Programming for AI

Abdul Haseeb

BS(AI)-IV

Programming Languages vastly used for AI

PYTHON

R

JAVA

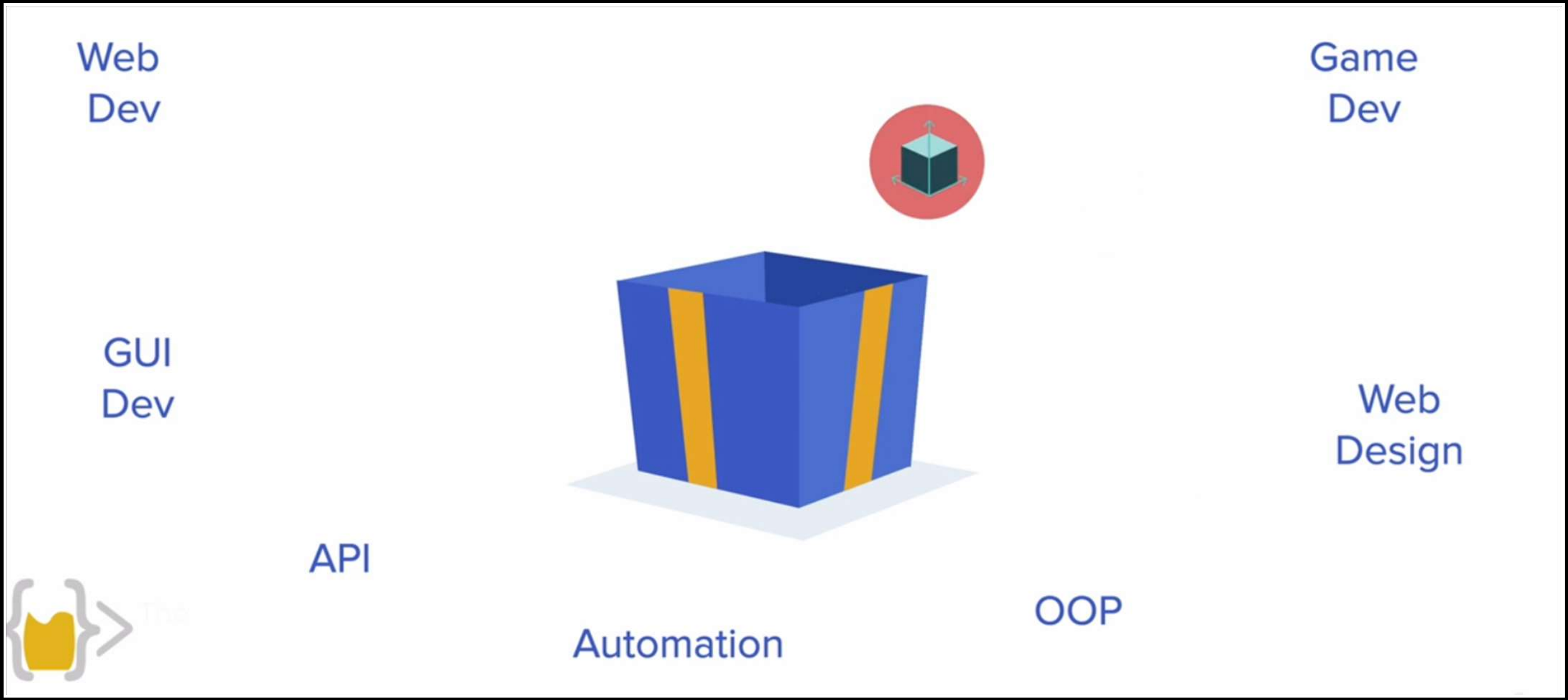
JULIA

Why to choose Python among all?

Simplicity: Python's readable and clean syntax makes it easy to learn and use, speeding up development.

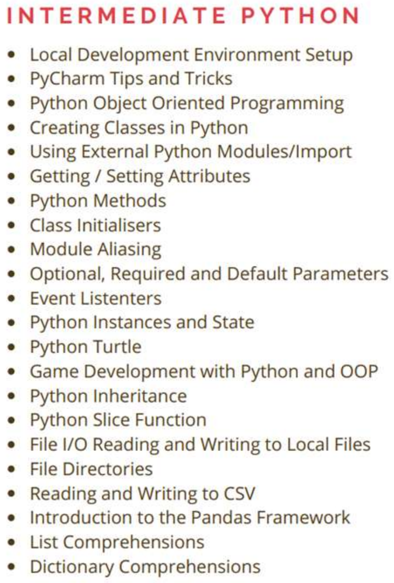
Community Support: Python has a large and active community, providing resources, tutorials, and solutions to common problems.

Wide Adoption: Python is used across industries and in academia, ensuring a strong job market and frequent updates to libraries and tools.



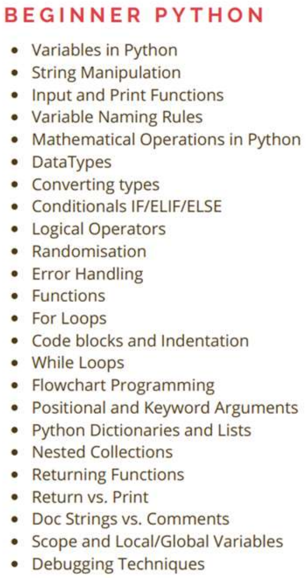
Expectations

Overview



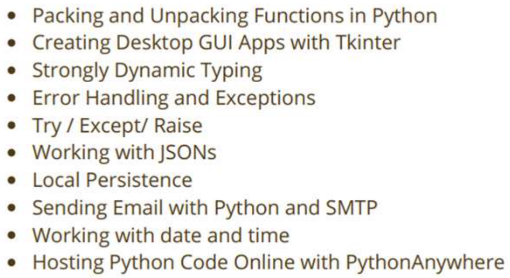
INTERMEDIATE PYTHON

- Local Development Environment Setup
- PyCharm Tips and Tricks
- Python Object Oriented Programming
- Creating Classes in Python
- Using External Python Modules/Import
- Getting / Setting Attributes
- Python Methods
- Class Initialisers
- Module Aliasing
- Optional, Required and Default Parameters
- Event Listenters
- Python Instances and State
- Python Turtle
- Game Development with Python and OOP
- Python Inheritance
- Python Slice Function
- File I/O Reading and Writing to Local Files
- File Directories
- Reading and Writing to CSV
- Introduction to the Pandas Framework
- List Comprehensions
- Dictionary Comprehensions



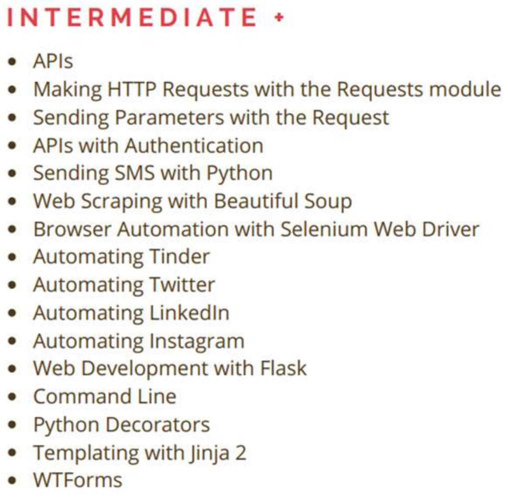
BEGINNER PYTHON

- Variables in Python
- String Manipulation
- Input and Print Functions
- Variable Naming Rules
- Mathematical Operations in Python
- DataTypes
- Converting types
- Conditionals IF/ELIF/ELSE
- Logical Operators
- Randomisation
- Error Handling
- Functions
- For Loops
- Code blocks and Indentation
- While Loops
- Flowchart Programming
- Positional and Keyword Arguments
- Python Dictionaries and Lists
- Nested Collections
- Returning Functions
- Return vs. Print
- Doc Strings vs. Comments
- Scope and Local/Global Variables
- Debugging Techniques



