

# Advanced Programming with Python

## Session 1

Pepe García [jgarciah@faculty.ie.edu](mailto:jgarciah@faculty.ie.edu)

# Advanced Programming with Python. Session 1

Pepe García

[jgarciah@faculty.ie.edu](mailto:jgarciah@faculty.ie.edu)

Ask me anything

# About the course

- 15 sessions

# About the course

- 15 sessions
- 1 workgroup assignment

# About the course

- 15 sessions
- 1 workgroup assignment
- Individual work (I'll be grading an individual project from you. It can be the term integration project, or something else.)



# Syllabus

# SESSION 1

Course presentation. in this session we will introduce the course, the syllabus, materials we're going to use, and grading system. Backend development in Python. We will understand what are the latest tools of Backend development in Python, from libraries for creating web servers, to others helping with the creation of development environments such as pipenv.



## HTTP in Python

We will use this session to understand the basics of web servers in Python using Flask, and how to create HTTP clients with Python. We will learn the following subjects:

- HTTP Request-Response cycle

## HTTP in Python

We will use this session to understand the basics of web servers in Python using Flask, and how to create HTTP clients with Python. We will learn the following subjects:

- HTTP Request-Response cycle
- HTTP Status codes

## HTTP in Python

We will use this session to understand the basics of web servers in Python using Flask, and how to create HTTP clients with Python. We will learn the following subjects:

- HTTP Request-Response cycle
- HTTP Status codes
- Flask routing

## HTTP in Python

We will use this session to understand the basics of web servers in Python using Flask, and how to create HTTP clients with Python. We will learn the following subjects:

- HTTP Request-Response cycle
- HTTP Status codes
- Flask routing
- Rendering JSON



## HTML Templating

How do we use HTML templates with Flask.

## Lab: creating a dynamic website

In this lab, we'll work in class to wind up creating a dynamic website in Flask, that uses HTML templates, loads CSS stylesheets, and reacts to dynamic routes.

## Web servers - Authentication

In this session we will learn about how to implement authentication in web applications.



## Connecting to databases

In these sessions we will learn how to make our Python applications connect to databases. We will also learn advanced techniques that help us deal with data in the database using object oriented programming.



## Lab: building a Twitter clone in Python

We will use this session to do some hands-on work. We will tackle a small project in class in which we will create a Twitter clone with Python.

## Group assignment

In this session students will do a group assignment. We will have time in class for working on it and ask questions.

## Building RESTful APIs in Python

APIs are a way for services online to provide an programatic interface so that they can be connected to different programs. In this session we'll learn what do we mean by RESTful and how to adapt our servers to interact in a RESTful fashion.

## Deployment

In this session we will learn how to deploy our flask applications to the cloud.

## Data Intensive Web Application

In this session we will introduce a new framework for data oriented web applications. Using this framework it will be easier to create dashboards and data rich applications with Python.

## Lab: Data Intensive Web Application

In this session we'll work together to get to build a dashboard application in Python



## QA session

In this session we will do a whirlwind tour over what we have learned in the course and we will have time to answer questions students may have



# Grading criteria

Criteria	Score
Class participation	<b>10%</b>
Workgroups	<b>30%</b>
Individual work	<b>60%</b>





What are we really going to learn in this course?

Let's draw!

What happens when you type an address in your browser?



HTTP is a request-response protocol. HTTP **clients send requests** and HTTP **servers answer with responses**.

## Uniform Resource Locators.



Depending on the intention of the request, HTTP describes different methods:



Depending on the intention of the request, HTTP describes different methods:

method	intention
<b>GET</b>	read to a resource
<b>POST</b>	update a resource
<b>PUT</b>	create a resource
<b>DELETE</b>	delete a resource

# Repo for today

<https://github.com/mcsbt-app-2022/session-1>