Introduction

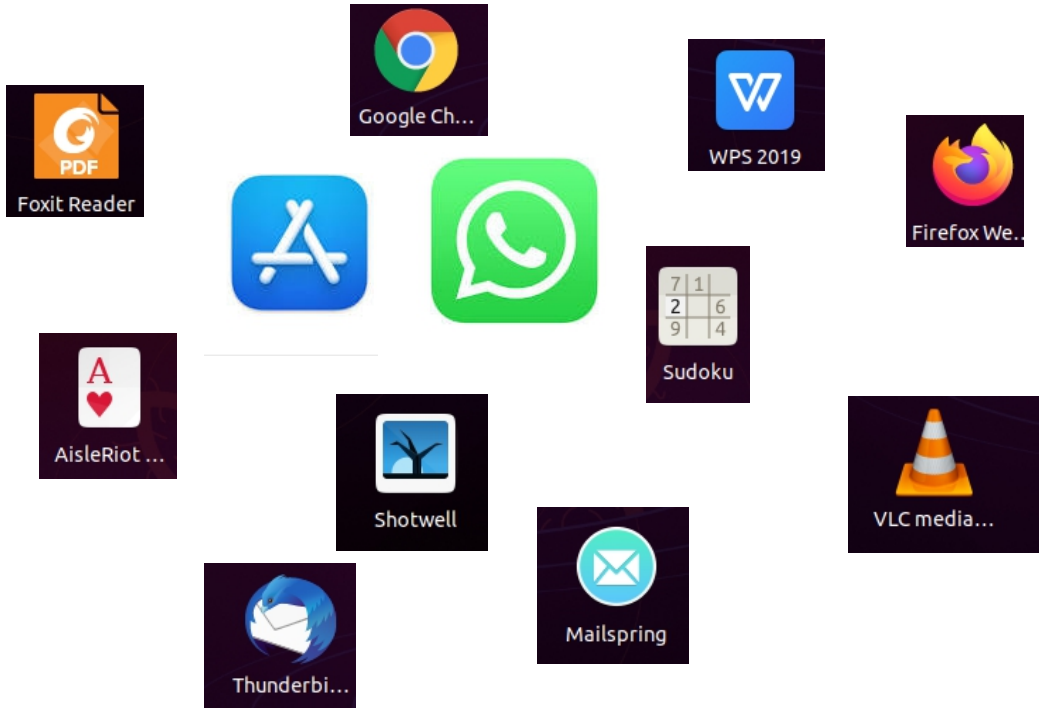
Course Content

- Introduction to Python
  - What is programming?
  - Why Python?
  - What is python interpreter?
  - Writing our first program in Python
  - Comments in python

- Python Basics
- Datatype in python
- Flow Control
- Functions
- Files
- Object Oriented Programming
- Introduction To Data Science

What is programming?

Programming is how *you* get computers to solve problems.



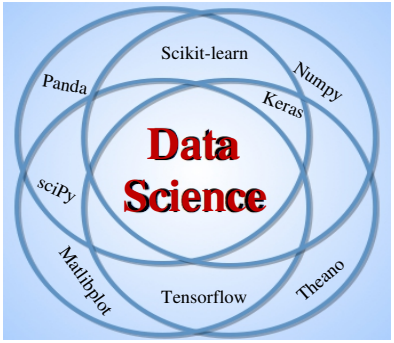
Why using python?


Easy to read/write
Elegant Syntax
Free & Opensource
Platform Independent
Active Community
Rich Libraries (pip/conda)


python

Tkinter

GUI


Scikit-learn
Panda
Numpy
Keras
Data
Science
sciPy
Matlibplot
Tensorflow
Theano


MySQL
Developement Services


mongoDB


TURBO
GEARS


django


Flask
web development,
one drop at a time


CherryPy

Python 2.x vs Python 3.x

Python 3.x

https://www.python.org/

python™

Donate | Search | GO | Socialize


```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
987
```

>_

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. »» Learn More

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

Start with our Beginner's Guide

Download

Python source code and installers are available for download for all versions!

