

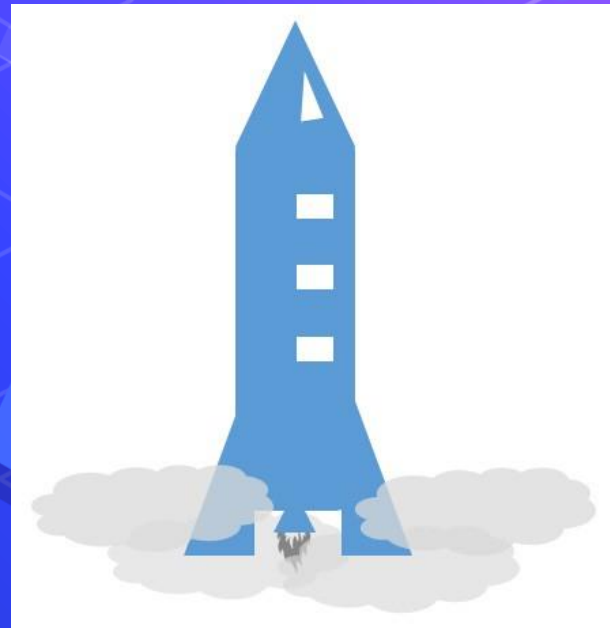
# Software Engineering & Deployment with Flask



# What is Deployment

The last step of a data science pipeline is model deployment, that means integrating the trained model within an application (web, mobile, desktop, hardware, game, etc...)

Business owners doesn't understand jupyter notebook  
Or coding stuff so they should interact with the model  
In an easy known way.



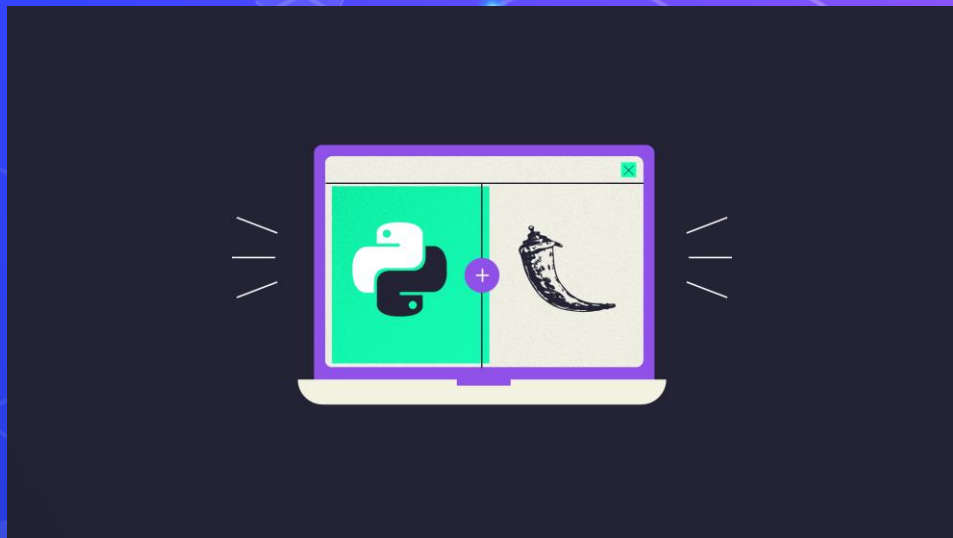
# What is Deployment

we will learn to create  
web applications and  
web services using Flask.

Awesome >\_



```
1 pip install flask gunicorn
```



# Agenda

- ⬡ Save and Load machine learning models
- ⬡ What is Network Topologies
- ⬡ What is Internet and Web Servers
- ⬡ HTTP Request/Response Cycle
- ⬡ Web applications with HTML and CSS
- ⬡ Web services with JSON
- ⬡ Flask Framework
- ⬡ Deploy on Heroku cloud



# Agenda

## Save and Load machine learning models

 What is Network Topologies

 What is Internet and Web Servers

 HTTP Request/Response Cycle

 Web applications with HTML and CSS

 Web services with JSON

 Flask Framework

 Deploy on Heroku cloud



# Save and Load machine learning models

```
1 from sklearn.linear_model import LogisticRegression
2 import joblib
3
4
5 model = LogisticRegression()
6 model.fit(X_train, Y_train)
7
8
9 # save the model to disk
10 joblib.dump(model, 'model.pkl')
11
12
13 # some time later...
14 # load the model from disk
15 loaded_model = joblib.load('model.pkl')
16 loaded_model.predict(X_test)
```



# Agenda

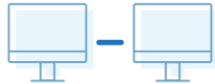
- ⬡ Save and Load machine learning models
- ⬡ What is Network Topologies
- ⬡ What is Internet and Web Servers
- ⬡ HTTP Request/Response Cycle
- ⬡ Web applications with HTML and CSS
- ⬡ Web services with JSON
- ⬡ Flask Framework
- ⬡ Deploy on Heroku cloud