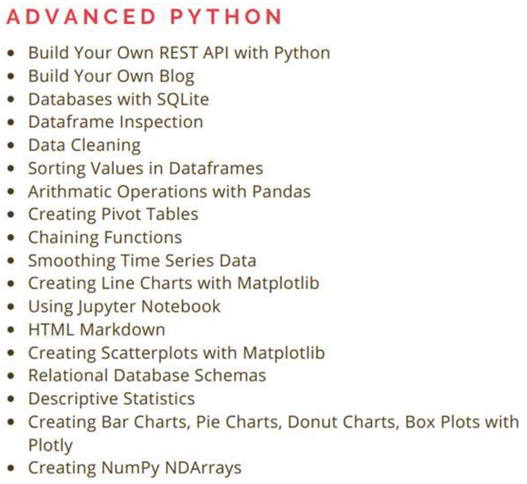
- Packing and Unpacking Functions in Python
- Creating Desktop GUI Apps with Tkinter
- Strongly Dynamic Typing
- Error Handling and Exceptions
- Try / Except/ Raise
- Working with JSONs
- Local Persistence
- Sending Email with Python and SMTP
- Working with date and time
- Hosting Python Code Online with PythonAnywhere

Overview



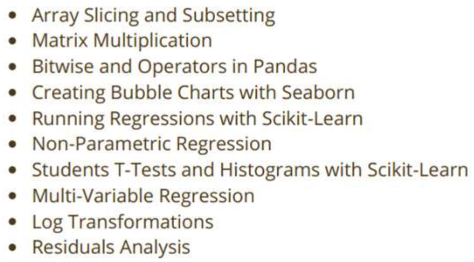
INTERMEDIATE +

- APIs
- Making HTTP Requests with the Requests module
- Sending Parameters with the Request
- APIs with Authentication
- Sending SMS with Python
- Web Scraping with Beautiful Soup
- Browser Automation with Selenium Web Driver
- Automating Tinder
- Automating Twitter
- Automating LinkedIn
- Automating Instagram
- Web Development with Flask
- Command Line
- Python Decorators
- Templating with Jinja 2
- WTForms



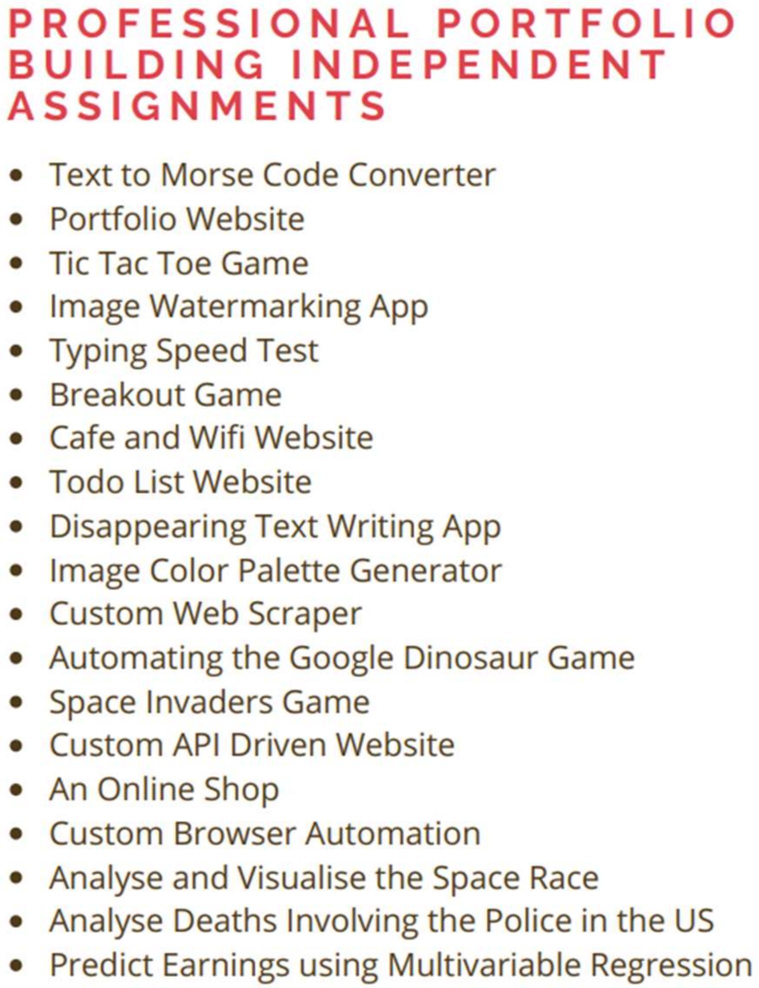
ADVANCED PYTHON

- Build Your Own REST API with Python
- Build Your Own Blog
- Databases with SQLite
- Dataframe Inspection
- Data Cleaning
- Sorting Values in Dataframes
- Arithmatic Operations with Pandas
- Creating Pivot Tables
- Chaining Functions
- Smoothing Time Series Data
- Creating Line Charts with Matplotlib
- Using Jupyter Notebook
- HTML Markdown
- Creating Scatterplots with Matplotlib
- Relational Database Schemas
- Descriptive Statistics
- Creating Bar Charts, Pie Charts, Donut Charts, Box Plots with Plotly
- Creating NumPy NDArrays



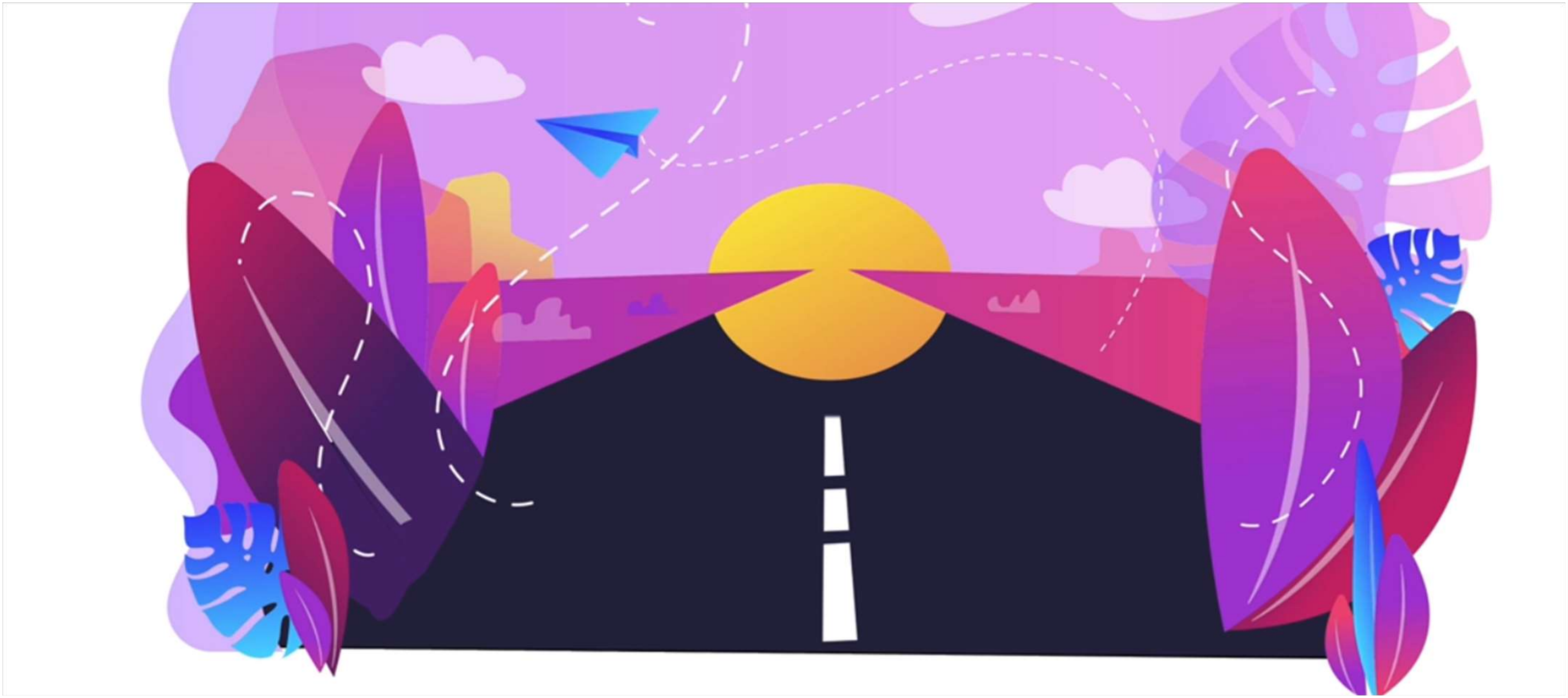
- Array Slicing and Subsetting
- Matrix Multiplication
- Bitwise and Operators in Pandas
- Creating Bubble Charts with Seaborn
- Running Regressions with Scikit-Learn
- Non-Parametric Regression
- Students T-Tests and Histograms with Scikit-Learn
- Multi-Variable Regression
- Log Transformations
- Residuals Analysis

