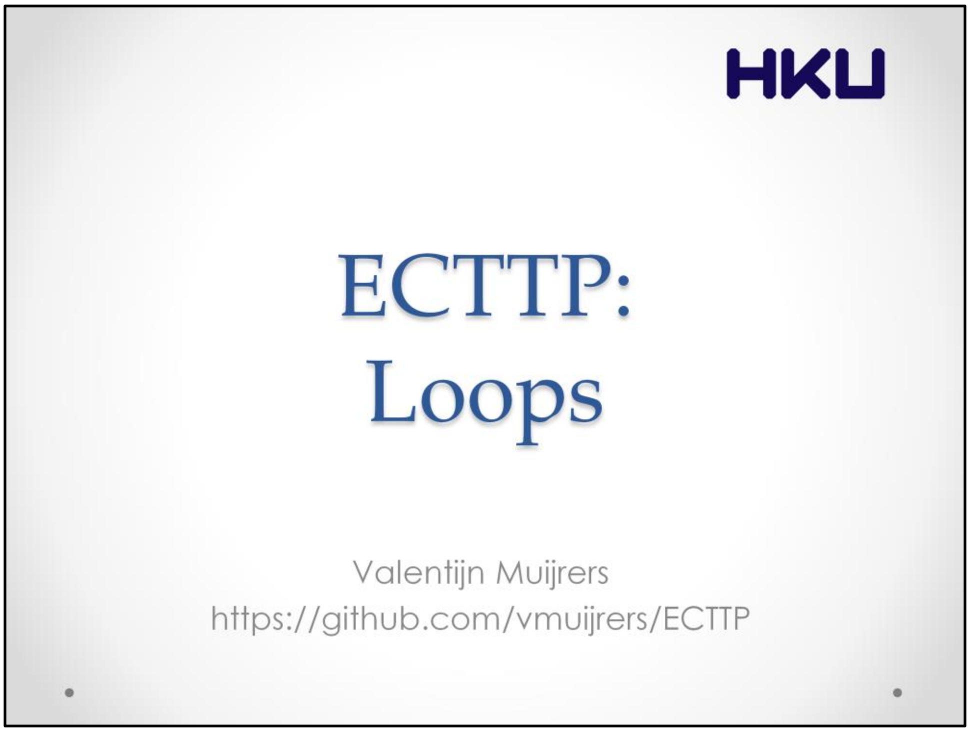
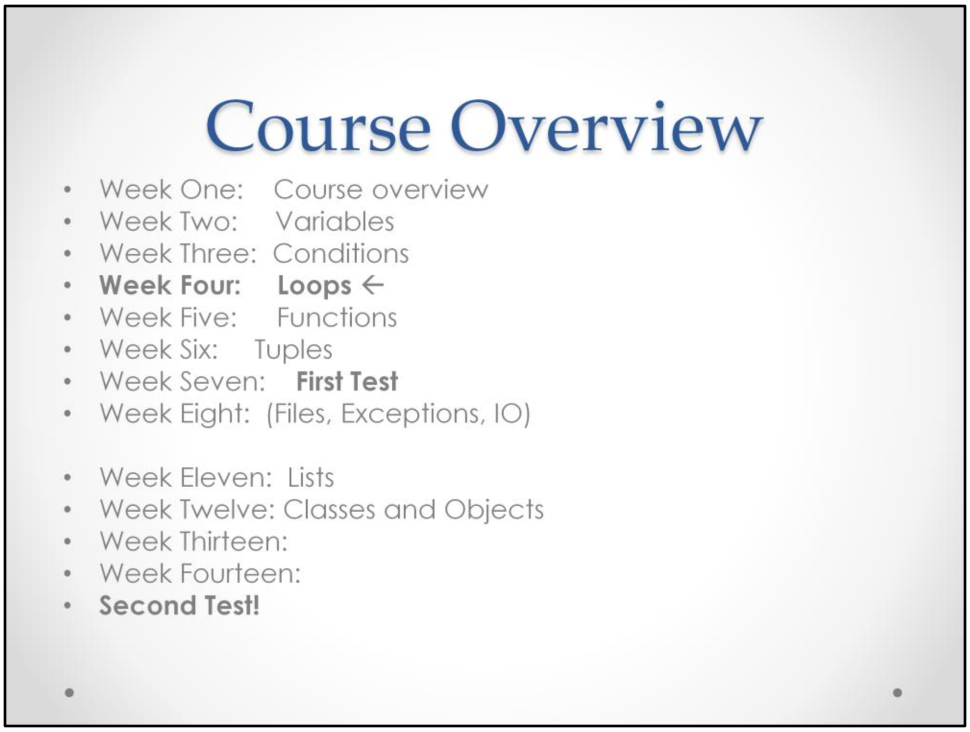


