ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

Chapter 1
Introduction to Python & Jupyter Notebook

Thu Huong Nguyen, PhD
Si Thin Nguyen, PhD

KHMT
Computer Science

http://vku.udn.vn/

About Authors

Thu Huong Nguyen

PhD in Computer Science at Université Côte d'Azur, France

Email: nthuong@vku.udn.vn

Address: Faculty of Computer Science, VKU

Si Thin Nguyen

PhD in Computer Science at Soongsil University, Korea

Email: nsthin@vku.udn.vn

Address: Faculty of Computer Science, VKU

Chapter Content

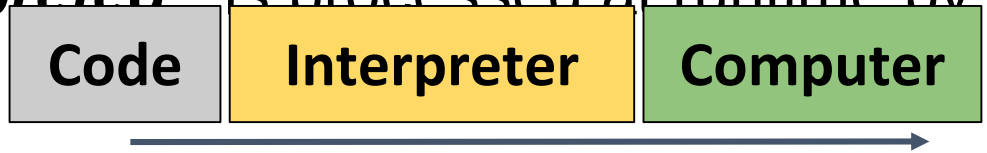
- What is Python?
- Philosophy of Python
- Brief Development History of Python
- Why Python?
- Who uses Python today?
- Python IDEs and Code Editors
- Jupyter Notebook
- The first Python Program

What is Python?

Python is …

➔ *a scripting language* characterized by

- ***Interpreted*** - is processed at runtime by the interpreter.



- ***Interactive*** − uses prompts and interact with the interpreter directly to write programs.
- ***Object-Oriented*** − supports Object-Oriented style or technique of programming that encapsulates code within objects

What is Python?

➔a *high-level* programming language designed emphasize

- ***code readability -*** designed to be highly readable.
- ***developer productivity*** - automates certain areas of computing systems to make the process of developing a program simple and fast.
- ***program portability*** - used to write software in a wide variety of application domains (e.g. web applications, GUI desktop applications, scientific and numeric application).

Philosophy of Python

- ➔ Based on ***minimalistic*** design philosophy in which complex tasks can be done in simple ways.
- ➔ *“Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered.”* - Guido van Rossum
- ➔ “***The Zen of Python***” is a collection of 19 guiding principles for writing computer programs in Python. (https://zen-of-python.info)

Brief Development History of Python

- ➔ Python was conceptualized by **Guido van Rossum** in the late 1980s.
- ➔ The first version of Python code (version 0.9.0) was released in February 1991.
- ➔ Python version 1.0 was released in January 1994.
- ➔ Python version 2.0 was introduced in October 2000.
- ➔ Python version 3.0 was released in December 2008

Why Python?

➔ Python is *easier* and *simpler* to learn and use than other languages.

- a *dynamically typed language*, i.e**.** need not declare the type of variables, declare the class like some other languages
- *simple syntaxes*, i.e. need not use any opening/closing braces, semicolons, etc.

Java

```
public class Main
{
public static void main(String[] args) {
int a= 10, b=20;
int result = a+b;
System.out.println("The  result a+b = " + result);    }
}
```

Python

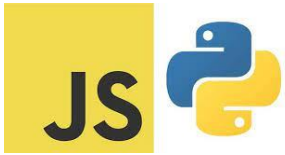
```
a=10

b=20

print ('The result a+b = ', a+b)
```

Why Python?

- Python is *open source*! Free! Massive online support from many resources and quality documentation worldwide.

