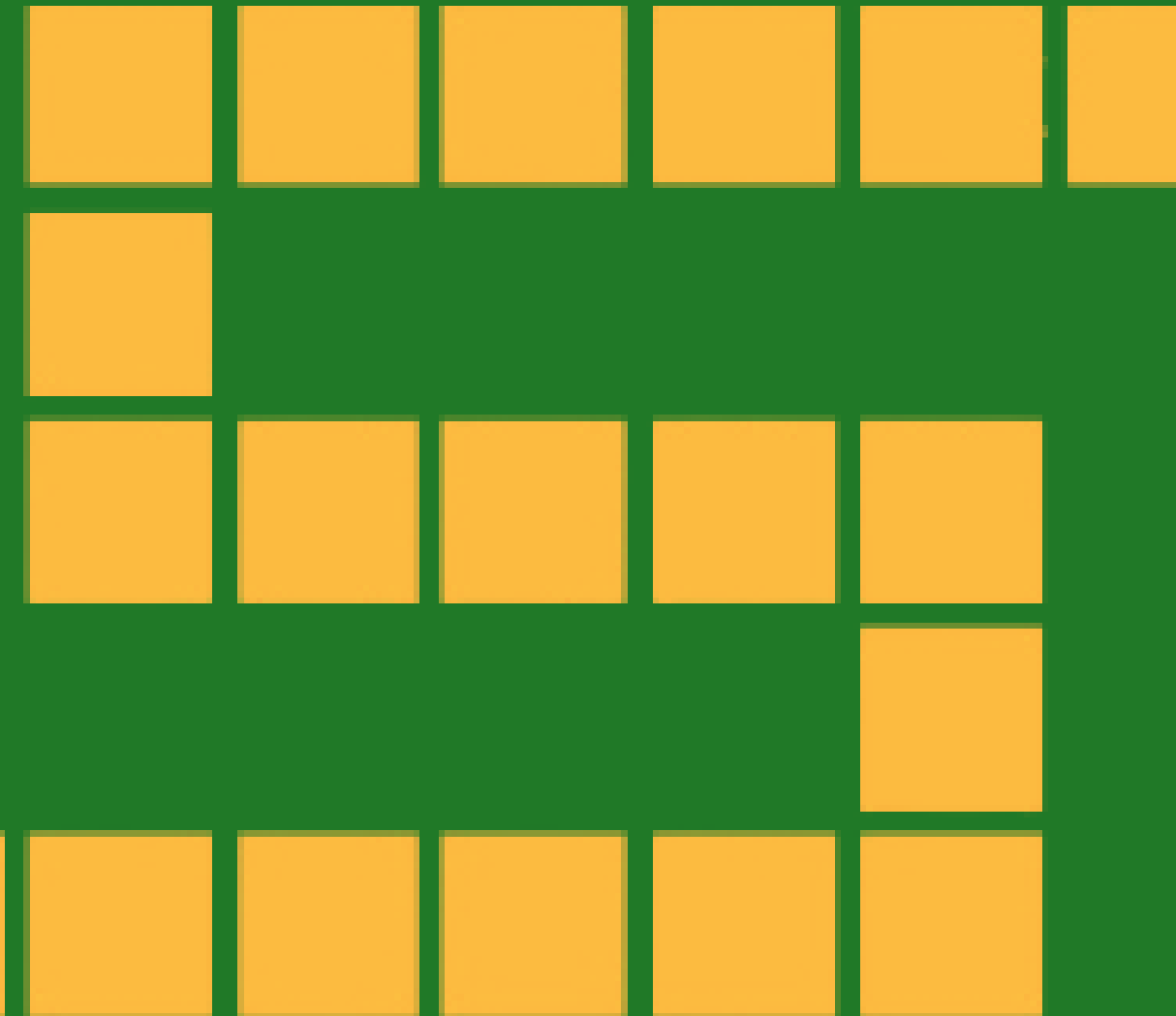
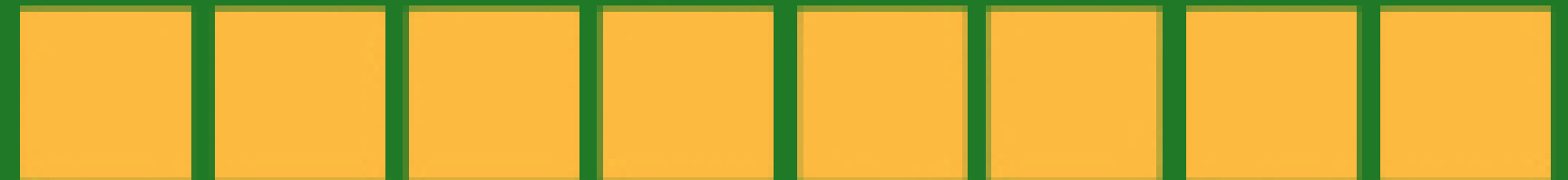
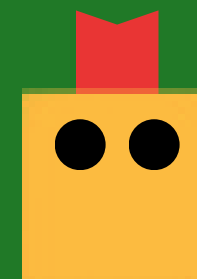