ECTTP: Loops

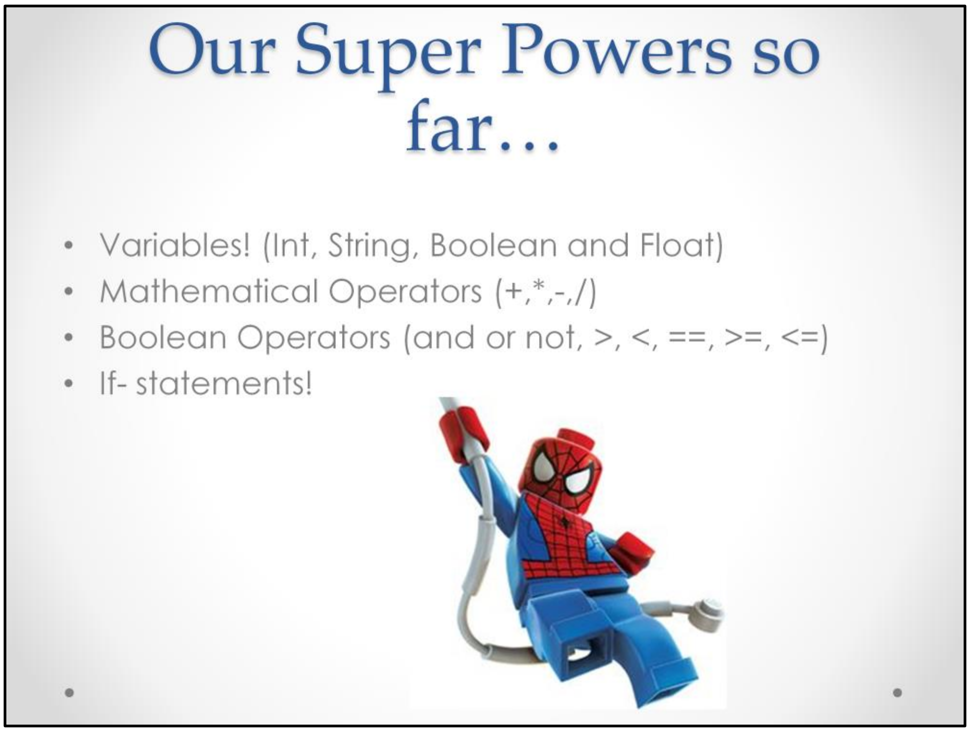
Valentijn Muijrers
https://github.com/vmuijrers/ECTTP



Course Overview

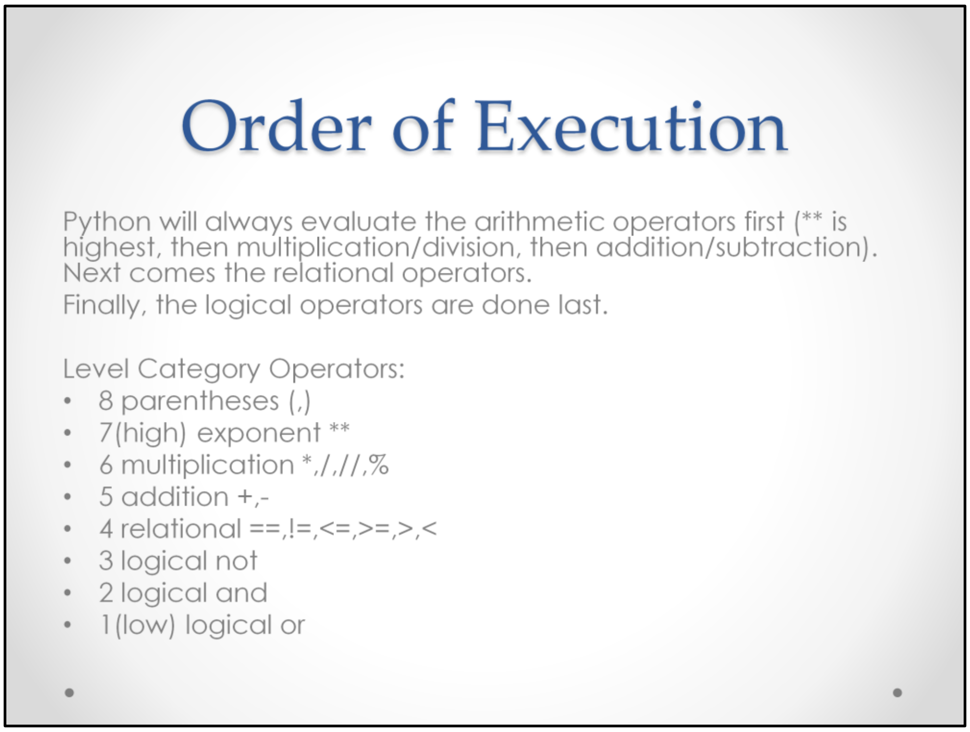
- Week One: Course overview
- Week Two: Variables
- Week Three: Conditions
- **Week Four: Loops ←**
- Week Five: Functions
- Week Six: Tuples
- Week Seven: **First Test**
- Week Eight: (Files, Exceptions, IO)

- Week Eleven: Lists
- Week Twelve: Classes and Objects
- Week Thirteen:
- Week Fourteen:
- **Second Test!**



