

# Web Scraping & Web Services



# Hello!

I am Eslam Ahmed

I am a software engineer.

You can find me at [jeksogsa@gmail.com](mailto:jeksogsa@gmail.com)



# Hello!

I am Eman Ehab

I am a ML research engineer.

You can find me at  
[emanehab.ieee@gmail.com](mailto:emanehab.ieee@gmail.com)



# Agenda

- What is Network Topologies
- What is Internet and Web Servers
- HTTP Request/Response Cycle
- HTML
- CSS
- Scrapping Concept
- **BeautifulSoap** Library
- REST API Web Services
- Work with JSON



# Agenda

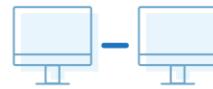
- What is Network Topologies
- What is Internet and Web Servers
- HTTP Request/Response Cycle
- HTML
- CSS
- Scrapping Concept
- BeautifulSoup Library
- REST API Web Services
- Work with JSON



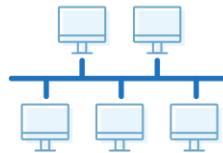
# What is Network Topologies

## Network Topology Types

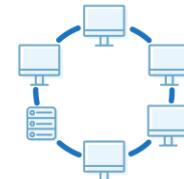
1 Point to point



2 Bus



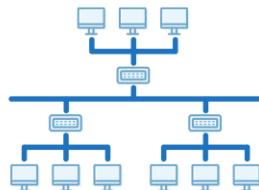
3 Ring



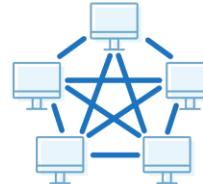
4 Star



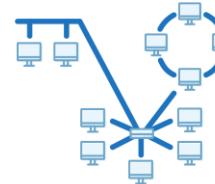
5 Tree



6 Mesh



7 Hybrid

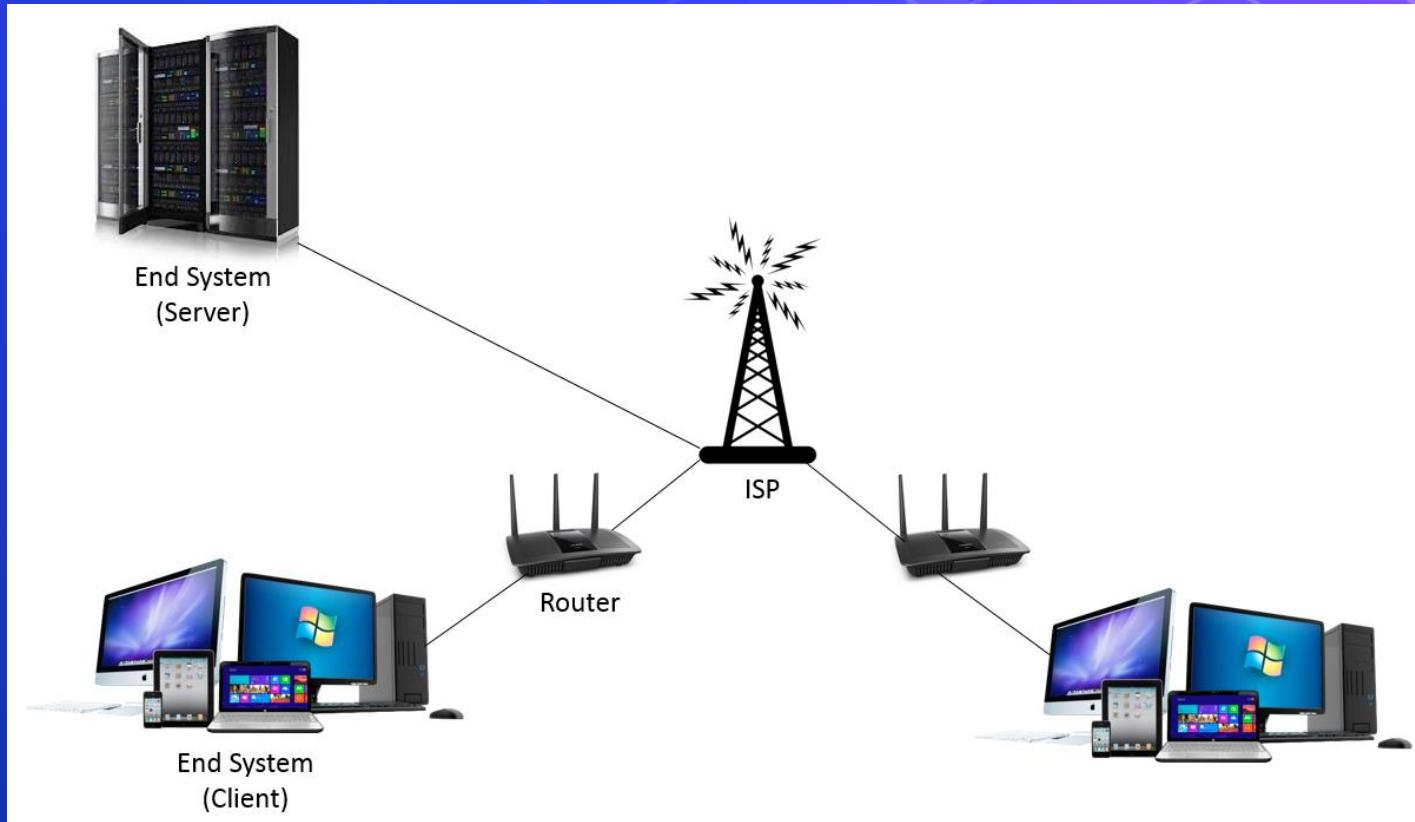


# Agenda

- What is Network Topologies
- **What is Internet and Web Servers**
- HTTP Request/Response Cycle
- HTML
- CSS
- Scrapping Concept
- **BeautifulSoap Library**
- REST API Web Services
- Work with JSON