Programming with

PYTHON

by SoCode

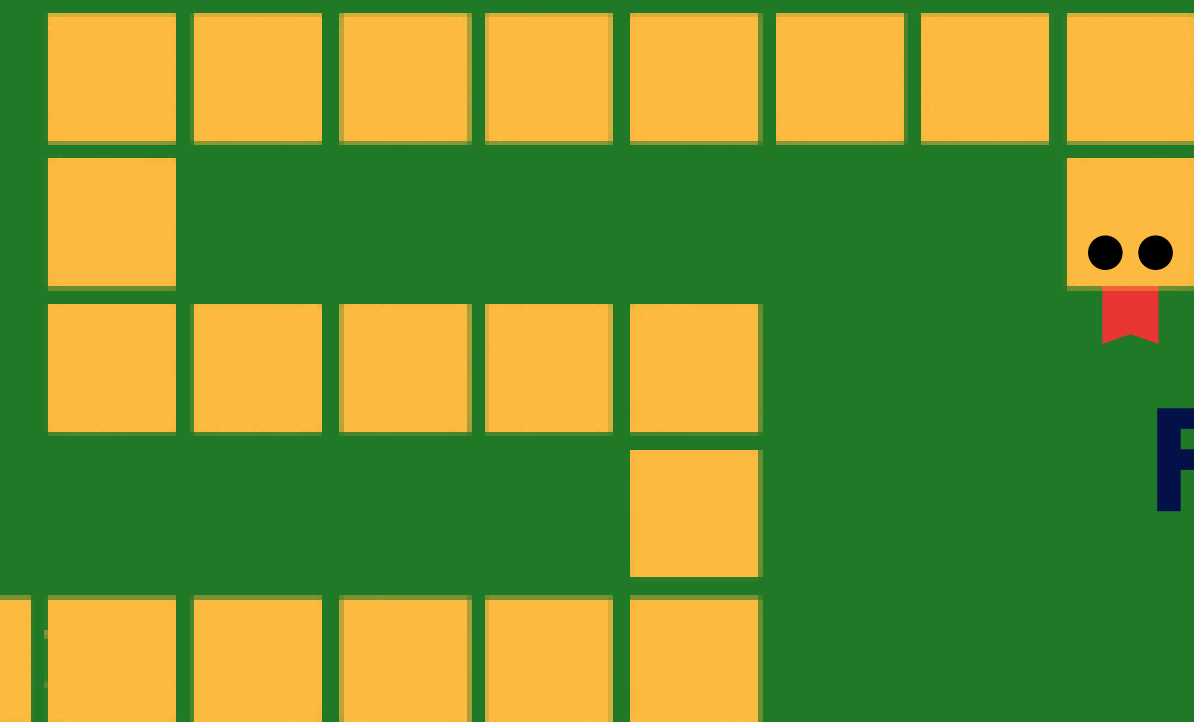




Introduction to Git & GitHub

Control Statements

»»» Data Structure



Functions <<<

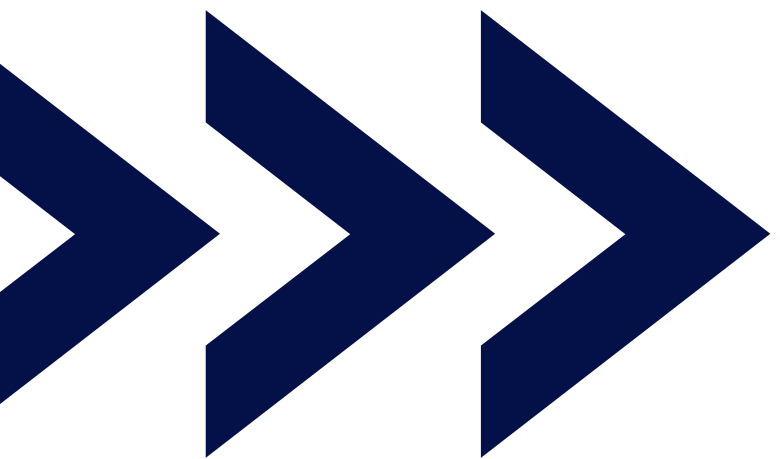
>>> **Files Manipulations**

Programming Oriented Object <<<

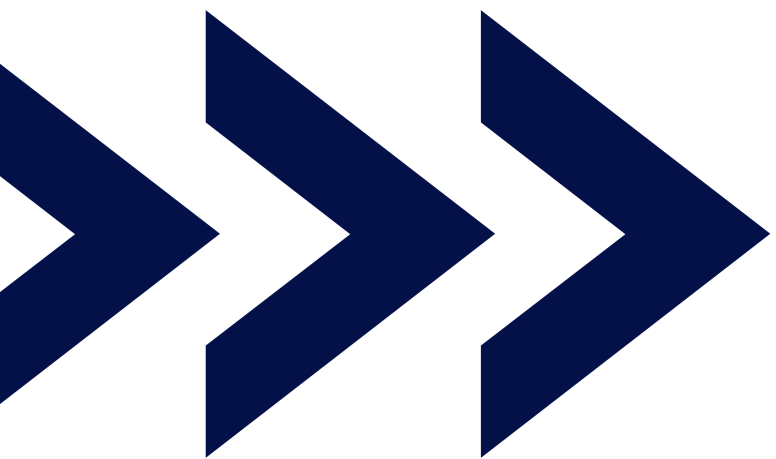
