Introduction to Python and IDLE

Objectives:

- What is Python
- IDLE Environment
- Python Basics
- Python Variables
- Data Types Overview
- String
- Basic Built-in Functions

1. What is Python

Python is an interpreted, object-oriented, high-level programming language.

Interpreted

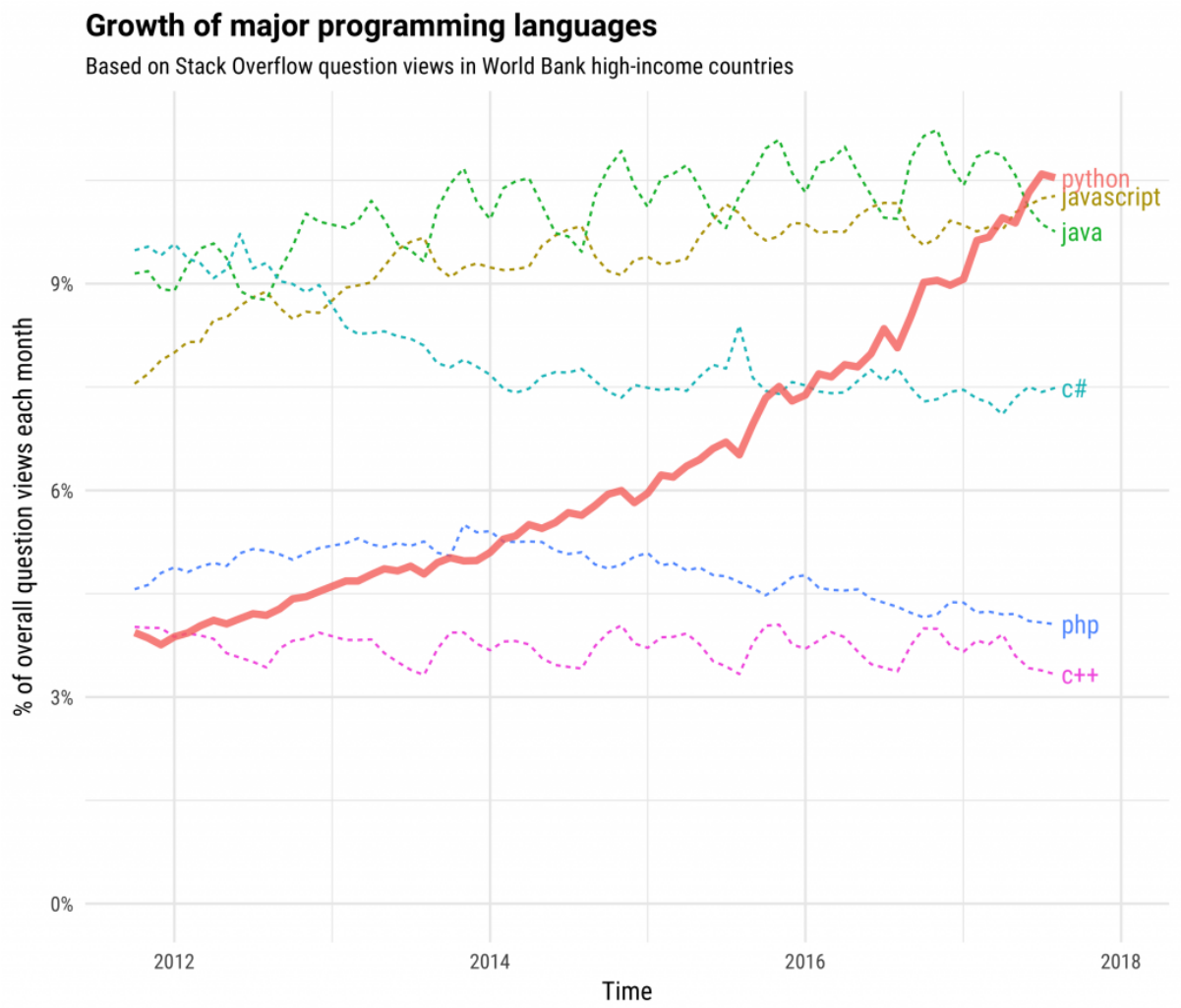
- No compilation step
- Fast edit-test-debug cycle