# What is Network Topologies

## Network Topology Types

1 Point to point



2 Bus



3 Ring



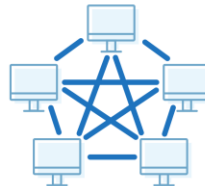
4 Star



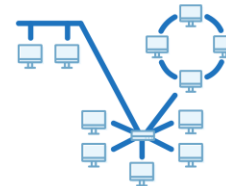
5 Tree



6 Mesh



7 Hybrid



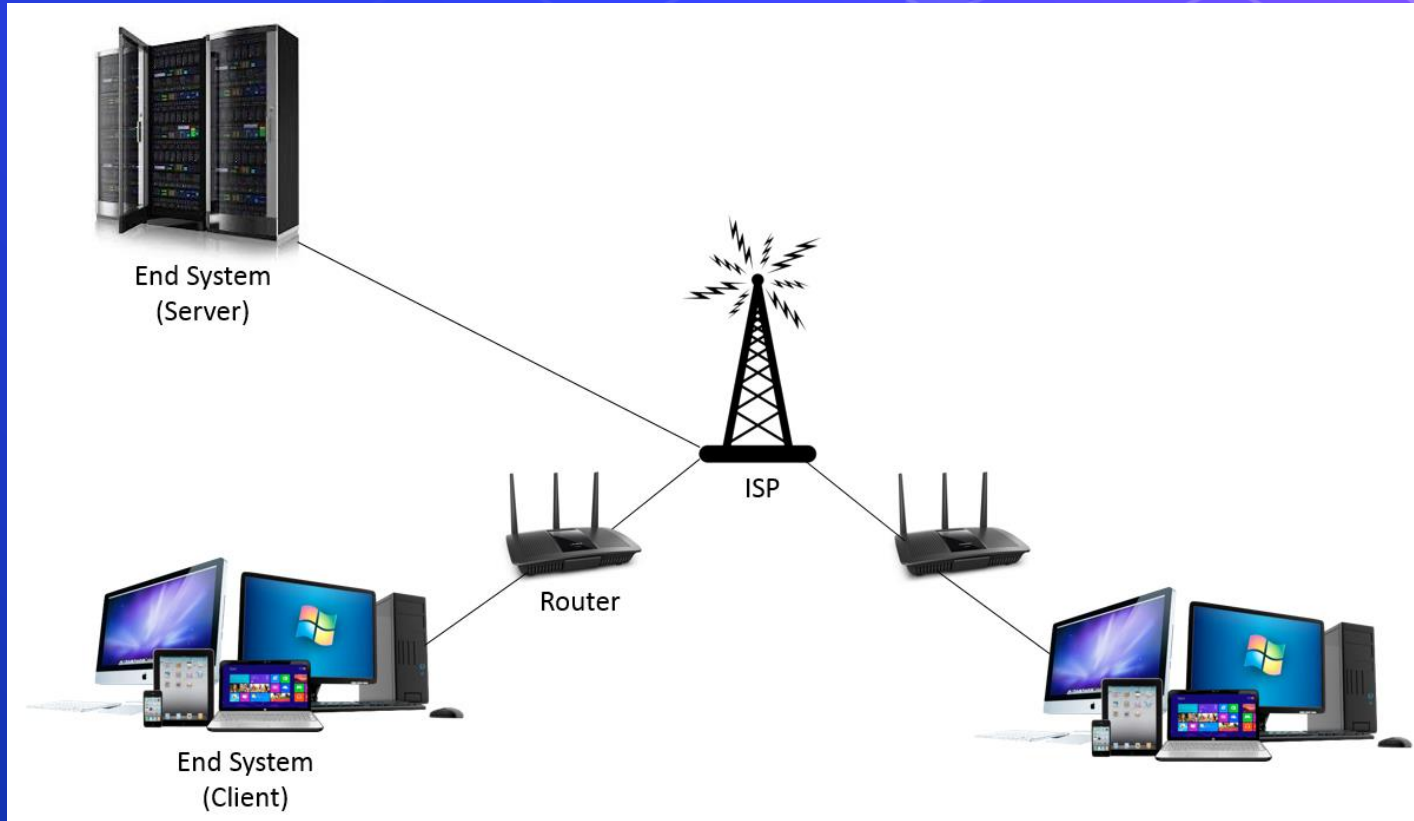


# Agenda

- ⬡ Save and Load machine learning models
- ⬡ What is Network Topologies
- ⬡ **What is Internet and Web Servers**
- ⬡ HTTP Request/Response Cycle
- ⬡ Web applications with HTML and CSS
- ⬡ Web services with JSON
- ⬡ Flask Framework
- ⬡ Deploy on Heroku cloud