>>> **WEB DEV with Python**

Data Science with Python <<<

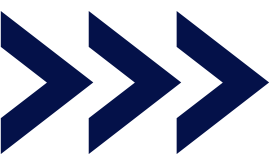




before we start...



Q&A Session



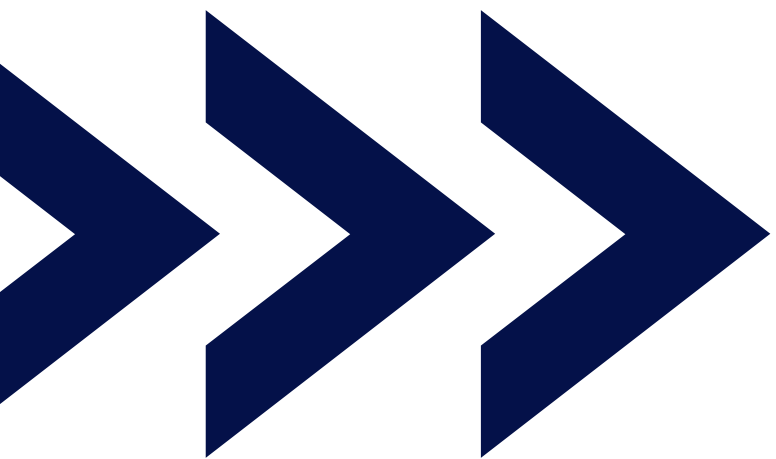
difference between constructor and method?



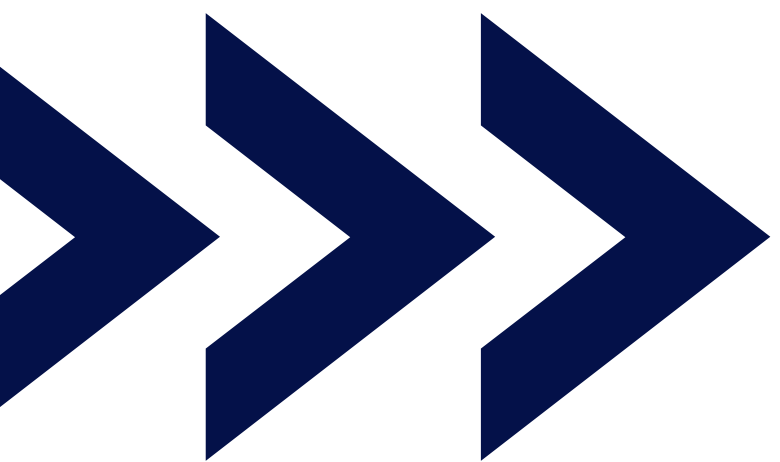
what's OOP?