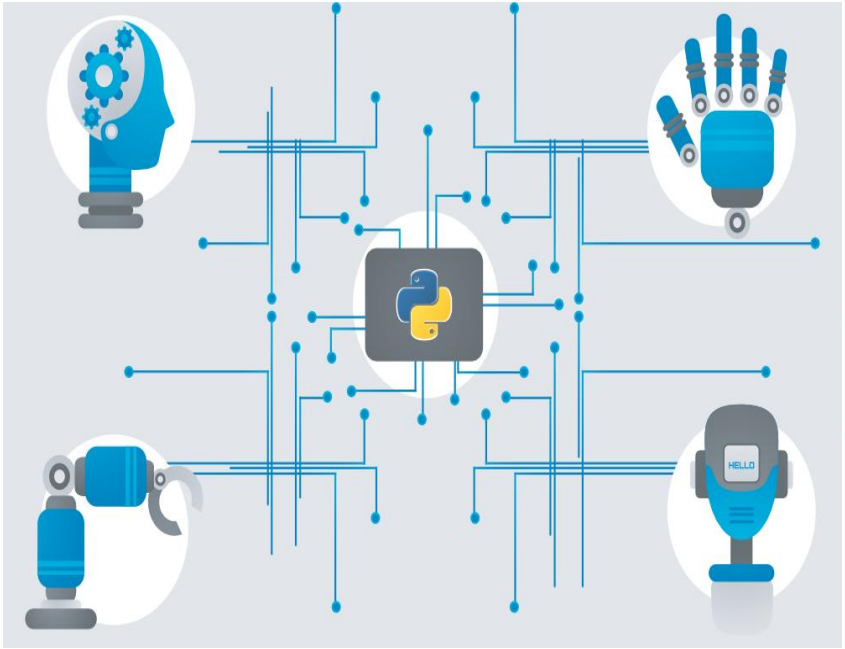
Python.Community

- Python projects can be integrated with other systems coded in different programming languages.



Why Python?

- ➔ Python is one of the best programming languages for Data science (DS) and AI.
  - ◆ Python's simplicity allows developers to put effort into solving AI problems instead of focusing on technical nuances of the language



Why Python?

TIOBE
the software quality company

About us | Join TIOBE | News | Coding Standards | TIOBE Index | Contact

Products | Quality Models | Markets | Request a demo

Jun 2022	Jun 2021	Change	Programming Language	Ratings	Change
1	2	↑	Python	12.20%	+0.35%
2	1	↓	C	11.91%	-0.64%
3	3		Java	10.47%	-1.07%
4	4		C++	9.63%	+2.26%
5	5		C#	6.12%	+1.79%
6	6		Visual Basic	5.42%	+1.40%
7	7		JavaScript	2.09%	-0.24%
8	10	↑	SQL	1.94%	+0.06%
9	9		Assembly language	1.85%	-0.21%

