



Introduction to Python & Flask

Presented by **Ahmed Eltaher**

TABLE OF CONTENTS

01

Intro To The Summer Training

What will we learn?

02

Python Syntax

How to code with python?

03

Problem Solving With Python

Building a basic calculator program

04

HTTP Requests And Responses

How the web works?

05

Flask Installation

Building a basic website using flask

01

Introduction To The Summer Training



Introduction To Summer Training

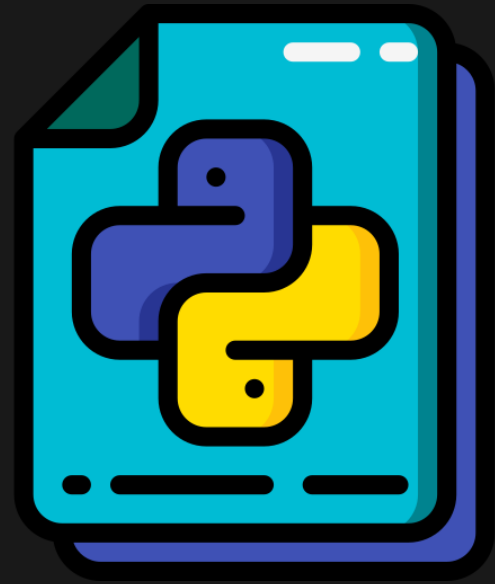
What will we learn?

- Learning how to build web applications using flask framework.
- What is the most common web applications vulnerabilities.
- How securely develop our website to avoid these vulnerabilities.
- Practical project at the end of the training .



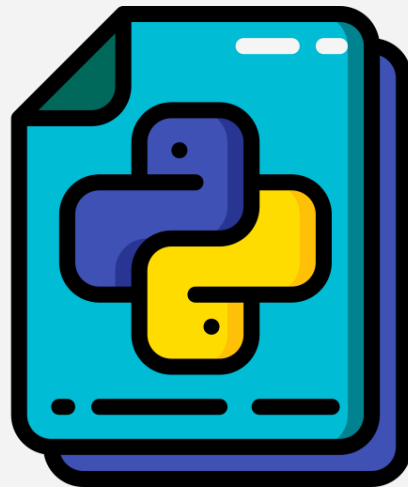
02

Python Syntax



What is Python?

- Python is an easy to learn, powerful programming language.
- It has efficient high-level data structures and a simple approach to OOP.
- Python is great for scripting and rapid application development in many areas on most platforms.



How to print a string using Python?

Using the "print()" function

Variables:-

- We identify string by putting "" .
- In the 3rd line we did an array, and we will see how we can print it after a few minutes! .

```
print("Hello world")  
#by running this it will print "hello world" so easily
```

```
name="CYBERUS"  
id=1337  
cars = ["BMW","mercedes","toyota"]
```

Input:

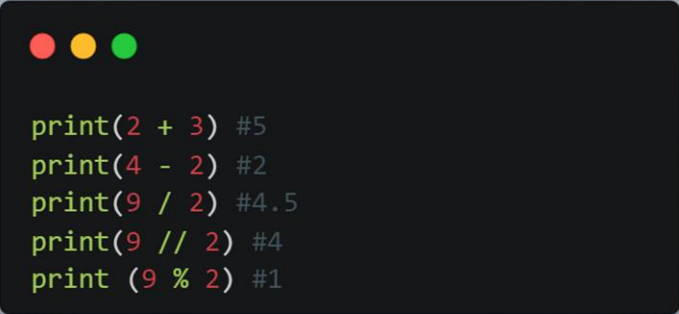
- Input function enables us to enter data.
- Before entering an integer value, we should specify it with "int" as you can see.
- Print(type()): it prints the type of the input so it's going to print "str" for the 3rd line and "int" for the 4th line.



```
name = input("Enter Your Name: ")  
age = int(input("Enter Your Age: "))  
print(type(name), name)  
print(type(age), age)
```


