



PYTHON FOUNDATIONS: FROM BASICS TO PRACTICE

SESSION 1:
GETTING
STARTED WITH
PYTHON BASICS

THE OBJECTIVES OF TODAY'S SESSION



USE PYTHON



UNDERSTAND
PYTHON



CREATE PYTHON
PROGRAMS



MASTER FUNDAMENTAL
SYNTAX



UNDERSTAND
DATA TYPES



APPLY BASIC
SYNTAX

...
...

TABLE OF CONTENT

01 What is Python?

02 Why Python?

03 How to use Python?

04 HELLO WORLD

05 Comments

06 Variables

07 Data types

08 Basic operators

09 Input and Output (I/O)

WHAT IS PYTHON?

Python is a language we use to talk to computers. Just like we use words and sentences to tell people what to do, we use Python commands to tell computers what to do.

We write these commands in a special text file called a program. When we run the program, the computer reads it and follows the instructions to do what we want.



WHY PYTHON ?

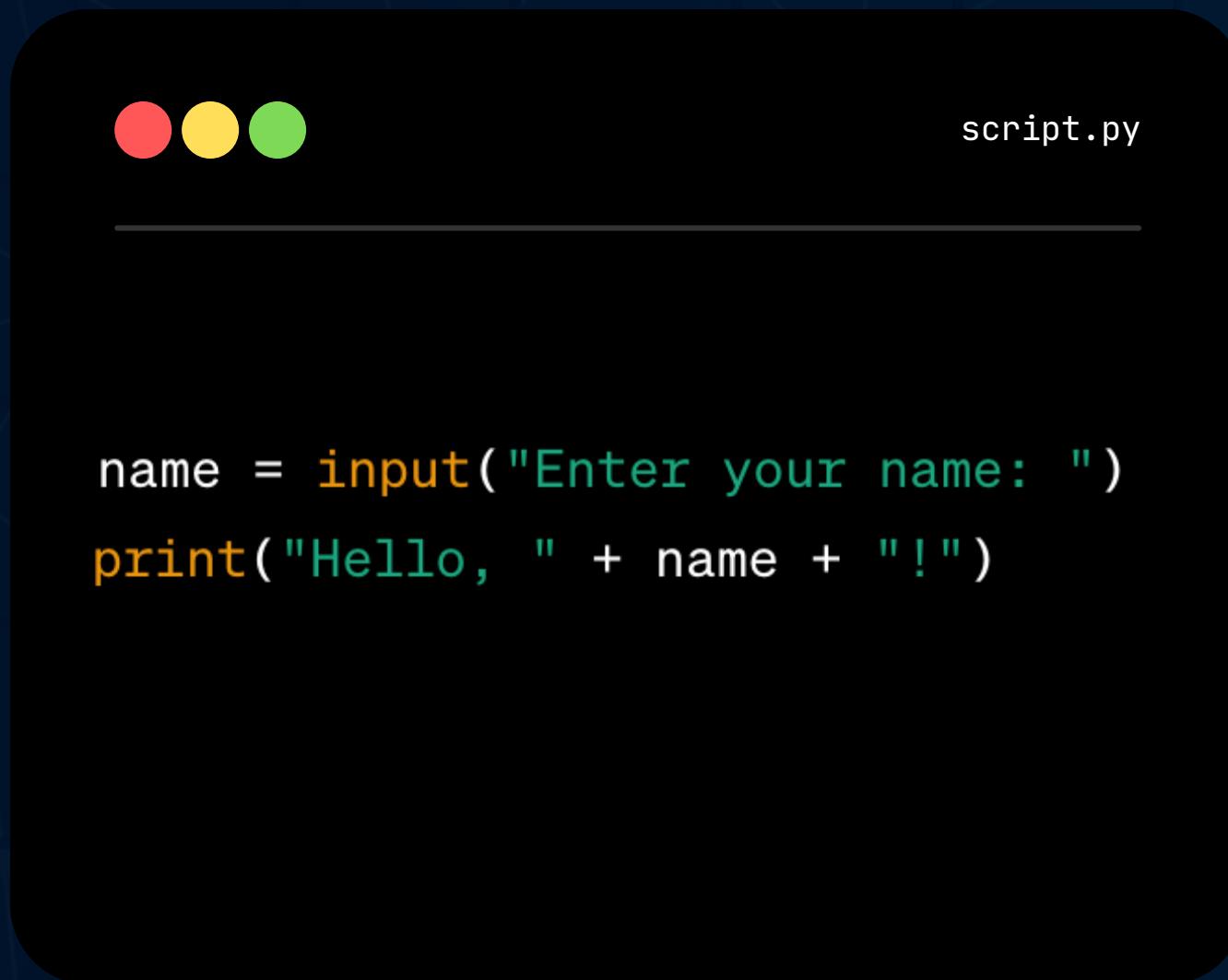
You're learning a new language, but it's not for talking with people, it's for talking with computers.

