



PYTHON FUNDAMENTAL MASTERCLASS

KICKSTART TO LEARNING PYTHON
AND
STEP BY STEP SYLLABUS

INTRODUCTION AND BASICS

- Introduction
- Interpreter vs. Compiler
- Understanding and Getting to Know Python
- More on Variables and Strings String Formatting
- Displaying Numbers, Variables and Strings

```
response = requests.get(url)

# Print the status code (if you want)
print(response.status_code)

# Print the status code (if you want)
print(f"Status: {response.status_code}")

# Use BeautifulSoup to parse the response
soup = BeautifulSoup(response.content, "html.parser")

# Find all images in the soup
images = soup.find_all("img", attrs={"alt": "image"})

# Print the number of images found
print(len(images))
```

PROGRAM FLOW CONTROL

- Conditions with IF ELIF ELSE
- FOR Loops
- Understanding CONTINUE and BREAK
- Augmented Assignments
- WHILE Loops
- Nesting Conditions and Loops

```
requests.get(url)

ing response.status_code (if you g
onse.status_code != 200:
nt(f"Status: {response.status_code

nt(f"Status: {response.status_code

BeautifulSoup to parse the respons
BeautifulSoup(response.content, "ht

ing Post images in the soup
= soup.find_all("img", attrs={"alt

Loading images
= 0
images:
```


UNDERSTANDING SEQUENCES

- Lists in Python
- Understanding Iterators
- Using Ranges
- Ordered Sets Using Tuples

```
response = requests.get(url)

# Print the status code (if you want)
print(response.status_code)

# Print the status code (if you want)
print(f"Status: {response.status_code}")

# Use BeautifulSoup to parse the response
soup = BeautifulSoup(response.content, "html.parser")

# Find all post images in the soup
images = soup.find_all("img", attrs={"alt": "Post image"})

# Print the number of images found
print(f"Found {len(images)} images")
```

UNDERSTANDING MAPPINGS

- Binary and Hex Numbers in Python
- Dictionaries and More
- Sets in Python

```
response = requests.get(url)

# Check response status code (if you get a non-200 status code)
if response.status_code != 200:
    print(f"Status: {response.status_code}")

# Parse the response with BeautifulSoup
soup = BeautifulSoup(response.content, "html.parser")

# Find all images in the soup
images = soup.find_all("img", attrs={"alt": "image"})

# Print the number of images found
print(f"Number of images: {len(images)}")
```


HANDLING FILES, INPUT AND OUTPUT

- Reading and Writing Text Files
- Appending to Files
- Writing Binary Files Manually
- Using PICKLE to Write Binary Files
- Shelves
- Manipulating Data with Shelves
- Updating With Shelves

```
response = requests.get(url)
# Load from the website
print(response.status_code) # if you get
if response.status_code != 200:
    print(f"Status: {response.status_code}")

# Using BeautifulSoup to parse the response
soup = BeautifulSoup(response.content, "html.parser")

# Find all post images in the soup
images = soup.find_all("img", attrs={"alt": "image"})

# Loading images
images = []
```

HANDLING FILES AND I/O – P1

- Modules and Import
- Standard Python Library
- Time and Date in Python
- Timezones and PYTZ
- Checking Path
- Functions in Python
- Scope in Functions
- Global Variables

```
response = requests.get(url)
# Check response status code (if you get a non-200 status code)
if response.status_code != 200:
    print(f"Status: {response.status_code}")

# Parse the response using BeautifulSoup to parse the response
soup = BeautifulSoup(response.content, "html.parser")

# Find all post images in the soup
images = soup.find_all("img", attrs={"alt": "image"})

# Print the number of images found
print(f"Number of images: {len(images)}")
```


HANDLING FILES AND I/O – P2

- Importing Techniques
- Underscores in Python
- Namespaces
- Recursion
- Nonlocal
- LEGB

```
response = requests.get(url)

# Print the status code (if you want)
if response.status_code != 200:
    print(f"Status: {response.status_code}")

# Parse the response with BeautifulSoup
soup = BeautifulSoup(response.content, "html.parser")

# Find all image tags in the soup
images = soup.find_all("img", attrs={"alt": "image"})

# Print the number of images found
print(f"Number of images: {len(images)}")
```


EXCEPTIONS

- Reviews
- Handling Exceptions
- Raising Exceptions
- Customising Exceptions

```
response = requests.get(url)

# Check response.status_code (if you get a non-200 status code)
if response.status_code != 200:
    print(f"Status: {response.status_code}")

# Parse the response using BeautifulSoup to parse the response
soup = BeautifulSoup(response.content, "html.parser")

# Find all Post images in the soup
images = soup.find_all("img", attrs={"alt": "Post image"})

# Print the number of images found
print(f"Number of images: {len(images)}")
```

OBJECT ORIENTED PROGRAMMING – P1

- OOP and Classes
- Instances, Constructors, Self and More
- Class Attributes
- Methods
- Non Public and Mangling
- DocString and Raw Literals
- Compile Time
- Getters and Properties

```
...load from the websi  
e = requests.get(url)  
  
ing response.status_code (if you g  
onse.status_code != 200:  
nt(f"Status: {response.status_code  
  
nt(f"Status: {response.status_code  
  
BeautifulSoup to parse the respons  
BeautifulSoup(response.content, "ht  
  
ing Post images in the soup  
= soup.find_all("img", attrs={"alt"  
  
Loading images  
= 0  
images:
```


OBJECT ORIENTED PROGRAMMING – P2

- Getters and Setters
- Encapsulation
- Inheritance
- Subclasses and Overloading
- Calling Super Methods
- Overriding Methods
- Polymorphism
- Duck Test

```
requests.get(url)

ing response.status_code (if you g
onse.status_code != 200:
nt(f"Status: {response.status_code

nt(f"Status: {response.status_code

BeautifulSoup to parse the respon
BeautifulSoup(response.content, "ht

ing Post images in the soup
= soup.find_all("img", attrs={"alt

Loading images
= 0
images:
```

OBJECT ORIENTED PROGRAMMING – P3

- Composition
- Aggregation
- Delegation

```
...load from the websi  
e = requests.get(url)  
  
ing response.status_code (if you g  
onse.status_code != 200:  
nt(f"Status: {response.status_code  
  
nt(f"Status: {response.status_code  
  
BeautifulSoup to parse the respon  
BeautifulSoup(response.content, "ht  
  
ing Post images in the soup  
= soup.find_all("img", attrs={"alt  
  
Loading images  
= 0  
images:
```


NOTES

WEBSITES

- www.python.org
- www.realpython.com

TEXTS

- Python Basics – ISBN: 9781775093329

SUGGESTED TIME

- 60 Hours

PREREQUISITES

- None

DOCUMENT HISTORY

Author	Version	Revision	Date / Time	Department	Validity
Mehdi Shokri	1.0.0		11-05-2023	Development	3 Months
Mehdi Shokri		1.0.1	12-05-2023	Development	3 Months

