Basic Python #1

AI Mentorship

Outline

1. Introduction to Version Control (Git)
2. Git: Create and clone repository
3. Git: Upload file to repository
4. Introduction to Python
5. Python: Create your first program
6. Python: Comments
7. Python: Variables
8. Python: Data types
9. Python: Math Operations
10. Python: Castings

Git

Git is a distributed **version-control** system for **tracking changes in source code** during software development.

It is designed for coordinating work among programmers, but **it can be used to track changes in any set of files**. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

Git Command

1. git clone [url]
   Clone (download) a repository that already exists on GitHub, including all of the files, branches, and commits

2. git add [file]
   Snapshots the file in preparation for versioning

3. git commit -m "[descriptive message]"
   Records file snapshots permanently in version history

4. git push
   Uploads all local branch commits to GitHub

Python

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Python

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

Python: Create Your First Program

1. Create file **[file name].py** in your folder
2. Write **print(“Hello, World!”)**
3. Save your file
4. Open terminal or command line
5. Make sure you are in the directory where you file saved
6. Run your program using command **python [file name].py**
7. Congratulations, you have written and executed your first Python program!

Python: Comments

Comments can be used to:

1. Explain Python code
2. Make the code more readable
3. Prevent execution when testing code

Comments starts with a #, and Python will ignore them:

```
#This is a comment
print("Hello, World!")
```

Python: Variables

Variables are containers for storing data values.
Unlike other programming languages, Python has no command for declaring a variable.

```
x = 5
y = "Hello, World!"
print(x)
print(y)
```

Python: Data Types

Text type : `str`

Number type : `int`, `float`

Boolean type : `bool`

Python: Data Types

String : is a sequence of characters, surrounded by either single or double quotation marks.

Int : is a whole number, positive or negative, without decimals, of unlimited length.

Float : is a number, positive or negative, containing one or more decimals.

Boolean : represent one of two values of `True` or `False`.

You can get the data type of any object by using the `type()`

```
x = 5
print(type(x))
```

Python: Data Types

Example	Data Type
x = “Hello, World!”	str
x = 10	int
x = 10.5	float
x = True	boolean

Python: Math Operations

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y

Python: Castings

There may be times when you want to specify a type on to a variable.

- `int()` - constructs an integer number from an integer literal, a float literal (by rounding down to the previous whole number), or a string literal (providing the string represents a whole number)
- `float()` - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)
- `str()` - constructs a string from a wide variety of data types, including strings, integer literals and float literals

```
x = int(2.8)           # y will be 2
y = float("4.2")       # w will be 4.2
z = str(3.0)           # z will be '3.0'
```