Overview

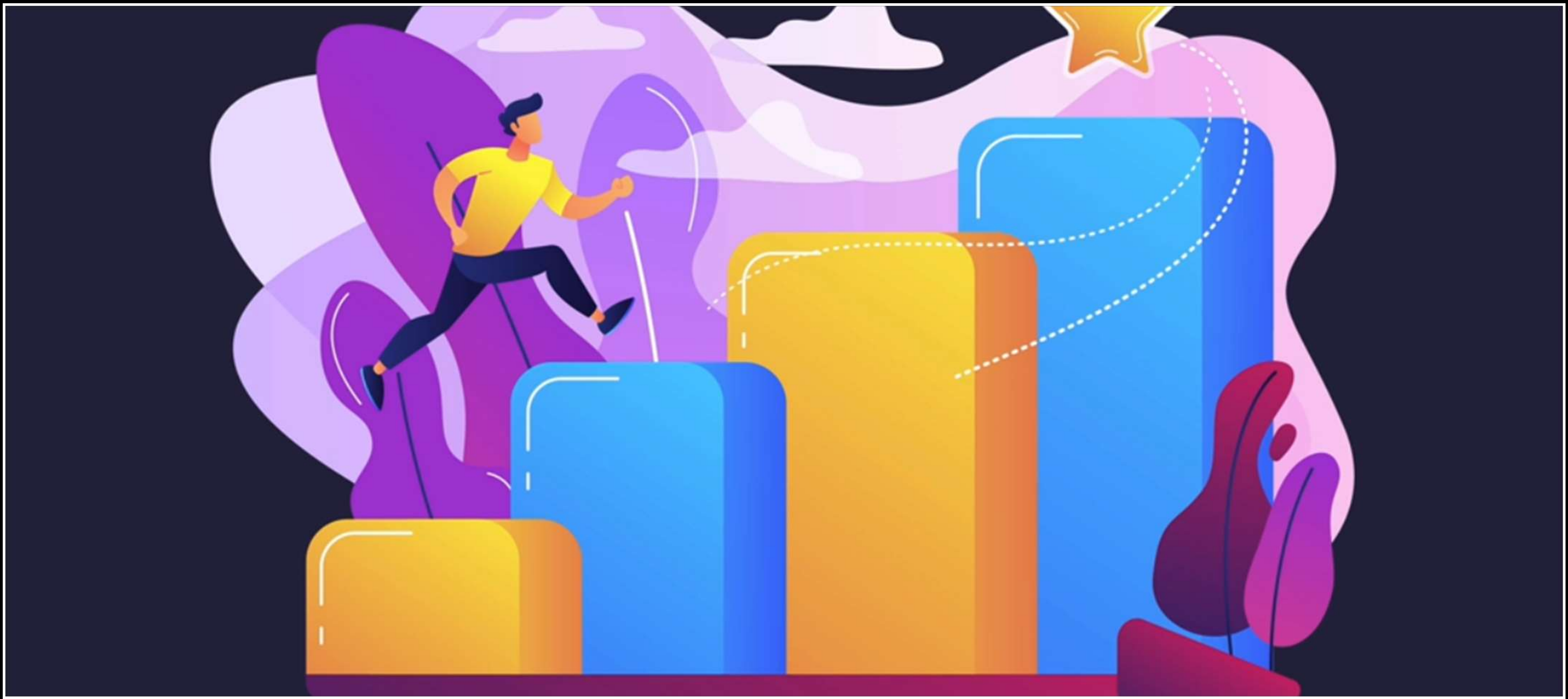


PROFESSIONAL PORTFOLIO BUILDING INDEPENDENT ASSIGNMENTS

- Text to Morse Code Converter
- Portfolio Website
- Tic Tac Toe Game
- Image Watermarking App
- Typing Speed Test
- Breakout Game
- Cafe and Wifi Website
- Todo List Website
- Disappearing Text Writing App
- Image Color Palette Generator
- Custom Web Scraper
- Automating the Google Dinosaur Game
- Space Invaders Game
- Custom API Driven Website
- An Online Shop
- Custom Browser Automation
- Analyse and Visualise the Space Race
- Analyse Deaths Involving the Police in the US
- Predict Earnings using Multivariable Regression