Arithmetic Operations:

The difference between the "/" operation and the "//" operation is when we use / it means that we are just dividing 2 numbers but by using // it means that it divides the 1st number by the 2nd number and returns the largest integer that is less than or equal to the result.



```
print(2 + 3) #5
print(4 - 2) #2
print(9 / 2) #4.5
print(9 // 2) #4
print(9 % 2) #1
```

If condition:

In python the else if is written as : elif
and the not operation is written and
`not` different than most
programming languages which is
written like `!`



```
age = int(input("Enter Your Age: "))
if age > 20:
    print("You Are a Grown person Now go Study Physics")
elif age >= 18:
    print("You Are an Adult go Play Valorant")
elif age == 16:
    print("You Are in mid situation Now Go Delete TikTok")
else :
    print("You Still a child go Play some Minecraft")
```

Loops:

For the while loop :

- It will print numbers from 1 to 9.

- For the 1st for loop:

It will print numbers 0 to 9.

- For the 2nd for loop:

It will print only the even numbers from 2 to 9.

- For the 3rd for loop:

Remember the previous array? we are printing it now!



```
x = 1
while x < 10:
    print(x)
    x += 1

for i in range(10):
    print(i)

for i in range(2, 10, 2):
    print(i)

cars = ["BMW", "mercedes", "toyota"]
for car in cars:
    print("My Favorite Car is: " + car)
```

Functions:

1st function:

It takes a double number and multiply it by 2.

2nd function:

it takes a string that will be used to ask the user for input and it reads an int input from the user and returns the int value entered and If the user enters something that can't be converted to an int the function returns 0.

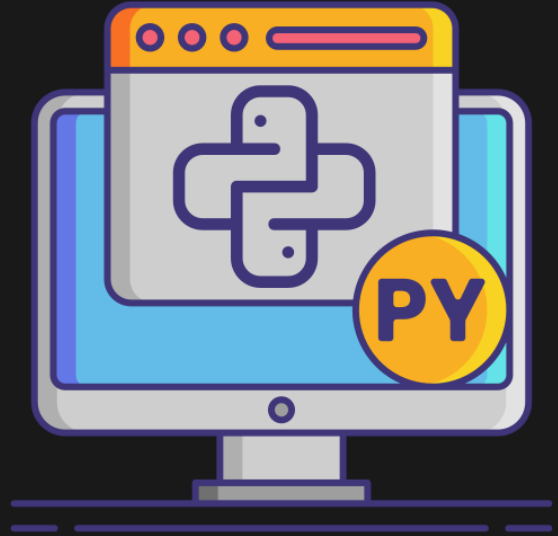
3rd function:

So simple function , it prints 2 strings.

```
def double(num):  
    return num * 2  
  
def getNumber(message):  
    try:  
        return int(input(message))  
    except:  
        return 0  
  
def printHelp():  
    print("Welcome To Our App")  
    print("Developed by Cyberus")  
  
print(double(2))  
print(getNumber("Enter Your Age: "))  
print(getNumber(message="Enter Your Age: "))  
printHelp()
```

03

Problem Solving With Python



Problem Description:

- You are tasked to create a simple calculator program in Python.
- The program should offer basic arithmetic operations such as addition, subtraction, multiplication, and division.
- Ensure the program handles division by zero gracefully.
- Users should be able to input two numbers and choose the desired operation.
- The calculator should display the result and allow users to perform more calculations or exit the program.

Problem Solving With Python

Now ,let's build our first python project!

1- create functions for the operations
to make it easier.

2- create a function for the calculator.

3- let user enter the operation.



```
def add(x, y):  
    return x + y  
  
def subtract(x, y):  
    return x - y  
  
def multiply(x, y):  
    return x * y  
  
def divide(x, y):  
    if y != 0:  
        return x / y  
    else:  
        return "Can't be divided"  
  
def calculator():  
    print("Simple Calculator")  
    print("Operations:")  
    print("1. + ")  
    print("2. - ")  
    print("3. * ")  
    print("4. / ")  
    print("5. Exit")  
  
    while True:  
        choice = input("Enter operation number :")
```

Problem Solving With Python

4- After letting user enter the operation let's make him enter the 2 numbers.

