Python Foundation I

Dr. Nan Meng (Naen)

Digital Health Laboratory
Department of Orthopaedics and Traumatology
University of Hong Kong

Contents

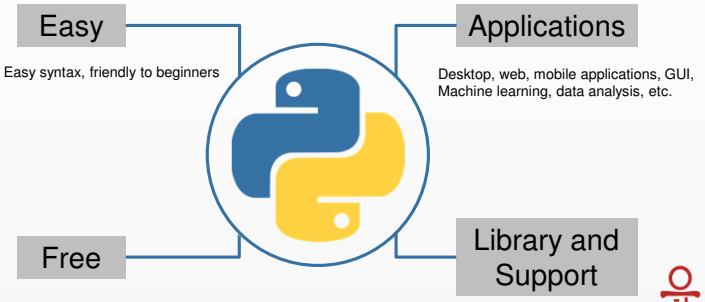
- Introduction to Python
- Anaconda, Notebook & Pycharm
- Packages in Python
- Data types, Variables and Operators
- Coding rules
- Printing formats
- Practices

Introduction to Python

- **Python** is a general purpose, high level interpreted language with easy syntax and dynamic semantics
- It was created by ***Guido van Rossum*** in the year 1989, and it is released in 1991.

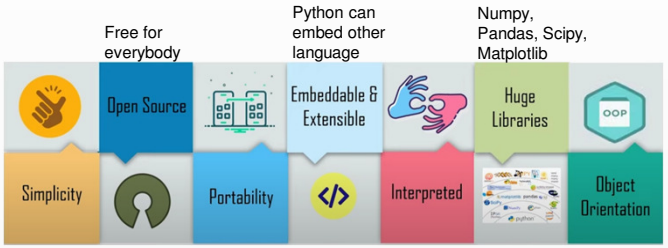
Introduction to Python

- Why is Python popular?



Introduction to Python

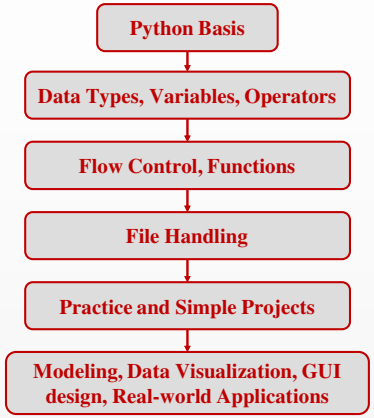
- Features



5

Introduction to Python

- Learning path



6

Packages in Python

- Numpy

NumPy

- **NumPy** stands for *Numerical Python* and it's a fundamental package for scientific computing in Python.
- **NumPy** provides Python with an extensive math library capable of performing numerical computations effectively and efficiently.

- In the year 2020, there is a Nature paper for **Numpy**

Harris, C.R., Millman, K.J., van der Walt, S.J. *et al.* **Array programming with NumPy**. *Nature* 585, 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2

7

Packages in Python

- Pandas

pandas

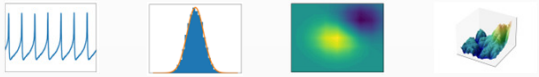
- **Pandas** is a package for data manipulation and analysis in Python. The name Pandas is derived from the econometrics term *Panel Data*.
- Pandas incorporates two additional data structures into Python, namely **Pandas Series** and **Pandas DataFrame**. These data structures allow us to work with *labeled* and *relational* data in an easy and intuitive manner.

8

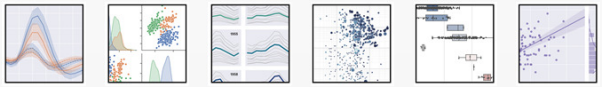
Packages in Python

- Matplotlib and Seaborn

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.



- Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.



9

Anaconda, Notebook & Pycharm

- Anaconda

- Anaconda is a **distribution** of packages built for data science. It comes with conda, a package and environment manager.

- Conda
- Package, dependency and environment management for multiple language—Python, R, Ruby, Lua, Scala, Java, etc.
- Support multiple systems: Windows, macOS and Linux.

10

Anaconda, Notebook & Pycharm

- Notebook

- Notebook is a **web application** that allows you to combine plain text, math equations, code, and figures all in one easily sharable document. Notebooks have rapidly become an essential tool when working with big data. You'll find them being used for **data cleaning and exploration**, **visualization**, **machine learning**, and **big data analysis**.

11

Anaconda, Notebook & Pycharm

- Pycharm

- PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

12

Data Types, Variables and Operators

- **Data types**
  1. Integers
  2. Floats
  3. Booleans
  4. Strings
  5. Lists
  6. Tuples
  7. Sets
  8. Dictionaries

13

Data Types, Variables and Operators

- **Operators**
  1. `+` Addition
  2. `-` Subtraction
  3. `*` Multiplication
  4. `/` Division
  5. `%` Mod: give the remainder after dividing
  6. `**` Exponentiation (note that ^ does not do this operation, as you might have seen in other languages)
  7. `//` Integer division: Divides and rounds down to the nearest integer

14

Data Types, Variables and Operators

- **Variables**
- Understanding how to use variables can turn Python into more than just a calculator. Using variables, as opposed to just raw numbers, has many advantages.

```
my_population = 74728
```

- Define the variable before accessing

```
x = 2
y = x
print(y)
```

```
x = 2
y = z
print(y)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-46-bf6aeaa77d9c> in <module>
----> 1 y = z
      2 print(y)

NameError: name 'z' is not defined
```


15

Coding Rules

1. Only use ordinary letters, numbers and underscores in your variable names. They can't have spaces, and need to start with a letter or underscore.
2. You can't use reserved words or built-in identifiers that have important purposes in Python
3. The pythonic way to name variables is to use all lowercase letters and underscores to separate words.

16

Printing Format

- Formatted String Literals
  1. Place a `f` or `F` before the opening quotation mark or triple quotation mark.
  2. Inside this string, write a Python expression between `{` and `}` characters that can refer to variables or literal values.
- Str.format()
  1. Enclose formatting directives with a pair of braces where variable will be substituted.
  2. Provide variable value / expression with function format.
- String Slicing and Concatenation

17

Practices

- In-class Practices
- Homework 

18

Conclusion

- Python: Features & Benefits
- Distribution & IDE: Anaconda, Notebook & Pycharm
- Packages: Numpy, Pandas, Matplotlib, Seaborn
- Data types, Variables and Operators
- Coding rules
- Printing formats: str.format()

19