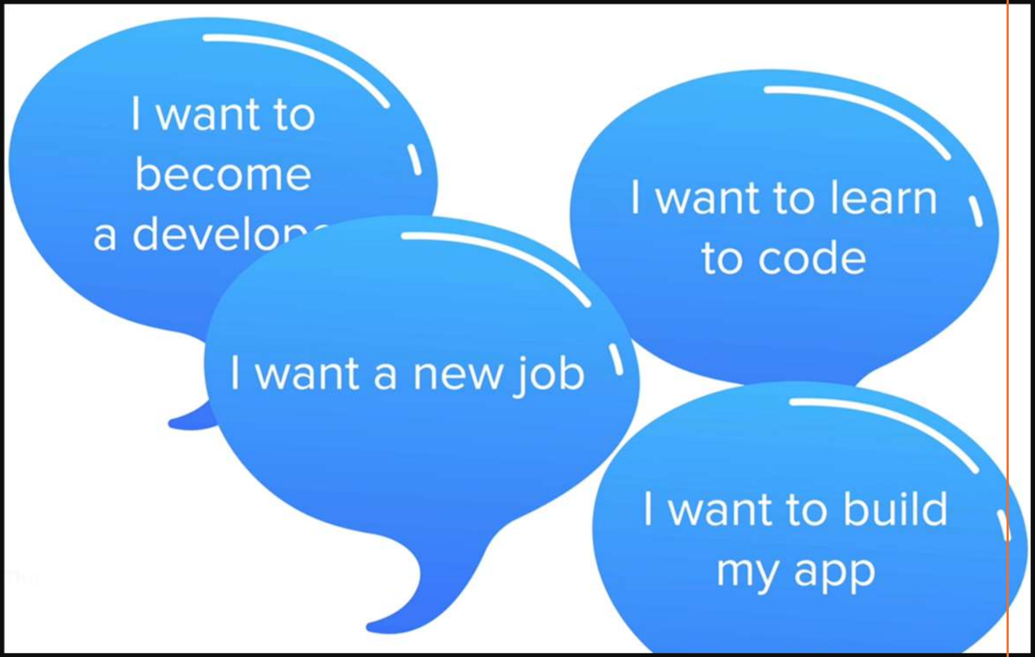


Start and Keep Going…



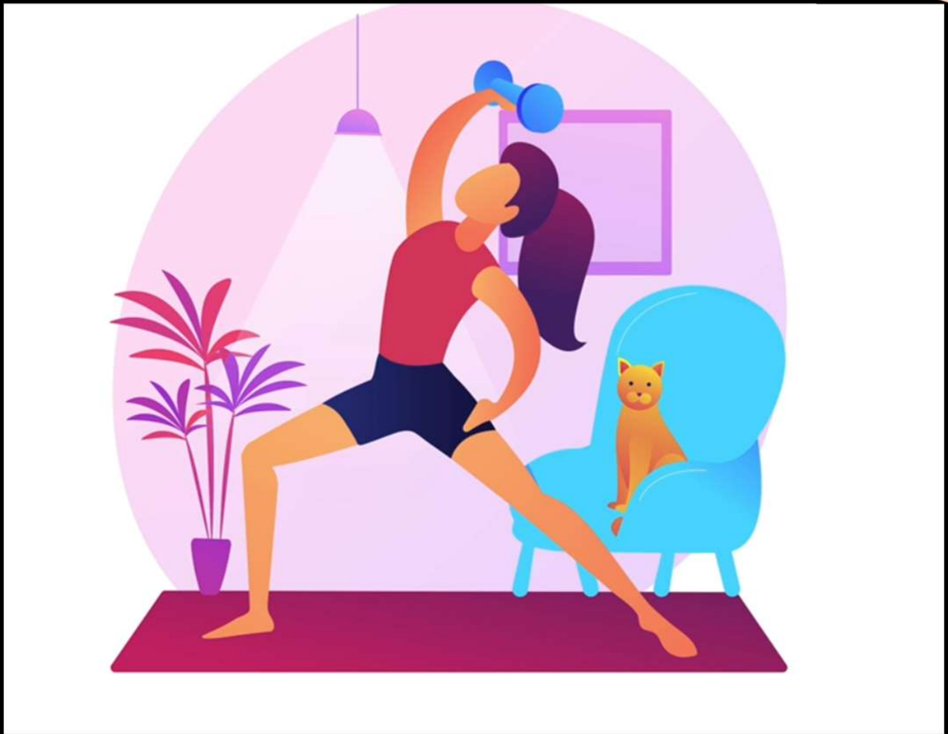
Challenge yourself

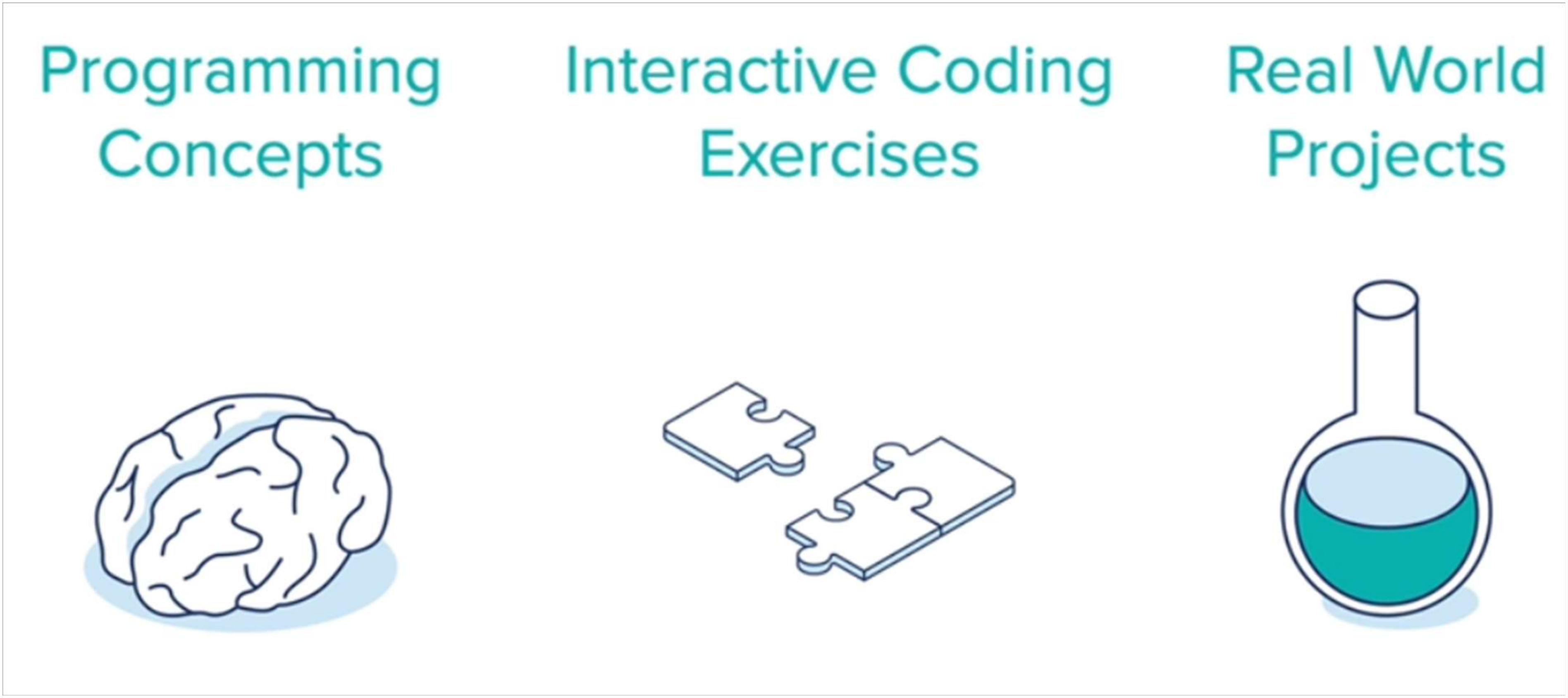
Without Actions, It’s all meaningless



What’s the best way to learn to code?

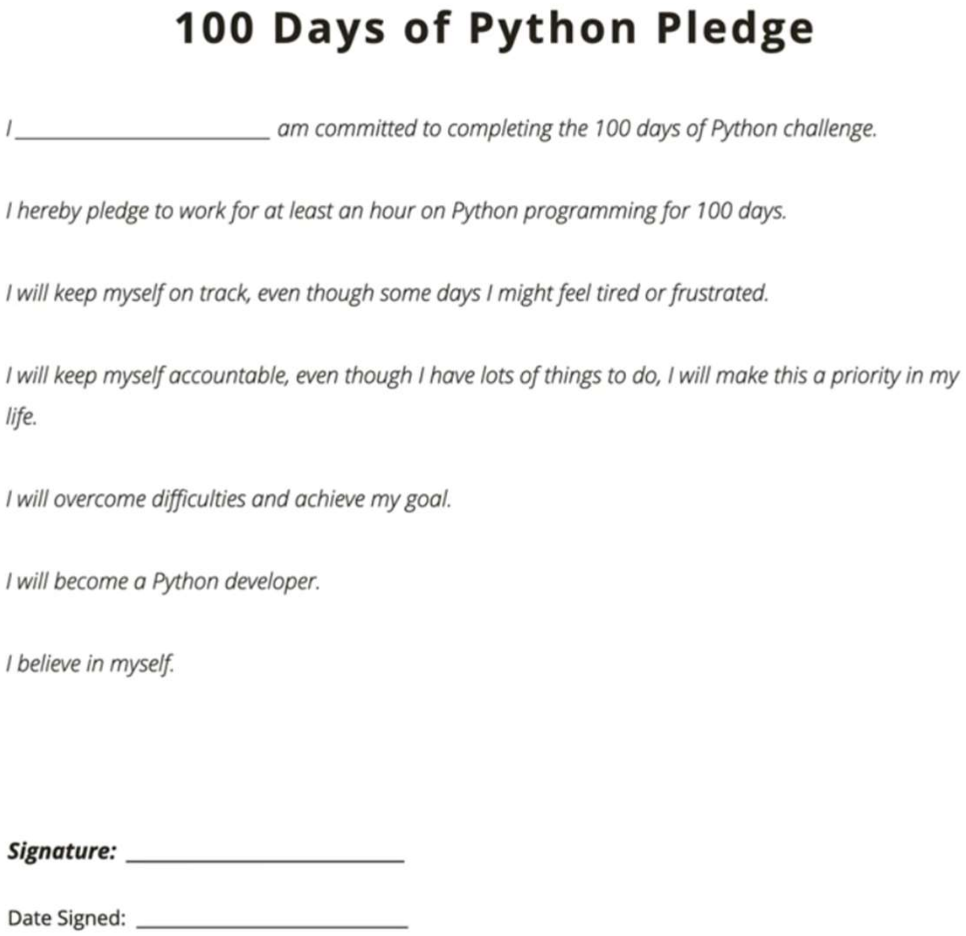
Code for an hour for 100 days at least





What to code each day?

Pledge and Make a contract to yourself



100 Days of Python Pledge

I ________________________ am committed to completing the 100 days of Python challenge.

I hereby pledge to work for at least an hour on Python programming for 100 days.

I will keep myself on track, even though some days I might feel tired or frustrated.

I will keep myself accountable, even though I have lots of things to do, I will make this a priority in my life.