Object-oriented

- Supports modules and packages
- Encourages program modularity and code reuse

High-level

- Abstracts many details from the computer
- Great built-in data structures
- Good for rapid application development



Reference: [https://hackernoon.com/how-is-python-different-from-other-programming-languages-63311390f8dd (https://hackernoon.com/how-is-python-different-from-other-programming-languages-63311390f8dd)](https://hackernoon.com/how-is-python-different-from-other-programming-languages-63311390f8dd)

1.1 Advantages

Simple and Easy to Learn

- Simple syntax closely resembling the English language
- Easy to read or easy to learn

Free and Open Source

- Free to use and distribute
- Development driven by the community collaboration

Many 3rd-Party Frameworks and Tools

- Module can be easily found to support fast software development

Portable and Extensible

- Supported by all major OS platforms; Develop once, run anywhere
- Interfaces with libraries written in other programming languages, e.g. Java, C/C++
- Python was first implemented using C

1.2 Disadvantages

- Pyhton is not as fast, especially compared to compiled languages
- Python does not scale well with multiprocessor or multicore systems

1.3 Applications

Web and Internet Development

- Web development frameworks, e.g. Django, Flask...
- Internet protocols, e.g. socket, HTML, JSON, email...

Scientific and Numeric Computing

- Data science and machine learning, e.g. NumPy, Pandas, Scipy...
- Data analysis and visualization, e.g. Plotly, Matplotlib, Seaborn...

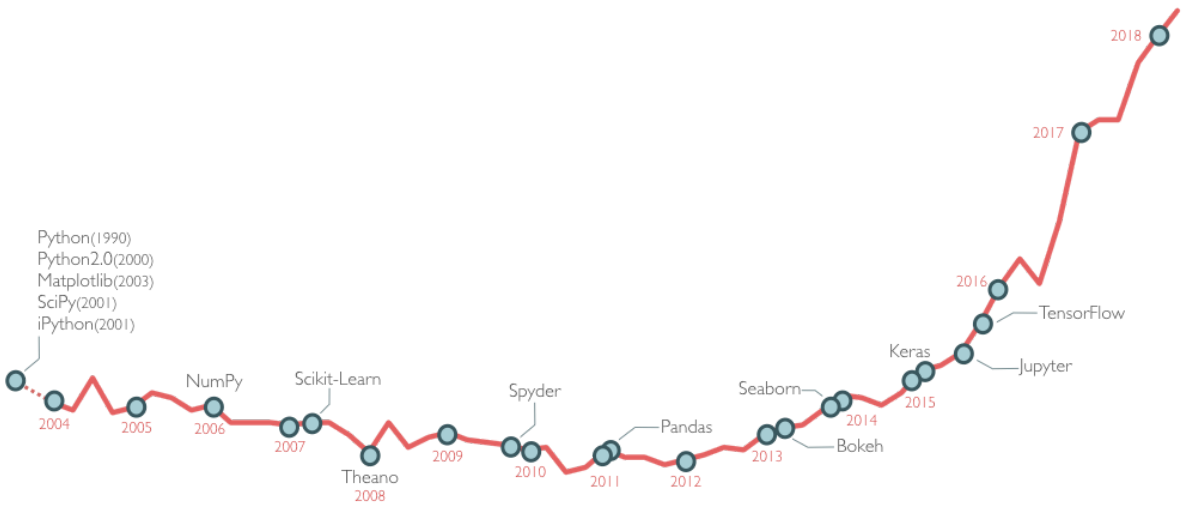
Scripting

- Small programs to automate simple tasks

Desktop GUI

- GUI libraries, e.g. Tk, Kivy, Qt...

Reference: [https://www.python.org/about/apps/ (https://www.python.org/about/apps/)](https://www.python.org/about/apps/)



Reference: https://medium.com/@atillaguzel/popularity-of-data-science-python-and-pythons-major-libraries-f7146e202e5d (https://medium.com/@atillaguzel/popularity-of-data-science-python-and-pythons-major-libraries-f7146e202e5d)

1.4 Python Versions

- Python 1.0.0 was released on 26 Jan 1994
- Python 2.0 was released on 16 Oct 2000
- Python 2.7 was released on 3 Jul 2010
- Python 3.0 was released on 3 Dec 2008
- Python 3.5 was released on 13 Sep 2015
- Python 3.6 was released on 23 Dec 2016
- Python 3.7 was released on 27 Jun 2018

There are 2 major versions in used, version 2.x and 3.x.