Our Super Powers so far…

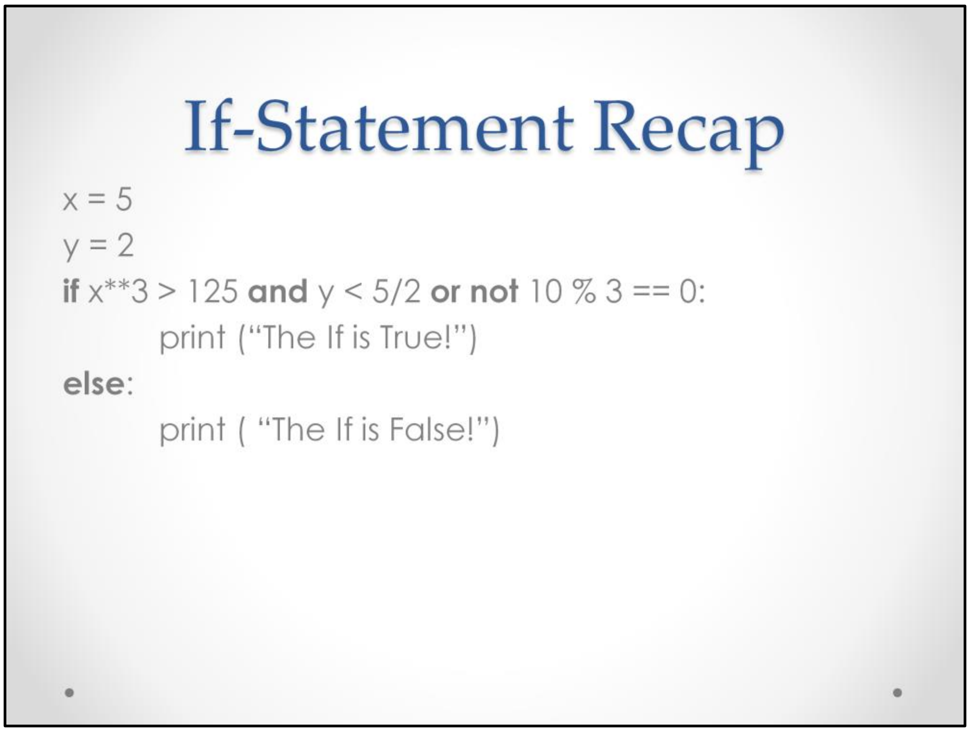
- Variables! (Int, String, Boolean and Float)
- Mathematical Operators (+,*,-,/)
- Boolean Operators (and or not, >, <, ==, >=, <=)
- If- statements!



Order of Execution

Python will always evaluate the arithmetic operators first (** is highest, then multiplication/division, then addition/subtraction). Next comes the relational operators.
Finally, the logical operators are done last.

Level Category Operators:
- 8 parentheses (,)
- 7(high) exponent **
- 6 multiplication *,/,//,%
- 5 addition +,-
- 4 relational ==,!=,<=,>=,>,<
- 3 logical not
- 2 logical and
- 1(low) logical or



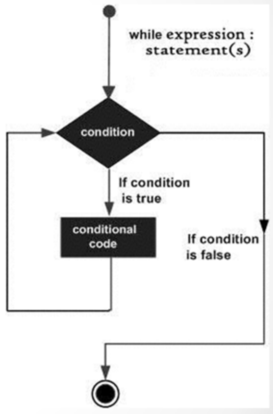
If-Statement Recap

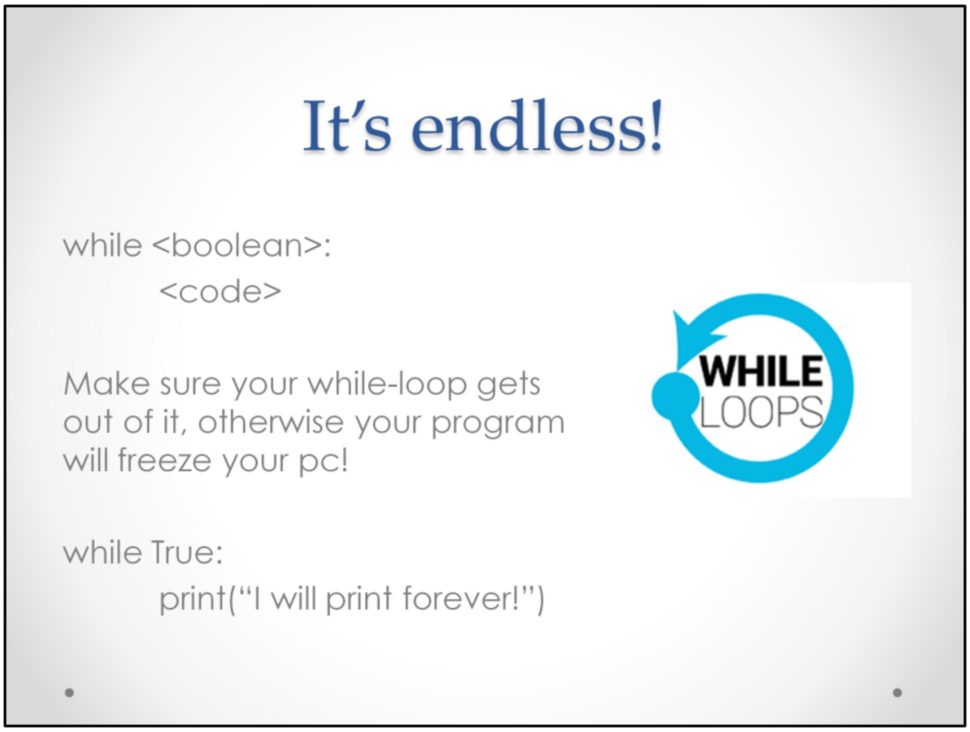
```
x = 5
y = 2
if x**3 > 125 and y < 5/2 or not 10 % 3 == 0:
        print (“The If is True!”)
else:
        print ( “The If is False!”)
```

While-loop

- While some condition is true, execute the code

```
x = 0
while x < 5 :
        x = x + 1
        print("I am looping!")
print("Aaaaand we'r e done")
```