# What is Internet and Web Servers

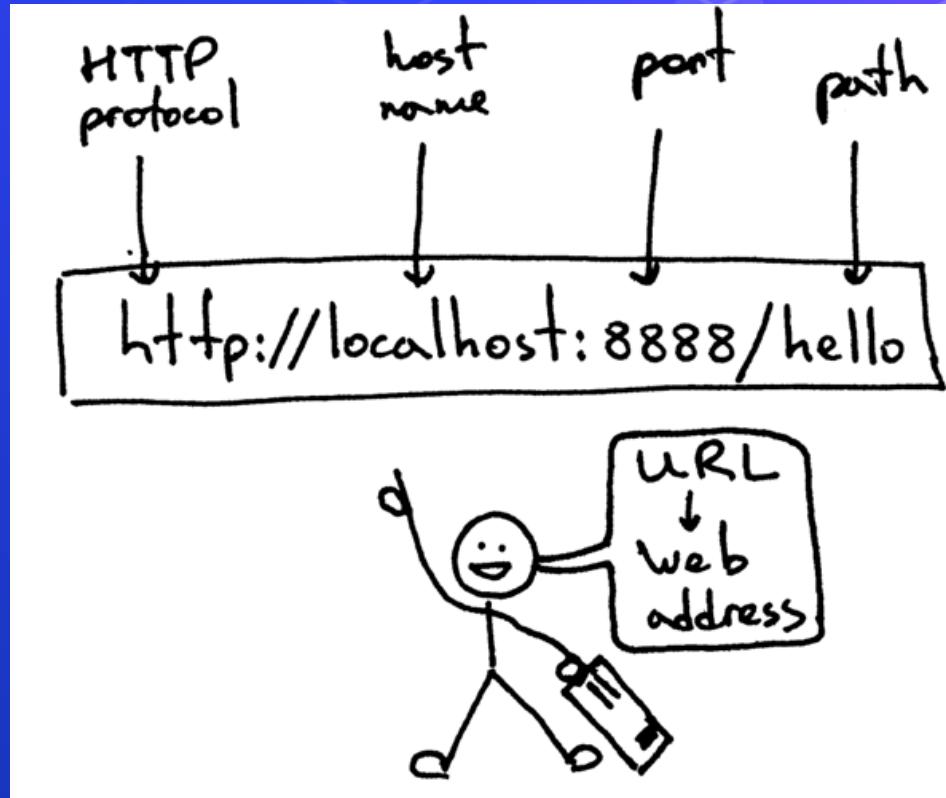


# Agenda

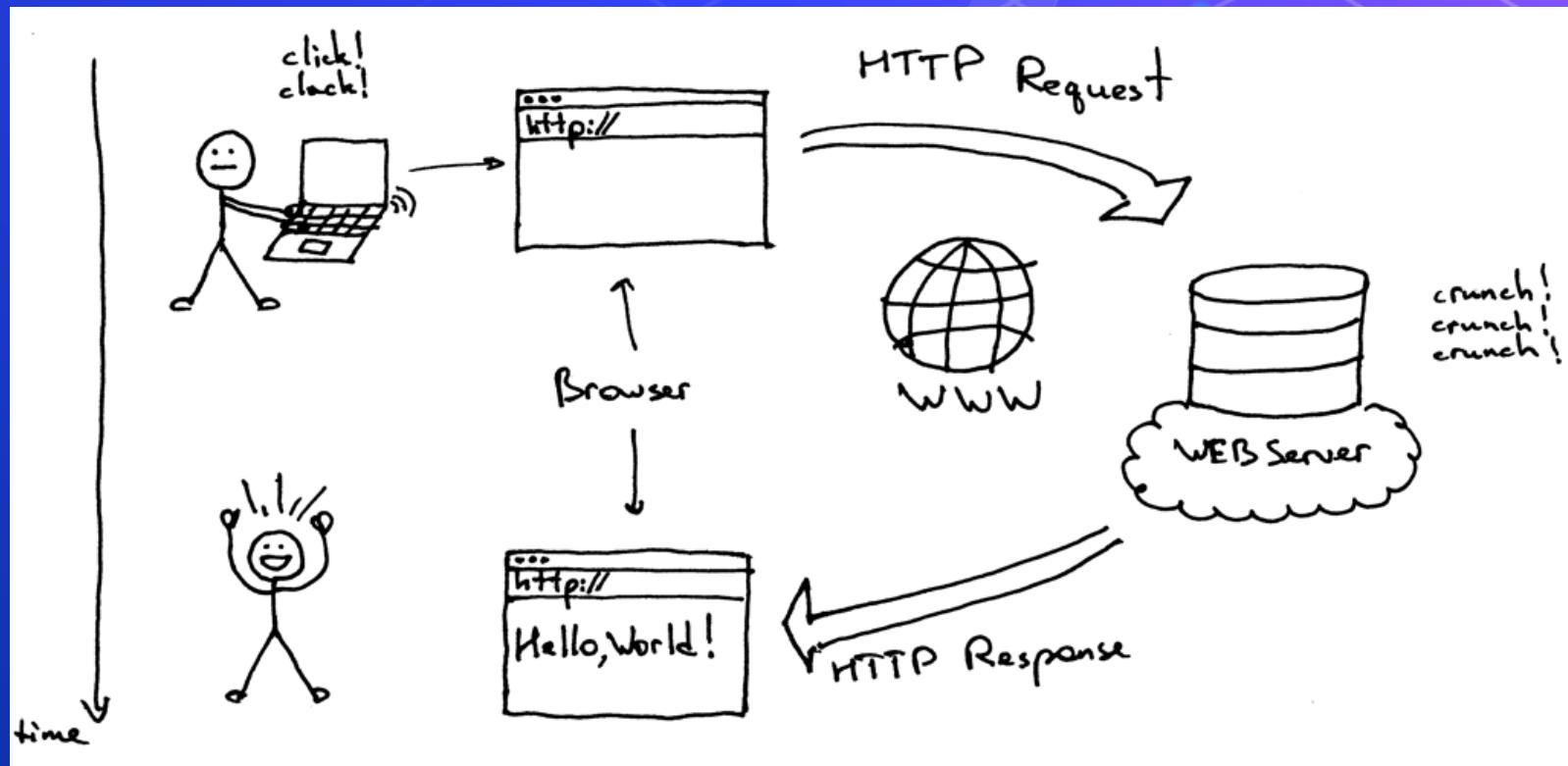
- What is Network Topologies
- What is Internet and Web Servers
- **HTTP Request/Response Cycle**
- HTML
- CSS
- Scrapping Concept
- **BeautifulSoap Library**
- REST API Web Services
- Work with JSON