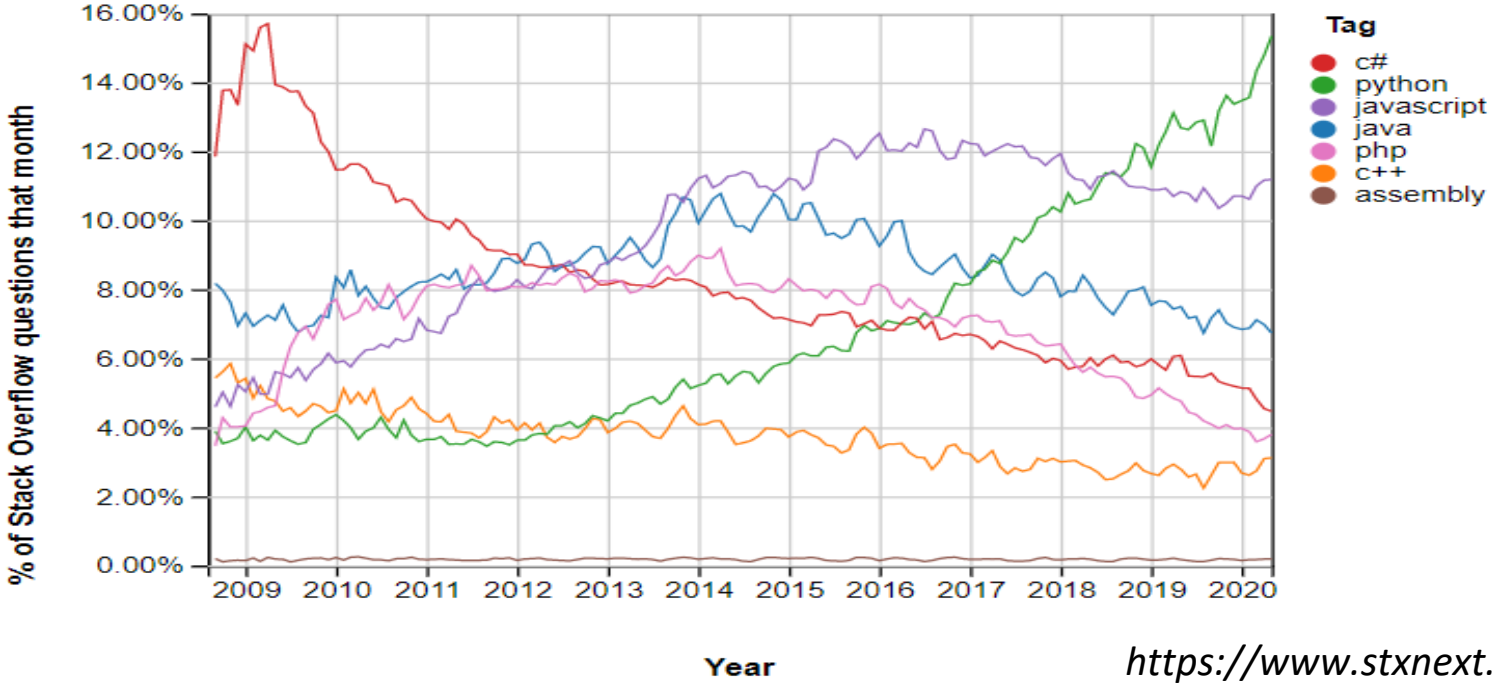
https://www.tiobe.com/tiobe-index/

Why Python?

- Python offers *extensive set of libraries for Machine Learning and plenty of data processing tools to handle the data.*

Who uses Python?

The incredible growth of Python is shown very clearly by *StackOverflow*:



https://www.stxnext.com

Who uses Python?

Source: logicfinder

Python IDEs & Code Editors

.....

Jupyter Notebook

- Introduction
- Jupyter Architecture
- Why uses Jupyter Notebook?

Introduction

- *Jupyter Notebook* formerly known as *IPython* (or Interactive Python), is an open source web application that is used to create and share documents that contain live code, equations, visualizations, and text.
- The name *Jupyter* is a reference to the three core languages supported by the project (**Ju**lia, **Py**thon, and **R**),
- Project Jupyter supports interactive data science and scientific computing across more than 40 programming languages.

Examples

jupyter IntroToJupyterPython (autosaved)

Logout

File Edit View Insert Cell Kernel Help | Not Trusted | Python 2

Code

There are many shortcuts available. Some of these include:

Basic navigation: up-arrow, down-arrow, enter, shift-enter, up/k, down/j Saving the notebook: s Cell types: y (code), m (markdown), 1-6 (heading level)
Add cells: a (cell above), b (cell below) Cell editing: x (delete), c(copy)

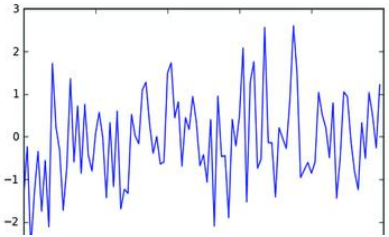
For a full list, see Help/Keyboard shortcuts on the top Menu.

Now let's try some examples! **To run: double-click in the code cell below, then press Shift+Enter:**

In [1]:
```
# This is a code cell. Click on it and press Shift+Enter. While it's executing, the prompt will turn into In [*]:

%pylab inline
print "Python is easy to learn!"

plot(randn(100))
```

```
Populating the interactive namespace from numpy and matplotlib
Python is easy to learn!
```

Out[1]: [<matplotlib.lines.Line2D at 0x49a0d68>]



jupyter spectrogram (autosaved)