# What is Internet and Web Servers

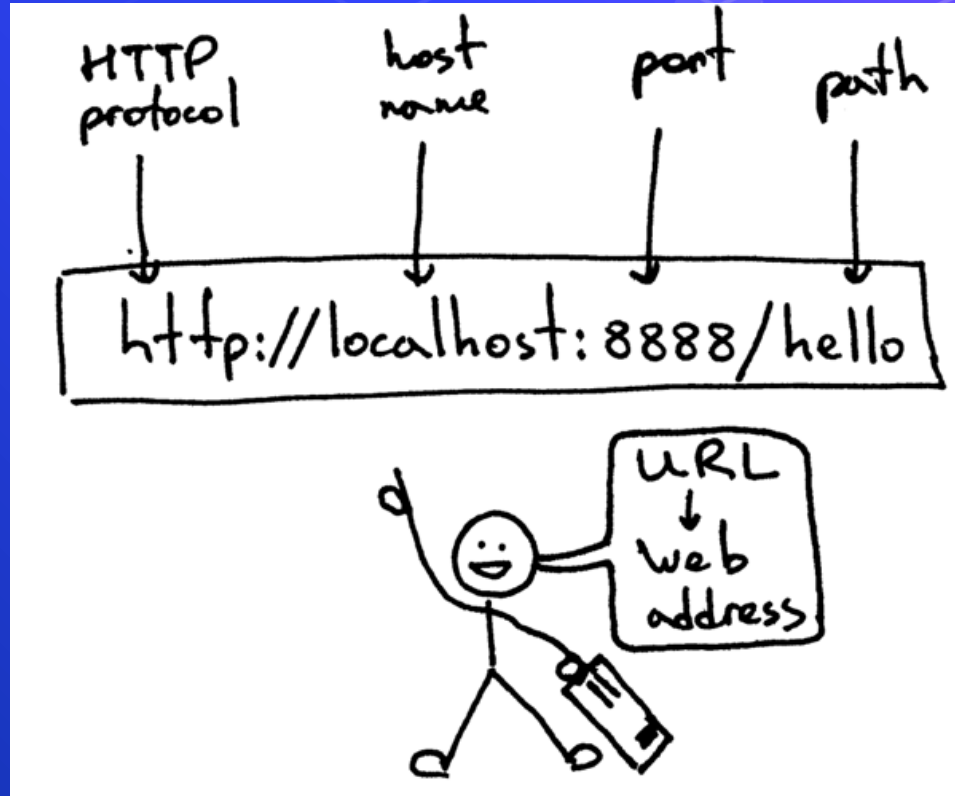


# Agenda

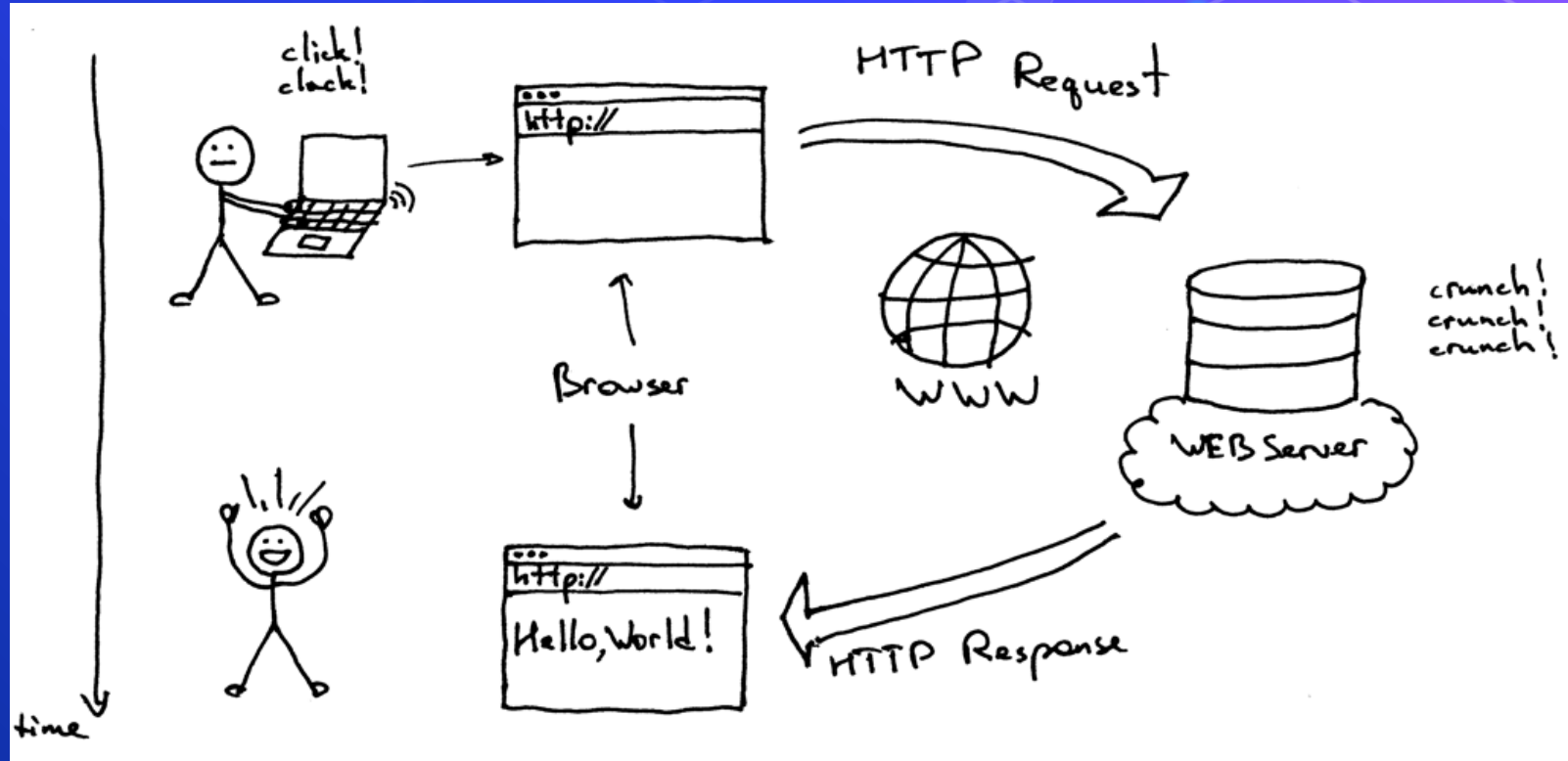
- ⬡ Save and Load machine learning models
- ⬡ What is Network Topologies
- ⬡ What is Internet and Web Servers
- ⬡ HTTP Request/Response Cycle
- ⬡ Web applications with HTML and CSS
- ⬡ Web services with JSON
- ⬡ Flask Framework
- ⬡ Deploy on Heroku cloud



# HTTP Request/Response Cycle



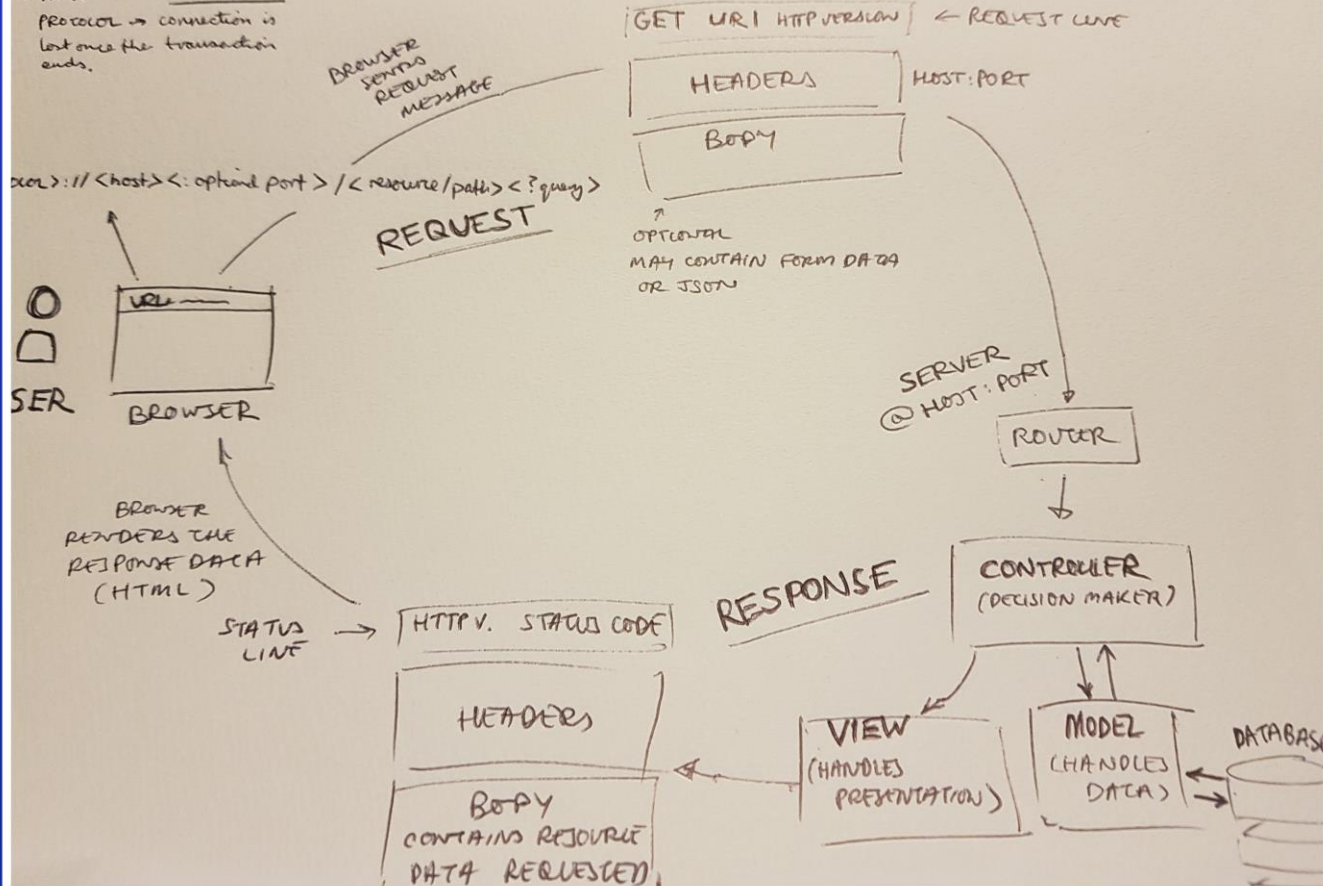
# HTTP Request/Response Cycle



# HTTP REQUEST-RESPONSE CYCLE

- HTTP is a STATELESS

PROTOCOL → connection is lost once the transaction ends.