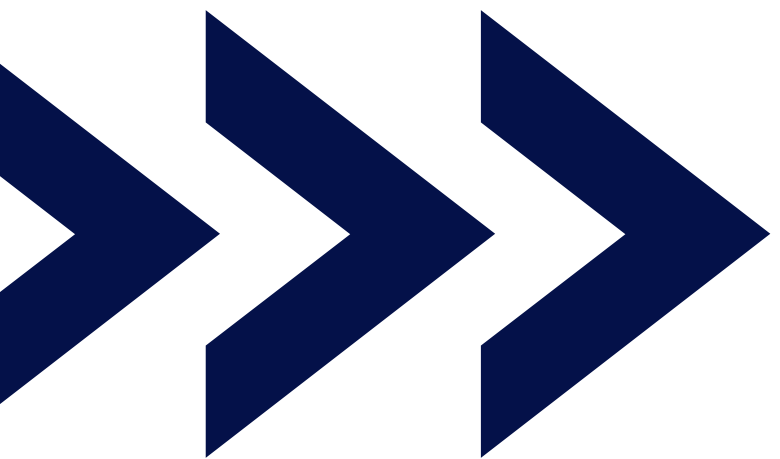
how to create a new object ?



Let's Start!



Introduction to WEB DEV



let's talk about how to?



what's a web dev?

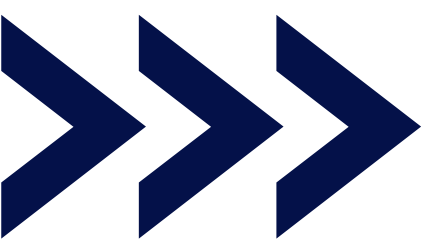
Web development is the process of creating and maintaining websites. It involves building the structure, design, and functionality of a website using various programming languages and technologies.



Types of web development

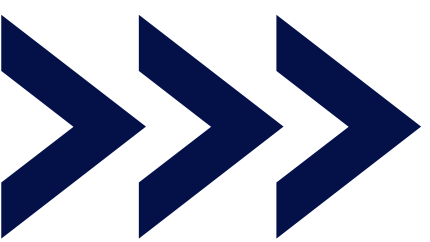
Front-end development: Creating the UI to have the UX.

Back-end development: Creating the server-side logic of a website



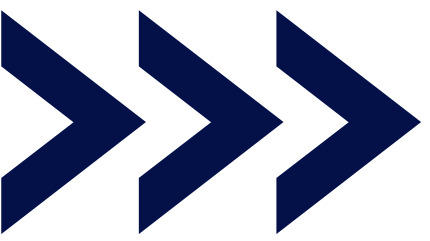
What can we do with it?

- **Create websites.**
- **Develop web applications.**
- **Build web-based tools: (project management, data analysis, or design).**
- **Create online games...**