# HTTP Request/Response Cycle

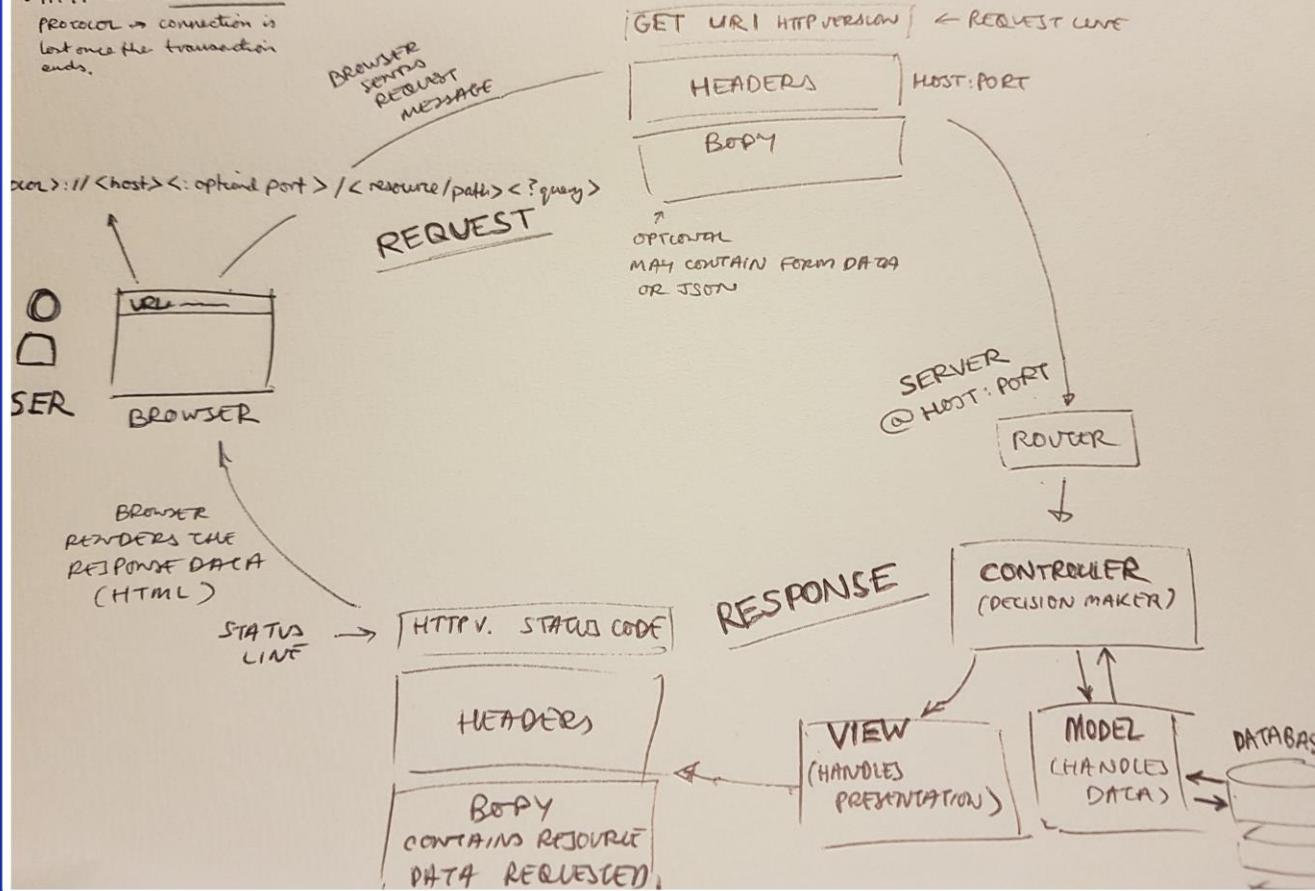


# HTTP Request/Response Cycle



# HTTP REQUEST-RESPONSE CYCLE

- HTTP is a STATELESS protocol → connection is lost once the transaction ends.



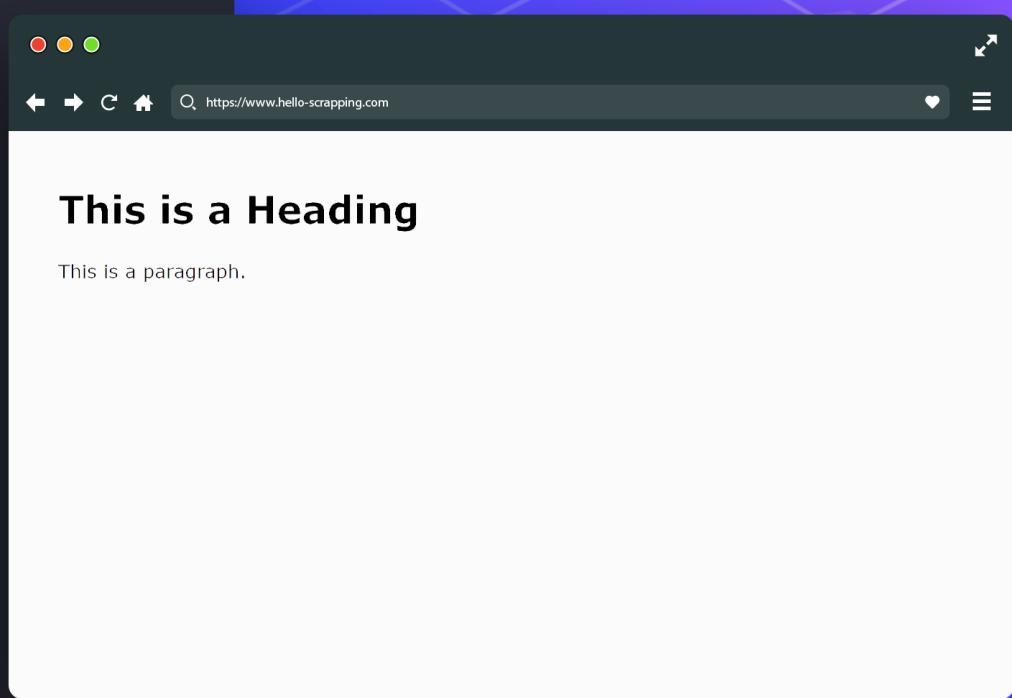
# Agenda

- What is Network Topologies
- What is Internet and Web Servers
- HTTP Request/Response Cycle
- HTML
- CSS
- Scrapping Concept
- **BeautifulSoap** Library
- REST API Web Services
- Work with JSON



# HTML (base snippet code)

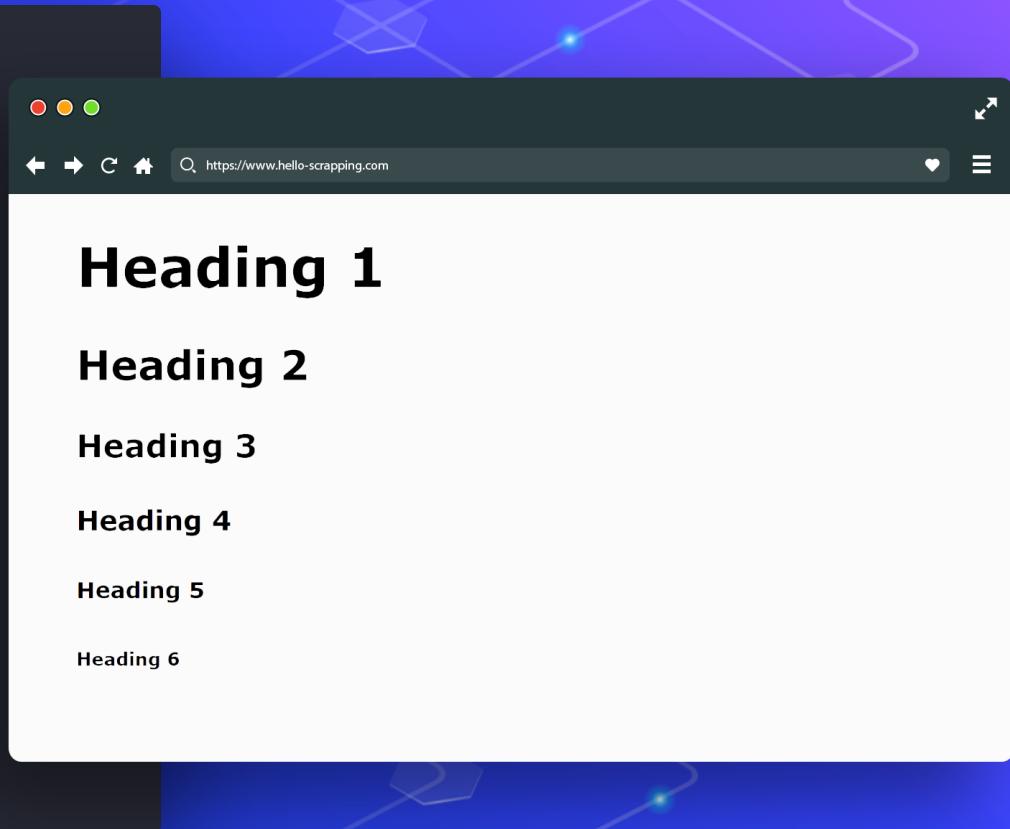
```
1 <!DOCTYPE html>
2 <html>
3
4   <head>
5     <title>Page Title</title>
6   </head>
7
8
9   <body>
10
11     <h1>This is a Heading</h1>
12     <p>This is a paragraph.</p>
13
14   </body>
15 </html>
16
```