File Edit View Insert Cell Kernel Help | Python 3

Markdown | CellToolbar

Simple spectral analysis

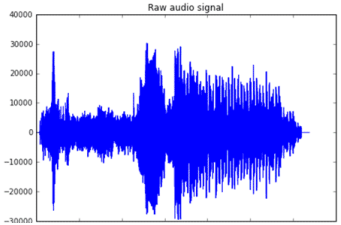
An illustration of the Discrete Fourier Transform

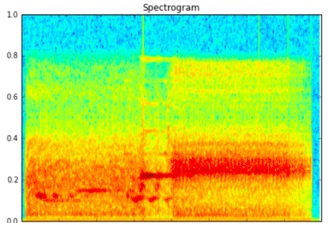
$$X_k = \sum_{n=0}^{N-1} x_n \exp^{\frac{-2\pi i}{N} kn} \quad k = 0, \ldots, N-1$$

In [2]:
```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

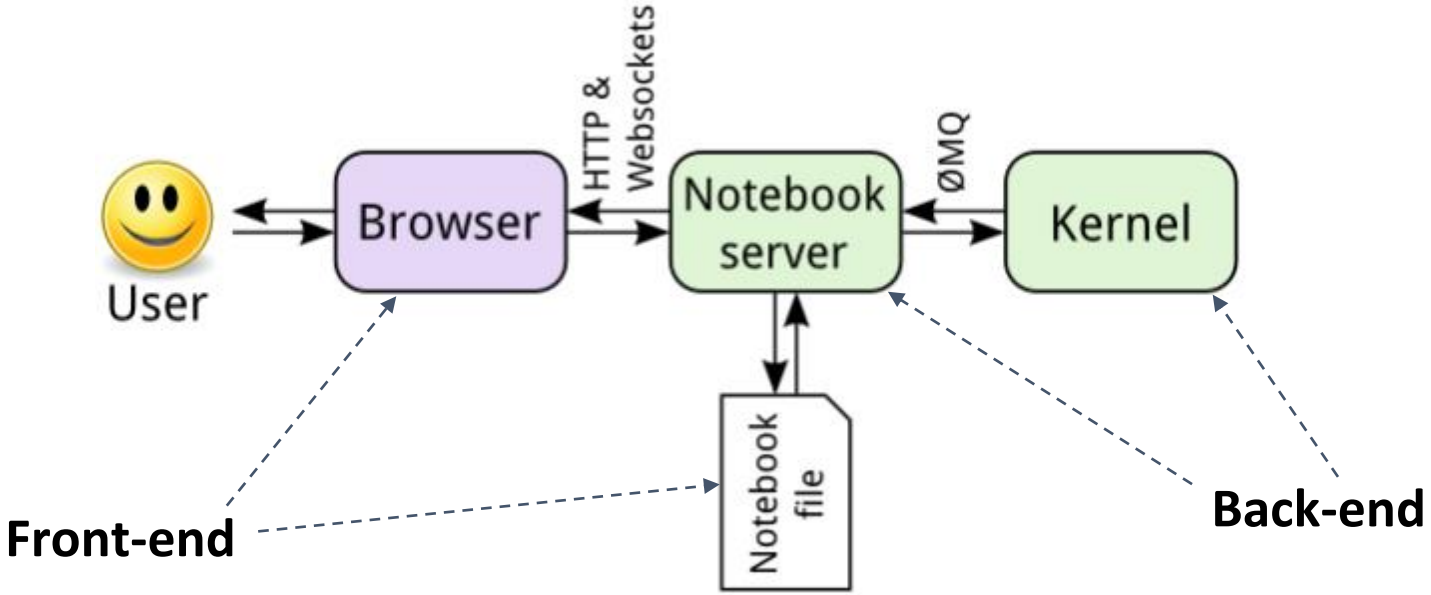
And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

In [5]:
```
fig, (ax1, ax2) = plt.subplots(1,2,figsize(16,5))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```





Jupyter Architecture



Jupyter Architecture

➔ **Front-end:** The user will work with the:

