Python Course

“ A Python Course Notes, Examples, Problem Solving, Advices, and some small projects “

By

Ahmad Jawabreh

What prerequisites should be in place before commencing the course?

1- Code Editor Selection: Numerous options are available, with the most renowned being VS Code, or alternatively, you can opt for the PyCharm IDE.

2- Python Installation and Configuration: Begin by installing and setting up Python from the official website.

3- VS Code Python Extension: If you choose VS Code as your code editor, install the Python extension to enhance your development environment.

4- Prerequisite Programming Knowledge: Ensure you possess basic programming knowledge before diving into the Python course.

5- Command Line Proficiency: Familiarize yourself with command line operations, as it is a beneficial skill for Python development. Some Command Line notes are attached in this repository.

6- Please review the file named "notes.txt" in this repository before commencing the course.

1.0 Print Statement:

The print statement is commonly employed as a useful means to display information on the debugging console. However, it is crucial to emphasize that the use of print statements is restricted when working on production code.

So, how can we print some information on the debugging console?

Example: print(“ Hello World”)
By running the app you’ll be able to see the “ Hello World

2.0 Implicit Line Continuation (Optional Semicolon):

If you’ve noticed in the last example that we are not using semicolon at the end of the line as we used to do, because Python's avoidance of semicolons in its code is a characteristic of a programming language feature known as "optional semicolons" or "implicit line continuation." In Python, the end of a line generally marks the end of a statement, and the use of semicolons to terminate statements is optional. The interpreter relies on indentation to determine the structure of the code, making semicolons unnecessary for line termination in most cases. This approach enhances readability and contributes to Python's clean and concise syntax.

3.0 Comments:

In Python, the hash sign is employed to comment on a specific line. For instance: # This is a single-line comment If you are using VS Code, you can conveniently comment on a specific line using the shortcut CTRL+/.

Note: It's essential to note that Python does not have explicit syntax for multiple-line comments. While some use triple quotes """ HERE """ for this purpose, it's crucial to recognize that this construct is, in fact, a string not assigned to any variable, making it an unconventional method for commenting in Python.

3.0 Data Types:

Python Data Types	
Integer	10 , 100 , -10 , -100
Floating	14.19 , 115.59 , -10.23 , -19.28
String	"Hello", "HI", "1232"
List (Array)	[1,2,3,4], ['A', 'B ', 'C'], [1,-1,'A']
Tuple	(1,2,3,4), ('A', 'B ', 'C'), (1,-1,'A')
Dictionary	{"One: 1", "Two: 2", "Three, 3"}
Boolean	2 = = 2

figure 1: Data Types in Python

- int (Integer): Any positive or negative number without any decimal or fractional part is integer
- float (Floating): Any positive or negative number with any decimal or fractional part is integer
- str (String): String type for representing text, e.g., "hello", 'Python'
- List: Ordered, mutable sequence of elements that can hold any type of data
- Tuple: Ordered, immutable sequence that can hold any type of data
- dict (Dictionary): A collection of key-value pairs
- bool: Boolean type representing True or False

3.1 List VS Tuple:

- List: Lists are mutable, meaning you can modify their elements, add new elements, or remove existing ones after the list is created. Use lists when you have a collection of items that may need to be modified, such as adding or removing elements. Lists are suitable for sequences where the length may change during the program's execution.

- Tuple: Tuples are immutable, and once a tuple is created, you cannot change, add, or remove elements. Use tuples when you want to create an immutable and ordered collection. Tuples are useful for representing fixed collections of items, like coordinates or settings, where you don't want the values to be changed accidentally.

4.0 Variables:

Python uses the concept of dynamic typing or duck type which is used also by other programming languages such as PHP and Dart. In Python, variables are dynamically typed, which means their type is determined at runtime based on the value assigned to them. Unlike statically-typed languages where you explicitly declare the variable type, Python allows more flexibility by inferring the type based on the assigned value. This dynamic typing contributes to the language's flexibility and ease of use.

Example:

Variable assignment in Python (dynamic-typed):
myVariable = “Hello World”

VS

Variable assignment in Dart (statically-typed):
String myVariable “Hello World”

4.1 Variables Rules:

4.1.1 Variables Naming Rules:

- Variable names can consist of letters (both lowercase and uppercase), digits, and the underscore character _.
- They cannot start with a digit.
- Variable names are case-sensitive, meaning myVar and myvar would be considered different variables.
- Avoid using Python reserved words (keywords) as variable names.

4.1.2 Variables Assignment Rules:

Variables are assigned using the assignment operator =. For example: my_variable = 10.

4.1.3 Data Types:

Python is dynamically typed, so you don't need to explicitly declare the data type of a variable. The interpreter infers the type based on the assigned value.

4.1.4 Convention for Variable Names:

It's a convention in Python to use lowercase names with underscores to represent variables (snake_case). For example: my_variable.

4.1.5 Avoid Single Character Names:

Unless it has a specific purpose (like loop counters), it's generally better to use descriptive names rather than single-character names.

4.1.6 Readability Counts:

Choose variable names that are descriptive and convey the purpose of the variable. This enhances the readability of your code.

Note: Python uses the concept of Variable Reassignment, where variables are mutable, meaning their values can be changed during the execution of the program.

5.0 Escape Character Sequences

- Backspace: using \b we can make a backspace in our string, where the character before the \b will be deleted

  Example: print “Hello \b World” OUTPUT: HelloWorld.

- Escape New Line: so if you want to write a string line in multiple lines in your code editor but you want to show it on the same lien when you print it you can use the backslash sign to do that

  Example: print(“ Hello \
  World\
  Baby”)
  OUTPUT: Hello World Baby

- Escape Backslash: if you want to write a backslash sign in you’re string you just need to put one more backslash before it

  Example: print(“ Hello \\ World”)
  OUTPUT: Hello \ World

- Escape Single and Double Quote Sign: if we want to write a single or double quote sign in our string we can use also the backslash sign

  Example: print(‘ Hello \‘World\’ ’)
  OUTPUT: Hello ‘World’

  Example: print(“ Hello \”World\” “)
  OUTPUT: Hello “World”

- Carriage Return: moves the cursor to the beginning of the line

  Example: print("Hello\rWorld")
  OUTPUT: World

  Example: print("123456\rAbcd")
  OUTPUT: abcd56

6.0 String Concatenation:

String concatenation in Python refers to the process of combining two or more strings into a single string. This can be done using the + operator, which concatenates or joins strings.

Example:

```
str1 = "Hello"
str2 = "World"
mistake = str1 + str2
result = str1 + " " + str2
print(result)
```

OUTPUT:

```
HelloWorld
Hello World
```

In this example, the strings "Hello" and "World" are concatenated with a space in between, resulting in the output.

7.0 Strings in Python