# HTML (header)

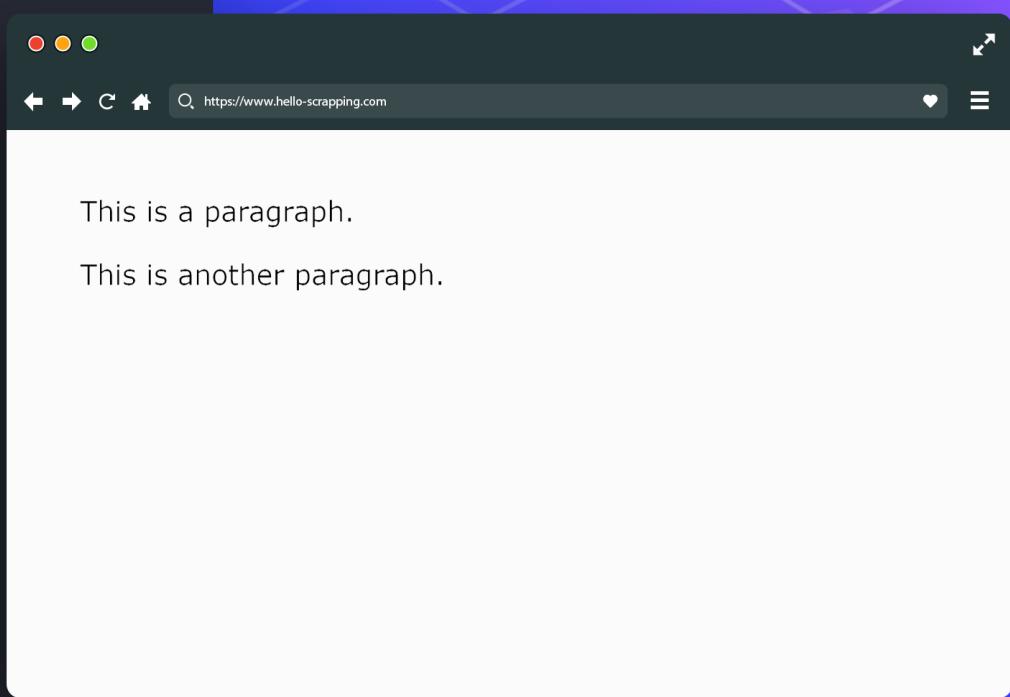


```
1 <!DOCTYPE html>
2 <html>
3
4     <head>
5         <title>Page Title</title>
6     </head>
7
8
9     <body>
10
11         <h1>Heading 1</h1>
12         <h2>Heading 2</h2>
13         <h3>Heading 3</h3>
14         <h4>Heading 4</h4>
15         <h5>Heading 5</h5>
16         <h6>Heading 6</h6>
17
18     </body>
19 </html>
20
```



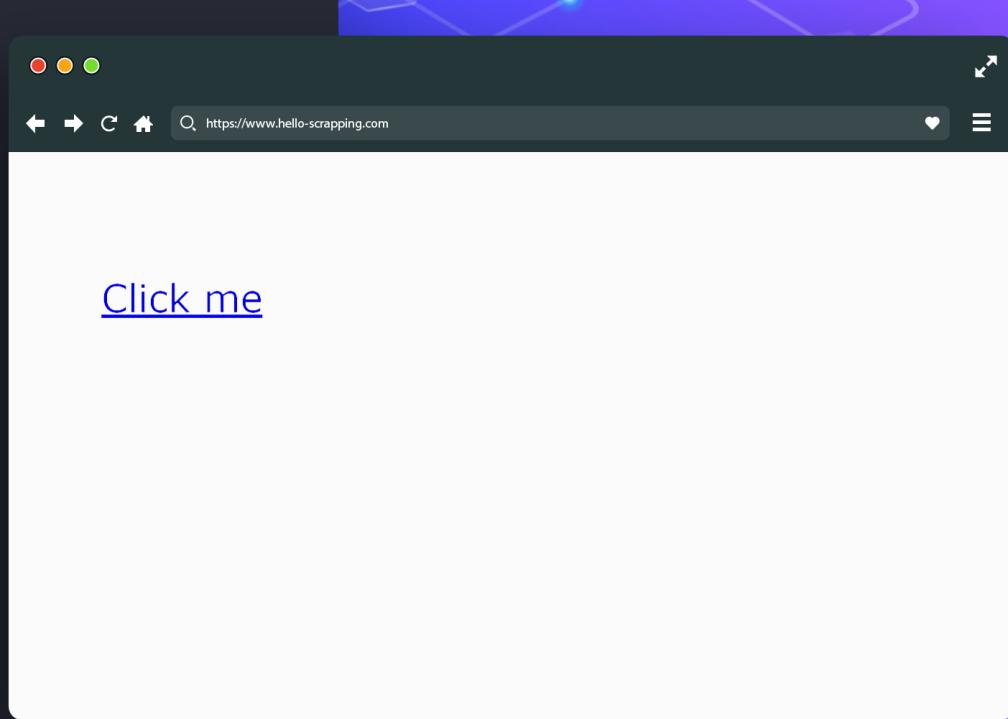
# HTML (paragraph)

```
1 <!DOCTYPE html>
2 <html>
3
4   <head>
5     <title>Page Title</title>
6   </head>
7
8
9   <body>
10
11     <p>This is a paragraph.</p>
12     <p>This is another paragraph.</p>
13
14   </body>
15 </html>
16
```



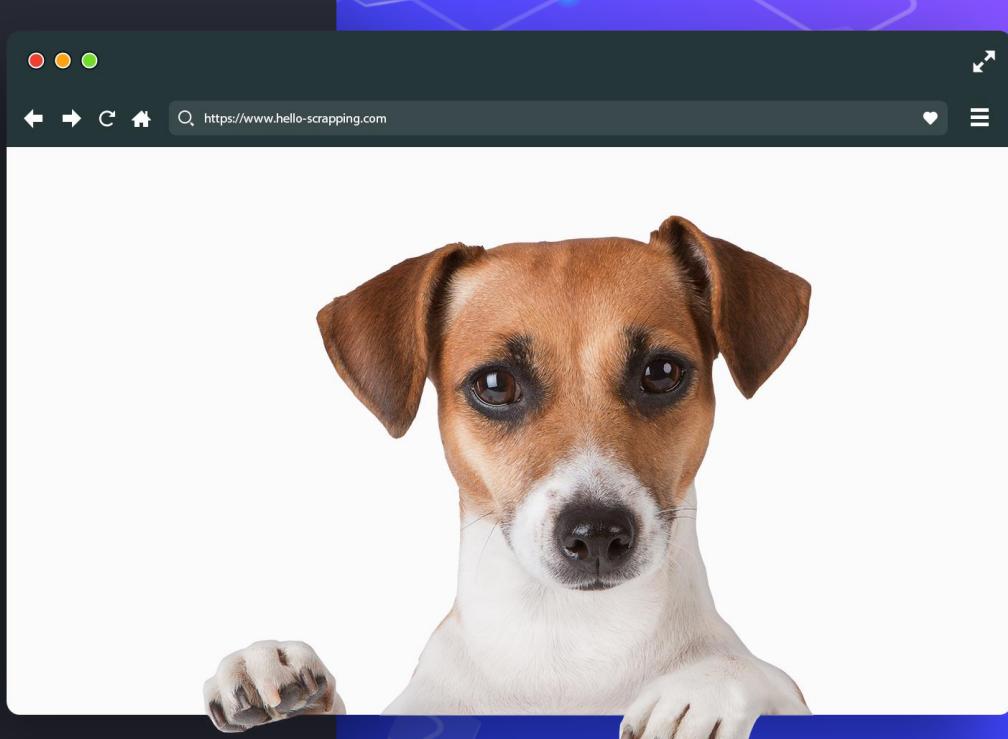
# HTML (link)

```
1 <!DOCTYPE html>
2 <html>
3
4     <head>
5         <title>Page Title</title>
6     </head>
7
8
9     <body>
10
11         <a href="url">Click me</a>
12
13     </body>
14 </html>
15
```