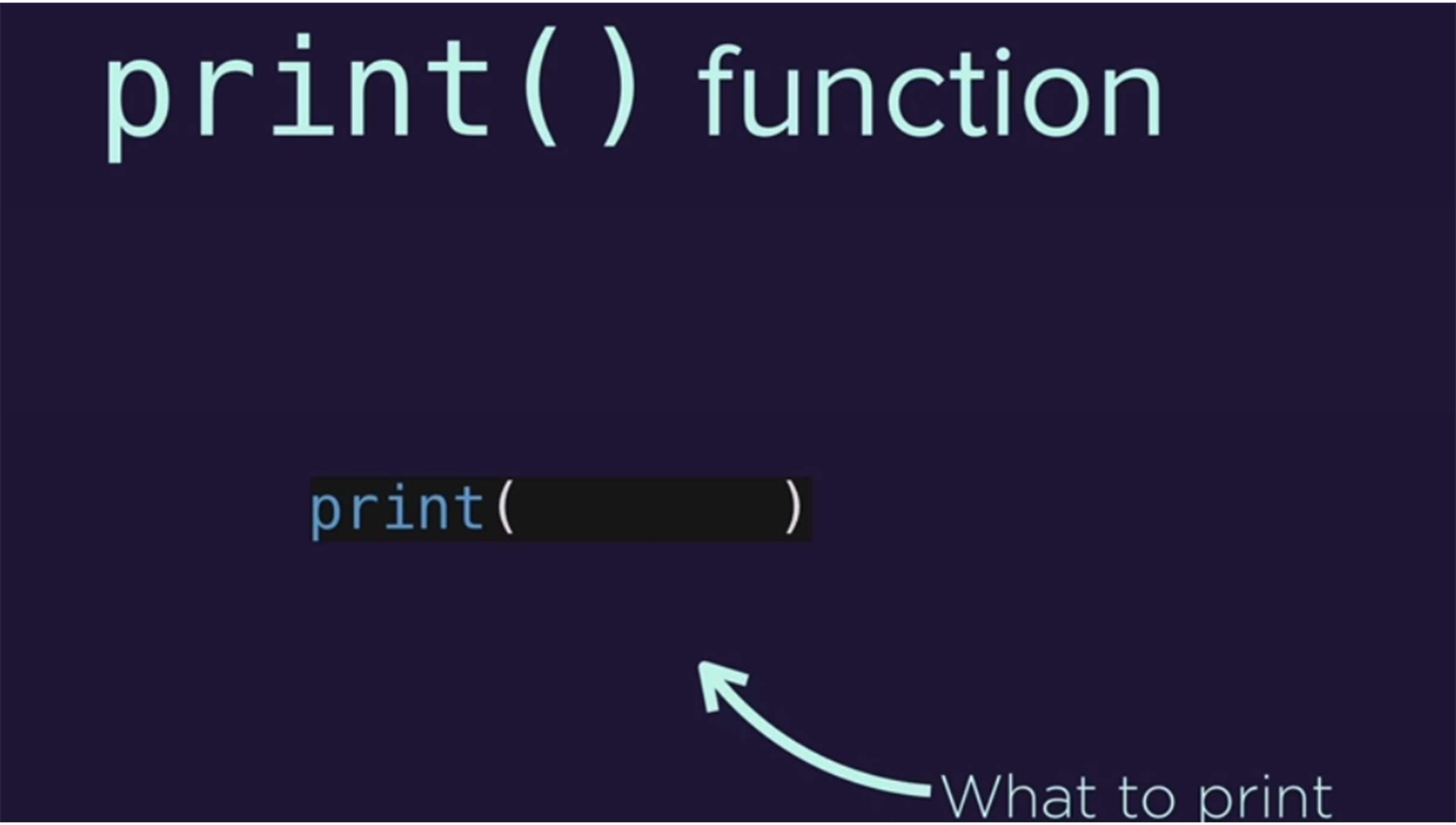
I will overcome difficulties and achieve my goal.

I will become a Python developer.

I believe in myself.