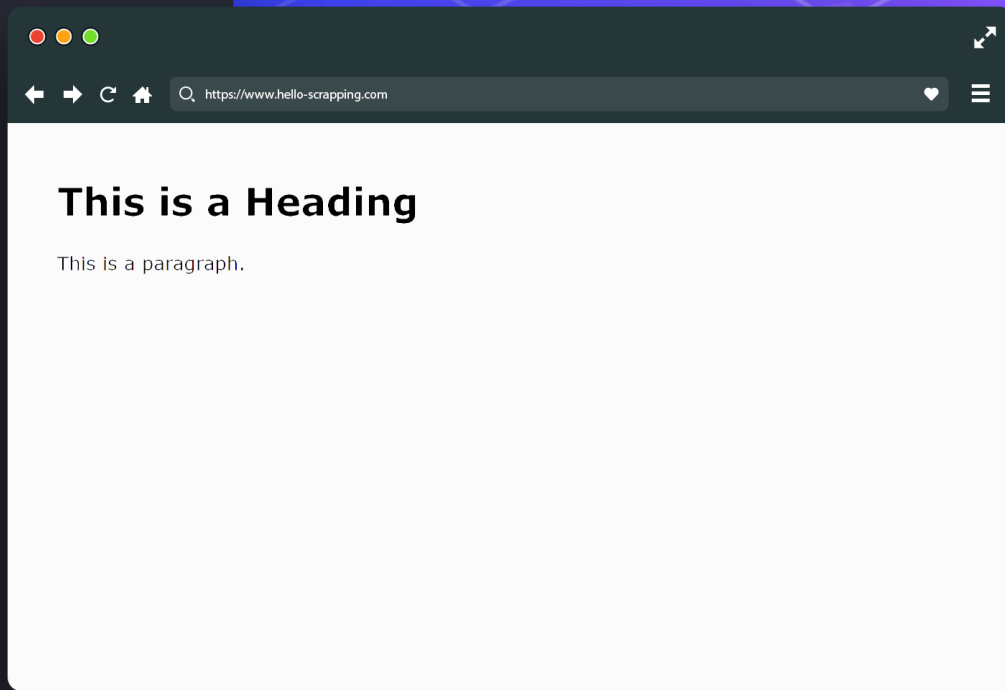
# Agenda

- ⬡ Save and Load machine learning models
- ⬡ What is Network Topologies
- ⬡ What is Internet and Web Servers
- ⬡ HTTP Request/Response Cycle
- ⬡ Web applications with HTML and CSS
- ⬡ Web services with JSON
- ⬡ Flask Framework
- ⬡ Deploy on Heroku cloud



# HTML (base snippet code)

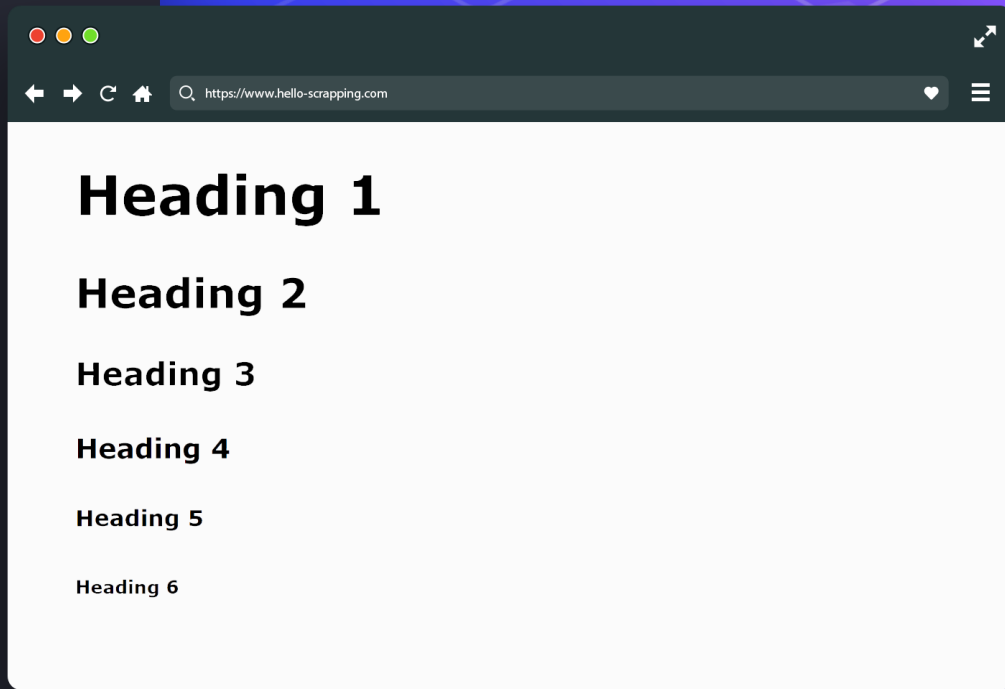
```
1 <!DOCTYPE html>
2 <html>
3
4   <head>
5     <title>Page Title</title>
6   </head>
7
8
9   <body>
10
11     <h1>This is a Heading</h1>
12     <p>This is a paragraph.</p>
13
14   </body>
15 </html>
16
```





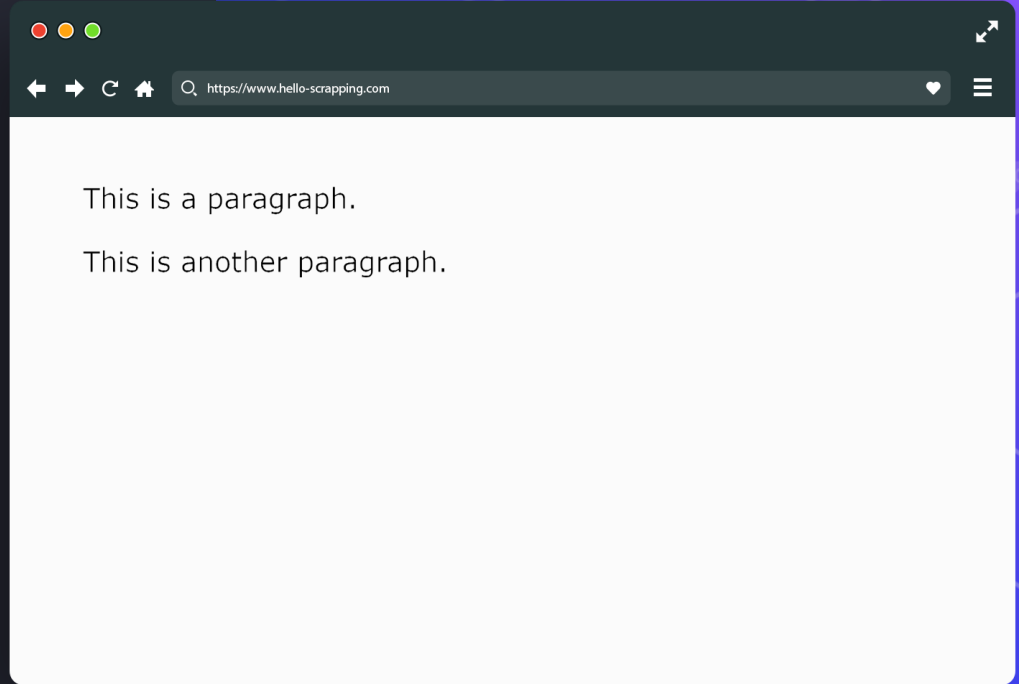
# HTML (header)

```
1 <!DOCTYPE html>
2 <html>
3
4   <head>
5     <title>Page Title</title>
6   </head>
7
8   <body>
9
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
```