- Version 3.x is not backward compatible with version 2.x
  - Legacy libraries/code must be re-written
- Version 2 will be End Of Life (EOL) in January 2020
  - no further updates nor bugfixes

You can check out the differences between version 2 and 3 in following site.

Reference: https://www.guru99.com/python-2-vs-python-3.html (https://www.guru99.com/python-2-vs-python-3.html)

2. IDLE

IDLE is Python's Integrated Development and Learning Environment. It is installed with Python installation.

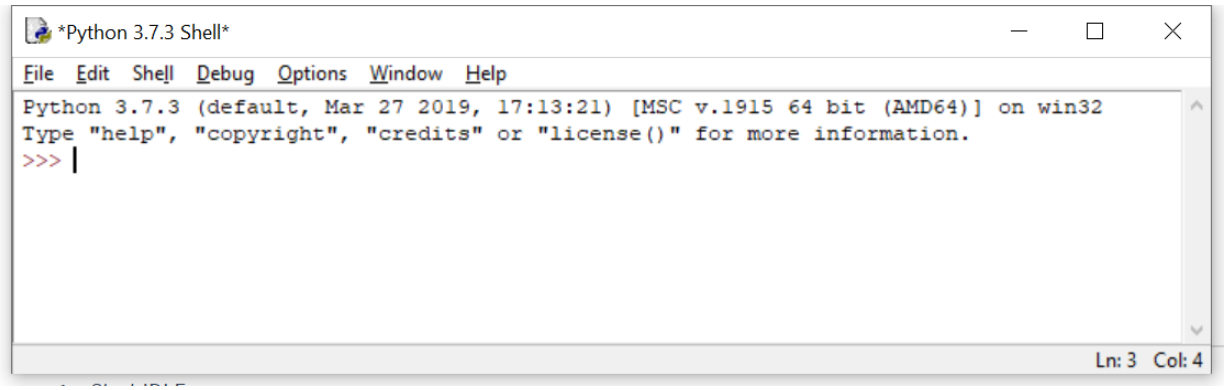
What does it provide?

- Cross-platform, available on Windows, Unix and MacOS.
- Interactive Python interpreter (shell window)
- Multi-window text editor with syntax highlighting, code completion
- Debugger with breakpoints, stepping, viewing of variables

Start IDLE

To start IDLE, press `WIN+S` , type `IDLE` to find it, and press `ENTER` key.

- The '>>>' indicates that Python is ready to read in a new command.



2.1 Read-Eval-Print-Loop (REPL)

The default window of IDLE is an interactive Read-Eval-Print-Loop (REPL) environment, where user can type command directly. The interpreter will

- Reads the command entered by user
- Evaluate and execute the command
- Print the output (if any) to the console
- Loop back and repeat the process

a. Hello World

Exercise:

Type following command and press ENTER key. This command prints out `Hello World` in output.

- Python will give us its output immediately

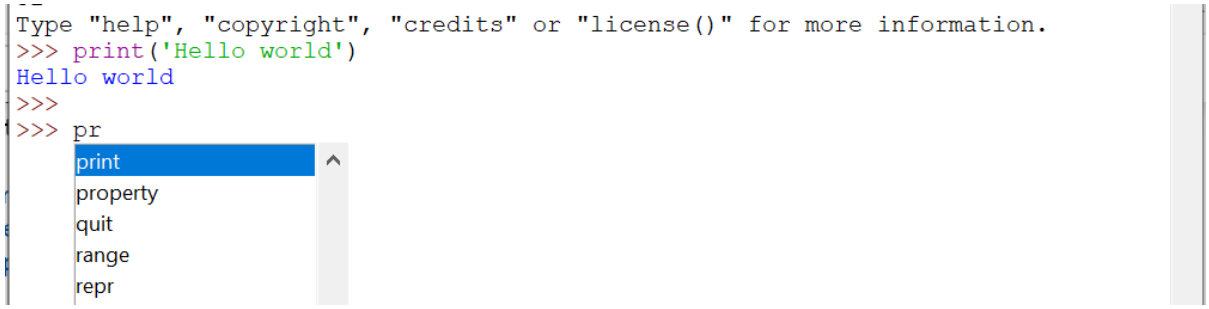
```
print('Hello World!')
```

b. Autocomplete

REPL has autocompletion feature. Use `TAB` key to activate autocompletion.