5- Using if condition we can know the result so easily and we are done!

```
if choice in ('1', '2', '3', '4'):
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))

    if choice == '1':
        print("Result:", add(num1, num2))
    elif choice == '2':
        print("Result:", subtract(num1, num2))
    elif choice == '3':
        print("Result:", multiply(num1, num2))
    elif choice == '4':
        print("Result:", divide(num1, num2))
elif choice == '5':
    print("Exiting the calculator.")
    break
else:
    print("Invalid input. Please try again.")

if __name__ == "__main__":
    calculator()
```

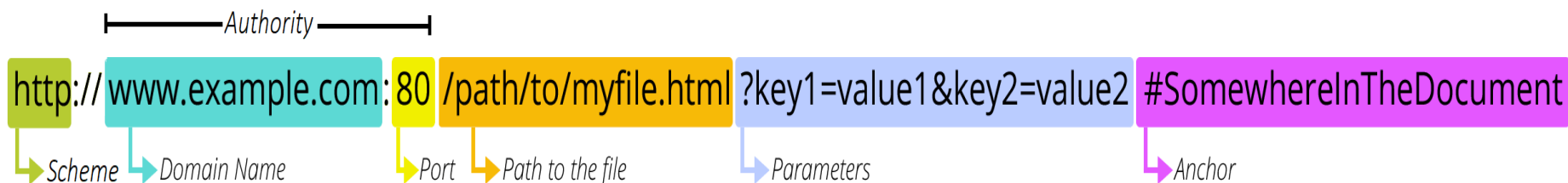

04

HTTP Requests And Responses



What is URL ?

URL (Uniform Resource Locator) is predominantly an instruction on how to access a resource on the internet.



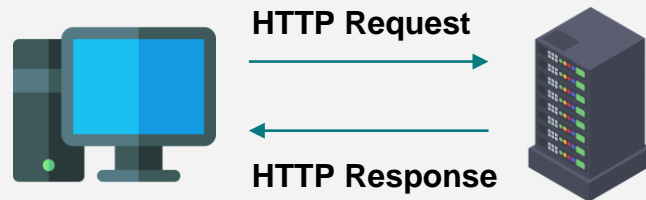
What is HTTP?

HTTP is a protocol that allows communication between clients (e.g., web browsers) and servers (e.g., web servers).

HTTP is responsible for retrieving web resources like HTML pages, images, videos, etc.

Client-Server Model :

A client (e.g., web browser) sends requests, and the server (e.g., web server) responds with the requested data.



HTTP Requests

HTTP request is made by the client to the server to retrieve a resource.

Basic structure of an HTTP request



```
GET /path/to/resource HTTP/1.1  
Host: example.com  
User-Agent: Mozilla/5.0 Firefox/87.0
```

Basic structure of an HTTP request

“GET” is the request method

“/path/to/resource” is the URL of the resource

“Host” is the website domain

“User-Agent” identifies the client making request



```
GET /path/to/resource HTTP/1.1
```

```
Host: example.com
```

```
User-Agent: Mozilla/5.0 Firefox/87.0
```

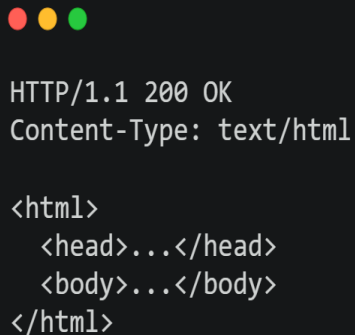
HTTP Methods:

- GET:
Used to request data from a specified resource.
- POST:
Used to submit data to be processed to a specified resource.
- PUT:
Used to update or replace existing data on the server with new data
- DELETE:
Used to request the removal of a specified resource from the server.