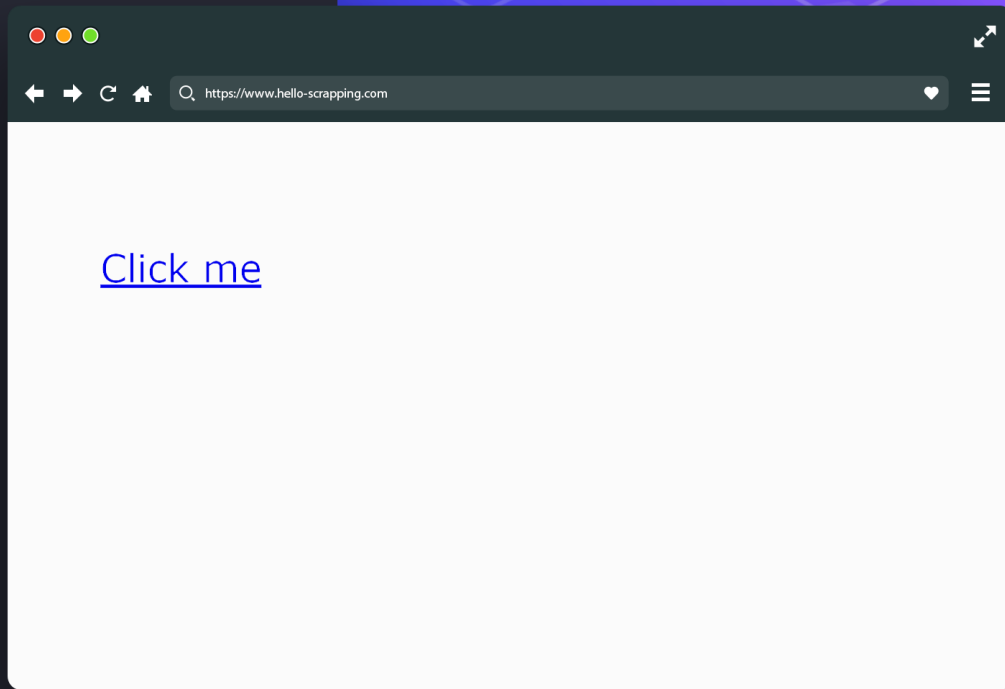
# HTML (paragraph)

```
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
```



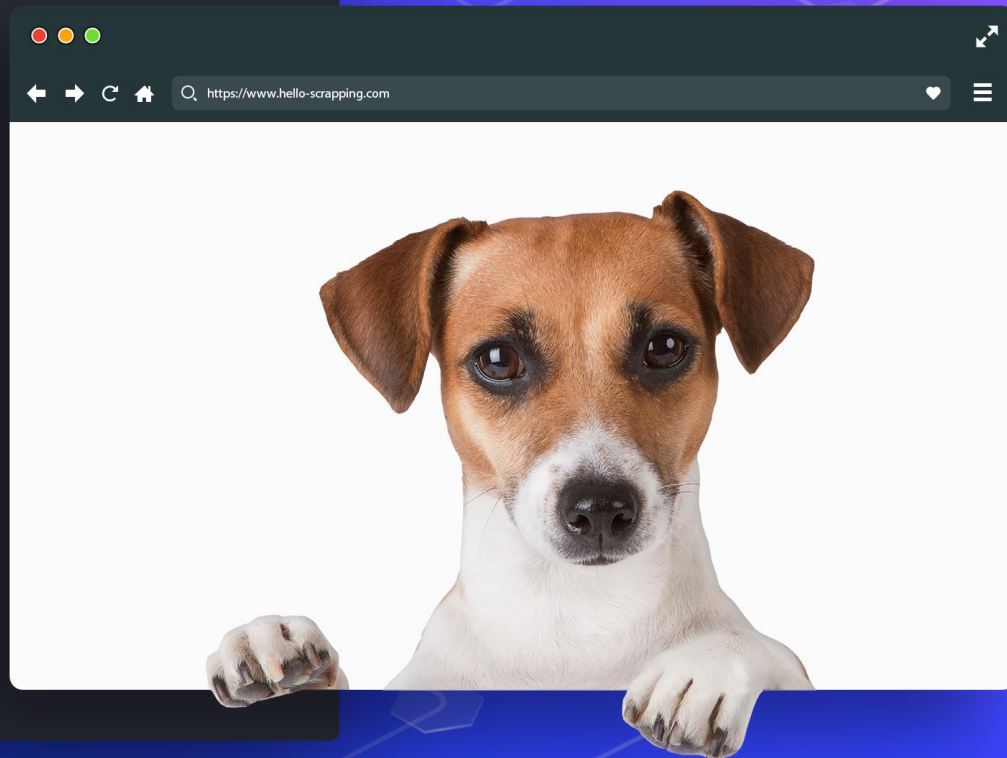
# HTML (link)

