Python3 vs Python2

2008 — Creator of Python is Guido van Rossum. He was unhappy with the architectural decisions made in Python. He wanted to change them and overhaul Python and released Python3. The goal was to be the present and future of the language.

Python3 is not backwards compatible. For most other languages, if a JavaScript script was written 5 years ago, it would typically be able to run fine even with all the new updates to JavaScript that had occurred over the years. The opposite is true for Python3. All the companies that had their scripts written in Python2 would not work with Python3.

This link describes the differences between Python 2.x and Python 3.x even further:
https://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html

This link includes information on the built-in functions within Python3.x:
https://docs.python.org/3/library/functions.html

So why learn Python3?

- It's easiest to learn Python3.x and learn the difference/quirks of Python2.x
- It's been almost 10 years. Many of the main criticisms are no longer valid.
- All the popular packages support 3.x now. It's no longer a sacrifice to use 3.x

Mac/Linux Command Line Fundamentals

Objectives:

- Define what the command line is
- Learn command line navigation and file structure
- Learn to manipulate files and folders via command line

What the command line does:

The command line is a faster and more powerful way to maneuver your operating system than by using a GUI, such as Windows Explorer or Mac Finder.

We use special keywords to do everything that we normally can with a GUI and more.

Operating systems organize their folders in a hierarchy (like a tree), with parents and children all relative to a base root directory.

Most commonly used commands in the Command Line:

- cd — Change directory, or in simpler terms, change which folder the user is in
- ls — this command will list everything that exists in the current working directory

- mkdir — Make directory, or make a new folder
- touch — this will allow the user to create a new file or update the last time it was modified
- mv — Move or rename, this command will allow the user to move a file from one directory to another, rename the file, or do both
- rm — Remove, this command will remove a file or empty directory
- pwd — Print Working Directory, this command will return the exact file path for the file or folder the user is currently working in

Windows Command Line Fundamentals

OSX and Linux operating systems are both based on Unix and share many of the same terminal commands.

Windows has an entirely different kernel (base) and has traditionally had its own command-line syntax for MS-DOS and command prompt.

However, Powershell has adopted aliases that mimic the most common Unix commands.

Powershell ships with all modern Windows systems (since Windows 7) and is a superior shell to command prompt. It compares favorably with Unix-based shells, such as bash.

Most of the commands listed for Mac OSX is similar to Windows. The main difference would be the **touch** command. Within Windows, the following command will work similarly to the touch command in Powershell:

```
function touch {New-Item -ItemType File -Name ($args[0])}
```

Normally, the command to create a new file within Powershell is:

```
New-Item -ItemType filename.py
```

Numbers, Operators, and Comments

Objectives:

- Understand the differences between ints and floats
- Work with simple mathematical operators
- Adding comments to code

Int vs Float

Integer and Float are both two different types of numerical data. An *integer* is a whole number with no decimal. A *float* or floating-point number is a number that has a decimal place and is used when more precision is needed. Integers, however, take up less memory than floating-point numbers.

Basic Math

Python follows the PEMDAS rule when it comes to mathematics.
These are the most commonly used math operators:

+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Exponentiation
%	Modulo
//	Integer Division

For the more weirder mathematical operators within Python, here are some examples of how they are used and the result that will be returned:

Exponentiation:

2 ** 3 = 8 i.e. 2 to the power of 3 equals 8

Modulo:

10 % 3 = 1 i.e. 10 mod 4 equals 1

Modulo is an operator that essentially will divide the numbers and return the remainder.

Integer Division:

10//3 = 2 i.e. 10 divided by 3 equals 3

Integer Division is an operator that will divide the numbers and ultimately only return the integer and not the remainder.

Comments

Comments within the Python script are a way for the user to remind themselves of certain things within the program. Although Python code should be easily readable—not just to the user, but to anyone else who interacts with the script—that is not always the case. They can be indicated with the # or triple quotes for multiple lines of comments.