Python 101

Python essentials.

Contents

Philosophy of Python

Two important PEPs we need to know are PEP20 and PEP8.

PEP20 tells us about “The Zen of Python”

https://www.python.org/dev/peps/pep-0020/

PEP8 defines the “Style Guide” or “How to format the code in python”

https://www.python.org/dev/peps/pep-0008/

The Zen of Python

It is PEP 20’s rules.

It only defines what should be the mindset when coding in python.

To summarize the PEP 20 we can say -

If it is hard to explain, it is not a good idea.

If it's easy to explain, it might be a good idea.

Mind that when we write code in Python readability and maintainability is top priority.

Python Interpreter

It simply interprets code line by line as you feed it into it.

Usually the interpreter exists at **/usr/local/bin/**

But it might exist at **/usr/bin/** as well. In order to see if your python interpreter file exists on this path run following commands-

```
> cd /usr/bin
```

```
> ls | grep “python”
```

We will be learning ***most of the*** basics on the interpreter.

Try out the following code on the interpreter

```
>>> import this
```

Running Python Scripts

Create a new file ending with **.py** at home (~) in the system and write the same code we wrote in interpreter in last page. I.e. “import this”

Open your terminal at the same path i.e. at home (~) or wherever you have saved the file.

On the terminal execute the following command-

```
> python file_name.py
```

A word about the extensions

We saved the file using extension **.py**

- Try remove that extension and then running the file again.
- Try adding some extension like **.java** then running the file again.

What does it mean? Do extensions of a file not serve any purpose?

Extensions do serve a purpose. They tell the operating system which program the file is associated with. But mind that here we are explicitly telling the Operating System which program to run the file with by typing ***python file_name*** *so it doesn't really matter what extension the file has.* ***But in normal cases we use .py as the python file's extension.***

Compiling Python Code

We already know python is bytecode interpreted high level language.

And we also know that in order to convert python code to bytecode we need to compile python's code.

We don't need to explicitly compile the python's code to bytecode and then interpret it, but irrespective of that we can do that by following command-

```
> python -m py_compile file_name.py
```

There is a new folder created by the name **__pycache__** which contains the compiled file.

Shebang

Shebang = she-bang = hash + bang (abbr. for exclamation mark)

Ie. Shebang = #!

Its used in ***NIX** systems as the first line in scripts to tell the Operating System that if the script is executed directly then irrespective of the extension run it with the program given in the specified path.

For python it is **#!/usr/bin/env python3**

What is a Class and an Object? Just a definition. Not starting of OOPS

Variable - Something that holds some value.

Methods - A bunch of code that does something with those variables.

Object - Something that contains a bunch of variables and methods.

Class - A template to create an object that defines what variables and methods will be in an Object.

Everything is an object in python

Everything is an Object so what?

Because everything is an object so everything in python has some variables and methods associated with it.

id of an object in python

id of an object is a unique identifier assigned by Python to that object.

We can see the id of that object using following commands -

```
> import this

> id(this)
```

Asking Python for Help

Start the Python interpreter on Terminal.

See first line -

```
Type "help", "copyright", "credits" or "license" for more information.
```

help(some_object) prints help related to that object. And as everything is object in python so help function can show help for virtually anything in python.

Seeing inside a Python Object

Again repeating **everything is object in python** and every object comprises of variables and methods.

In order to see what are the variables and methods in an object in python (ie. Anything in Python) we can use **dir** in function.

```
> dir(some_object)
```

Interview Questions

Tell me something about PEPs?

PEPs are Python Enhancement Proposals. Most important ones are PEP 20 and PEP 8.

PEP 20 contains the ideology or Philosophy of Python also called “The Zen of Python”.

PEP 8 contains guidelines for styling python code ie. how to write python code.

Where is Python Interpreter in *NIX systems?

/usr/local/bin

or

/usr/bin

What is the extension of a Python file?

.py is the extension of the files that we write code in. ie. the script files.

and

.pyc is the extension of the files that python automatically compiles for interpreting the bytecode.

What do you understand by the shebang?

Shebang is the first comment line used in most of the scripting languages to show that irrespective of the extension if the file is run directly then it should be run with the program given after the shebang. (#!)

It is valid only for *NIX systems.

Can you compile Python code?

Yes.

We can compile the code to Bytecode using py_compile module in python.

And can we compile it directly to machine code (0/1) ? **Technically Speaking Yes, Speaking based on Feasibility and Practicality No.** We can do it using Cython(based on Pyrex) or Psyco or Pyrex. But Cython remains as the only stable tool. We should not be confused between Cython and CPython.

Cython is a whole new language, it is a superset of python with capabilities of C. It’s not providing a layer of abstraction over C code instead it is providing a way to write C and Python side by side.

Distinguish Class and Object.

Class is a blueprint for a building.

Object is the building itself.

What is id in Python? Is it address of an object in memory?

No.

But it is computed based on the address of an object in memory.

It is only advisable to be used as a unique identifier for python objects.

What do help and dir do?

Help shows content that can be used as a help text for a python object and dir shows the attributes of a python object.