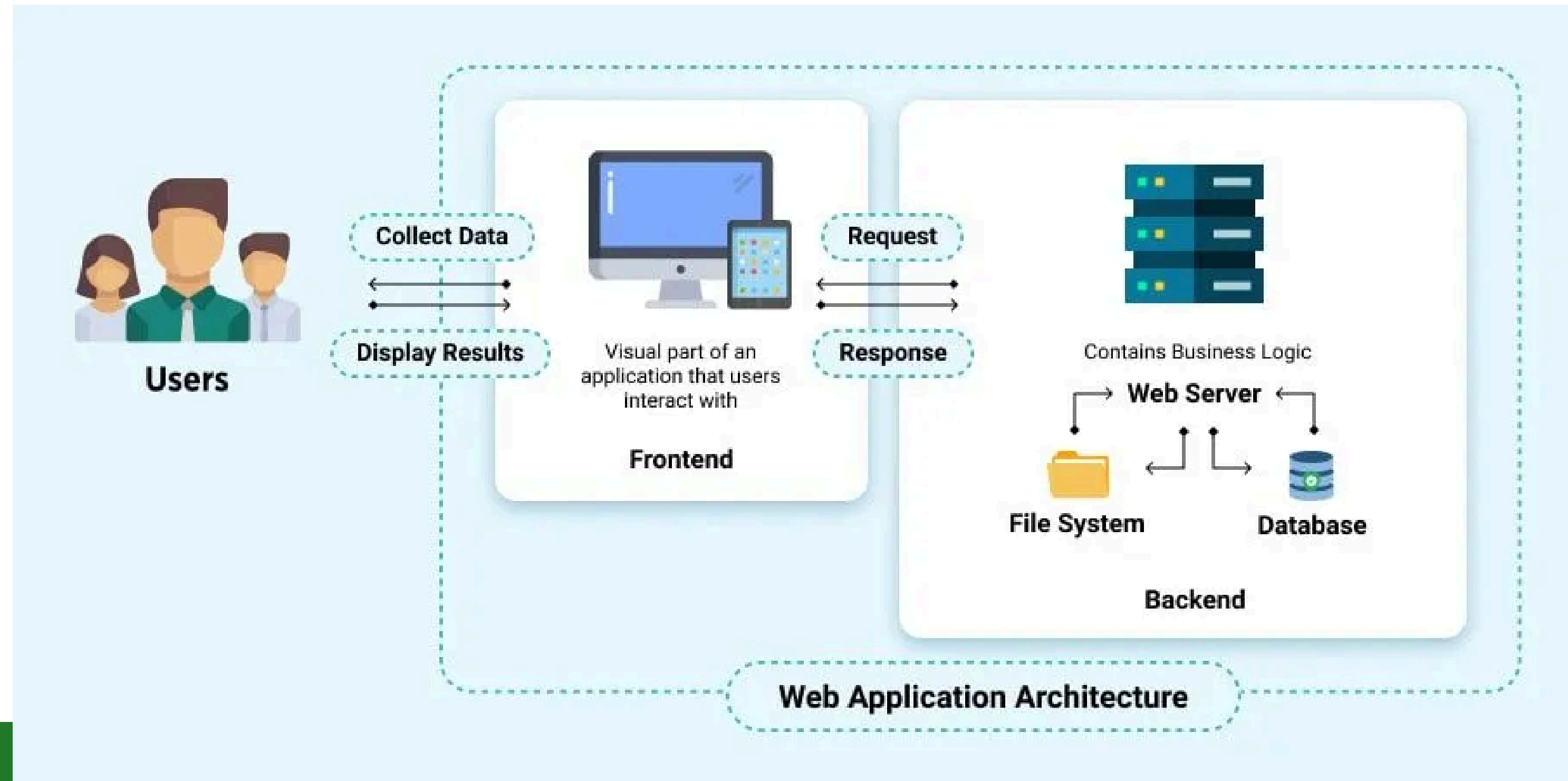


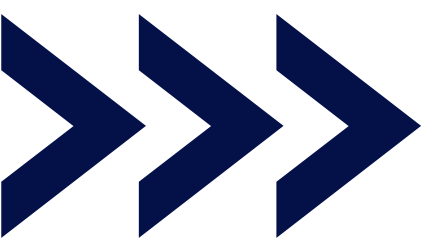
What can we do with it?

- **Create websites.**
- **Develop web applications.**
- **Build web-based tools: (project management, data analysis, or design).**
- **Create online games...**