Latest: Python 3.10.7

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

docs.python.org

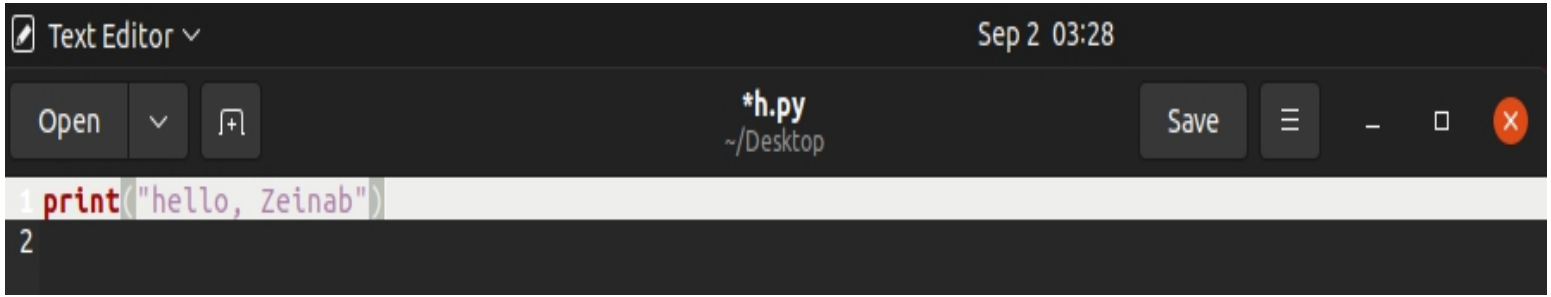
Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.