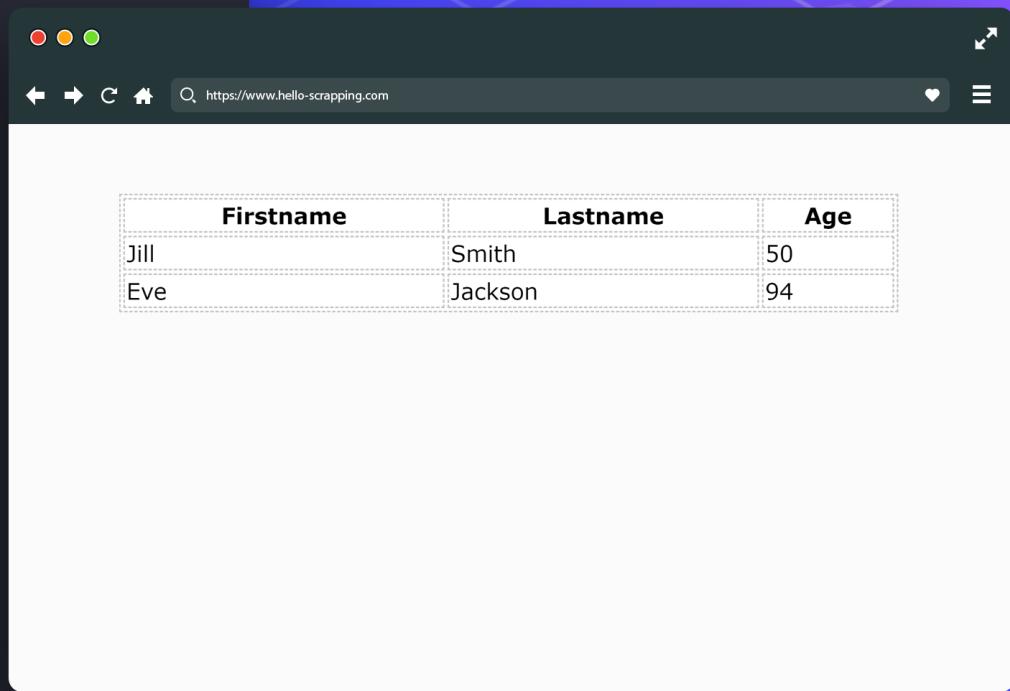
# HTML (image)

```
1 <!DOCTYPE html>
2 <html>
3
4     <head>
5         <title>Page Title</title>
6     </head>
7
8
9     <body>
10
11         
12
13     </body>
14 </html>
15
```



# HTML (table)

```
1 <table style="width:100%">
2   <tr>
3     <th>Firstname</th>
4     <th>Lastname</th>
5     <th>Age</th>
6   </tr>
7   <tr>
8     <td>Jill</td>
9     <td>Smith</td>
10    <td>50</td>
11  </tr>
12  <tr>
13    <td>Eve</td>
14    <td>Jackson</td>
15    <td>94</td>
16  </tr>
17 </table>
18
```

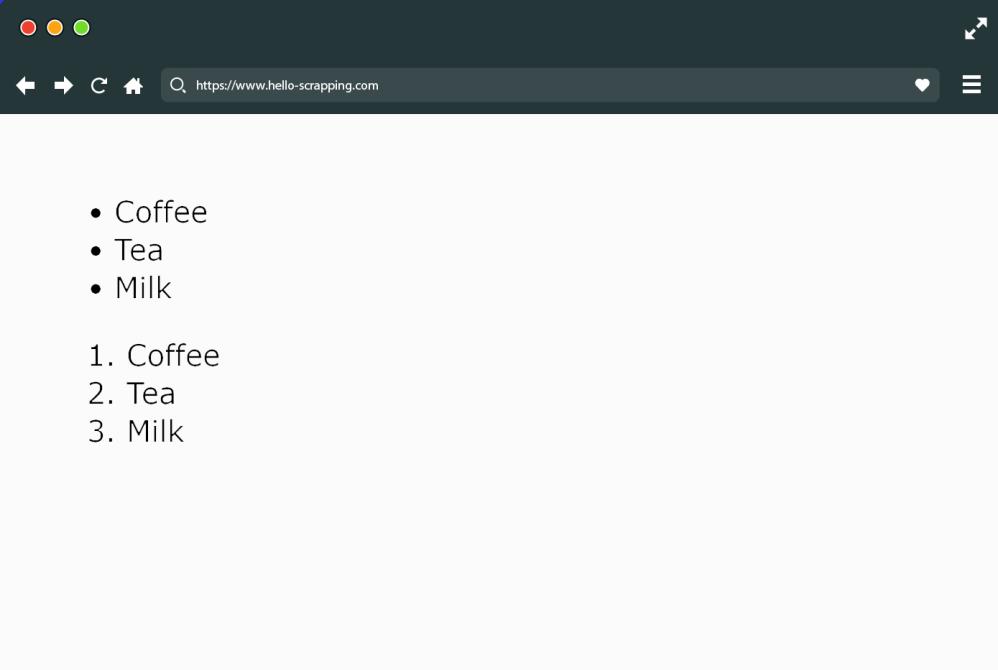


A screenshot of a web browser window displaying a simple HTML table. The browser interface includes a title bar with three dots, a back/forward button, a refresh button, and a search bar containing the URL <https://www.hello-scraping.com>. The main content area shows a table with three columns: Firstname, Lastname, and Age. The table has two rows of data: one row for Jill Smith (age 50) and another for Eve Jackson (age 94). The table is styled with a width of 100% and no border.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94

# HTML (list)

```
1 <ul>
2     <li>Coffee</li>
3     <li>Tea</li>
4     <li>Milk</li>
5 </ul>
6
7 <ol>
8     <li>Coffee</li>
9     <li>Tea</li>
10    <li>Milk</li>
11 </ol>
```



