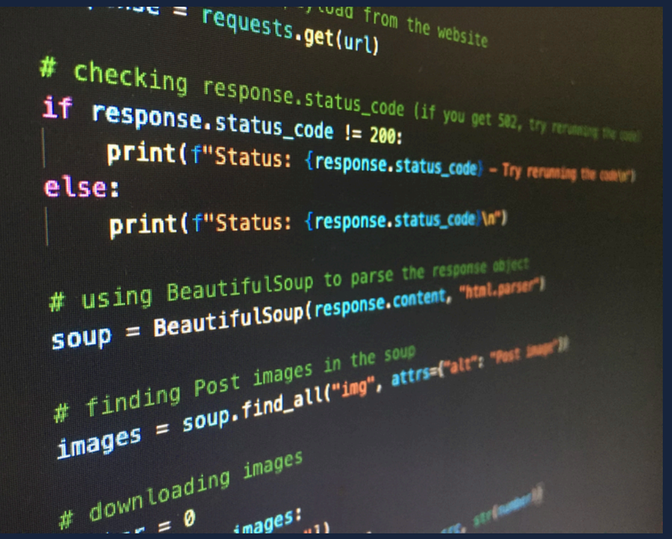
PYTHON FOR AUTOMATION

PYTHON FUNDAMENTALS

PDSC WORKSHOP DAY 1



Overview

- Introduction to Python and Automation
- Why Automate?
- Future of Automation
- Setting up Python Environment
- Python libraries for automation
- What to automate?
- How to automate?
- Example of Automation:
  Downloading Multiple Images in a JPG/PNG Format

PDSC

What is Python?

- Multi-purpose (Web, GUI, Scripting & automation etc.)
- Object Oriented
- Interpreted open source
- Strongly typed and Dynamically typed
- Focus on readability and productivity

Guido Van Rossum
(Creator of Python)

What is Automation?

- Use of technology to perform tasks with minimal human intervention
- Examples: Web scraping, data entry, repetitive tasks

PDSC

Why automate?

- Efficiency: Save time by automating repetitive tasks
- Accuracy: Reduce human error
- Productivity: Free up time for more important tasks
- Consistency: Perform tasks the same way every time

Future of Automation

- Technological Advances:
  AI and Machine Learning integration
- Workforce Impact:
  Shift towards more creative and strategic roles
- Industry Applications:
  Healthcare, finance, manufacturing, and more
- Continuous Evolution:
  Ongoing improvements and new automation tools

Let’s set up our environment!

PDSC

Python Libraries for Automation

- **Selenium**: Web browser automation
- **BeautifulSoup**: Web scraping
- **Requests**: Sending HTTP requests
- **Pandas**: Data manipulation and analysis
- **Schedule**: Task scheduling
- **PyAutoGUI**: GUI automation

What to Automate?

- Data Entry and Extraction:
  - Automate form filling, data scraping
- File Operations:
  - Renaming, moving, and organizing files
- Email Automation:
  - Sending and managing emails.
- Report Generation:
  - Automated report creation
- Monitoring and Alerts:
  - System monitoring, sending alerts

How to Automate?

PDSC

- Identify Repetitive Tasks: Analyze tasks that consume a lot of time
- Choose the Right Tools: Select appropriate libraries and tools for the task
- Write the Script: Develop the automation script in Python
- Test the Script: Ensure it performs as expected
- Deploy and Monitor: Put the script into production and monitor its performance

Example of Automation:

Downloading Multiple Images in JPG/PNG Format

Steps:

1. Install Required Libraries:
    - pip install requests beautifulsoup4
2. Import Libraries:
    - import requests
    - from bs4 import BeautifulSoup
3. Fetch Web Page:
    - response = requests.get('URL')
4. Parse HTML:
    - soup = BeautifulSoup(response.text, 'html.parser')

5. Find Image URLs:
    - img_tags = soup.find_all('img')’
6. Download Images:

```
for img in img_tags:
    img_url = img['src']
    img_data = requests.get(img_url).content
    with open('filename.jpg', 'wb') as handler:
        handler.write(img_data)
```

Python Recap

Text should always be quoted inside “ “

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

print() function prints the given value into the console

Variables

```
job = "programming"
```

Variable

Value

Data Types

Integer
String
Float

```
num = 10
fruit = "mango"
percent = 85.5
```

```
print(type(fruit))
```

```
<class 'str'>
```

PDSC

Converting data type

```
age = "19" #here, age is a string value
age = int(age) #now, age is converted to int
age = float(age) #now, it is converted to float
age = str(age) #back to string



print(type(age))
```

Operators

PDSC

Arithmetic Operators

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

Comparison Operators

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Comparison of string is in alphabetical order while of numbers is in mathematical.

PDSC

Logical Operators

Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

PDSC

Assignment Operators

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3

PDSC

Taking Input from User

```
input("Enter value")
```

The value returned by this function is always a **string**.

Functions

a block of code which only runs when it is called.

Syntax:

```
def my_function():
  print("Hello from a function")

my_function() #calling the function
```

Built-in functions: print(), type(), str(), sorted(), min(), max()

Lists

used to store multiple items in a single variable

Example:

```
mylist= ["apple", "pear", "mango"]
print(mylist)
```

Other collection data types:

- Tuple: a collection which is ordered and unchangeable. Allows duplicate members.
- Set: a collection which is unordered, unchangeable, and unindexed. No duplicate members.
- Dictionary: a collection which is ordered and changeable. No duplicate members.

Pyautogui

Pyautogui

```python
import pyautogui

# Move the mouse to the specified coordinates (x, y)
pyautogui.moveTo(100, 100, duration=1)
```

Pyautogui

```python
import pyautogui

# Click the mouse at the current location
pyautogui.click()

# Click the mouse at the specified coordinates (x, y)
pyautogui.click(200, 200)
```

Pyautogui

```
import pyautogui

# Type a string of text
pyautogui.write('Hello, world!', interval=0.1)
```

Pyautogui

```python
import pyautogui

# Press a single key
pyautogui.press('enter')

# Press multiple keys
pyautogui.hotkey('ctrl', 'c')  # Copy
pyautogui.hotkey('ctrl', 'v')  # Paste
```

Pyautogui

```
import pyautogui

# Click and drag the mouse from one position to another
pyautogui.moveTo(100, 100)
pyautogui.dragTo(200, 200, duration=1)
```

Pyautogui

```
import pyautogui

# Scroll the mouse up or down
pyautogui.scroll(300)   # Scroll up
pyautogui.scroll(-300)  # Scroll down
```

Pyautogui

```
import pyautogui

# Display an alert message box
pyautogui.alert('This is an alert box.')
```

Pyautogui

```python
import pyautogui

# Move the mouse to the specified coordinates (x, y)
pyautogui.moveTo(100, 100, duration=1)
```

Pyautogui

```python
import pyautogui

# Move the mouse to the specified coordinates (x, y)
pyautogui.moveTo(100, 100, duration=1)
```

Pyautogui

```python
import pyautogui

# Move the mouse to the specified coordinates (x, y)
pyautogui.moveTo(100, 100, duration=1)
```