Exercise:

- Type `pr` in the REPL, and press `TAB` key.



- Type following command and press `ENTER`

```
this_is_a_variable = 99
```

- Type `th` and press on `TAB` key to see autocompletion in action.

c. Get Help

To get help on a particular command, use `help()` function.

Exercise:

- Type `help(print)` to find out information on `print()` function.

```
>>>
>>> help(print)
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.

>>>
```

d. Copy a Line

Exercise:

- Use `UP` arrow key to move up to previous command `this_is_a_variable = 9` , press `ENTER` key.
- The command at the cursor will be copied.
- Edit the command to be `this_is_a_variable = 99` and press `ENTER` key.

Exercise:

- Type following command, press `Enter` key twice.

```
for x in range(5):
    print(x)
```

- Use `UP` arrow key to move up to `for x in range(5)` , press `ENTER` key.
- The command is copied. modified `print(x)` to `print(x**2)` , press `ENTER` key twice.

```
>>> for x in range(3):
        print(x)


0
1
2
>>> for x in range(3):
        print(x**2)


0
1
4
>>>
```

e. Command History

You can iterate through history commands on REPL. Following are shortcuts to move to previous/next history command.

- Press `ALT + P` keys to get last history command.
- Press `ALT + N` keys to see next history command.
- To exit from history, press `CTRL + C` keys.

Exercise:

- Move cursor up to `help(print)` line and run it again.

2.2 Python Modules

- REPL is good for quick evaluation of single statement.
- But for multilines of code, it is common to put them in a Python file, which is called `module` .

Youtube

- https://www.youtube.com/watch?v=5hwG2gEGzVg (https://www.youtube.com/watch?v=5hwG2gEGzVg)

In [1]:
```
from IPython.display import IFrame
IFrame('https://www.youtube.com/embed/5hwG2gEGzVg', width=400, height=300)
```

Out[1]:

Create Code File

Exercise:

- On IDLE, go to menu `File > New Window` (or press `Ctrl + N`) to start a new code window
- Enter following code in the code window

```
print("hello " + name)
```

- Press `Ctrl + S` to save the code as a Python file `ex01-2`
- Go to menu `Run > Run Module` (or press `F5`) to run the module
  - The code runs and output at REPL window
  - There is an error in the code at line 3

```
>>>
================ RESTART: C:/Users/zqi2/Desktop/ex01-02.py ================
Traceback (most recent call last):
  File "C:/Users/zqi2/Desktop/ex01-02.py", line 1, in <module>
    print("hello " + name)
NameError: name 'name' is not defined
>>>
```

- Replace the code as following:

```
name = 'world'
print("hello " + name)
```

- Save the code by key `Ctrl + S`
- Run the code again by key `F5`

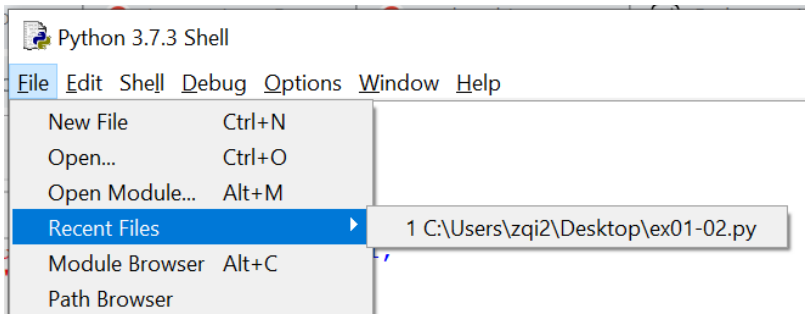
```
>>>
================ RESTART: C:/Users/zqi2/Desktop/ex01-02.py ================
hello world
>>>
```

- Close Code Window.

Open an Existing File

Exercise:

- You can open file `ex01-02.py` again by either option:
  - From menu `File > Recent Files`
  - From menu `File > Open File...`



3. Recap

Discuss and answer following questions:

- Which version of Python you should learn if you just start learning Python?
- What does IDLE stand for?
- How to run a Python module in IDLE?
- State one advantage of Python programming language.
- State one disadvantage of Python programming language.