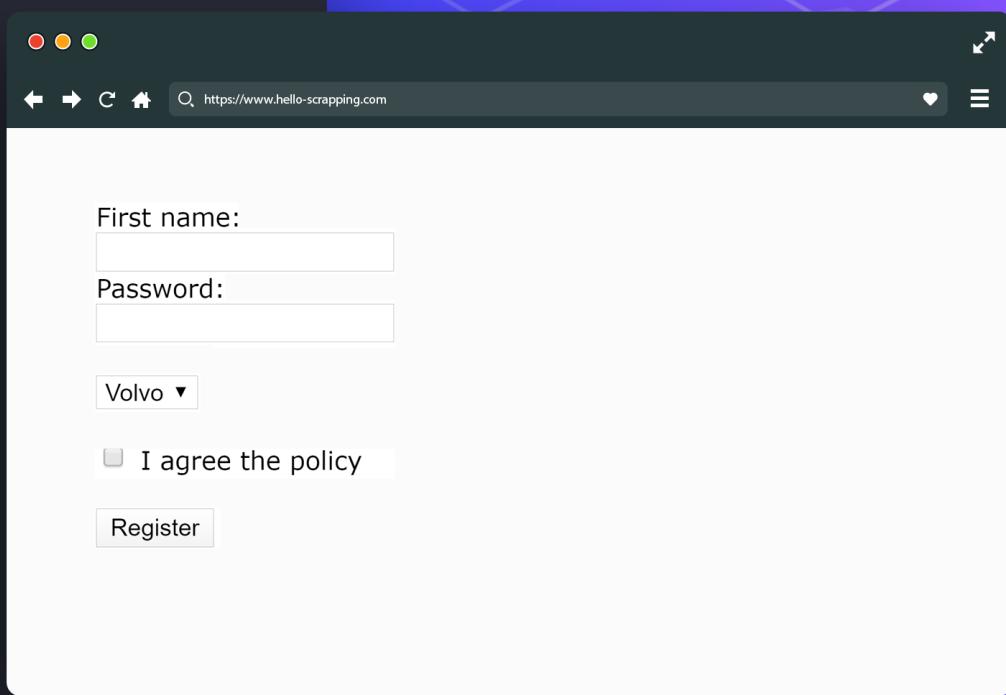
The screenshot shows a web browser window with a dark theme. The address bar displays the URL <https://www.hello-scrappling.com>. The main content area shows two lists:

- Coffee
- Tea
- Milk

1. Coffee
2. Tea
3. Milk

# HTML (form)

```
1 <form>
2
3     First name:
4     <input type="text">
5
6     Password:
7     <input type="text">
8
9     <select>
10        <option value="volvo">Volvo</option>
11        <option value="saab">Saab</option>
12        <option value="fiat">Fiat</option>
13        <option value="audi">Audi</option>
14    </select>
15
16    <input type="checkbox">
17    I agree the policy
18
19    <input type="submit" value="Register">
20
21 </form>
```



The screenshot shows a web browser window with the URL <https://www.hello-scraping.com>. The page displays an HTML form with the following elements:

- A text input field labeled "First name:"
- A text input field labeled "Password:"
- A dropdown menu currently set to "Volvo"
- A checkbox labeled "I agree the policy" which is unchecked
- A submit button labeled "Register"

# Other HTML Elements

- Div
- Span
- Video
- Audio
- Iframe
- Header
- Footer
- Canvas
- ...



# Agenda

- What is Network Topologies
- What is Internet and Web Servers
- HTTP Request/Response Cycle
- HTML
- CSS
- Scrapping Concept
- **BeautifulSoap** Library
- REST API Web Services
- Work with JSON



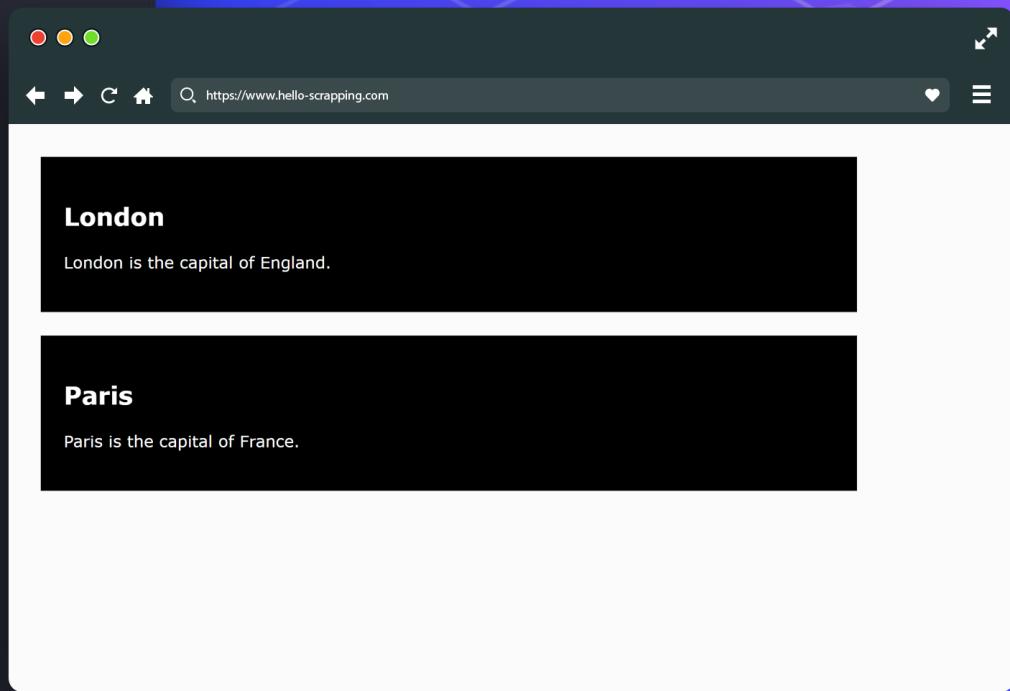
# CSS

- Text & Fonts
- Colors
- Backgrounds
- Borders
- Margin & Padding
- Width & Height
- Gradient
- Shadows
- ...