Signature: ________________________

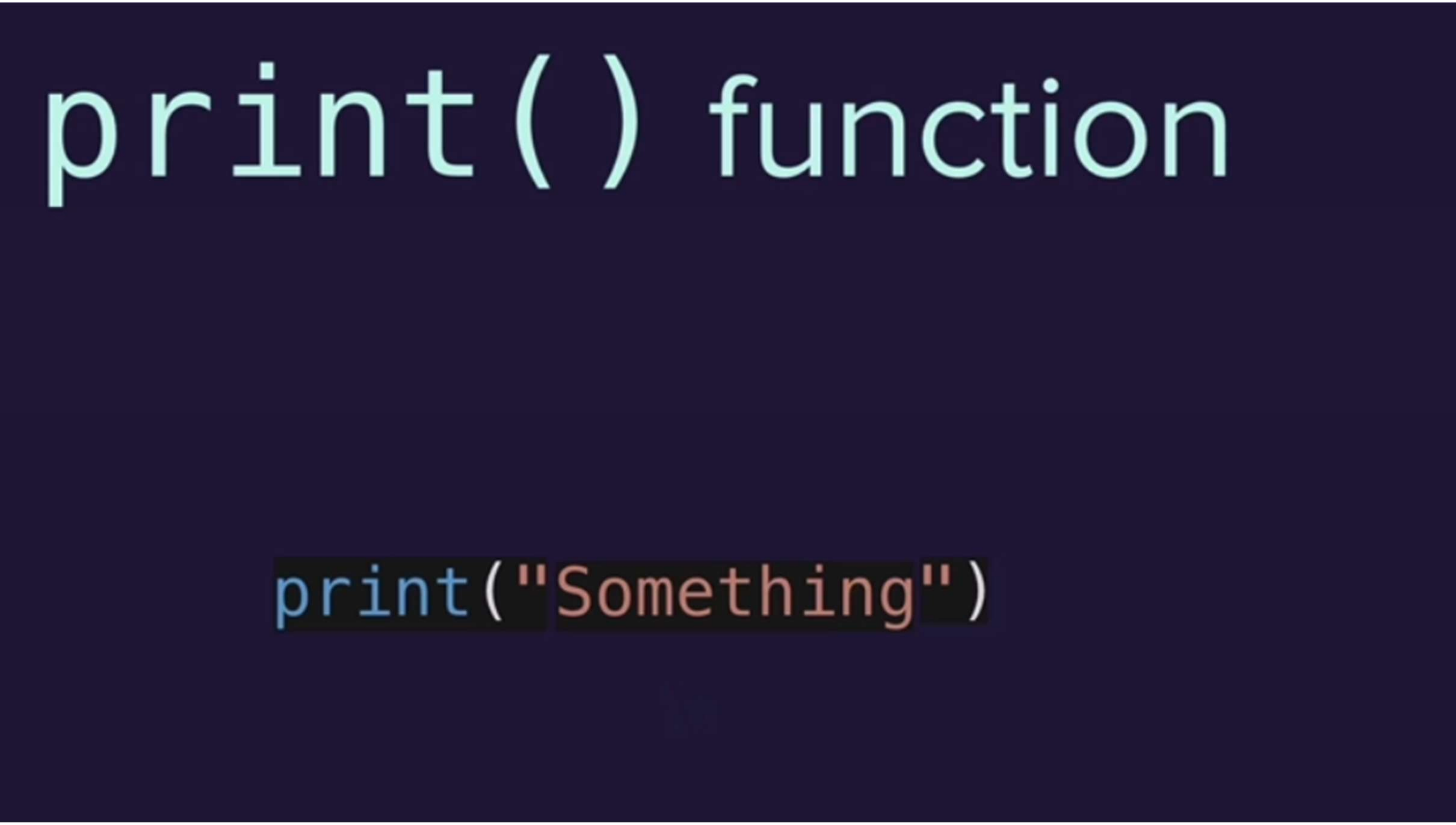
Date Signed: ________________________



print() function

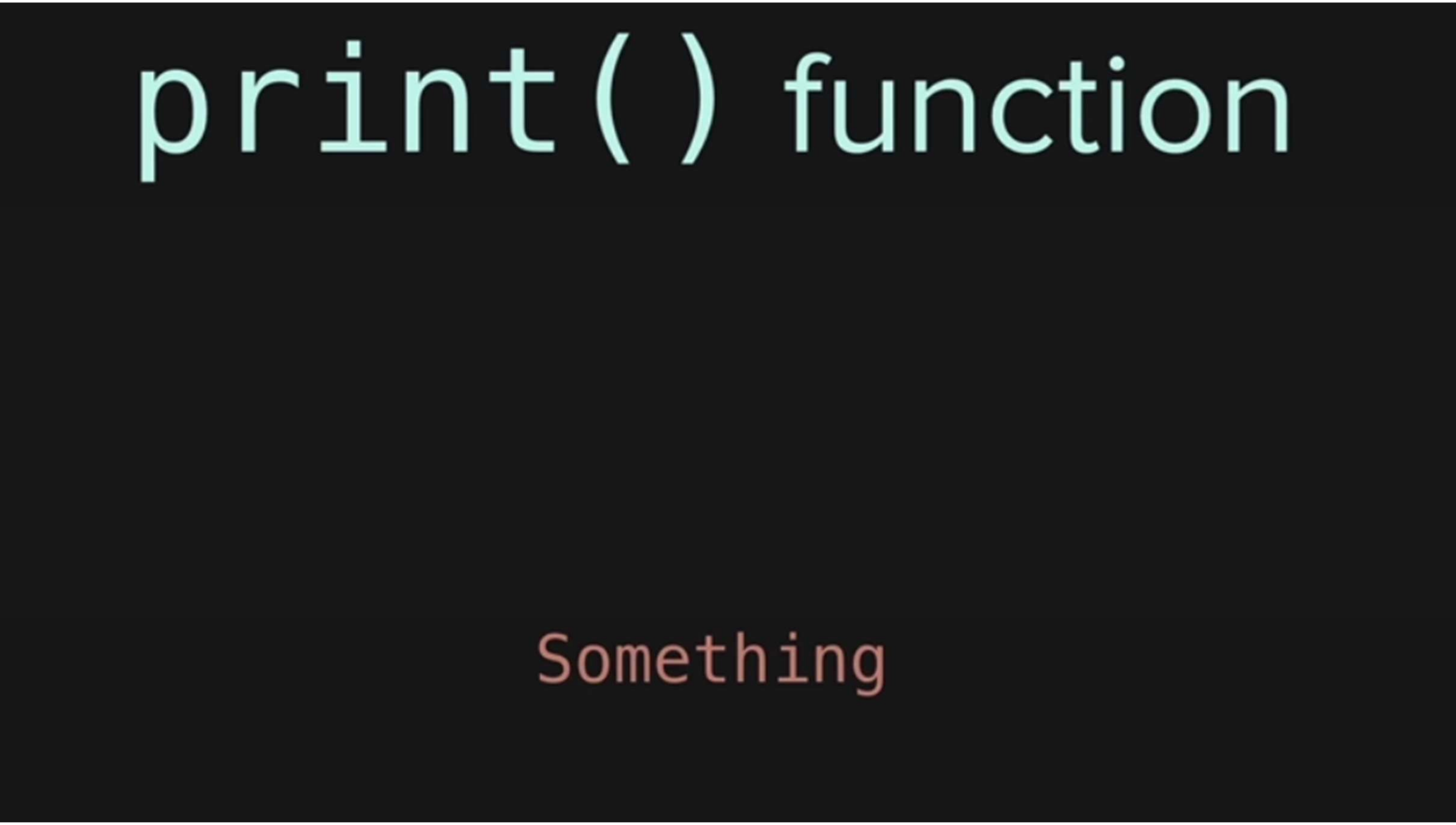
```
print(          )
```

What to print



print() function

```
print("Something")
```

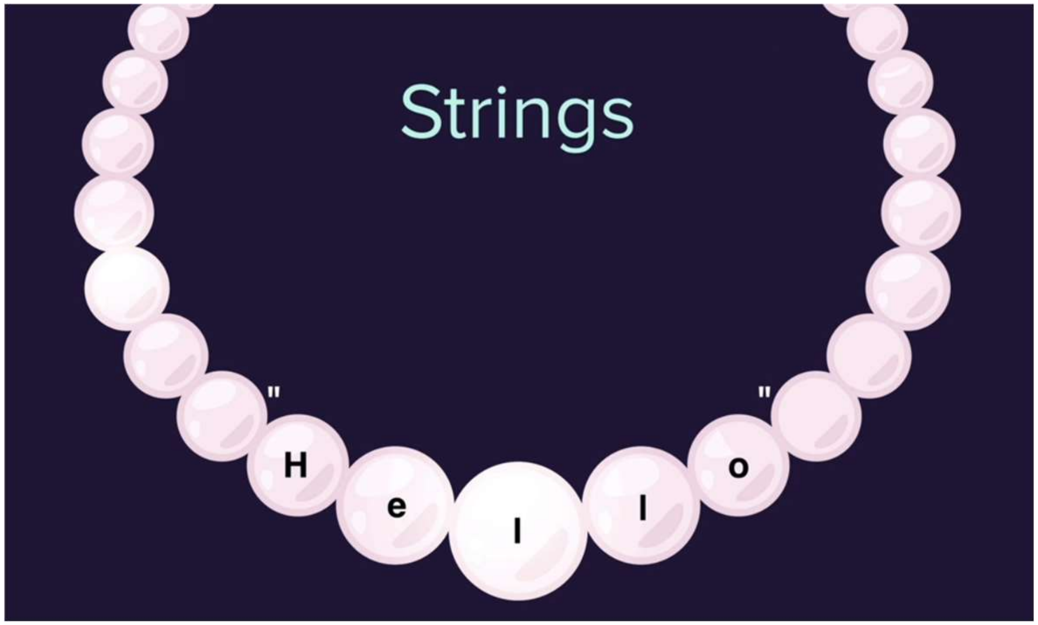


print() function

```
Something
```

“ ” tell the interpreter it is a text

- Text is called string in Programming world…, “ ” tell the start and end



Don’t get confused if you get errors..

1

Remove “ from the start and read the error.

2

Remove ” from the end and read the error.

3

Also copy the error and paste it in browser and head over to stack overflow.

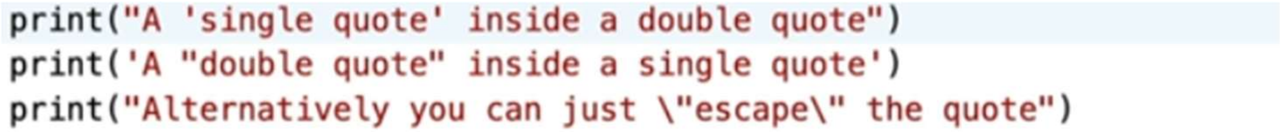
Print Modifiers

```
print("She said: "Hello" and then left.")
```

```
print('She said: "Hello" and then left.')
```