Just like some languages are easier to learn, Python is like the friendliest language for talking to computers.

It makes learning to talk to computers fun and less intimidating, especially when you're just getting started.



PYTHON VS JAVA



script.py

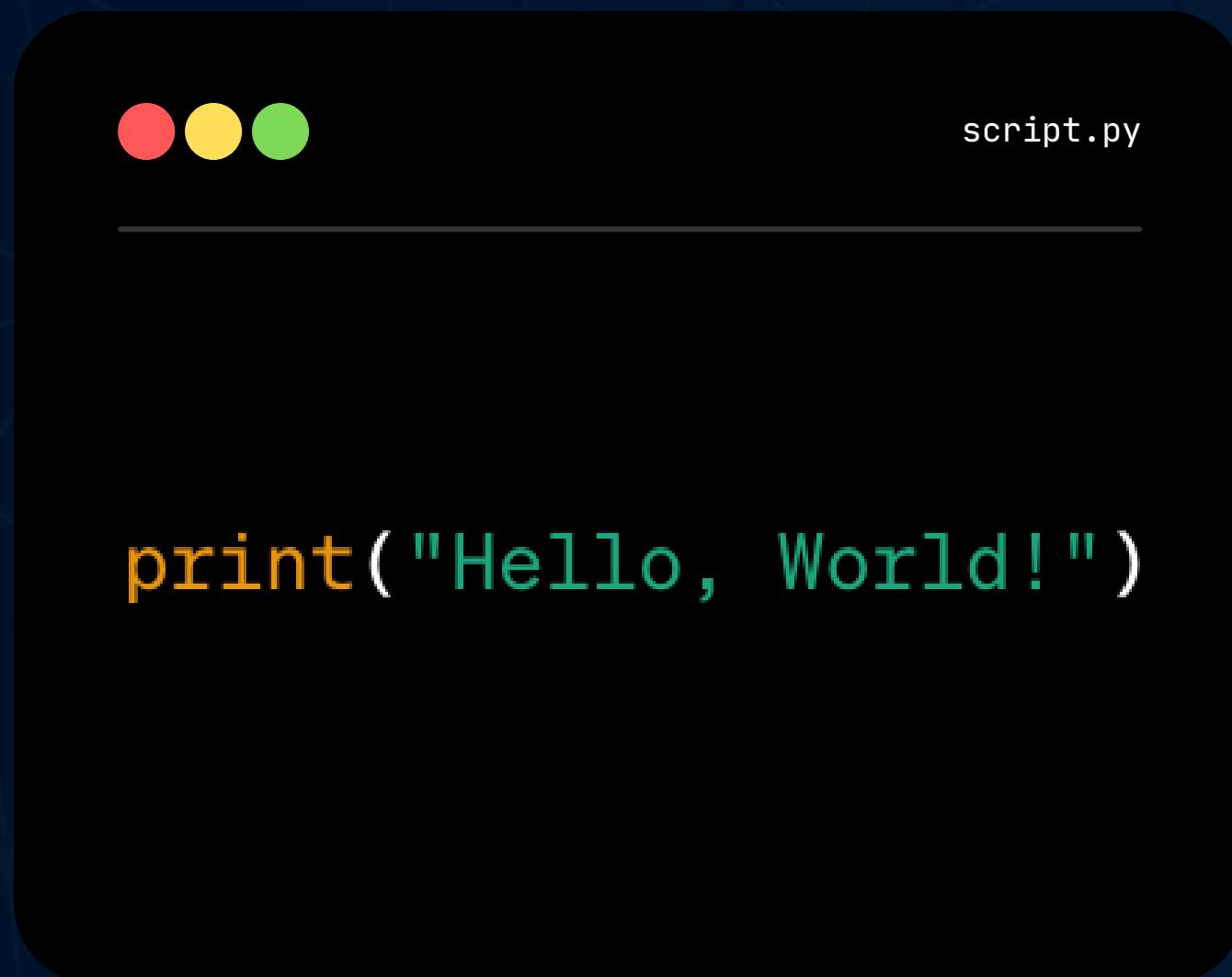
```
name = input("Enter your name: ")  
print("Hello, " + name + "!")
```