# Classes and ID

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       .cities
6     {
7       background-color: black;
8       color: white;
9       margin: 20px;
10      padding: 20px;
11    }
12  </style>
13 </head>
14 <body>
15
16 <div class="cities">
17   <h2>London</h2>
18   <p>London is the capital of England.</p>
19 </div>
20
21 <div class="cities">
22   <h2>Paris</h2>
23   <p>Paris is the capital of France.</p>
24 </div>
25
26 </body>
27 </html>
```



# Check Also

- JavaScript
- React & Angular
- Ajax
- Web Sockets
- ...



# Agenda

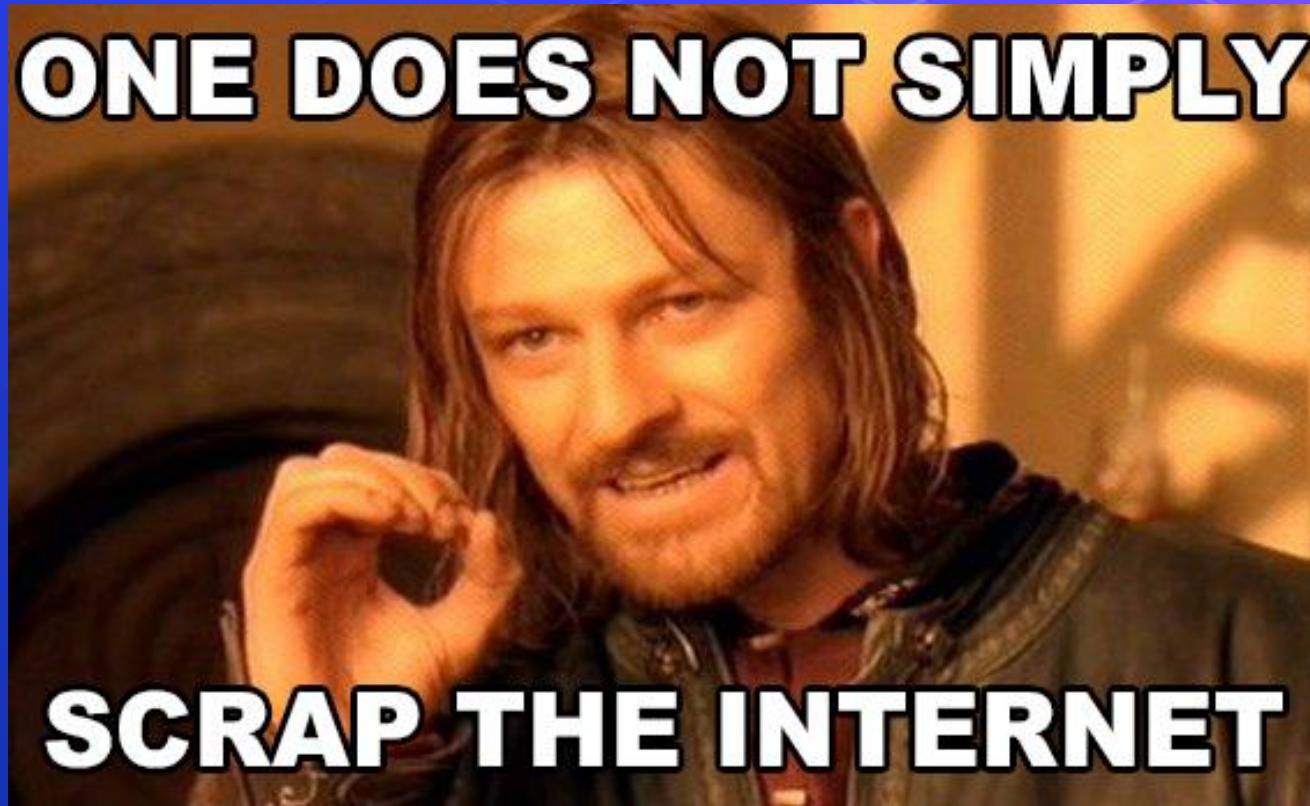
- What is Network Topologies
- What is Internet and Web Servers
- HTTP Request/Response Cycle
- HTML
- CSS
- Scraping Concept
- BeautifulSoup Library
- REST API Web Services
- Work with JSON



# Scraping Concept



## Scraping Concept



# Agenda

- What is Network Topologies
- What is Internet and Web Servers
- HTTP Request/Response Cycle
- HTML
- CSS
- Scrapping Concept
- [BeautifulSoap Library](#)
- REST API Web Services
- Work with JSON



# BeautifulSoup Library

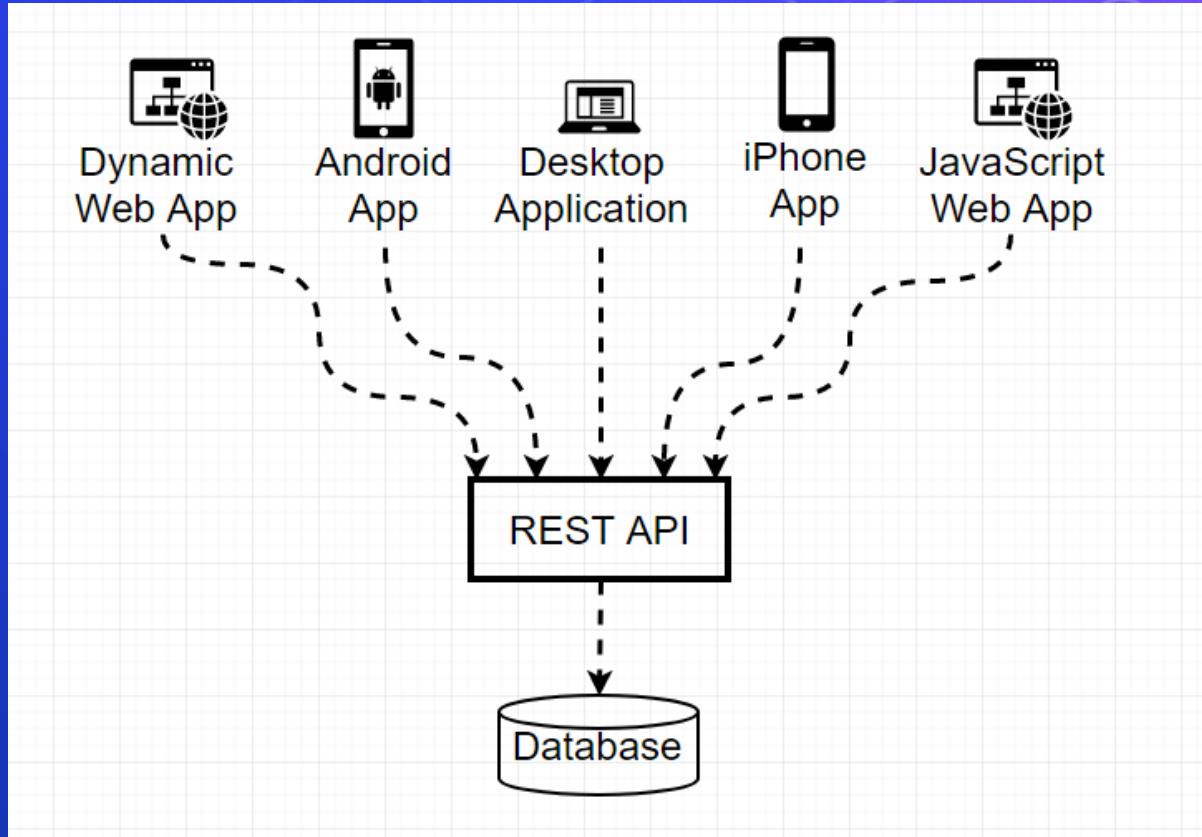
```
1 from bs4 import BeautifulSoup
2
3
4
5 page = requests.get("URL")
6
7 soup = BeautifulSoup(page.content, 'html.parser')
8
9 weather = soup.find('div', attrs={'id': 'weather_div'})
10
11 temp = weather.find_all('p', attrs={'class': "temp"})
12
13 print(f"Temperture is {temp.get_text()}")
14
```

# Agenda

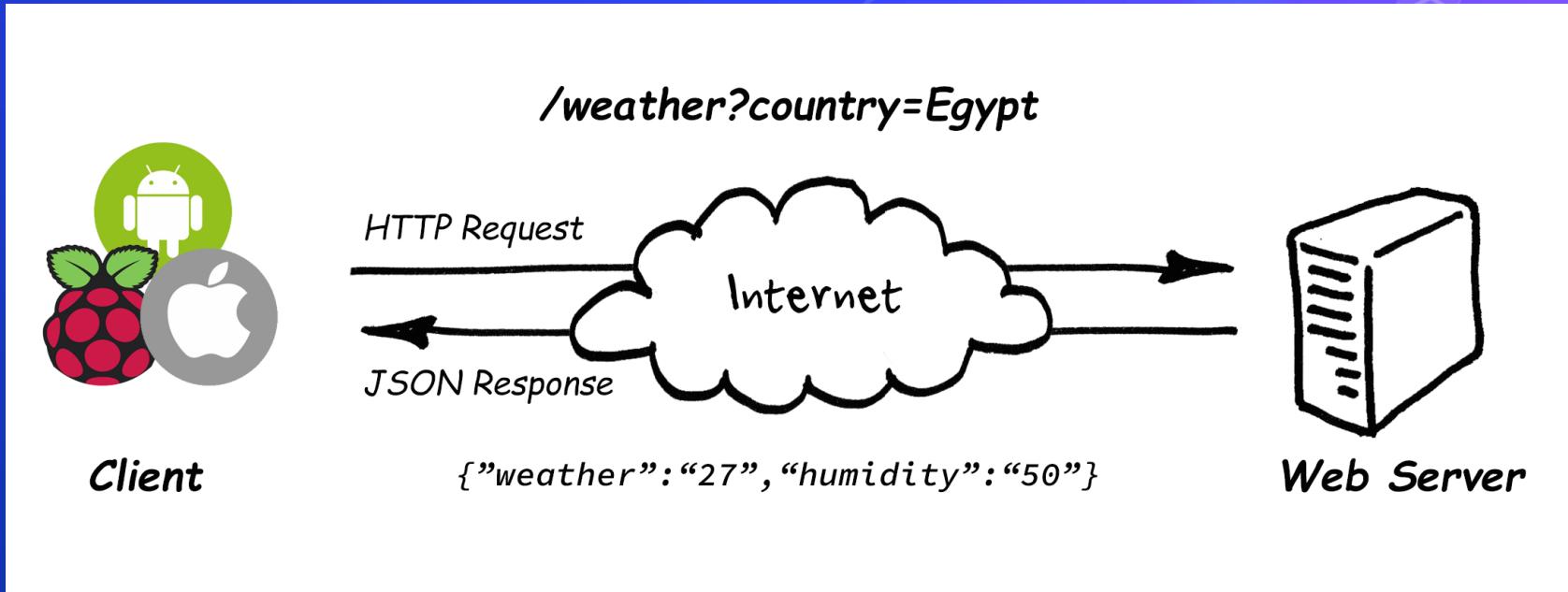
- What is Network Topologies
- What is Internet and Web Servers
- HTTP Request/Response Cycle
- HTML
- CSS
- Scrapping Concept
- BeautifulSoup Library
- REST API Web Services
- Work with JSON



# REST API Web Services



# REST API Web Services



# REST API Web Services

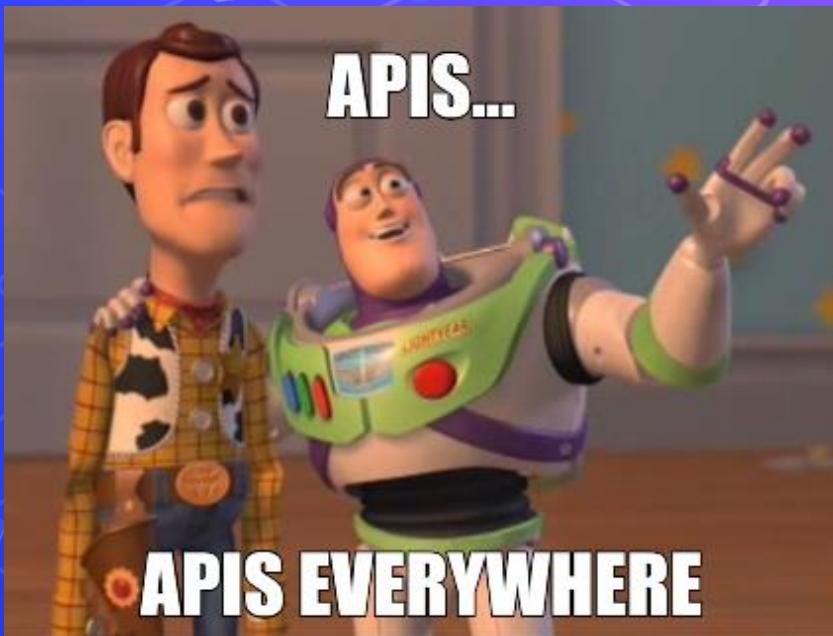
<https://any-api.com/>

<https://github.com/public-apis/public-apis>

<https://cloud.google.com/products/ai>

<https://www.twilio.com/>

....



# Agenda

- What is Network Topologies
- What is Internet and Web Servers
- HTTP Request/Response Cycle
- HTML
- CSS
- Scrapping Concept
- **BeautifulSoap Library**
- REST API Web Services
- Work with JSON



# Work with JSON

