UBS

Public

Introduction to Python

Katowice, IntQuant 2019

Alicja Cieślewicz
alicja.cieslewicz@ubs.com

INTQUANT

March 12, 2019

Python

- interpreted
- high-level
- general purpose

- created by
  Guido van Rossum
- first released in 1991

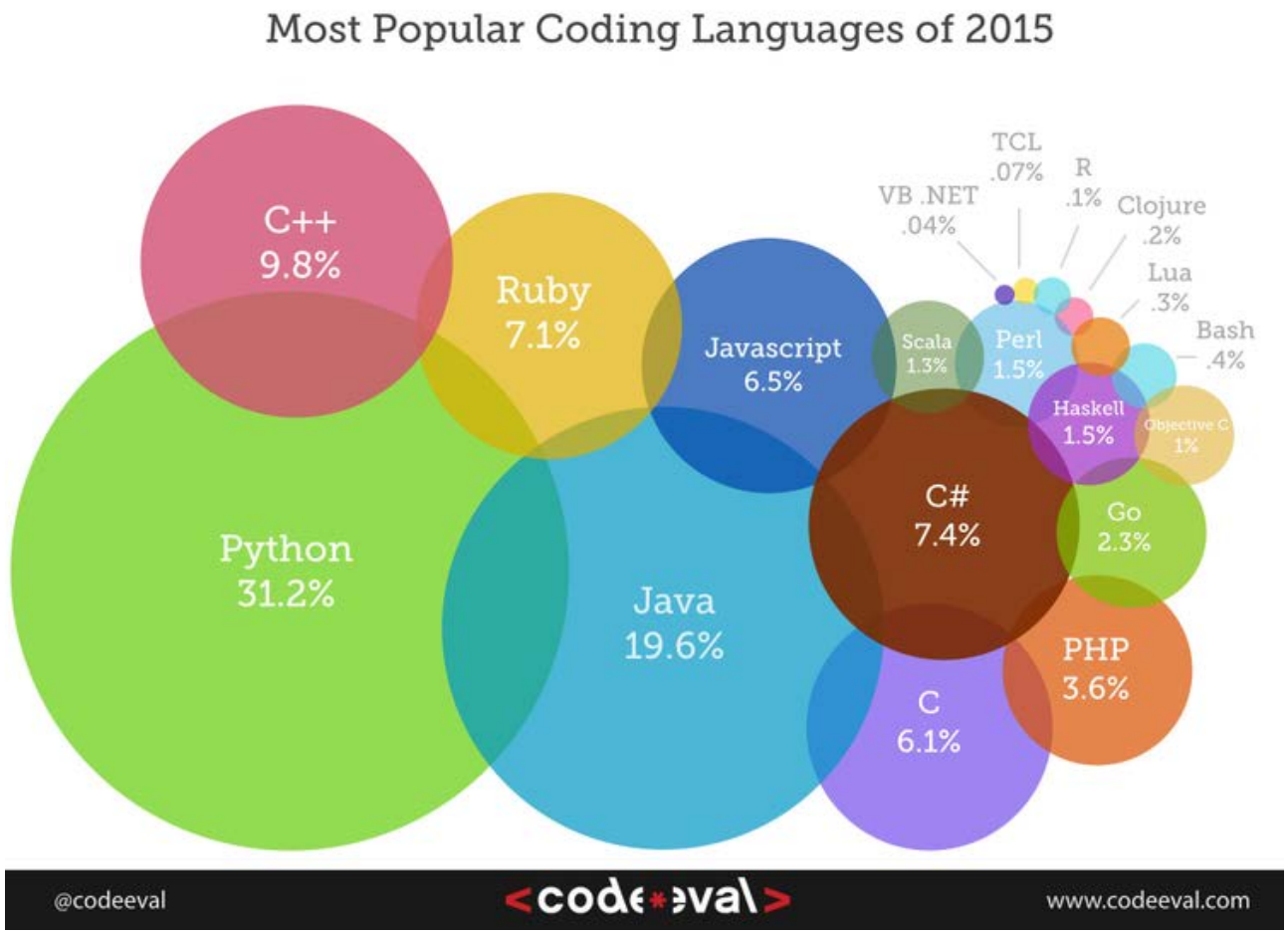

Table of contents

https://www.tutorialspoint.com/python/index.htm

https://www.codecademy.com

Operators

Arythmetic

Operator	Description	Example
+ Addition	Adds values on either side of the operator.	a + b = 30
- Subtraction	Subtracts right hand operand from left hand operand.	a – b = -10
* Multiplication	Multiplies values on either side of the operator	a * b = 200
/ Division	Divides left hand operand by right hand operand	b / a = 2
% Modulus	Divides left hand operand by right hand operand and returns remainder	b % a = 0
** Exponent	Performs exponential (power) calculation on operators	a**b =10 to the power 20
//	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity) –	9//2 = 4 and 9.0//2.0 = 4.0, -11//3 = -4, -11.0//3 =

Comparison

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(a == b) is not true.
!=	If values of two operands are not equal, then condition becomes true.	(a != b) is true.
<>	If values of two operands are not equal, then condition becomes true. (python 2.7)	(a <> b) is true. This is similar to != operator.
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true.
<	If the value of left operand is less than the value of right operand, then condition becomes true.	(a < b) is true.
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	(a >= b) is not true.
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true.

https://www.tutorialspoint.com/python/python_basic_operators.htm

Variables & Iterators & Loops

Python has five standard data types:

- Numbers
- String
- List
- Tuple
- Dictionary

```
number_one = 1  # int
number_two = 2.0  # float

string = 'Hello World!'

empty_list = []
my_list = [1, 5.0, 'bla bla']

empty_tuple = ()
my_tuple = (4, 5, 6)  # I'm a tuple and I'm immutable.

empty_dictionary = {}
my_dict = {'name': 'Alice',
           'eyes colour': 'blue',
           'height': 175.,
           'favourite numbers': [3,7]}
```

Loop Type & Description
while loop Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.
for loop Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
Nested loops You can use one or more loop inside any another while, for or do... while loop.

Control Statement & Description
break Terminates the loop statement and transfers execution to the statement immediately following the loop.
continue Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.
pass The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

```
>>> for item in my_list:
...     print(item * 2)
...
2
10.0
bla blabla bla
>>> for item in my_list:
...     if type(item) == int:
...         print(item * 2)
...
2
>>> i=0
... while i<3:
...     print(i)
...     i += 1
...
0
1
2
```

```
>>> for i in range(5):
...     print(i, end=' ')  # print i, --> python2.7
... print(i)
...
0 1 2 3 4 4
>>> for i in range(len(my_list)):
...     print(my_list[i])
...
1
5.0
bla bla
```

https://www.tutorialspoint.com/python/python_variable_types.htm

https://www.tutorialspoint.com/python/python_loops.htm

UBS