Web App Architecture

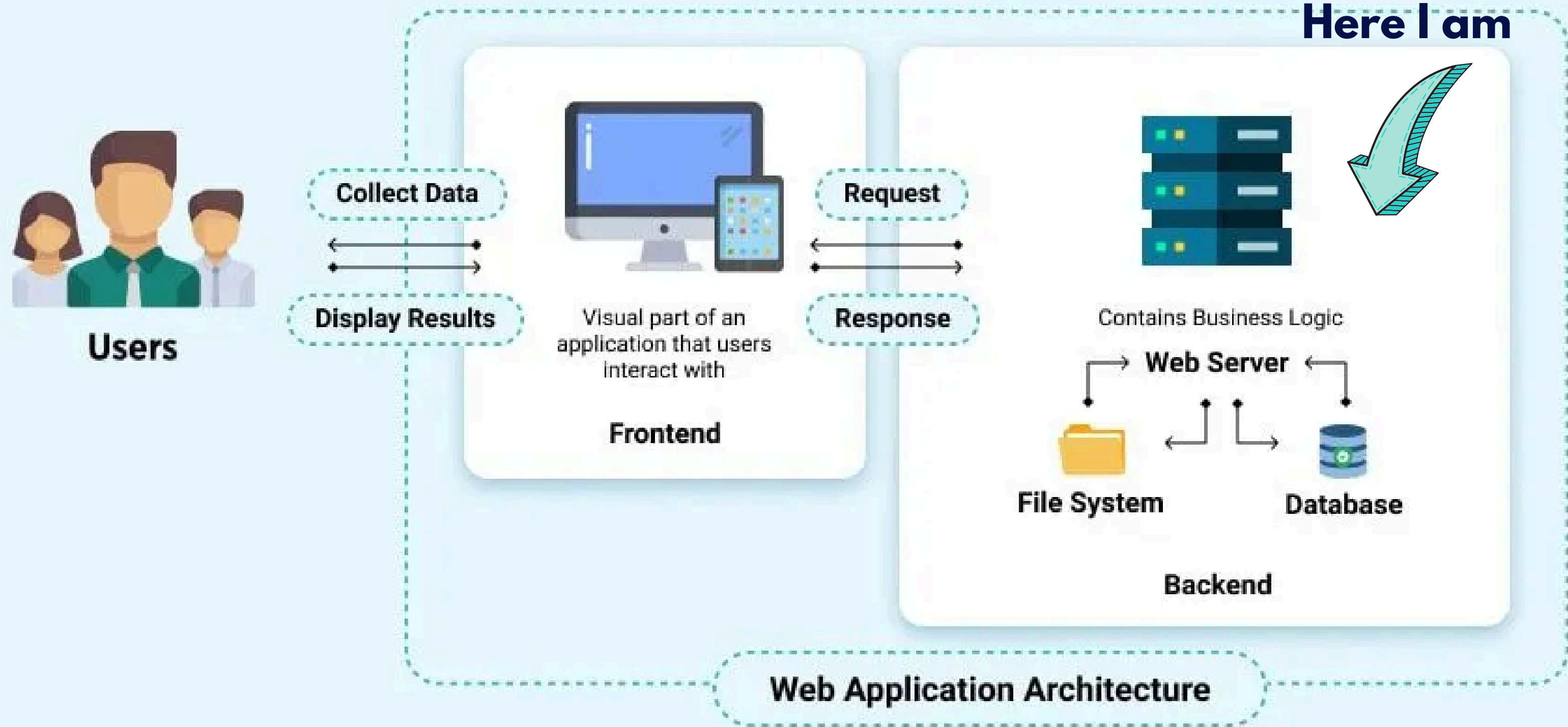




So where locates the Python?

Python where are you?

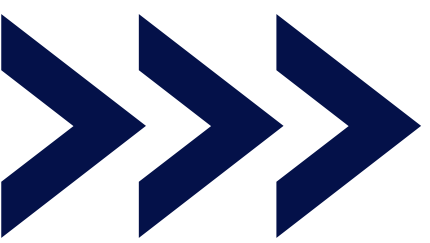
Here I am





Why Choosing Python as your server pal?

- **Readability and Simplicity and Rapidity.**
- **Versatility.**
- **Large and Active Community.**
- **Extensive Libraries and Frameworks (like Flask).**
- **Strong Ecosystem for Data Science and Machine Learning.**
- **Cross-Platform Compatibility.**
- **Integration with Other Technologies**



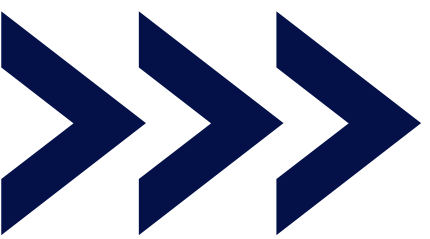
Now the question is how to design all the development process?



Let's play a game?