```
1 {
2     "employees": [
3         {
4             "id": "1",
5             "employee_name": "Ahmed",
6             "employee_salary": "320800",
7             "employee_age": "61"
8         },
9         {
10            "id": "2",
11            "employee_name": "Amr",
12            "employee_salary": "170750",
13            "employee_age": "63"
14        },
15        {
16            "id": "3",
17            "employee_name": "Sara",
18            "employee_salary": "86000",
19            "employee_age": "66"
20        }
21    ]
22 }
23 }
```



# Work with JSON

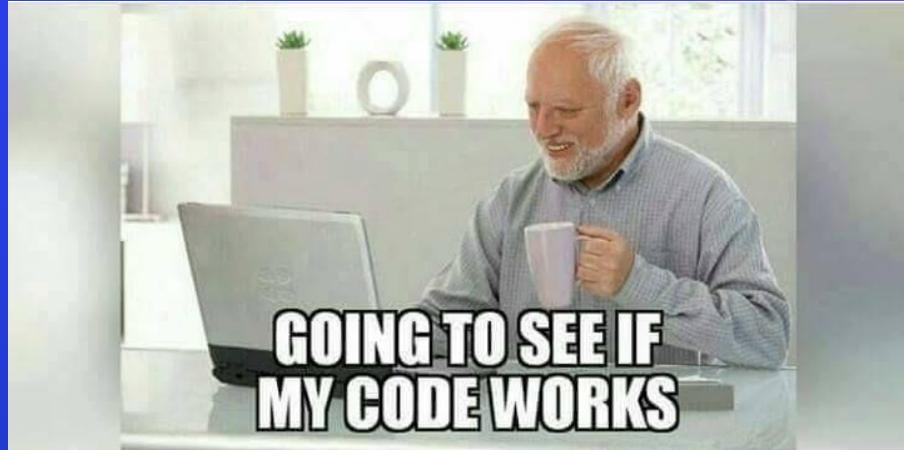
Install JSON Viewer extension  
For chrome or firefox

<https://chrome.google.com/webstore/detail/json-viewer/gbmdgpbipfallnflgajpalibnhdgobh>

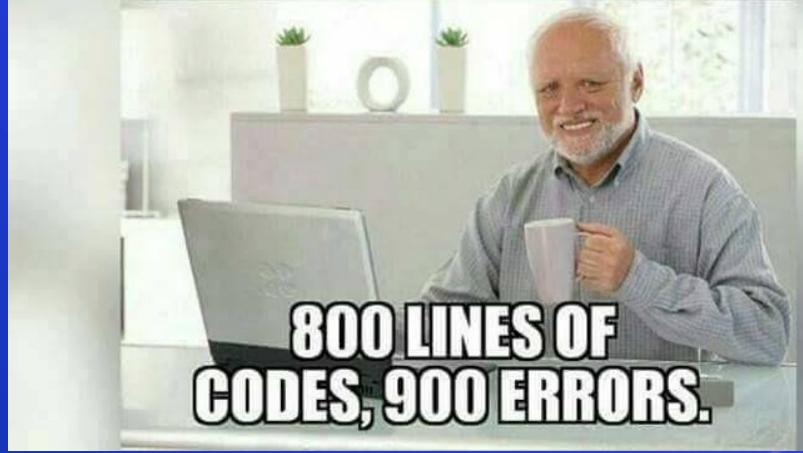
# Work with JSON

```
1 import json
2
3 # some JSON string:
4 json_text = '{ "name":"John", "age":30, "city":"New York"}'
5
6 # parse json_text:
7 user = json.loads(json_text)
8
9 # the result is a Python dictionary:
10 print(user["age"])
```





**GOING TO SEE IF  
MY CODE WORKS**



# Questions ?!



# Thanks!

>\_ Live long and prosper

