

INTRODUCING PYTHON

FULL STACK SKILLS BOOTCAMP

INTRODUCING PYTHON

- **Lesson Overview:**

- In this lesson, we will be introduced to:

1. Installing Python & Running in VSCode
2. Virtual Environments
3. Debugging in VSCode
4. Variables, Loops, and Functions



A SHORT HISTORY OF PYTHON

- **Created by Guido van Rossum in 1991**
- **Designed for readability and simplicity**
- **Common uses**
 1. Web Development (Django, Flask)
 2. Data Science & AI (Pandas, NumPy, TensorFlow)
 3. Automation & Scripting
 4. Game Development & Cybersecurity
 5. Artificial Intelligence



WHY LEARN PYTHON?

- More readable and concise than JavaScript
- Used heavily in AI, ML, and automation
- Strong community and vast ecosystem

INSTALLING PYTHON & RUNNING IN VSCODE

- **Installation:**

Download Python from python.org

Add Python to system PATH during installation

- **Setting up VSCode:**

Install the **Python** extension

Open a .py file and select the Python interpreter).

- **Running Python Code in VSCode:**

Run a script: `python script.py`

Use the VSCode terminal or built-in debugger

VIRTUAL ENVIRONMENTS

- Why use Virtual Environments?
 1. Keeps dependencies isolated
 2. Prevents version conflicts
 3. Best practice for Python projects
- Creating & Activating a Virtual Environment:
demo...

```
# Create a virtual environment
python -m venv myenv

# Activate it (Mac/Linux)
source myenv/bin/activate

# Activate it (Windows)
myenv\Scripts\activate
```

DEBUGGING FUNCTIONS IN VSCODE

- **What is Debugging?**
 - Debugging is the process of identifying and fixing errors in your code.
 - Helps you understand how your program executes step by step.

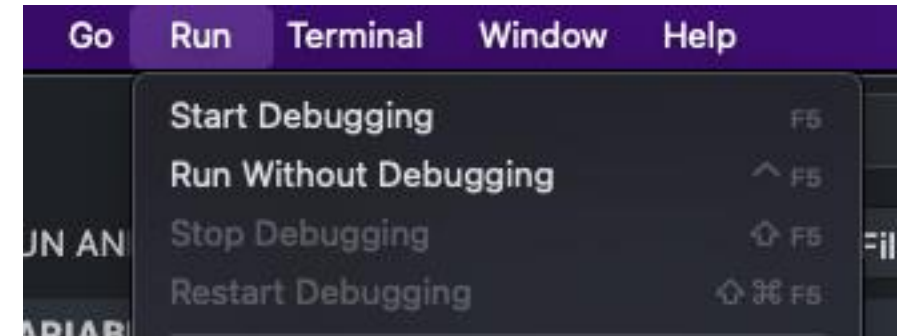
WHY USE A DEBUGGER IN VSCODE?

- See variable values at each step
- Pause code execution at breakpoints
- Find and fix logical errors efficiently

```
1  import questionnaire
2
3  def main_menu():
4      choice = questionnaire.select(
5          "Select an option:",
6          choices=[
7              "Option 1: View Data",
8              "Option 2: Edit Data",
9              "Option 3: Exit"
10         ]
11     ).ask()
12
```


SETTING UP DEBUGGING IN VSCODE

- Open your Python file in VSCode
- Add breakpoints (Click left of the line number)
- Run the debugger (Click Run → Start Debugging or press F5)
- Step through the code using Step Over (F10), Step Into (F11)



COMMON DEBUGGING FEATURES

- **Breakpoints** – Stops execution at a specific line
- **Step Over** – Executes the next line without stepping into functions
- **Step Into** – Steps inside function calls
- **Watch Variables** – Observe how values change
- **Call Stack** – See the function execution order

demo...



VARIABLES, LOOPS & FUNCTIONS

- **What are Variables?**

Variables store values in memory.

No need to declare types (dynamic typing like JavaScript).

```
name = "Alice"  
age = 25  
pi = 3.14
```

PYTHON LOOPS – FOR LOOP

- **Python for loop**

Used to iterate over sequences like lists or strings.

```
for i in range(5):  
    print("Hello", i)
```

PYTHON LOOPS – WHILE LOOP

- **Python while loop**

Executes as long as a condition is True.

```
count = 0
while count < 5:
    print("Count:", count)
    count += 1
```

DEFINING FUNCTIONS IN PYTHON

- **Python functions**

Like JavaScript, Functions help organize reusable code.

```
def greet(name):  
    return f"Hello, {name}!"  
  
print(greet("Alice"))
```

CONCLUSION

- Debugging in VSCode helps find errors easily.
- Variables store data dynamically in Python.
- Loops (for, while) help automate repetitive tasks.
- Functions make code reusable and organised.

QUESTIONS?