

# Python 101

Curs 5 - Python and web  
Data

The background features abstract geometric shapes: a large blue circle on the left, a smaller blue circle at the bottom right, and several thin, light gray circular arcs. Small gray dots are placed at the intersections of these arcs.

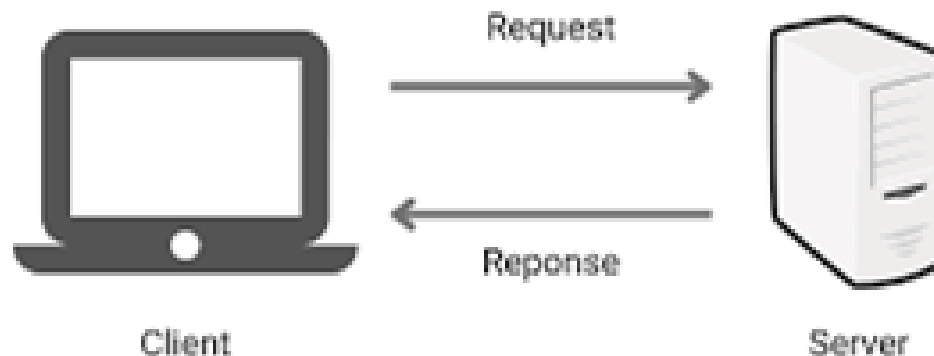
# Quiz time!

# Ce înseamnă web?

- Noun: a network of silken thread spun especially by spiders. (english dictionary)
- O rețea (pânza) de calculatoare care comunică prin intermediul Internetului.
- Majoritatea limbajelor de programare au biblioteci și framework-uri care permit programului să acceseze Internetul.

# Relatia Client Server

- O aplicație web este de fapt formată din două aplicații: clientul și serverul.
- Ele comunică prin intermediul unui API. (application process interface)



# Client

- Rulează pe dispozitivul utilizatorului.  
(calculator, telefon, tabletă, etc)
- Cere resurse de la server prin cereri http.

# Server

- Rulează de obicei pe un calculator special al companiei/organizației care deține produsul.
- Răspunde la cererile clienților.
- Pentru o aplicație/serviciu nu exista neapărat un singur server.

# API

- Pentru clienți serverul este un black box.
- API este un intermediar între client și server.
- Descrie tipul de cereri pe care clientul le poate face către server.
- Primește cereri de la client și le trimite serverului.

# Cereri de tip HTTP

- **Get**
  - Cere serverului date.
- **Post**
  - Cere serverului să stocheze resurse în baza de date.
- **Put**
  - Cere serverului să modifice resurse din baza de date.
- **Patch**
  - Cere serverului să modifice parțial resurse din baza de date.
- **Delete**
  - Cere serverului să șteargă resurse din baza de date.



# Cereri de tip HTTP

- Exista aplicații prin intermediul cărora putem formula cereri http:
  - Postman
  - Orice Browser
  - Etc
- Sau putem folosi module/biblioteci ale limbajelor de programare:
  - Python request module

# Modulul Request

- Face cereri de tip HTTP.
- Util în scripting, boți și automatizare.
- Documentația completă se găsește [aici](#).
- Principalele tipuri de requesturi sunt:
  - GET
  - POST



# Modulul Request

```
import requests  
r = requests.get('https://ro.wikipedia.org/wiki/Python')  
print(r.text)
```

- Metodele din request returnează obiecte de tip Response.
- Aceste obiecte conțin informații despre cererea pe care am făcut-o.

# Modulul Request

- Uneori suntem nevoiți să trimitem parametrii adiționali într-o cerere http.
- Putem face asta pasând un dicționar argumentului opțional params.

```
import requests
```

```
params = {'key1': 'value1', 'key2': ['value2', 'value3']}
```

```
r = requests.post(url, params=params)
```

- Request știe să implementeze toate tipurile de cereri http.

# Modulul Request

- Request știe să implementeze toate tipurile de cereri http:
  - `request.get()`
  - `request.post()`
  - `request.put()`
  - `request.patch()`
  - `request.delete()`
- Aceste metode funcționează exact la fel, având exact aceleași parametrii.
- Ele diferă prin modul în care server-ul la care facem cererea le tratează, primind răspunsuri diferite în funcție de tipul request-ului.

# Modulul JSON

- Este o modalitate de a stoca și trimite date.
- Exemplu json:
  - '{"name":"John", "age":30, "car":null}'
- Este de fapt un string pe baza căruia putem modela:
  - O clasă în JavaScript
  - Un dicționar în Python

# Modulul JSON

- import json
- Vom folosi metodele:
  - json.loads(x) care returnează un dicționar dacă x este un string json.
  - json.dumps(x) care returnează un json dacă x este json serializabil.

# Modulul JSON

- Tipuri serializabile:
  - Dicționar
  - Listă
  - Tuplu
  - String
  - Int
  - Float
  - Bool
- Clasele nu sunt json serializabile.



The background features several thin, light gray circular arcs and three small gray dots. A blue curved shape is visible on the left edge, and a blue triangle is in the top right corner.

# Pauză

# Flask Framework

- Ce este un framework?
  - O colecție de module și pachete care simplifică dezvoltarea de aplicații.
- Flask este o soluție pentru dezvoltarea aplicațiilor web în Python, furnizând un API simplu pentru crearea unui server.
- Nu este recomandat pentru aplicații complexe.
- Documentația completă se găsește [aici](#).

# Flask Framework

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route("/")  
def hello_world():  
    return "<p>Hello, World!</p>"
```

- Flask folosește view functions.
- Ele descriu codul ce va răspunde unui request.
- Creăm o instanță a clasei flask.
- Folosim decoratorul `@app.route(URL)` pentru a specifica ce URL va declanșa apelul funcției `hello_world`.

# Flask Framework

- Pornim aplicația cu flask --app main run
- Apoi putem sa deschidem un browser la adresa <http://127.0.0.1:5000/> unde ni se va afișa mesajul Hello World!
- By default Flask setează host-ul la 127.0.0.1 si portul 5000.
- Putem să modificăm aceste valori plasând parametrii in linia de comandă: flask --app main run -host ip -port port

# Flask Framework

## Decoratori Flask

- `@app.route('/add', methods=['GET'])`
- `@app.route('/login', methods=['POST'])`
- `@app.route('/product/<name>', methods=['GET'])`

## Python Request:

- `request.get('http://127.0.0.1:5000/add', json=json)`
- `request.post('http://127.0.0.1:5000/login',  
json=json)`
- `request.get('http://127.0.0.1:5000/product/phone')`
- `request.get('http://127.0.0.1:5000/product/laptop')`

# Flask Framework

- Anumite cereri http trimit parametrii, JSON-uri sau chiar fişiere.
- Putem accesa datele primite cu ajutorul variabilei globale request.

```
from flask import Flask, request  
app = Flask(__name__)  
@app.route('/add', methods=['GET'])  
def f():  
    json = request.get_json()  
    f = request.files["name"]
```

# Flask Framework

- Flask știe să răspundă la cereri și cu pagini html.
- Fișierul index.html trebuie stocat într-un director numit *templates*.

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

```
@app.route("/")  
def hello_world():  
    return render_template('index.html')
```

# Postman Demo





The background features several large, thin, light gray circles and arcs. A blue curved shape is visible on the left edge, and a blue triangle is in the top right corner.

# Întrebări?