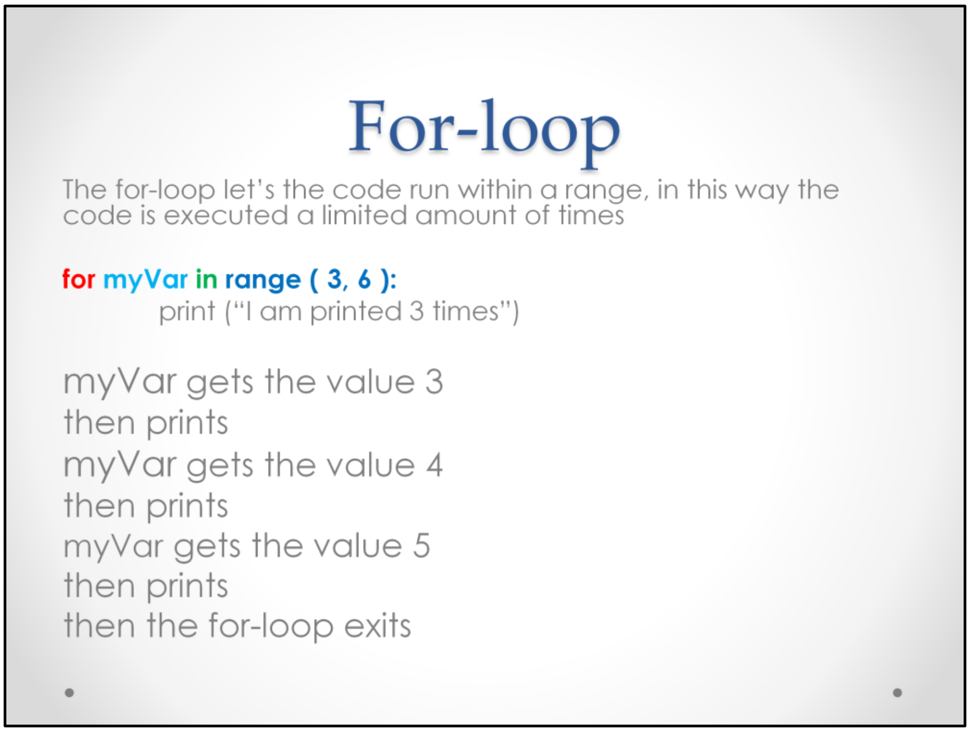


It’s endless!

```
while <boolean>:
        <code>
```

Make sure your while-loop gets out of it, otherwise your program will freeze your pc!

```
while True:
        print(“I will print forever!”)
```

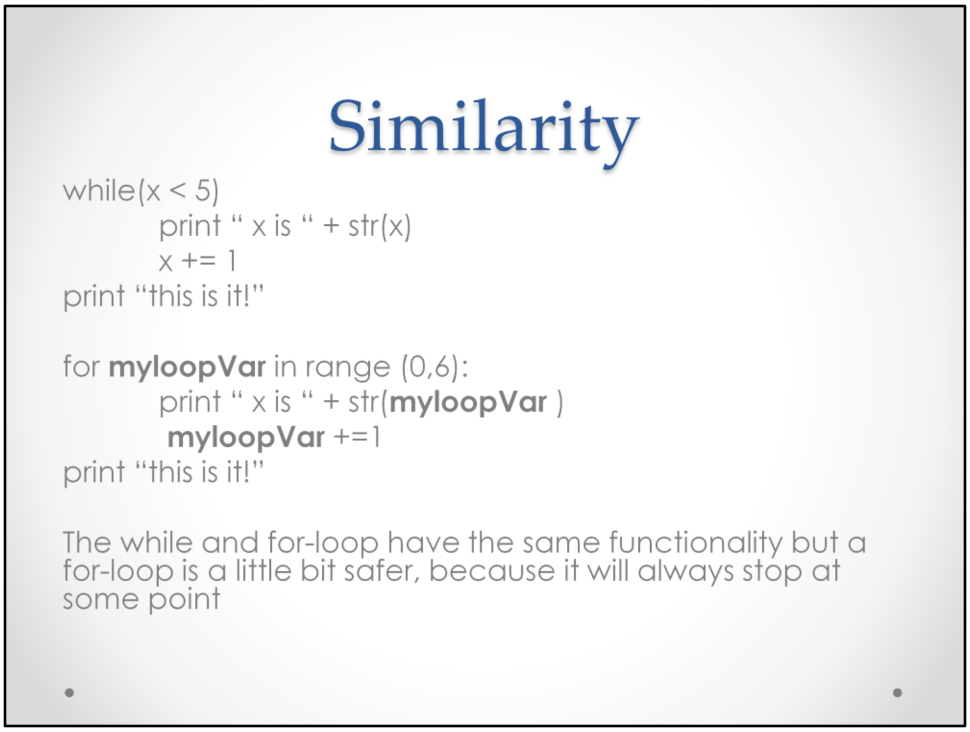


For-loop

The for-loop let's the code run within a range, in this way the code is executed a limited amount of times

```
for myVar in range ( 3, 6 ):
        print (“I am printed 3 times”)
```