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



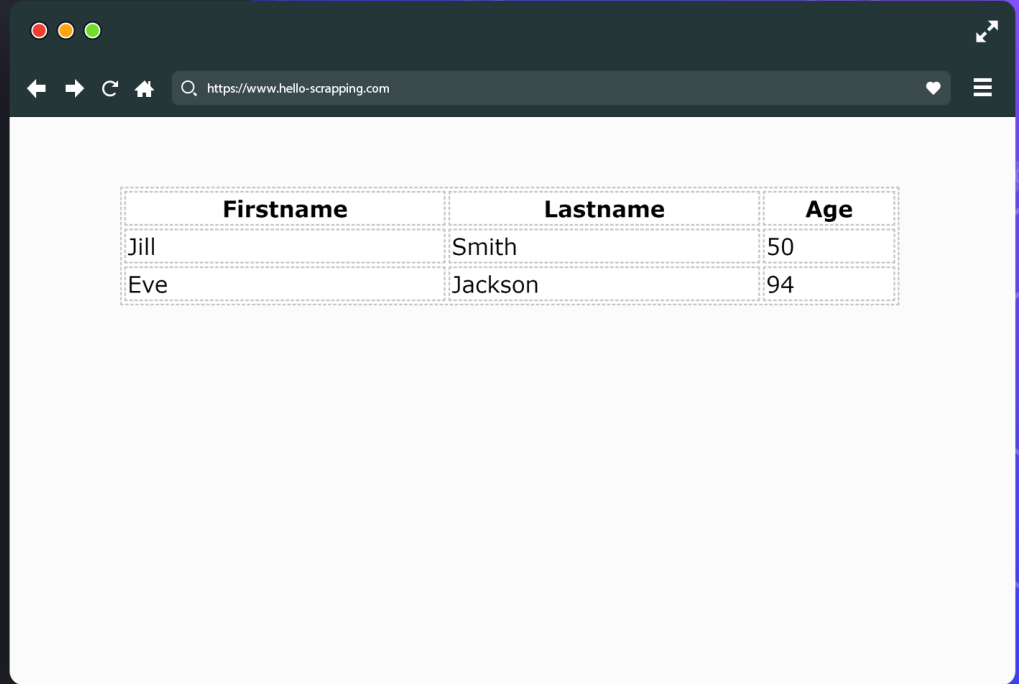
# HTML (image)

```
1 <!DOCTYPE html>
2 <html>
3
4   <head>
5     <title>Page Title</title>
6   </head>
7
8   <body>
9
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
```

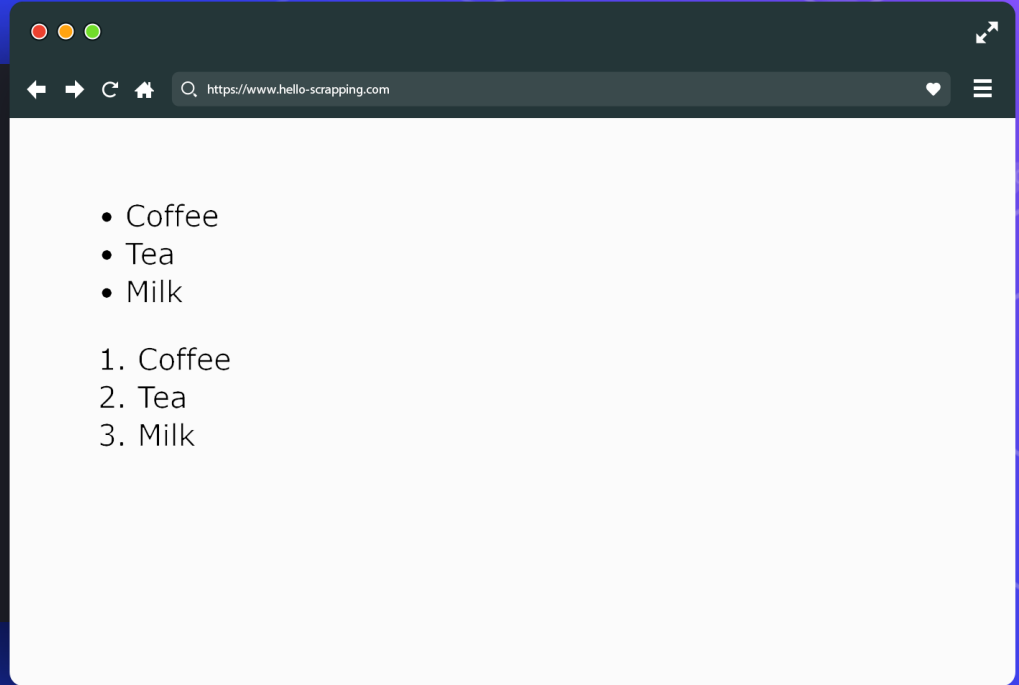


A screenshot of a web browser window showing the rendered HTML table. The browser's address bar displays the URL <https://www.hello-scraping.com>. The table is rendered with three columns: Firstname, Lastname, and Age. The first row contains the values Jill, Smith, and 50. The second row contains the values Eve, Jackson, and 94.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94

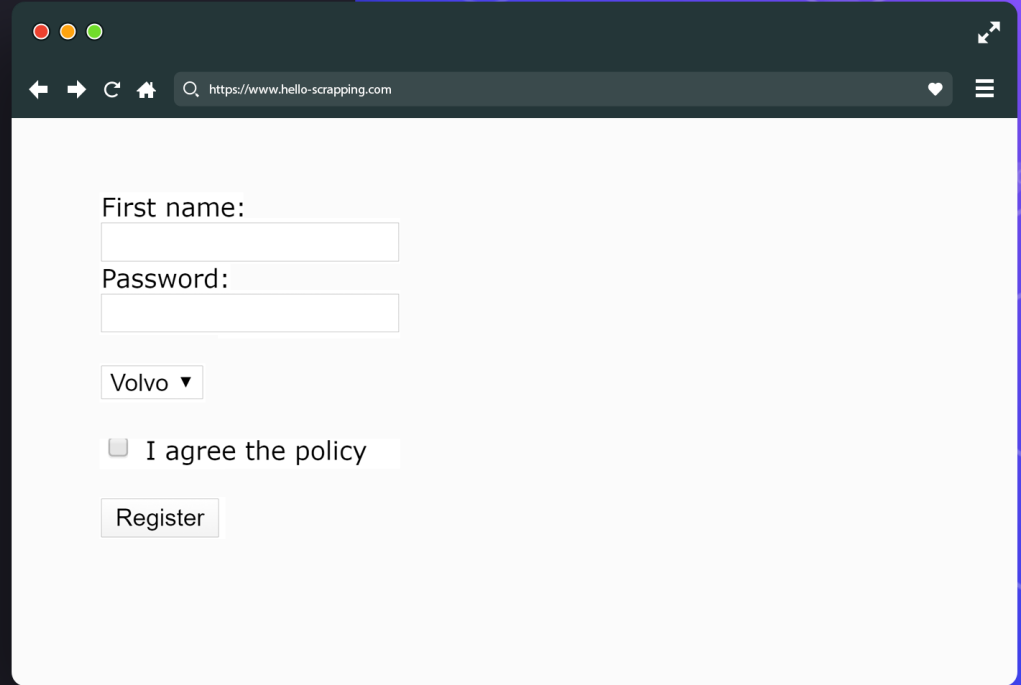
# HTML (list)

```
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>
```