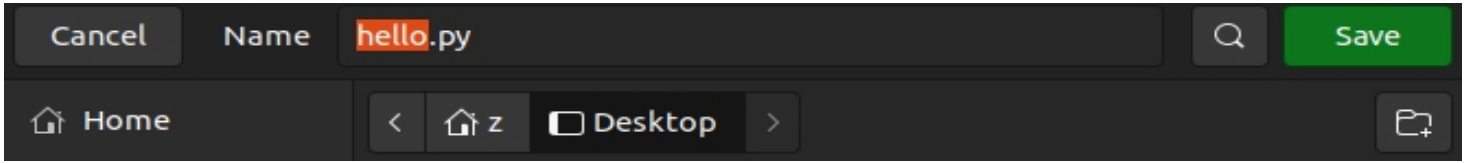
jobs.python.org

Writing your first Python Program

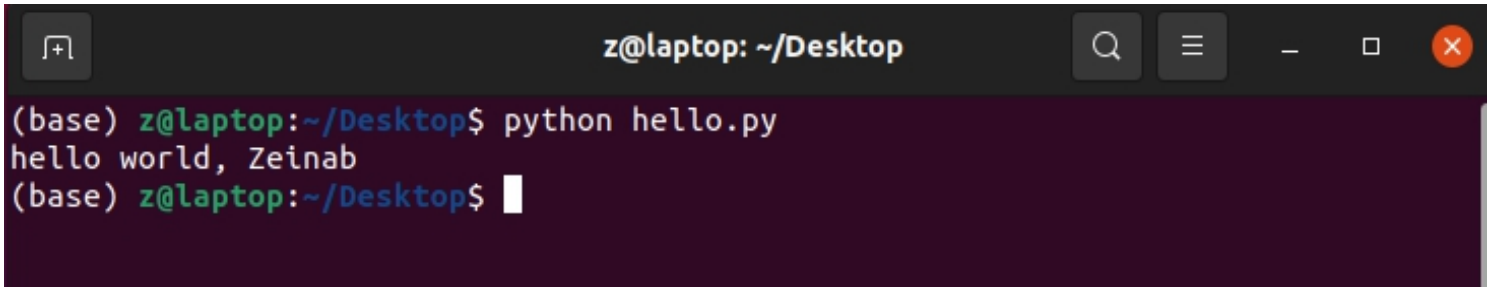
Step1: Write the program



Step2: save the file as FileName.py



Step3: Execute the program

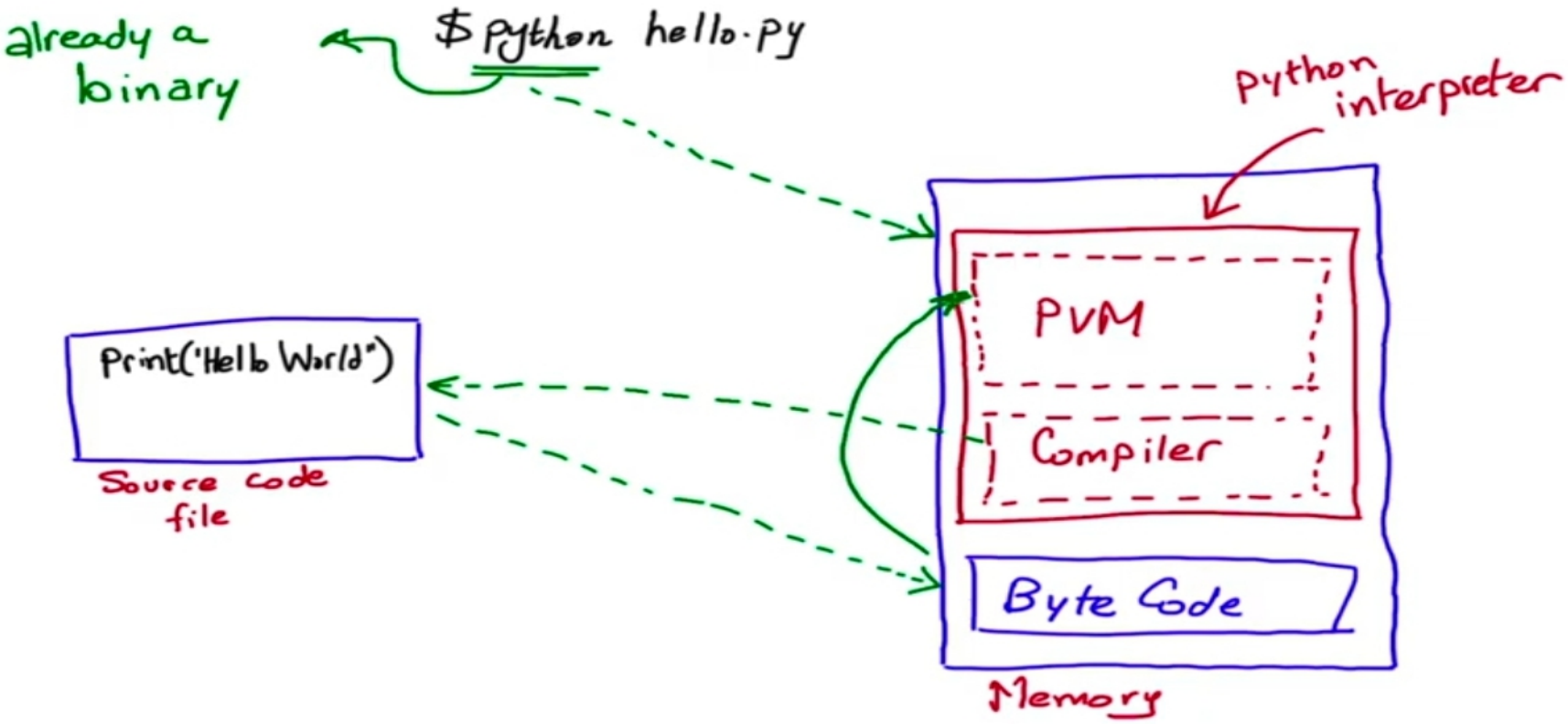


Python in interactive mode

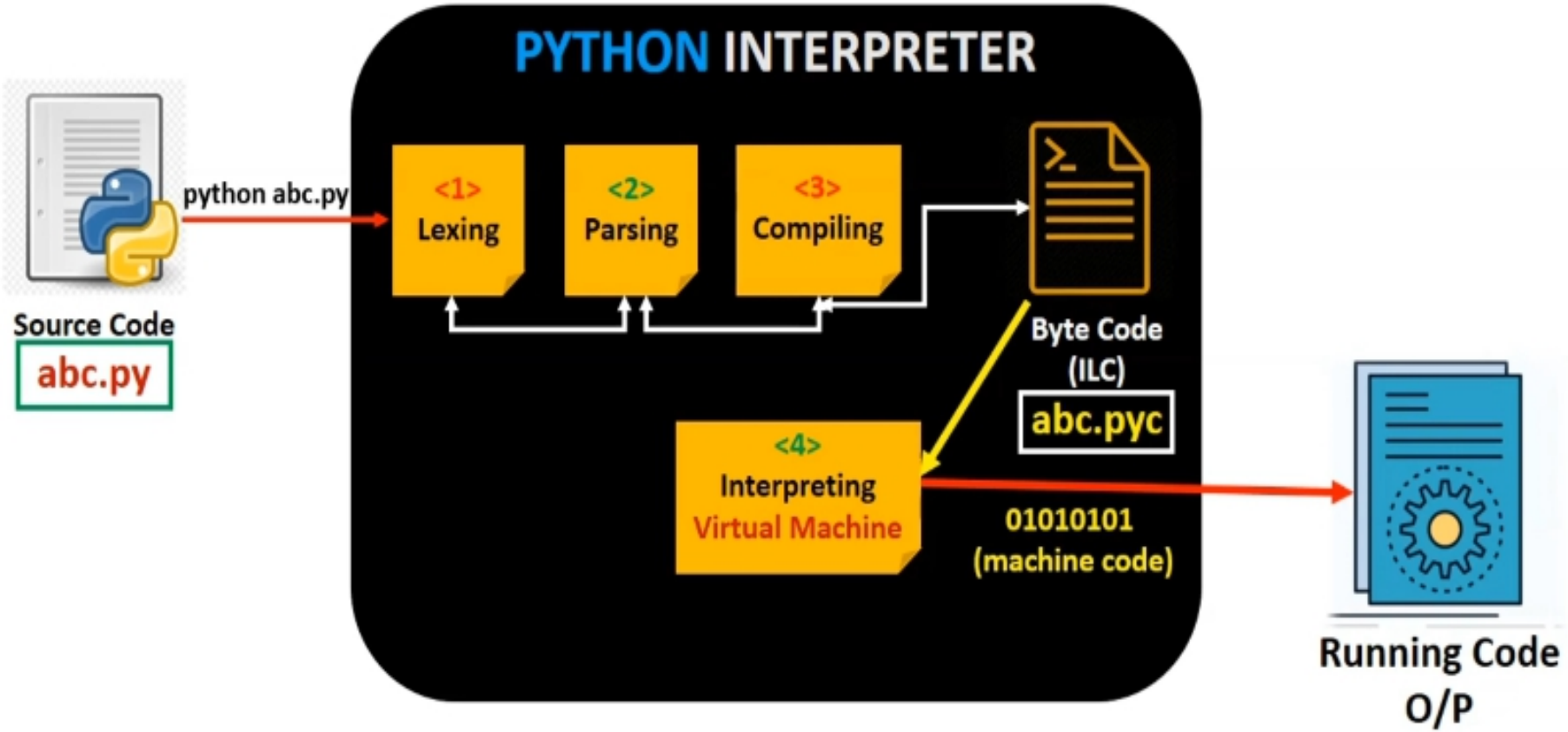
```
(base) z@laptop:~$ python
Python 3.7.6 (default, Jan  8 2020, 19:59:22)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+3
5
>>> 45+44
89
>>> print("hello")
hello
>>>
```

Python Interpreter

How does Python Interpreter Work?



Simulating Python Interpreter



Printing *.pyc* file

```
(base) z@laptop:~$ gedit hello.py
(base) z@laptop:~$ python hello.py
hello world, Zeinab
(base) z@laptop:~$ python -m py_compile hello.py
(base) z@laptop:~$ ls __pycache__
aaa.cpython-37.pyc  hello.cpython-37.pyc
(base) z@laptop:~$ cat __pycache__/hello.cpython-37.pyc
B
◆◆Ac◆@s
       ed◆dS)zhello world, ZeinabN)◆print◆rrhello.py<module>◆(base)
(base) z@laptop:~$ python -m dis hello.py
  1           0 LOAD_NAME                0 (print)
              2 LOAD_CONST               0 ('hello world, Zeinab')
              4 CALL_FUNCTION            1
              6 POP_TOP
              8 LOAD_CONST               1 (None)
             10 RETURN_VALUE
(base) z@laptop:~$
```

hello.pyc


Open
*hello.cpython-37.pyc
~/__pycache__
Save
1 B
2
3 \00\00\00\00\F5\98Ac\00\00\00\E3\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00@\00\00\00s
\00\00\00e\00d\00\83\00\00\00d\00S\00)\00z\00hello world, ZeinabN)\00
\DA\00print\A9\00r\00\00\00r\00\00\00\00\FA hello.py\DA <module>\00\00\00\00\F3\00\00\00\00
4

Types of python interpreter

Integrated Development Environments: IDE

- To write a python program, you will need:

  An editor: to write your code

  An interpreter: to execute your code

  A debugger: to debug your code

- These are the basic components for any IDE (Integrated Development Environment).