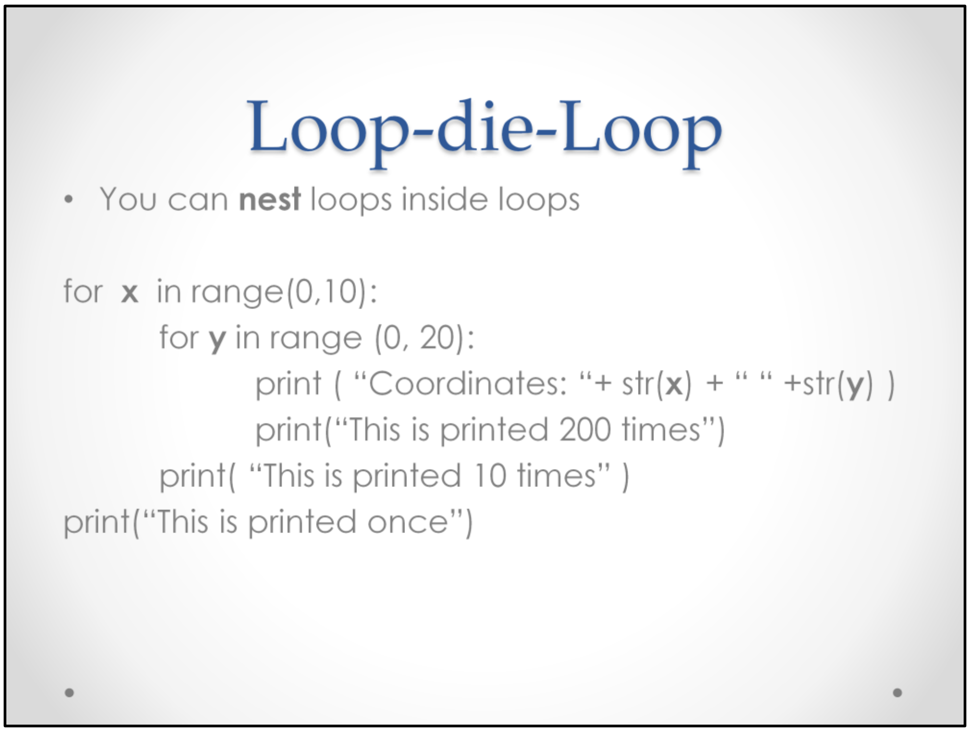
myVar gets the value 3
then prints
myVar gets the value 4
then prints
myVar gets the value 5
then prints
then the for-loop exits



Similarity

```
while(x < 5)
          print " x is " + str(x)
          x += 1
print "this is it!"

for myloopVar in range (0,6):
          print " x is " + str(myloopVar )
           myloopVar +=1
print "this is it!"
```

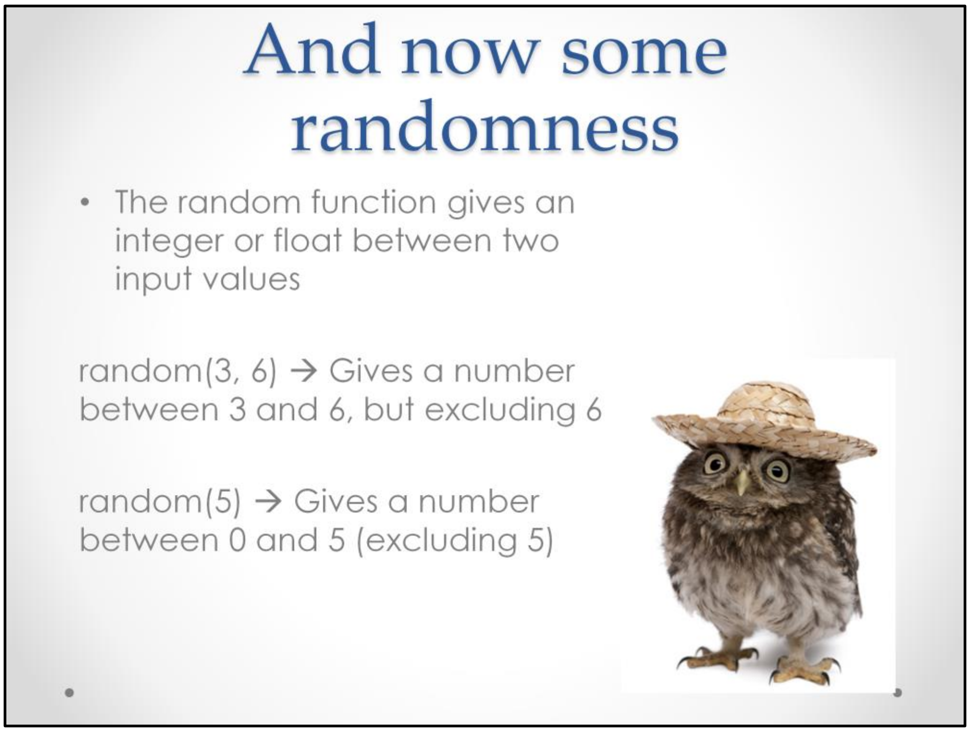
The while and for-loop have the same functionality but a for-loop is a little bit safer, because it will always stop at some point



Loop-die-Loop

- You can **nest** loops inside loops

```
for  x  in range(0,10):
        for y in range (0, 20):
                print ( “Coordinates: “+ str(x) + “ “ +str(y) )
                print(“This is printed 200 times”)
        print( “This is printed 10 times” )
print(“This is printed once”)
```

