

Python: 101

Introduction

- Python is:
- high-level programming language
- open source, object oriented, dynamic
- used across FIXME (e.g., data science, web development, machine learning)
- runs across all major platforms (e.g., Windows, Mac, Linux)
- interpreted, *extendable* and interactive
- many libraries available (e.g., **numpy**, **scikit-learn**), great community
- stable!

Getting Python

- Official releases: <https://www.python.org/downloads/>
- Python distributions
- Anaconda: <https://www.anaconda.com/>
- ActivePython: <https://www.activestate.com/activepython>
- Enthought: <https://www.enthought.com>
- Provide: **python**, **ipython**, IDEs, assorted tools, pre-installed libraries, ...

(using Python3)

Python Program

- Collection of statements (sentences)
- That follow a specific grammar
- Strict rules
- Set of allowed
- Tell the Python interpreter what to do

Hello, world!

- Run from the interpreter

```
>>> print("Hello World!")  
Hello World!
```

- Run from the command line

```
print("Hello World!")
```

```
$ python hello.py
```

- Run from Spyder
- FIXME

Variables

- Examples:

```
counter = 10      # integer assignment  
pi = 3.14         # floating point  
name = "Peter"   # string
```

- Standard data types:
- numbers: 0, 1, 3.14, ...
- string: "string", "one string", "another string" (' or ")
- list: [0,1,2,3], ["list", "of", "words"]
- tuple: (0.0, 0.0), (1, 2)
- dictionary: {1: "one", 2: "two"}, {"key": "value", "another", "value"}

Basic Operators

- Arithmetic operators:
- addition
- subtraction
- multiplication
- division
- modulus
- exponent
- Comparison operators:
- equal
- not equal
- greater than
- less than
- greater than or equal

- less than or equal
- Logical Operators
- and
- or
- not

Conditionals

- if statement
- if .. else statement
- ...

Loops

- Loops
- while loop
- for loop
- nested loops
- Loop control
- break statement
- continue statement
- pass statement

Numbers

- int
- long
- float