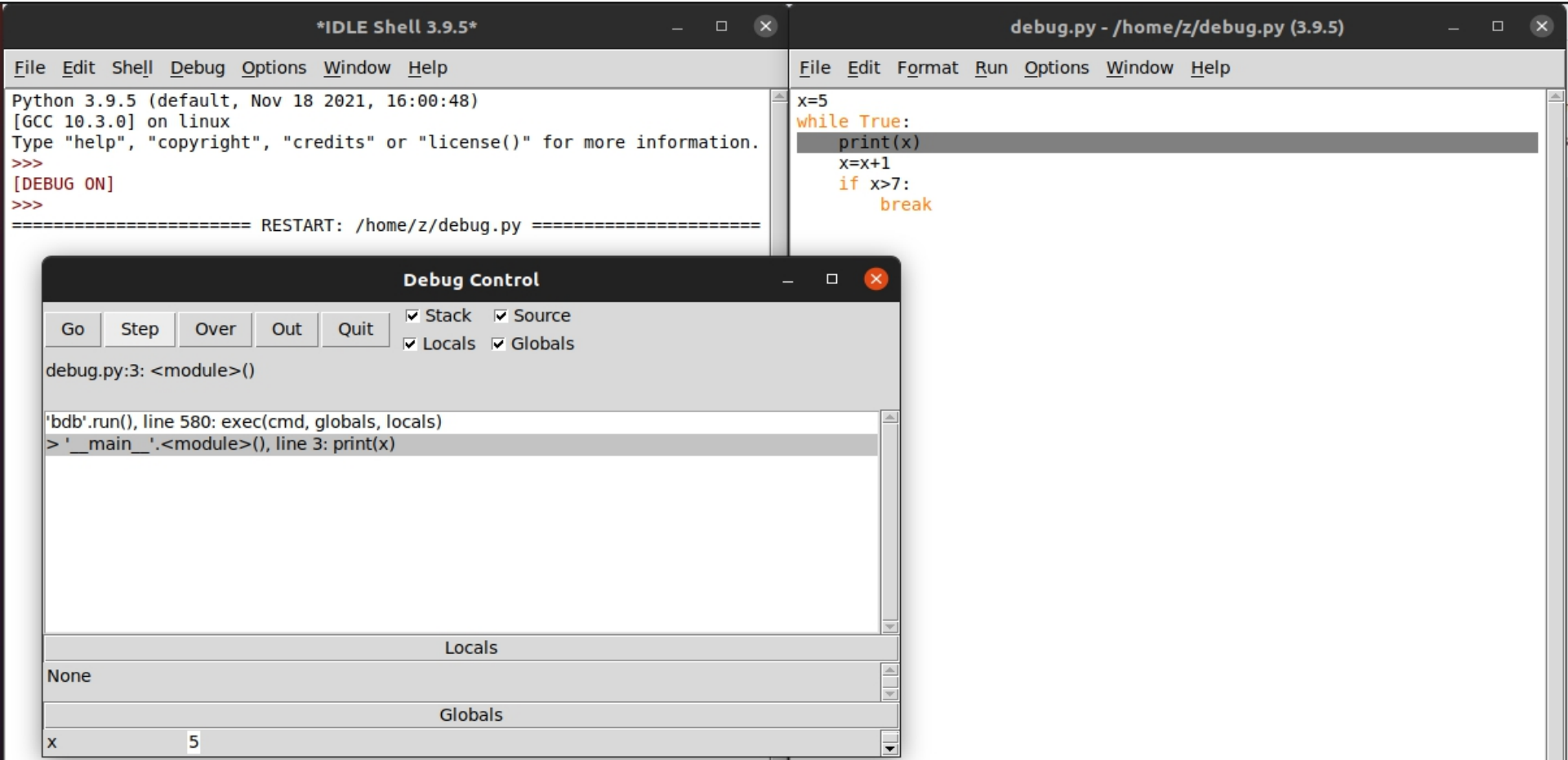
- IDLE (Integerated Development and Learning Environment) is the default python IDE.

