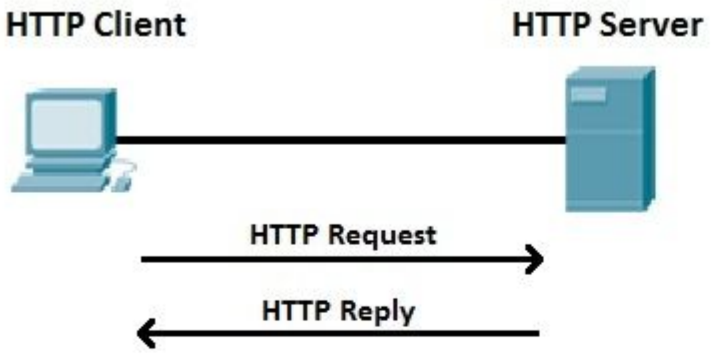
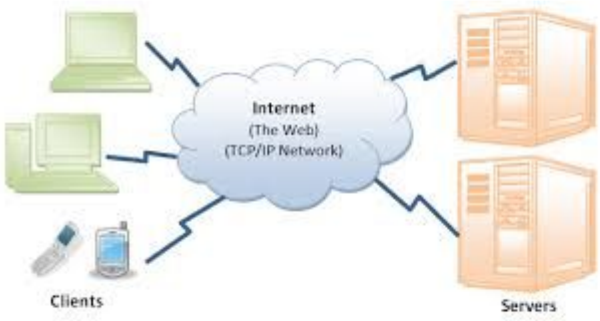
pyladies Bratislava #6

Web server with Flask

HTTP - HyperText Transfer Protocol

- protocol for websites
- client/server
- request/response





Server-side code

- That’s what we need to write in order to host a website
- receives requests - sends back responses

Flask

web development,
one drop at a time

Getting started with Flask

- Does all the heavy lifting for us
- We can only care about the important things

- `pip install flask`

- Running on http://127.0.0.1:5000/

```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def home():
    return "Hello, World!"

if __name__ == "__main__":

    app.run(debug=True)
```

https://medium.freecodecamp.org/how-to-build-a-web-application-using-flask-and-deploy-it-to-the-cloud-3551c985e492

Templates - Jinja

- .html files
- Put inside `templates/` folder

```python
from flask import Flask, render_template

@app.route("/")
def home():

    return render_template("home.html")
```

```html
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Flask Tutorial</title>
  </head>
  <body>
    <h1> My First Try Using Flask</h1>
    <p> Flask is Fun </p>
  </body>
</html>
```

Templates

- Context variables

```
@app.route("/")
def home():

    return render_template("home.html", name=”Maria”)


@app.route("/about")
def about():

    return render_template("about.html")
```

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Flask Tutorial</title>
  </head>
  <body>
    <h1> Hello {{name}} </h1>
    <ul class="menu">
              <li><a href="{{ url_for('home') }}">Home</a></li>
              <li><a href="{{ url_for('about') }}">About</a></li>
            </ul>

  </body>
</html>
```

Templates

- Parent templates

```
{% block content %}

        <p>Some default text here, that can be overridden </p>

{% endblock %}
```

```
{% extends "template.html" %}
```

CSS

- .css files
- Put inside `static/css/` folder

```
<link rel="stylesheet" href="{{ url_for('static',    filename='css/template.css') }}">
```

Add database if you want

- Remember sqlite session couple months ago?

```
@app.route('/signUp',methods=['POST'])
def signUp():

    # read the posted values from the UI
    _name = request.form['inputName']
    _email = request.form['inputEmail']
    _password = request.form['inputPassword']

    if _name and _email and _password:

        connection = sqlite3.connect("/home/maria/pyLadies/projects/FlaskApp/test.db")
        cursor = connection.cursor()
        sql = ‘INSERT INTO tbl_user (user_name, user_username, user_password) \
              ‘VALUES ("{}", "{}", "{}")’
        sql.format(_name, _email, _password)
        cursor.execute(sql)
        connection.commit()
        connection.close()

        return json.dumps({'message':'User created successfully !'})
    else:
        return json.dumps({'message':'Something went wrong :( '})
```

More tutorials

https://pythonspot.com/flask-web-app-with-python/

https://medium.freecodecamp.org/how-to-build-a-web-application-using-flask-and-deploy-it-to-the-cloud-3551c985e492

https://code.tutsplus.com/tutorials/creating-a-web-app-from-scratch-using-python-flask-and-mysql--cms-22972

Websites with python

Other web frameworks:

- Django
- web2py
- Google App Engine
- TurboGears
- Pylons
- Pyramid
- Masonite