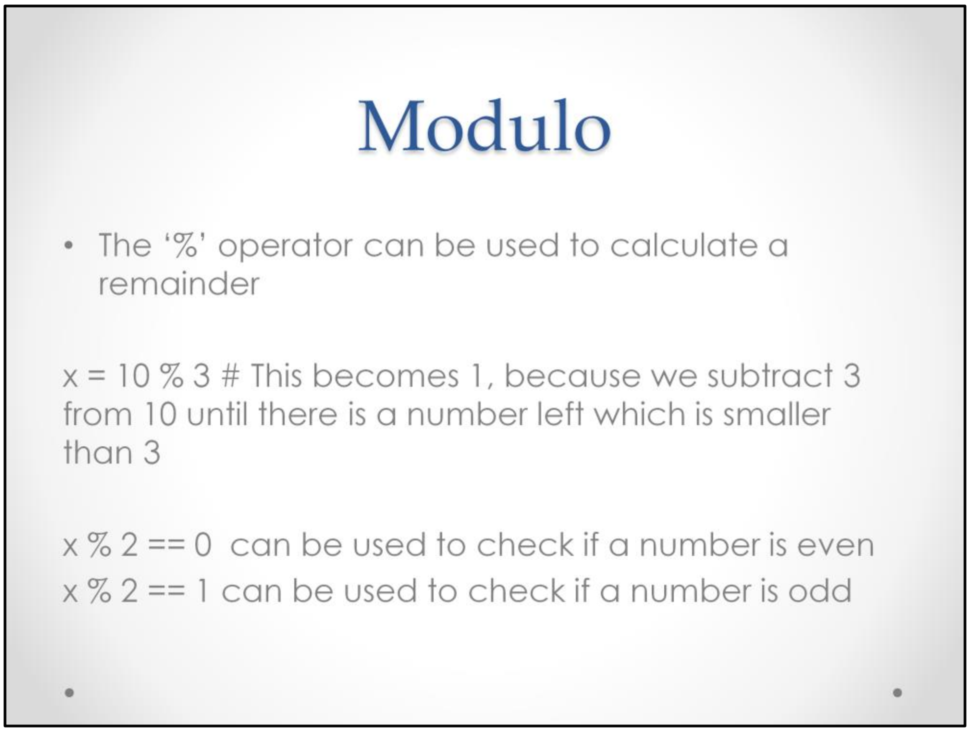


And now some randomness

- The random function gives an integer or float between two input values

random(3, 6) → Gives a number between 3 and 6, but excluding 6

random(5) → Gives a number between 0 and 5 (excluding 5)

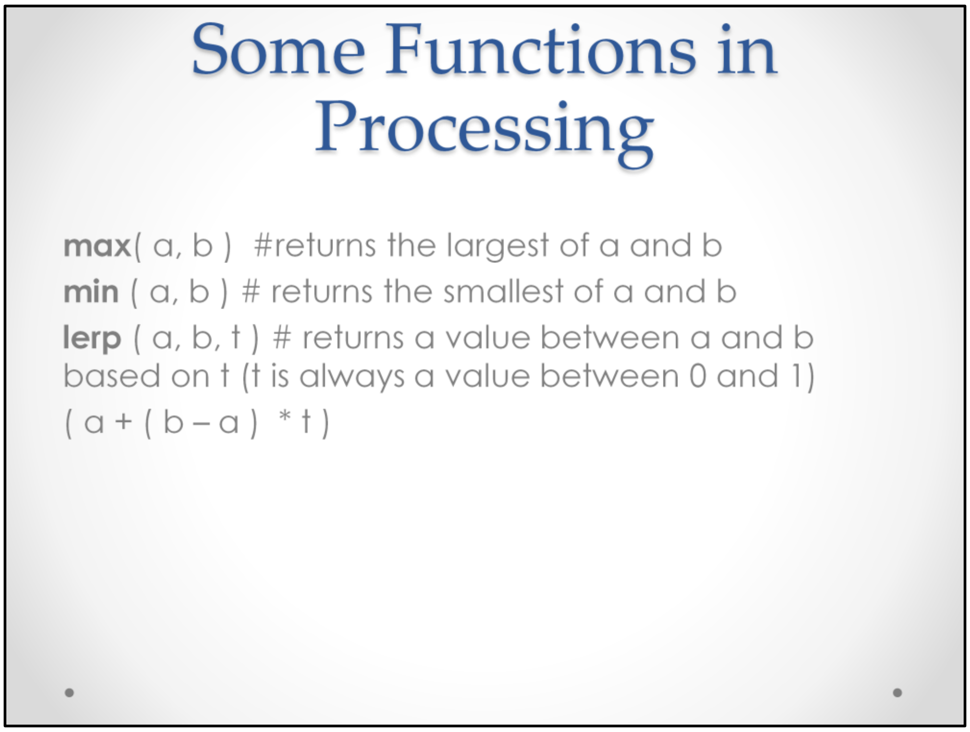


Modulo

- The ‘%’ operator can be used to calculate a remainder

x = 10 % 3 # This becomes 1, because we subtract 3 from 10 until there is a number left which is smaller than 3

x % 2 == 0 can be used to check if a number is even
x % 2 == 1 can be used to check if a number is odd



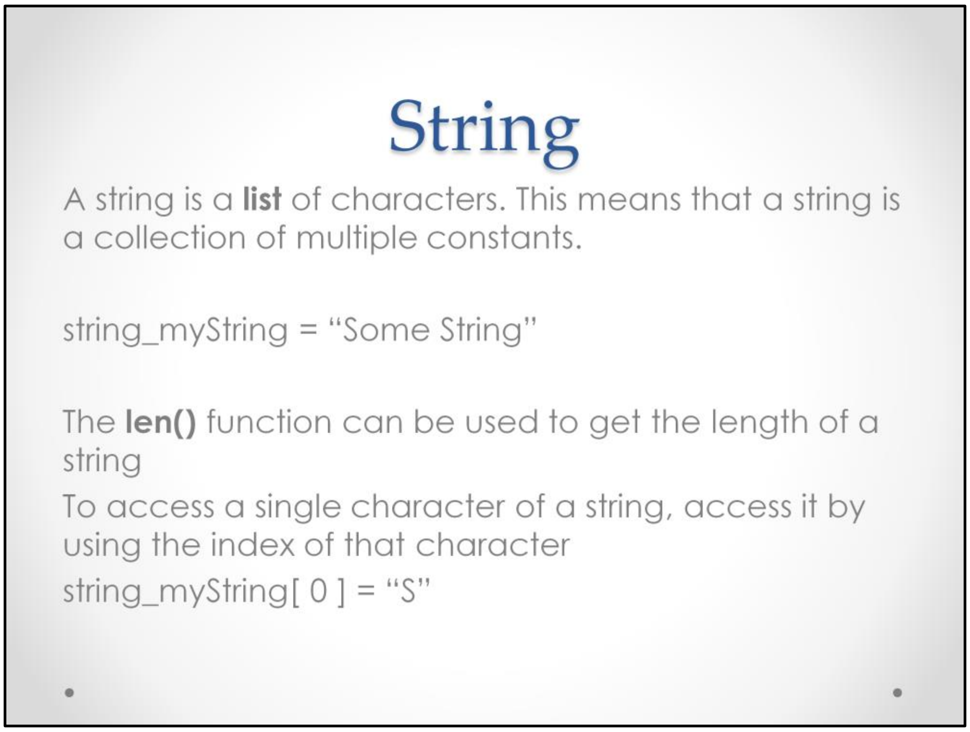
Some Functions in Processing

max(a, b) #returns the largest of a and b

min (a, b) # returns the smallest of a and b

lerp (a, b, t) # returns a value between a and b based on t (t is always a value between 0 and 1)