script.java

```
import java.util.Scanner;  
  
public class Greeting {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter your name: ");  
        String name = scanner.nextLine();  
        System.out.println("Hello, " + name + "!");  
        scanner.close();  
    }  
}
```

PYTHON VS C++



script.py

```
print("Hello, World!")
```



script.java

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, World!" << endl;
    return 0;
}
```

PYTHON APPLICATION FIELDS



PYTHON INSTALLING

Visit the Python Official Website: <https://www.python.org/downloads/> and download the version suitable for your operating system. Follow the installation instructions provided to set up Python on your machine.

The screenshot shows a web browser displaying the Python official website at [python.org/downloads/](https://www.python.org/downloads/). The page has a dark blue header with tabs for Python, PSF, Docs, PyPI, Jobs, and Community. Below the header is the Python logo and navigation links for About, Downloads, Documentation, Community, Success Stories, News, and Events. A prominent yellow button labeled "Download Python 3.11.5" is visible. To the right, there's a graphic of two parachutes descending from clouds, each carrying a wooden crate. Text on the page encourages users to download the latest version for macOS and provides links for other operating systems and development versions.

python.org/downloads/

Python PSF Docs PyPI Jobs Community

python™

About Downloads Documentation Community Success Stories News Events

Download Python 3.11.5

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python 3.12? [Prereleases](#), [Docker images](#)

PYTHON INSTALLING

Another option is the **Anaconda distribution**, which not only installs Python but also includes a variety of third-party libraries that are useful for data science and scientific computing.
Anancoda Website: <https://www.anaconda.com/>

Anaconda Installers

 Windows	 Mac	 Linux
Python 3.11  64-Bit Graphical Installer (898.6 MB)	Python 3.11  64-Bit Graphical Installer (610.5 MB)  64-Bit Command Line Installer (612.1 MB)  64-Bit (M1) Graphical Installer (643.9 MB)  64-Bit (M1) Command Line Installer (645.6 MB)	Python 3.11  64-Bit (x86) Installer (1015.6 MB)  64-Bit (Power8 and Power9) Installer (473.8 MB)  64-Bit (AWS Graviton2 / ARM64) Installer (727.4 MB)  64-bit (Linux on IBM Z & LinuxONE) Installer (340.8 MB)

COMMON IDES



PYCHARM

PyCharm is a dedicated Python IDE providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.



JUPYTER NOTEBOOK

An interactive environment often used for data analysis and research. It combines code, visualizations, and explanatory text in a single document.



VS CODE

A lightweight and versatile code editor with strong Python support through extensions.

HELLO WORLD



HelloWorld.py

