whisper words of wisdom"  
    print "Let it be"  
  
def chours_2():  
    print "Let it be, let it be"  
    print "Let it be, let it be"  
    print "Yeah, there will be an answer let it be"  
  
def vers_1():  
    print "When I find myself in times of trouble"  
    print "Mother Mary comes to me"  
    print "Speaking words of wisdom, let it be"  
  
def vers_2():  
    print "And in my hour of darkness"  
    print "She is standing right in front of me"  
    print "Speaking words of wisdom, let it be"  
  
def vers_3():  
    print "And when all the brokenhearted people"  
    print "Living in the world agree"  
    print "There will be an answer, let it be"  
  
def vers_4():  
    print "For though they may be parted"  
    print "There is still a chance that they will see"  
    print "There will be an answer, let it be"  
  
def vers_5():  
    print "And when the night is cloudy"
```

Demo egna moduler

Att skriva ut saker

Hur gör vi det snyggast?

```
name = "Anton"
age = 31
city = "Lund"

print(name + " är " + str(age) + "år och bor i " + city)
#=> Anton är 31år och bor i Lund
```

```
name = "Anton"
age = 31
city = "Lund"

print("{} är {}år och bor i {}".format(name, age, city))
#=> Anton är 31år och bor i Lund
```

```
name = "Anton"
age = 31
city = "Lund"

print(f"{name} är {age}år och bor i {city}")
#=> Anton är 31år och bor i Lund
```

```
name = "Anton"
```

```
age = 31
```

```
city = "Lund"
```

```
print("{} är {}år och bor i {}".format(name, age, city))
```

A diagram consisting of three curved arrows. The first arrow starts from the variable 'name' in the first line of code and points to the first curly brace '{ }' in the format string. The second arrow starts from the variable 'age' in the second line of code and points to the second curly brace '{ }' in the format string. The third arrow starts from the variable 'city' in the third line of code and points to the third curly brace '{ }' in the format string.

```
#=> Anton är 31år och bor i Lund
```

Inlämningsuppgiften!