(a + (b – a) * t)



String

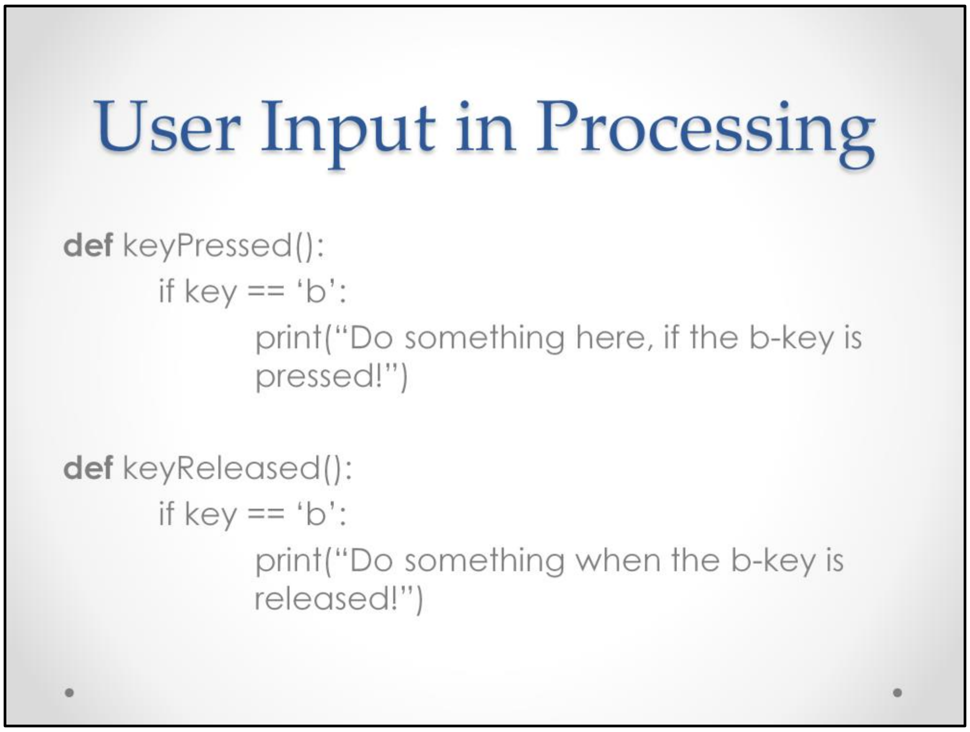
A string is a **list** of characters. This means that a string is a collection of multiple constants.

string_myString = “Some String”

The **len()** function can be used to get the length of a string

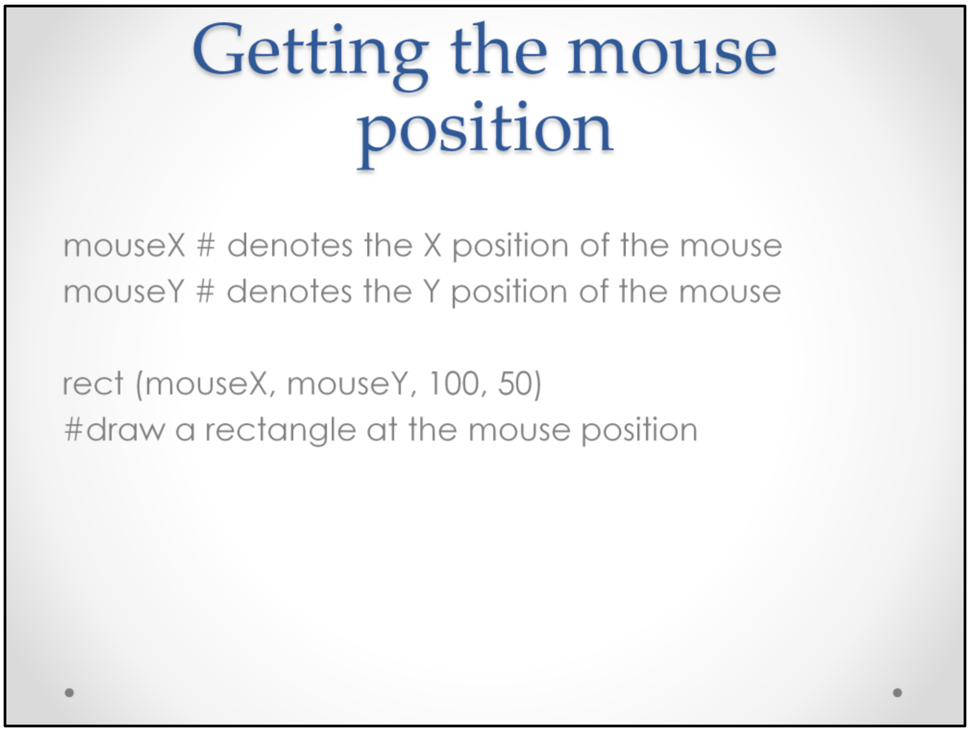
To access a single character of a string, access it by using the index of that character

string_myString[0] = “S”