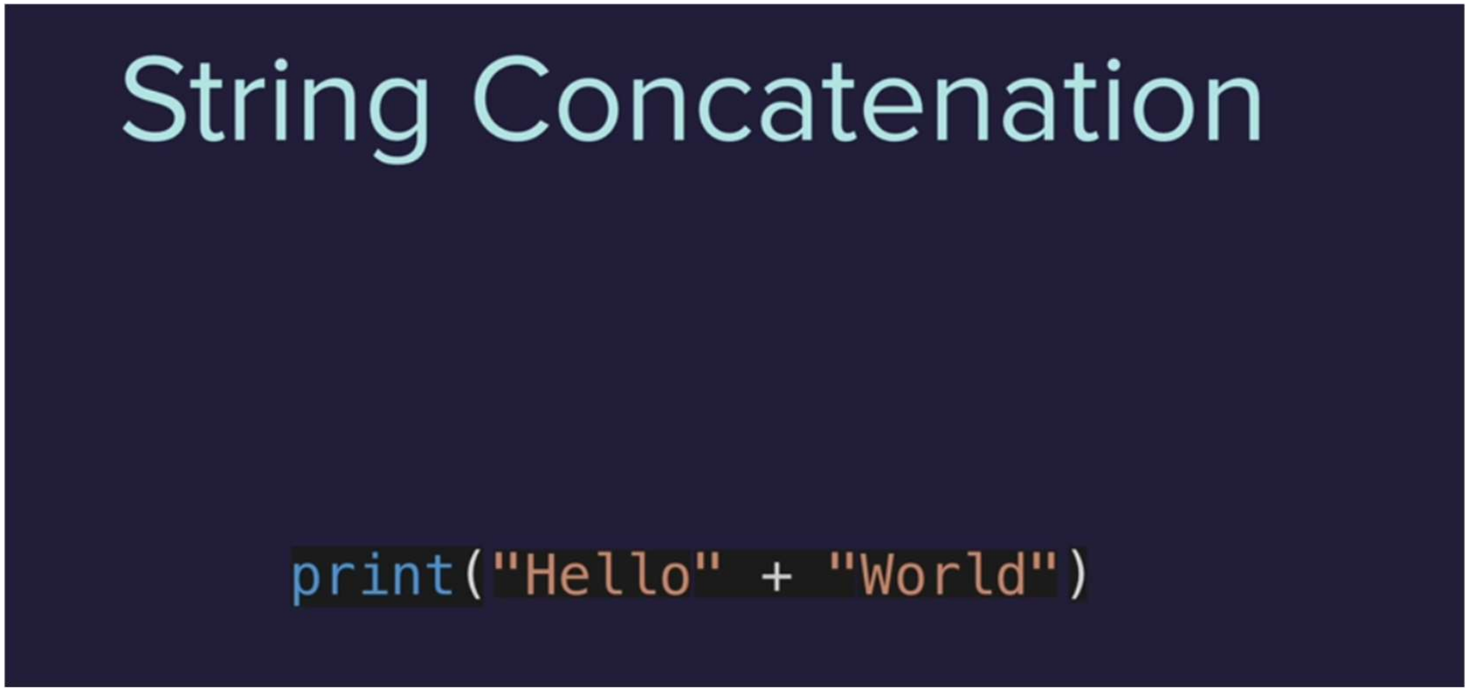
```
print("She said: \"Hello\" and then left.")
```

Make sure to try these out, and also print numbers without double quotes



```
print("A 'single quote' inside a double quote")
print('A "double quote" inside a single quote')
print("Alternatively you can just \"escape\" the quote")
```

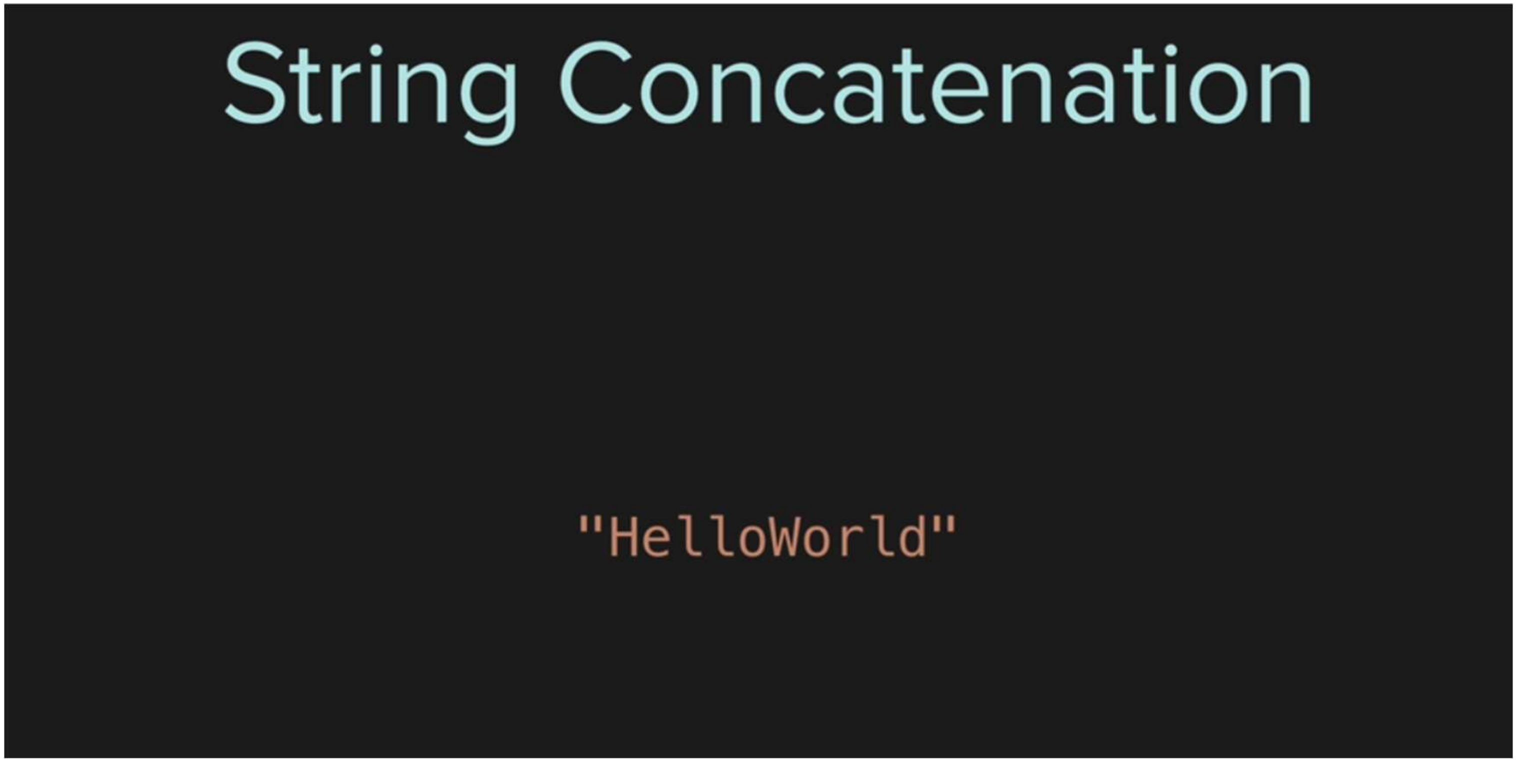
String Manipulation and code Intelligence



String Concatenation

```
print("Hello" + "World")
```

String Manipulation and code Intelligence



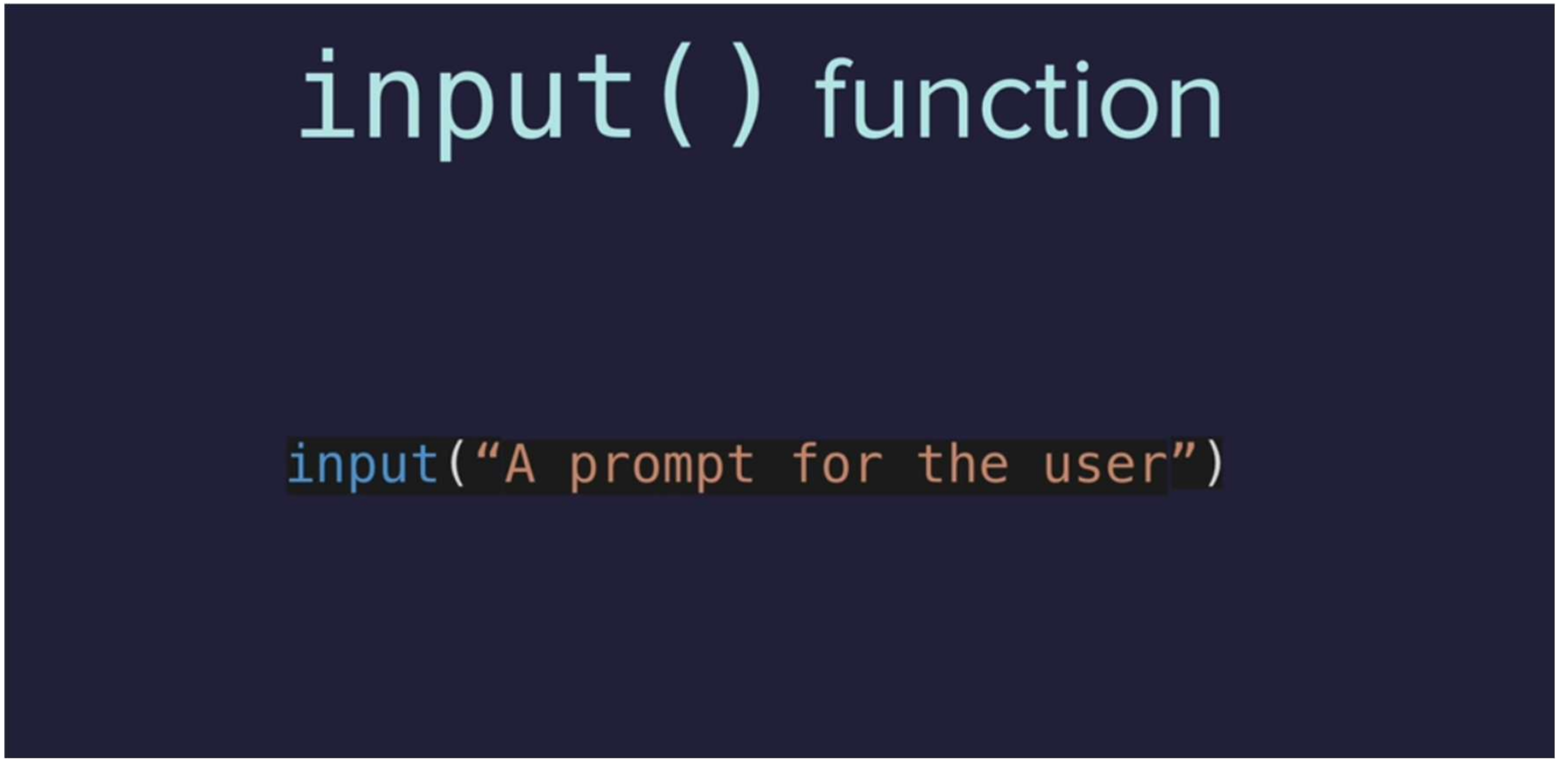
String Concatenation

```
"HelloWorld"
```

