Web Development

Technical Summer School 2019, IIT Bombay – Parth Patil

Part 4 – Introduction to Python,

Generating HTML Dynamically

Static Content

- Everything we've done till now
- HTML on server stays the same – in a file
- Too many pages for similar content
- Can't show user specific content
- Too much to write!

Dynamic Content

- Generated on the fly
- Can be different depending request parameters like
  - URL
  - User
  - Browser/Operating System
  - Country
- Avoid repetition with **templates**
- Involves server-side processing

Python

- One of the most popular
- Strongly, dynamically typed language
- Interpreted by the line

Python – Hello World!

```
$ python --version
Python 3.6.5

# hello.py
print("Hello World!")

$ python hello.py
Hello World!
```

Interactive - CalcAgain

```
def calculate(a, b):
    """Print the sum and product of two numbers."""
    print("Sum is ", a + b)
    print("Product is ", a * b)

# Get inputs
num1 = float(input("Input 1 - "))
num2 = float(input("Input 2 - "))

# Print sum and product with a Thank You message
calculate(num1, num2)
print("Thank You!")
```

Interactive - CalcAgain

```
def calculate(a, b):
    """Print the sum and product of two numbers."""
    sum = "Sum is " + str(a + b)
    prod = "Product is " + str(a * b)
    return sum + "\n" + prod

# Get inputs
num1 = float(input("Input 1 - "))
num2 = float(input("Input 2 - "))

# Print sum and product with a Thank You message
print(calculate(num1, num2))
print("Thank You!")
```

Interactive - CalcAgainWrite

```
def calculate(a, b):
    """Print the sum and product of two numbers."""
    sum = "Sum is " + str(a + b)
    prod = "Product is " + str(a * b)
    return sum + "\n" + prod
# Get inputs
num1 = float(input("Input 1 - "))
num2 = float(input("Input 2 - "))
# Write sum and product with a Thank You message
with open('myfile.txt', 'w') as file:
    file.write(calculate(num1, num2))
    file.write("\n")
    file.write("Thank You")
```

Interactive - Arrays

```
my_array = [2, 5, 'haha', 89]

print(my_array[1])


names = ['Bombay', 'Madras', 'Kanpur', 'Delhi', 'Kharagpur']

print('Here is a list of some IITs:')

for name in names:
    print('IIT', name)
```

Interactive - Loops

```
# Print 0 to 9
for i in range(0, 10):
    print(i)

# Print 0 to 9
current = 0
while current < 10:
    print(current)
    current += 1

# Print 0 to 9 skipping alternately
for i in range(0, 10, 2):
    print(i)
```

Dictionary

```
my_dict = {
    'name': 'Jon Snow',
    'house': 'Stark?',
    'home': 'Winterfell'
}

print('My name is', my_dict['name'])
print('I live in', my_dict['home'])
```

Interactive – Array of Dicts

```
everyone = [
    {'name': 'Daenerys', 'house': 'Targaryen', 'home': 'Dragonstone?'},
    {'name': 'Arya', 'house': 'Stark', 'home': 'Winterfell'},
    {'name': 'Cersei', 'house': 'Lannister', 'home': "King's Landing"},
    {'name': 'Yara', 'house': 'Greyjoy', 'home': 'Iron Islands'}
]
```

http://home.iitb.ac.in/~varunpatil/webdev/got-dict.py

```
for p in everyone:
    print(p['name'], 'of house', p['house'], 'from', p['home'])
```

Comma Separated Values

```
name,house,home
Bran,Stark,Winterfell
Tyrion,Lannister,King's Landing
Theon,Greyjoy,Winterfell
Samwell,Tarly,Horn Hill
```

http://home.iitb.ac.in/~varunpatil/webdev/got-csv.csv

```
import csv

with open('got-csv.csv', 'r') as csvfile:
    dreader = csv.DictReader(csvfile)
    for p in dreader:
        print(p['name'], 'of house', p['house'], 'from', p['home'])
```

Interactive – Dummy HTML

```
def get_html():
    return """
<html>
    <body>
        <h1> This is a heading </h1>
        <p> This is a paragraph </p>
    </body>
</html>
"""

# Write HTML
with open('dummy.html', 'w') as file:
    file.write(get_html())
```

Templates

- Files with HTML and some variables
- Variables replaced with actual data
- Makes editing HTML easier
- Liking CSS/JS

```
<body>
    <h1> {{ heading }} </h1>
    <p> {{ information }} </p>
</body>
```

Serving Dynamically

- Store only the template and data
- Create the “rendered” HTML only when asked for
- Instead of saving the HTML, send it to the client
- Allows changing data (very) frequently
- Can recognize user and generate specific content

Thank You!