```
print("Hello World!")
```

The "Hello World!" program is a time-honored tradition in the programming world. It's often the first program people write when learning a new programming language.

The purpose of this program is simple: it's a way to check if your programming environment is set up correctly and that you can write, run, and see output from a program.

COMMENTS



#Comments

script.py

Comments are very important while writing a program. They explains the codes in lines for programmers to better understand a program.

In Python, we use the hash (#) symbol to start writing a comment.

Python Interpreter ignores comments during execution.



VARIABLES



WHAT IS A VARIABLE?

In programming, a **variable** is like that labeled box. You give it a name, put information (numbers, words, etc.) inside, and then you can use that information by using the variable's name.

Just like changing what's in the box, you can change the information in the variable as your program goes on.

WHY USE VARIABLES?

1. STORING INFO:

Think of variables as memory spots. They hold important stuff, like names or game scores.

2. BEING FLEXIBLE:

Variables let us change values easily. We swap a value inside, and our program adapts without redoing everything.

3. MAKING SENSE:

Using variables keeps code human-friendly. Instead of puzzling over numbers, we see names that tell us what's going on.

4. USING AGAIN:

Once you save info in a variable, you can use it again and again. No need to say the same thing many times.

5. DATA MANIPULATION:

Variables let us do things like adding numbers or putting words together.

6. FIXING STUFF:

If you need to update or change data, you only have to do it in one place.



USING VARIABLES

script.py

```
age = 25
```

- **Declaring a Variable:** When you want to create a variable, you give it a name. Imagine it as naming a container. For example, you could name a container 'age' to hold someone's age.
- **Assigning a Value:** Once you've named your container (variable), you can fill it with something. You do this by using the = sign. For example, age = 25 puts the value 25 inside the 'age' container.
- **Using Variables:** With your container full, you can use what's inside. It's like taking out what's in the container. For example, you might use print(age) to show the age on the screen.

INTEGERS

Integers are like whole numbers. You can use them for counting and performing arithmetic operations like addition, subtraction, multiplication, and division.

FLOATS

FLOATS are numbers with decimal points. They're used for more precise calculations, including mathematical operations involving fractions.

DATA TYPES

STRINGS

Think of these as words or sentences. If you put something in double quotes like "Hello!", it's a string.
you can use them to represent text and manipulate it in various ways.

BOOLEANS

These are like the answers to yes/no questions. "True" or "False." They help you make decisions in your code.

BASIC OPERATORS

1. ADDITION `+`

2. SUBTRACTION `-`

3. MULTIPLICATION `*`

4. DIVISION `/`

5. MODULO `%`

6. EXPONENT `**`





INPUT AND OUTPUT (I/O)

script.py

```
name = input("What's your name? ")
print("Hello, " + name + "!")
```

INPUT (GETTING INFORMATION):

Input is like asking a question and getting an answer from the user.

It allows your program to interact with people.



INPUT AND OUTPUT (I/O)

script.py

```
age = 25
print("You are " + str(age))
```

OUTPUT (SHOWING INFORMATION):

Output is sharing information with the user. It lets your program communicate its results, messages, or data to people.

HOMEWORK ASSIGNMENT: CALCULATE STUDENT AVERAGE

You've learned about variables, data types, basic operators, and input/output in Python. Now it's time to put your skills to the test!

Your task is to write a Python program that helps calculate a student's average grade in three subjects: math, physics, and science. Follow these steps:

1. Ask the user for the student's name.
2. Ask the user to enter the grades for the student in each of the three subjects.
3. Calculate the average of the three grades.
4. Display a message that includes the student's name and their calculated average.



OUTPUT

Enter student's name: Sarah

Enter math grade: 15

Enter physics grade: 8

Enter science grade: 9

Student Sarah has an average grade of 10.67.

HAPPY CODING AHEAD!

Every line of code you write is a step towards mastering Python. Keep practicing, keep exploring, and keep pushing your boundaries.

Thank you for joining us on this journey!

Sarah Haddad
haddads686@gmail.com