# HTML (form)

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



https://www.hello-scrapping.com

First name:

Password:

Volvo ▼

☐ I agree the policy

Register

# Other HTML Elements

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





# CSS

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



# Check Also

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

JS

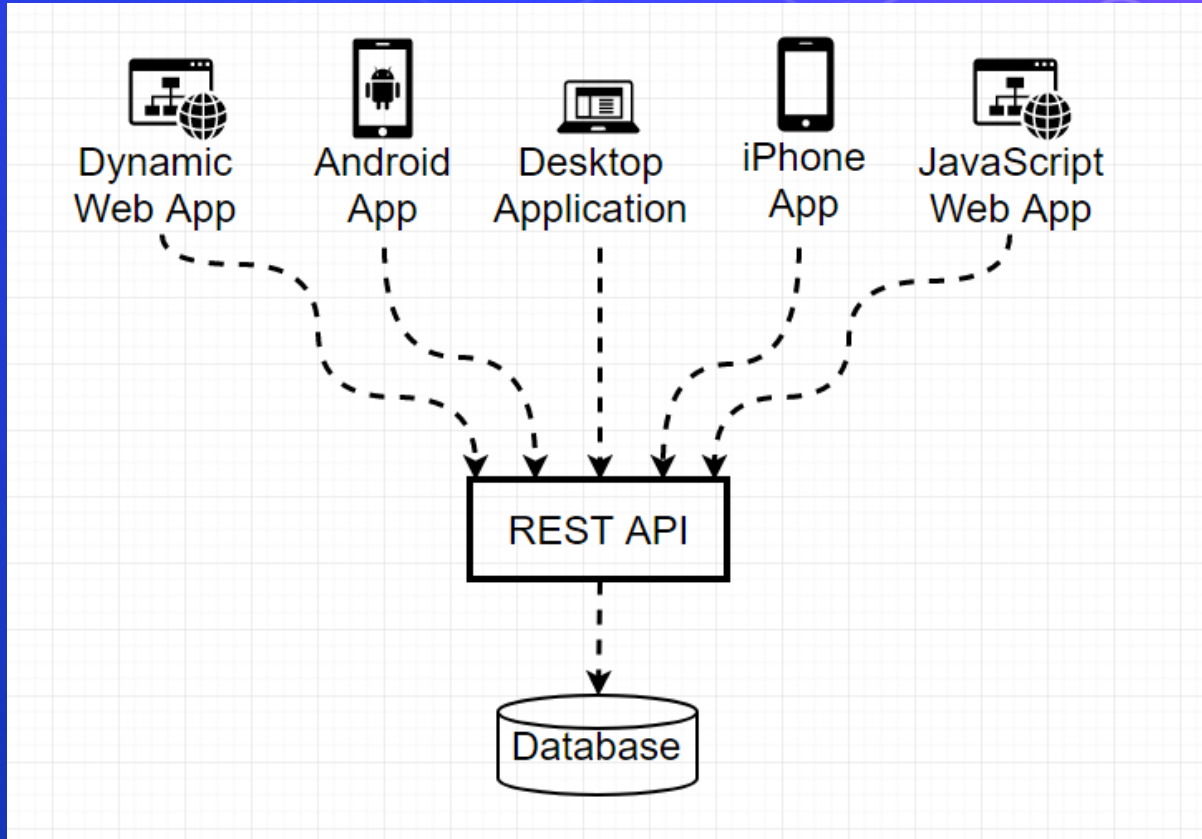


# Agenda

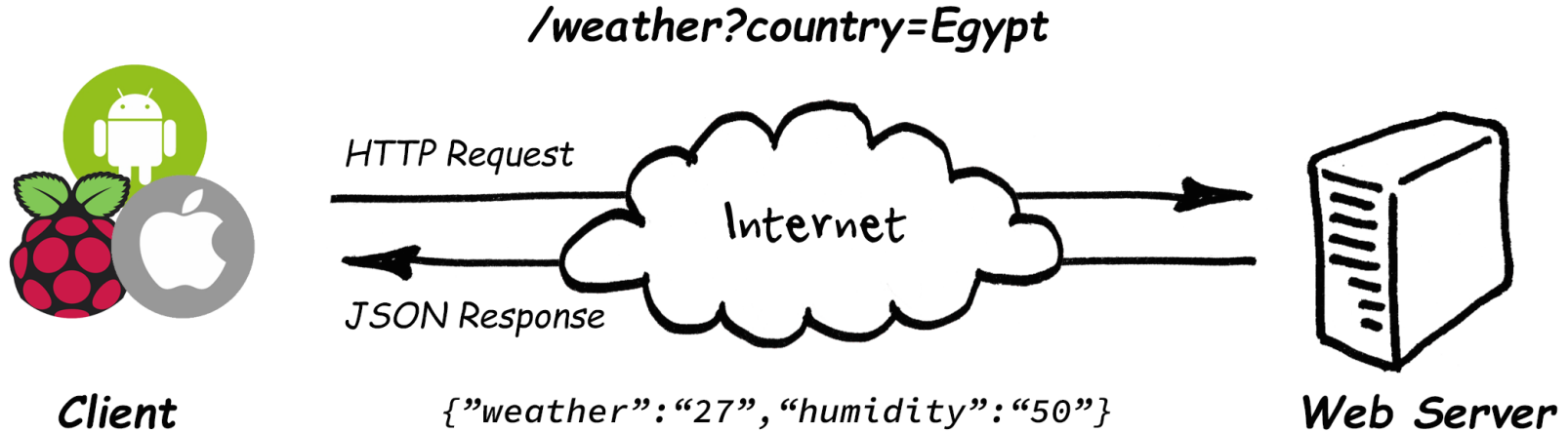
- ⬡ Save and Load machine learning models
- ⬡ What is Network Topologies
- ⬡ What is Internet and Web Servers
- ⬡ HTTP Request/Response Cycle
- ⬡ Web applications with HTML and CSS
- ⬡ Web services with JSON
- ⬡ Flask Framework
- ⬡ Deploy on Heroku cloud