- PyCharm and Spyder are the most popular python IDEs

- With the advent of web applications, a new generation of IDEs for interactive languages such as Python has been developed.

- Jupyter Notebook is a web based interactive computing notebook environment.

IDLE


IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (default, Nov 18 2021, 16:00:48)
[GCC 10.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
[DEBUG ON]
>>>
========== RESTART: /home/z/debug.py ==========
debug.py - /home/z/debug.py (3.9.5)
File Edit Format Run Options Window Help
x=5
while True:
print(x)
x=x+1
if x>7:
break
Debug Control
Go
Step
Over
Out
Quit
Stack
Source
Locals
Globals
debug.py:3: <module>()
'bdb'.run(), line 580: exec(cmd, globals, locals)
> '__main__'.<module>(), line 3: print(x)
Locals
None
Globals
x
5

Installing Anaconda

To download Anaconda

https://docs.anaconda.com/anaconda/install/

To setup on windows

https://www.youtube.com/watch?v=9V2IvJWZJ3I
https://docs.anaconda.com/anaconda/install/windows/

To setup on linux

https://linuxize.com/post/how-to-install-anaconda-on-ubuntu-20-04/?fbclid=IwAR1DViMtho63EbOYqQnetGdDv6YzvlqjSstBp2dgnk0vqC7RE7xMvizrPds

Writing Program in Jupyter lab

Python Comments

Python Comments: One line comment

- **Comment starts** with a hash character #

- Comments can be used to **explain code** and make it more readable.

```
In [3]: #this is my first program
        print("hello, Zeinab")

        hello, Zeinab
```

- Comments can be used to **prevent execution when testing code**.

```
In [5]: #print("hello, Zeinab")
        print("This is the first lecture in Python course")
        print("Its main topic is introduction about python")

        This is the first lecture in Python course
        Its main topic is introduction about python
```

Python comments: Multi-line comment

Using either:

- **Three double quotes """**

```
In [16]: """ This is a multi-line comment
         it is used to write comment in more than one line
         """

         print("finish")
```

```
finish
```

- **Three single quotes '''**

```
In [17]: ''' This is a multi-line comment
         it is used to write comment in more than one line
         '''

         print("finish")
```

```
finish
```

Python Comments Convention

Convention in using Python comments

```
In [1]: # Block Comments

In [2]: # Inline Comments

In [ ]: # Documentation Comments
```

Convention in using *Block* Comments

In [5]:
```
# Block Comments
for x in range(0,10):
    # this is going to loop for 10 times
    # used for looping the given range of values
    # also to print some value
    print(x)

class Employee:
    # this is a class block comments
    # line 2 comments
    # line 3 comments
    pass
```

```
0
1
2
3
4
5
6
7
8
9
```

Convention in using *Inline* comments

In [2]:
```
# Inline Comments

age = 10  # Kid's Age
name = 'Sabari Balaji'  # Trainer Name
```

Convention in using *Documentation* comments

In [8]:
```
# Documentation Comments

def calculate_mean_median():
    '''
    here the documentation comments are written
    lines
    lines
    lines
    lines
    '''
    print("Hi")

print(calculate_mean_median.__doc__)
```

```
    here the documentation comments are written
    lines
    lines
    lines
    lines
```

In [ ]:
```
def calculate_mean_median():
    """
    this is also a documentaion comments
    """
```

Using Python As a Calculator

Using python as a calculator

```
2+5
```

```
7
```

```
2*5
```

```
10
```

```
2-5
```

```
-3
```

```
2/5    # division, the output is float
```

```
0.4
```

```
2//5    # the // opertator is floor division, i.e., discard the fraction part
```

```
0
```

```
2**5    # power
```

```
32
```

```
2%5  # the modelus operator % returns the reminder of the division
```

```
2
```

The _ variable

```
[1]: 2**5   # power
```

```
[1]: 32
```

```
[2]: 2%5  # the modelus operator % returns the reminder of the division
```

```
[2]: 2
```

```
[29]: print(_)
```

```
2
```

```
[30]: 0>5
```

```
[30]: False
```

```
[31]: print(_)
```

```
False
```

```
[32]: ''' There is a full support for floating point;
operators for mixed type operands convert the integer operand to floatig point'''

4 * 3.75 - 1
```

```
[32]: 14.0
```