HTTP Responses:

When the server receives an HTTP request, it responds with an HTTP response.



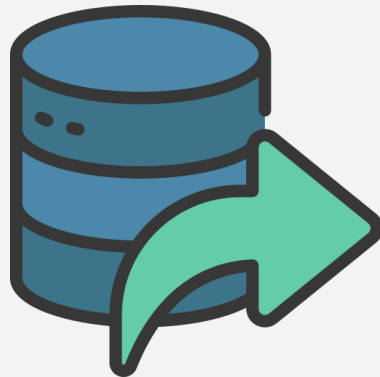
```
HTTP/1.1 200 OK  
Content-Type: text/html  
  
<html>  
  <head>...</head>  
  <body>...</body>  
</html>
```

HTTP Status Codes:

- 200 OK
- 404 Not Found
- 500 Internal Server Error

HTTP Headers:

HTTP requests and responses contain headers, which provide additional information about the data being sent.



05

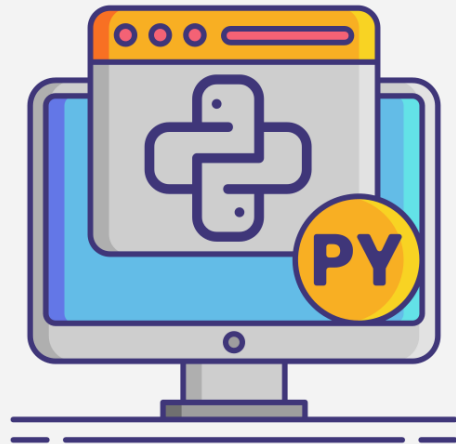
Flask Framework



Flask

Why flask ?

- Flask is a web framework written in Python that simplifies web application development.
- Flask is lightweight, flexible, and easy to use, making it an excellent choice for small to medium-sized projects.
- It will help us to build our web applications fast as we will then focus on the security aspects in them.



How to install flask ?

- ◆ Open a Terminal or Command Prompt
- ◆ Install flask using pip:



```
pip install flask
```

- ◆ You can verify by using this command :



```
flask --version
```

Developing a very simple “Hello World” website :

A basic Flask app that responds with "Hello, World!" when accessed through the root URL.

```
from flask import Flask

app = Flask(__name__)

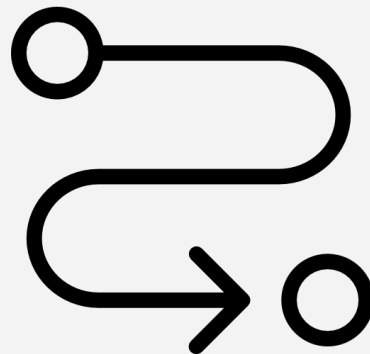
@app.route('/')
def hello():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run()
```

Introduction to routing in flask:

Routing is the process of mapping URLs to view functions in the application.

Flask uses the `@app.route()` decorator to associate a URL with a specific view function.