# REST API Web Services



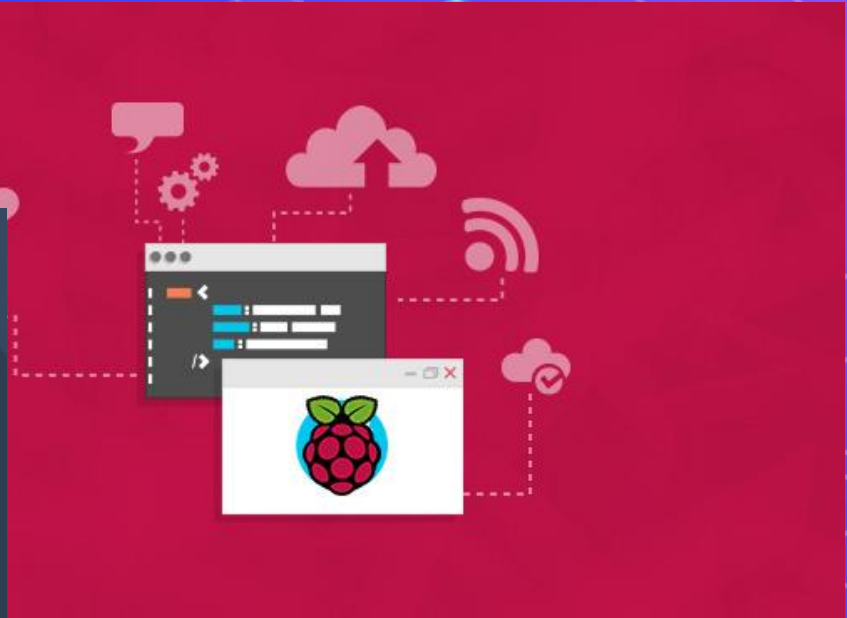
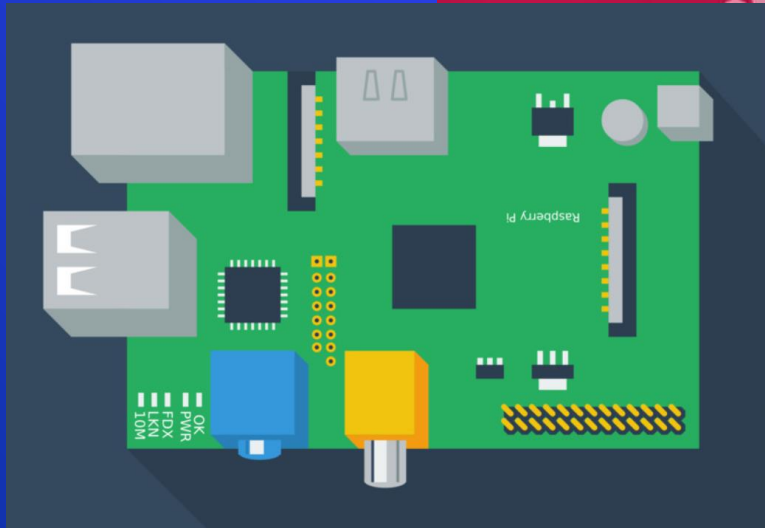
# REST API Web Services



# REST API Web Services



# REST API Web Services



# Work with JSON

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





# Work with JSON

Install POSTMAN

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

Install JSON Viewer extension for chrome or firefox

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



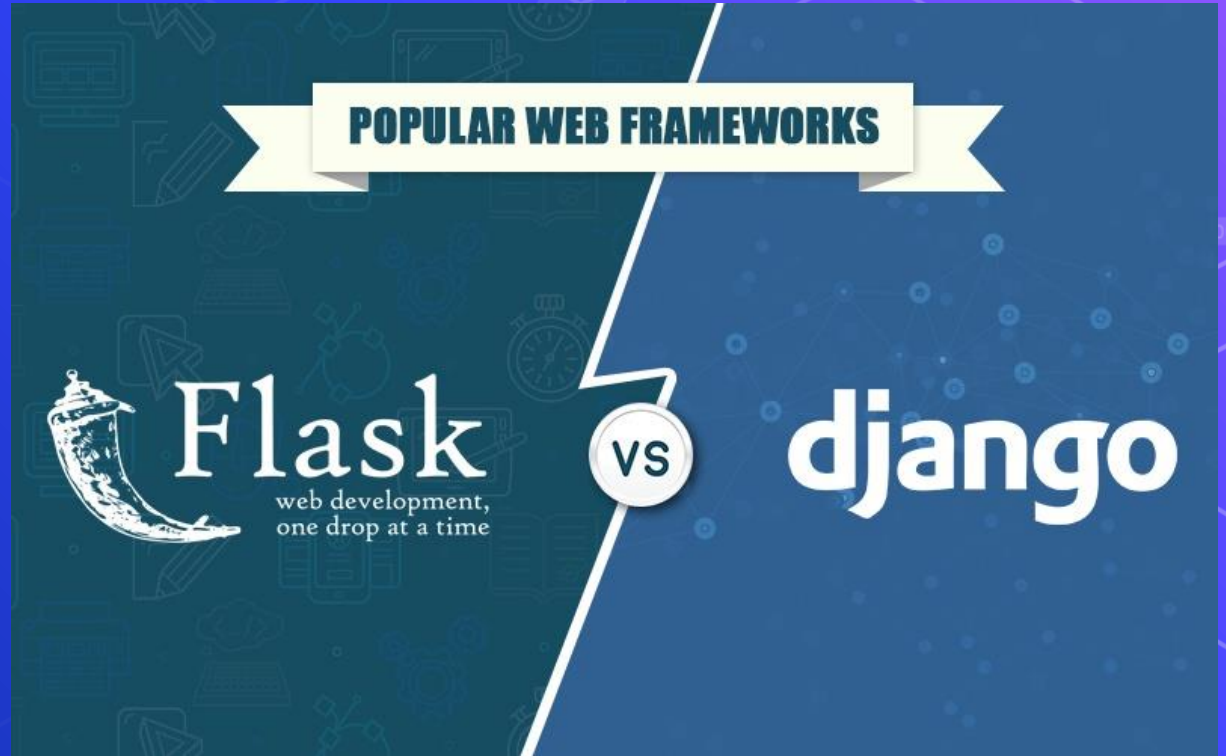
# Agenda

- ⬡ Save and Load machine learning models
- ⬡ What is Network Topologies
- ⬡ What is Internet and Web Servers
- ⬡ HTTP Request/Response Cycle
- ⬡ Web applications with HTML and CSS
- ⬡ Web services with JSON
- ⬡ **Flask Framework**
- ⬡ Deploy on Heroku cloud



# Flask Framework

Lets code >\_



# Agenda

- ⬡ Save and Load machine learning models
- ⬡ What is Network Topologies
- ⬡ What is Internet and Web Servers
- ⬡ HTTP Request/Response Cycle
- ⬡ Web applications with HTML and CSS
- ⬡ Web services with JSON
- ⬡ Flask Framework
- ⬡ Deploy on Heroku cloud



# Deploy on Heroku cloud

- 1- Create a new account.
- 2- Create a new app.
- 3- `pip freeze > requirements.txt`
- 4- Make Procfile.
- 5- Follow the deployment instructions.

That's it easy stuff xD



# Questions ?!



# Thanks!

>\_ Live long and prosper