- **You own your dev cooperation.**
- **You starting a project of building a web app of a school.**
- **What will you do first?**





Defining strategy?

- 1. Understand Project Goals: what's the objective from this app?**
- 2. Conduct Market Research: what's the best solution out there?**
- 3. Create a Detailed Project Plan: tasks, features, functionalities, and timelines.**



Defining strategy?

so what's your strategy from this project?

Defining Architecture?

- 1. Presentation Layer (frontend, backend and the relation between them).**
- 2. Business Logic Layer (data processing, validation, and calculations using programming language like python , java..).**
- 3. Data Access Layer (data base).**
- 4. Integration Layer (API,http...)**



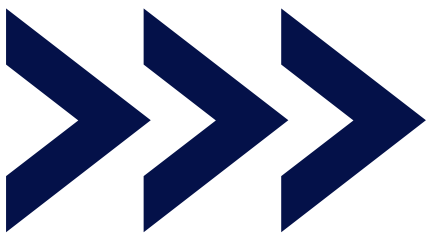
Defining Architecture?

Based on the objective you signed, what is the proposed architecture?

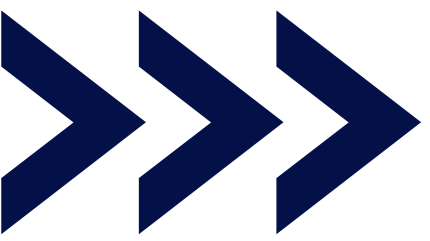


Design Process?

- 1. Use The design languages: UML Diagram (DCL, DCU..)**
- 2. Using the design tools like: LucidChart.**
- 3. UX UI like Figma.**
- 4. Get instant feedback.**
- 5. Repeat, iterate, develop more.**

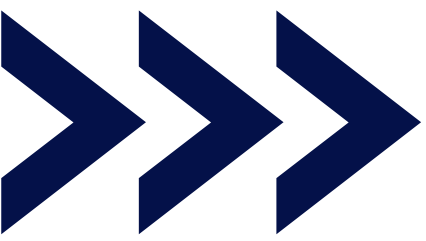


**Feel sure about? Are you clients
satisfied?**



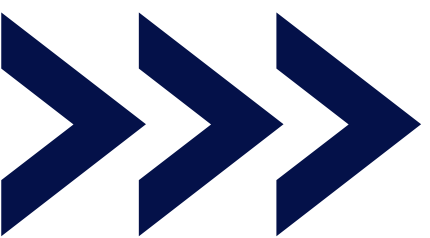
Development Process?

- 1. Frontend: choose your frontend framework or library or language: HTML, CSS, JS, React JS, Next JS...**
- 2. Backend Languages: choose your programming language: Python, Flask, Django, Pyramid, Jva, Node JS...**
- 3. Database: Mysql, Mango, Postgresql or if it's cloud add docker!**
- 4. Don't forget your API**



Frontend?

- 1. HTML: HyperText Markup Language (and it's not a programming language)**
- 2. CSS: Cascad Style Sheet: to add styles and colors.**
- 3. JS: JavaScript: to add Behavior.**
- 4. libraries: React JS**
- 5. Frameworks: Next JS**



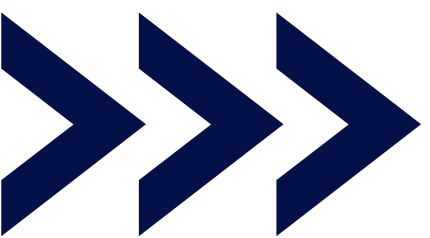
Backend

Server Side: Python, Node js... or use Frameworks Like Flask..



Whats' Flask?

Flask is a popular Python web framework known for its simplicity, flexibility, and lightweight nature. It provides a solid foundation for building web applications while giving developers significant control over the development process.