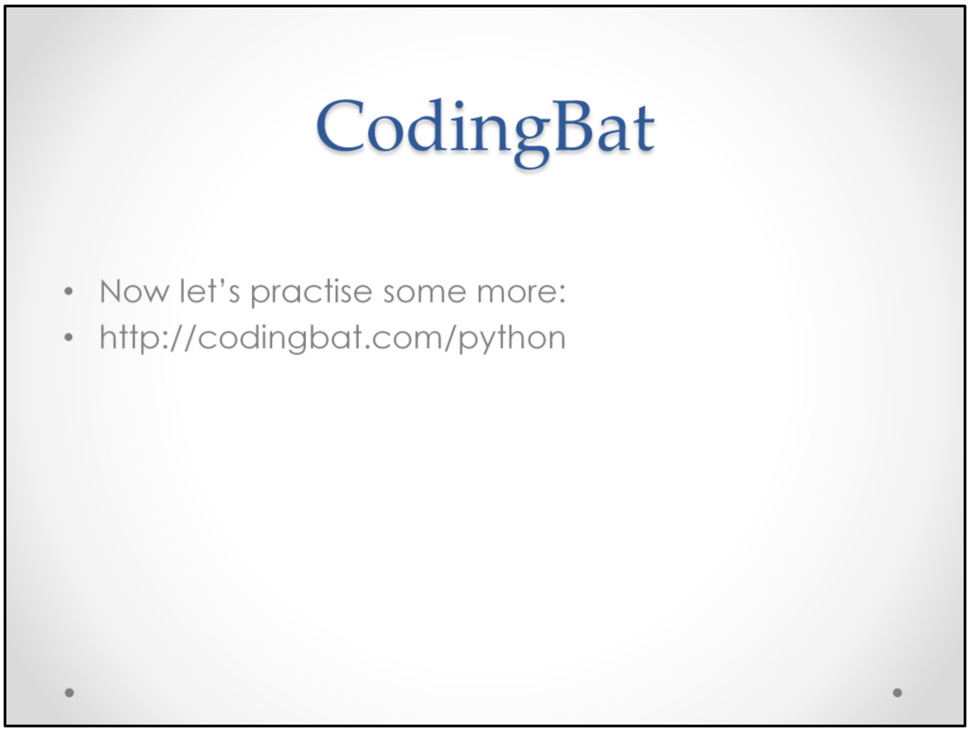
User Input in Processing

```
def keyPressed():
        if key == ‘b’:
                print(“Do something here, if the b-key is
                pressed!”)

def keyReleased():
        if key == ‘b’:
                print(“Do something when the b-key is
                released!”)
```



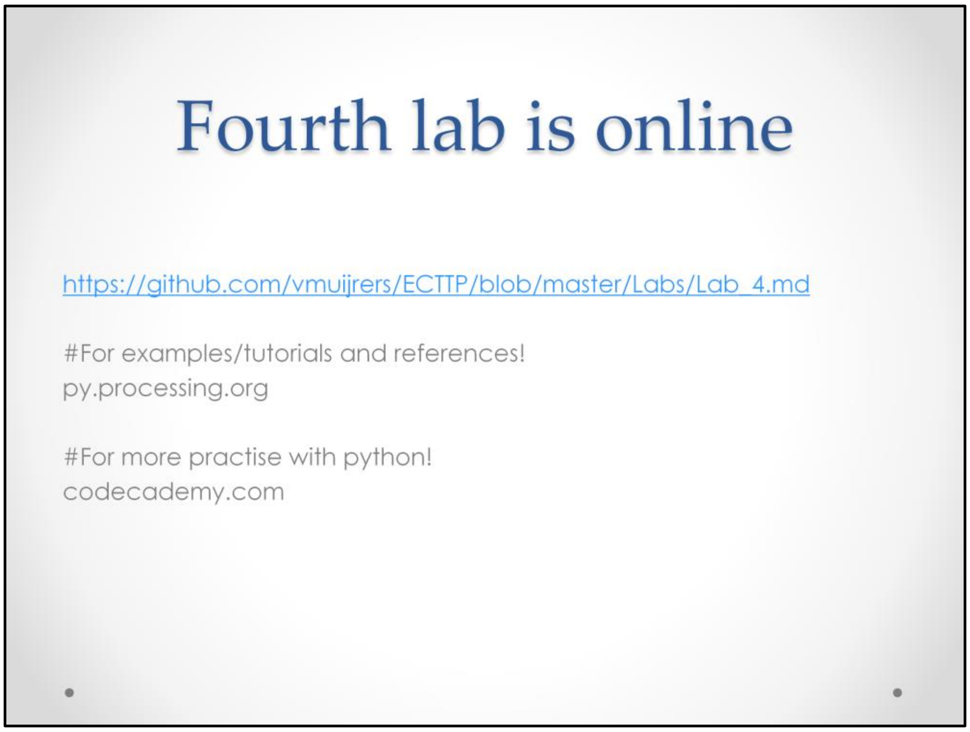
Getting the mouse position

```
mouseX # denotes the X position of the mouse
mouseY # denotes the Y position of the mouse

rect (mouseX, mouseY, 100, 50)
#draw a rectangle at the mouse position
```



CodingBat

- Now let's practise some more:
- http://codingbat.com/python



Fourth lab is online

https://github.com/vmuijrers/ECTTP/blob/master/Labs/Lab_4.md

#For examples/tutorials and references!
py.processing.org

#For more practise with python!
codecademy.com