Spaces are very sensitive in python

- Indentation errors may arise!

Return Value is always a string



input() function

```
input(“A prompt for the user”)
```

We can convert the return value although

Let’s try to do it..

You can't manipulate with the data if you don't store it some where

input("What is your name?")

After running this line, you will never be able to access the return value from the input

Variables

Variables

James:012 534 5769

Remember Variables are Changeable

- name=“Haseeb”
- print(name)

- name=“Ali”
- print(name)

When Naming Variables...

Remember to follow the Variable naming Rules

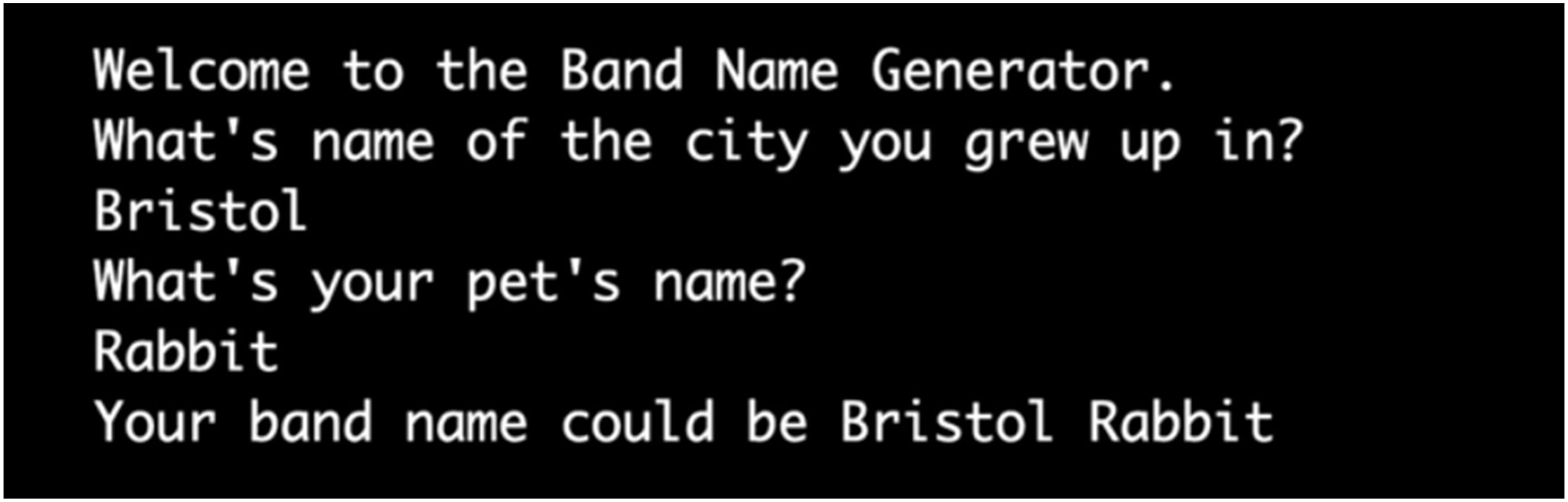
When Accessing the Variables..

- Remember to use the name with which you defined the variable, any misspelling will result in a syntax error

Add clarity to your code for yourself and others..

- Comments are a nice way to do..
- Begin with #

Coding Challenge: Suggest a Band Name to Your Friend



```
Welcome to the Band Name Generator.
What's name of the city you grew up in?
Bristol
What's your pet's name?
Rabbit
Your band name could be Bristol Rabbit
```

Let’s Try the code snippets..

```
print(len(“Hello”))
//Works Fine
```

```
print(len(1234))
// Gives Error, Need to understand data types
```

Let's explore Primitive Data Types

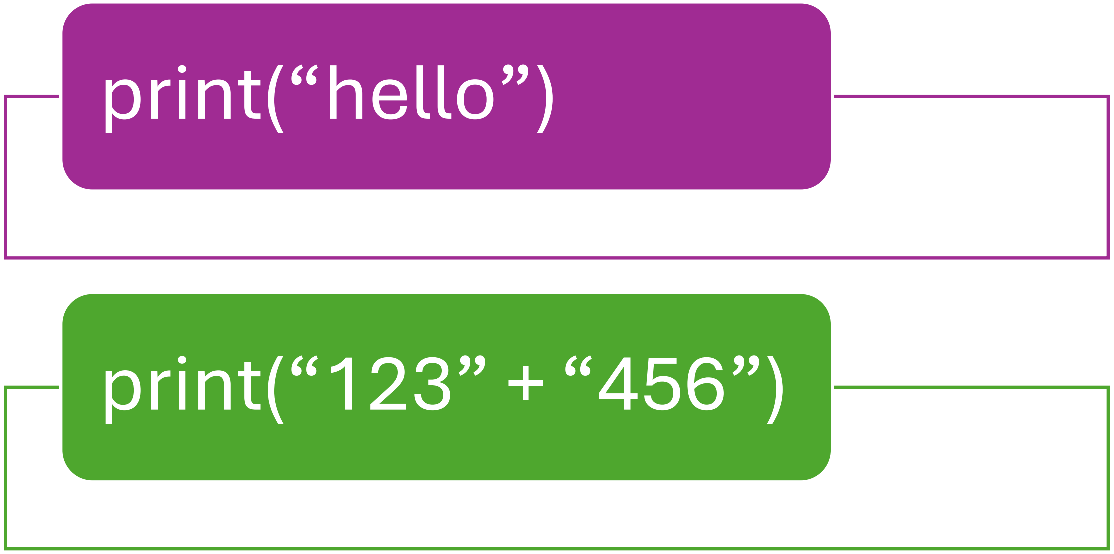
String

Integer

Float

Boolean

String



```
print(“hello”)
```

```
print(“123” + “456”)
```

Integer

All whole Numbers

print(123456789)

print(123+456)

print(123_456_678)   //Alternative for: 123,456,678 (Large Integer)

Boolean

Either **True** or **False**

Type Error (Occurs due to Type Checking)

```
Num_char=len(input(“Please Enter your name))
```

```
print(“Your name has”+ Num_char + “Characters”)
```

String doesn’t accept Integers along with it.

Type Conversion

Convert one data type into other

Let’s do some examples

Mathematical Operations

- Addition (+)
- Subtraction (/)
- Multiplication (*)
- Division (/)
- Integer Division (//)
- Power (**)

f-string

score=0

height=1.8

isWinning=True

Converting each of them to string would be hectic

So let’s try this:

print(f“Your score is {score},Your height is {height}, You are winning is {isWinning}”)