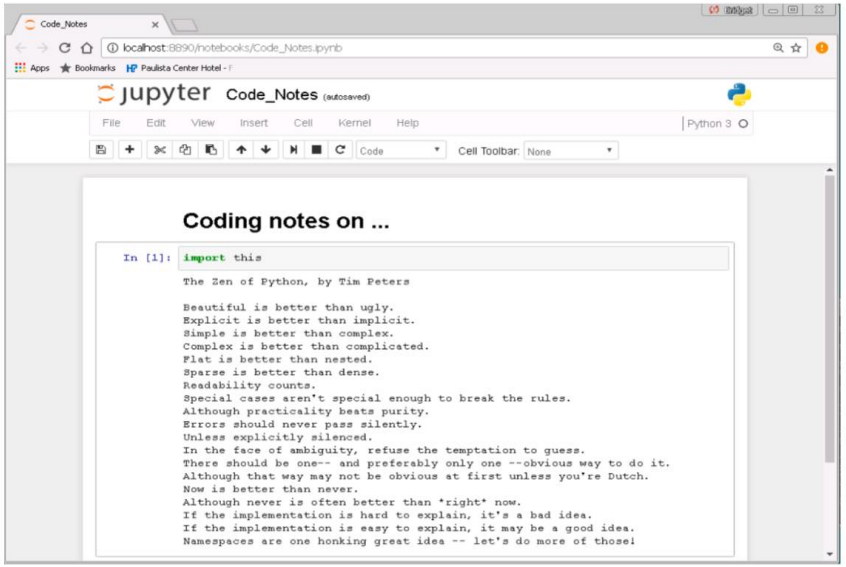
- **Web Application:** Browser-based tool for interactive development of notebook documents
- **Notebook Document:** A representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

Jupyter Architecture

- **Back-end:** The user doesn't directly interact with, but should at least be aware of
  - **Kernel:** A separate process responsible for running user code. For the purposes of the course, we will be working on Python kernels, although Jupyter is capable of interfacing with other programming languages as well.
  - **Notebook Server:** Communicates with kernel and routes the Python programming language to the web browser.

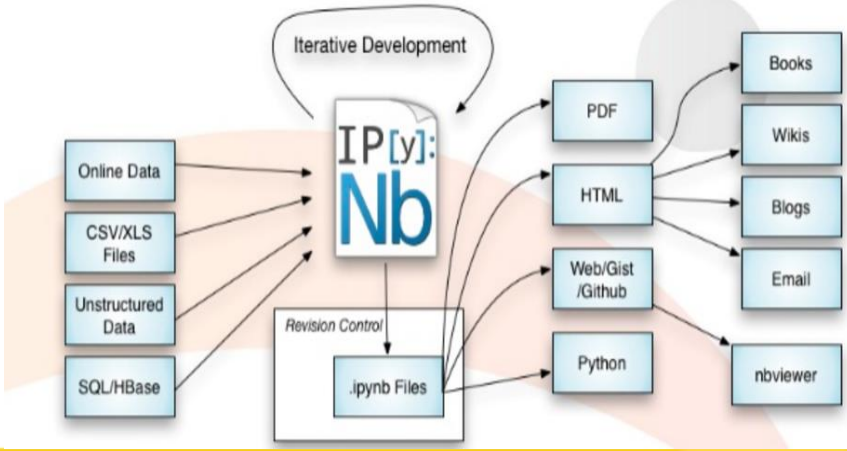
Why uses Jupyter Notebook?

- **Documentation and literate programming** by combining rich-text narrative *concepts* & machine-readable *code as* Coding Diary. The notebook itself is a data-structure with metadata that can be easily read and parsed.



Why uses Jupyter Notebook

- **Exploration & development**: Intermediate steps are saved in a clean, well documented format
- **Communication/Collaboration**: sharing research with peers, collaborators, reviewers, public
- **Publishing**: It is simple and quick switch between the development & publishing stage



Google Colab

- just a **cloud-based** version of the **Jupyter Notebook**
- is a product from Google Research allowing anybody to write and execute arbitrary python code through the browser
- a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs.

https://colab.research.google.com/