Developing A Website Using Flask

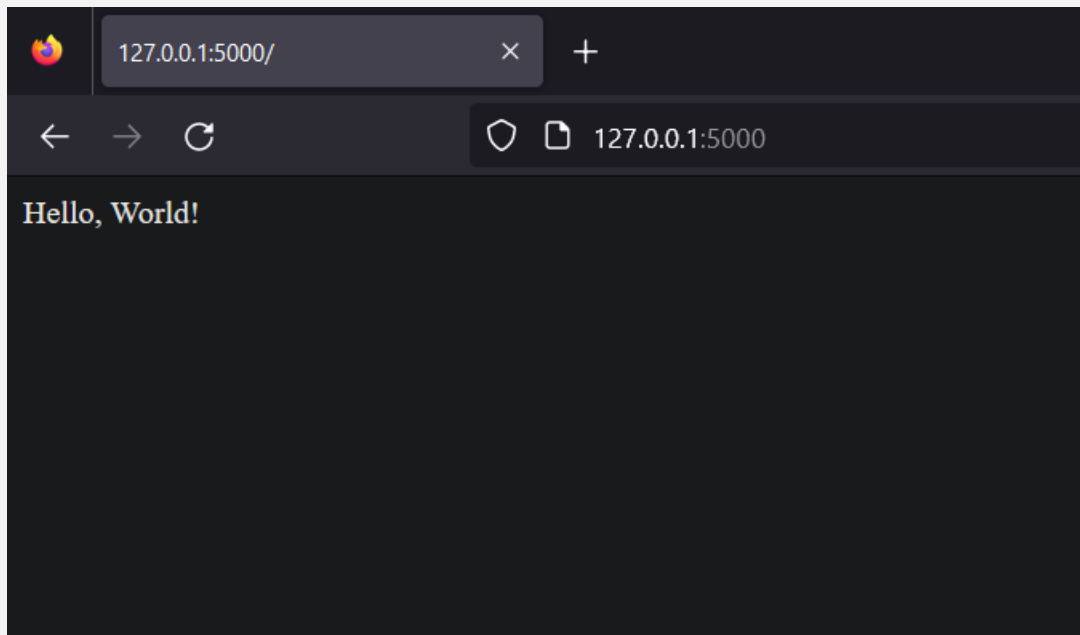
```
from flask import Flask, request
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

@app.route('/echo', methods=['POST'])
def echo():
    app.logger.debug(request.json)
    return f'Hello {request.json["name"]}'

if __name__ == '__main__':
    app.run(debug=True)
```

By visiting our website :



Problem Description:

- build a Flask application for managing student records.
- Your mission is to create a simple web application that allows adding new students to the system
- and retrieving a list of all students stored in an in-memory Array.

Problem Solving With Flask

```
from flask import Flask, request, jsonify

app = Flask(__name__)

# Sample database (in-memory)
students = []

@app.route('/add_student', methods=['POST'])
def add_student():
    data = request.get_json()
    if data and 'name' in data and 'age' in data:
        name = data['name']
        age = data['age']
        students.append({'name': name, 'age': age})
        return jsonify({'message': 'Student added successfully'}),
    201 else:
        return jsonify({'error': 'Invalid data'}), 400

@app.route('/students', methods=['GET'])
def get_students():
    return jsonify(students)

if __name__ == '__main__':
    app.run(debug=True)
```

What is SQLite ?

- SQLite is a lightweight, serverless, self-contained, and open-source database system.
- It is widely used for small to medium-sized applications and is integrated into many programming languages.



Why use SQLite with Flask?

- Flask is a micro web framework for Python, and SQLite is an ideal choice for small to medium-sized Flask projects.
- SQLite offers a simple setup and requires minimal configuration.
- It's perfect for development and testing, as well as prototyping.



We will use a module called “sqlite3” it is a part of Python standard library and comes preinstalled with Python

Creating our database file:

```
def connect_to_database(name='database.db'):
    import sqlite3
    return sqlite3.connect(name)

def init_db(connection):
    cursor = connection.cursor()

    cursor.execute('''
        CREATE TABLE IF NOT EXISTS users (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            username TEXT NOT NULL UNIQUE,
            password TEXT NOT NULL
        )
    ''')

    connection.commit()
```

Let's add database to our website:

We should now import the database file that we have created:

```
import db
```

Then we will modify the main of our program to be :

```
if __name__ == '__main__':  
    connection = db.connect_to_database()  
    db.init_db(connection)  
    app.run(debug=True)
```



Thank You

Presented by Ahmed Eltaher

RESOURCES

<https://www.python.org/>

<https://flask.palletsprojects.com/>

<https://en.wikipedia.org/wiki/HTTP>

<https://www.sqlite.org/index.html>