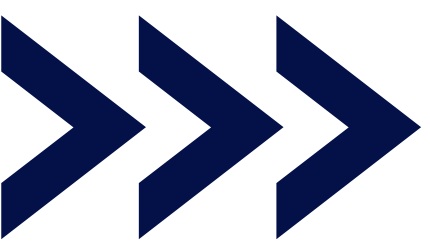
Key Features

- **Minimalism.**
- **Flexibility with all types of architecture.**
- **Microframework.**
- **Routing.**
- **Development Server.**
- **Community Support.**



Installation

```
pip install Flask
```

Basic Application

```
from flask import Flask

app = Flask(__name__)

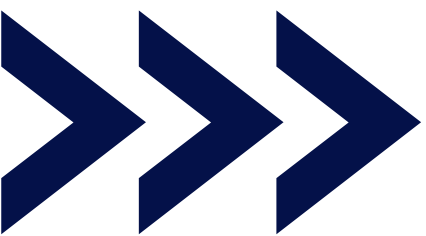
@app.route('/')
def hello_world():
    return 'Hello, World!'

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

 **run the app**

`python app.py`

this will start the app at: `http://127.0.0.1:5000/`

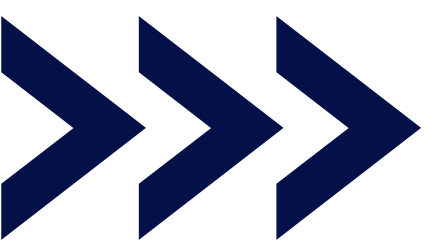


Applications that were built by Flask:

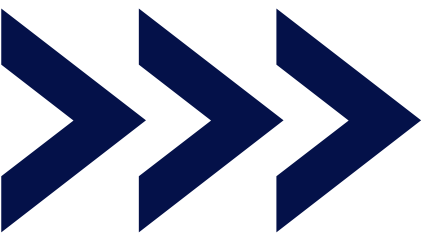
Trivago.

Samsung.

Netflix...

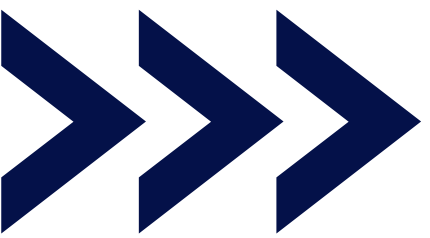


Connecting aInterface with Python?



HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Hello, World!</title>
</head>
<body>
  <h1>Hello, World!</h1>
  <p>This is a simple HTML page.</p>
</body>
</html>
```



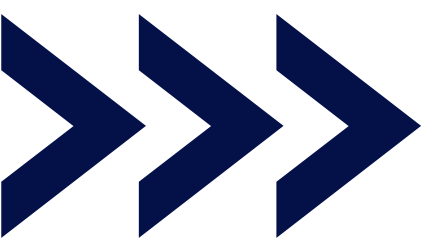
Python

```
from flask import Flask, render_template

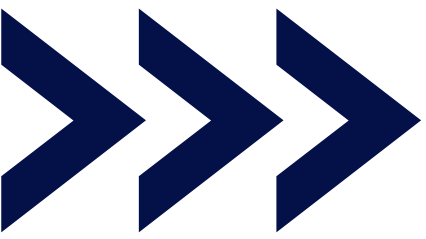
app = Flask(__name__)

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

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



Don't forget to run!



HTML

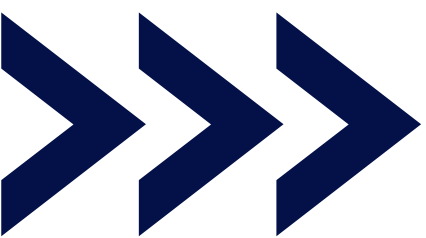
```
<!DOCTYPE html>
<html>
  <head>
    <title>Accueil</title>
  </head>
  <body>
    <h1>{{ message }}</h1>
  </body>
</html>
```




Python

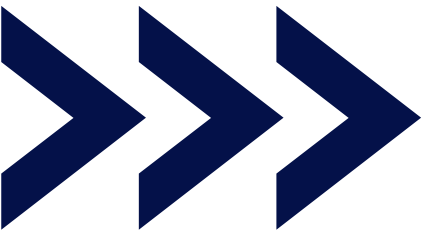
```
from flask import render_template

@app.route("/")
def home():
    return render_template("index.html", message="Bienvenue sur mon site web!")
```



app.route()

app.route() decorator is used to define URL routes within a Flask web application. It associates a specific URL pattern with a Python function that will be executed when that URL is accessed.



app.route()

```
from flask import Flask

app = Flask(__name__)

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

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



Till Next Week!



See you then!

