Project of Software Development

Lesson 4 - 6Feb
Web Development & Introduction to OutSystems

Lisbon School of Economics & Management
Universidade de Lisboa

LISBOA | UNIVERSIDADE DE LISBOA

Requirements for the group members

Each group will have 6 member. For the students also enrolled in e-Business Models and Technologies (EBUSI), create 5 groups which must meet the following requirements:

- 3 groups with 3 students from GSI (**pool A**) + 2 students non-GSI (**pool B**) + 1 student not enrolled in EBUSI (**pool C**)
- 2 groups with 4 students from GSI (**pool A**) + 1 student non-GSI (**pool B**) + 1 student not enrolled in EBUSI (**pool C**)

Students from pool C can join a group only after the presentation of the business ideas, in order to decide the group according to the business idea.

Pool A (GSI Students):

63132	ALEXANDER OKOLIE
62779	ALICE HACHENBURG
63091	ANA GARRIDO LIMA
63119	ASAD ULLAH
63154	CHINEDU NWANKWO
62753	ELIAS DOMINIK MOHR
62437	FLORIAN STICH
62583	HENGFENG ZHANG
62627	HUGO MARTIM RIBEIRO COSTA
63201	KIBRIA ANTARA ZAMAN
62597	MASAB MUSTAFA
63145	MD. SAJJAD BHUIYAN TAKI
62408	NINA ZIMMER
63189	SIYUAN YUAN
62606	XIHE YU
63162	YAĞMUR ÇELIK
63087	ZEINEB SIALA

Pool B (non-GSI enrolled in EBUSI):

63437	JAROSLAV ARON
58460	ABDUL AZEEZ FAZILUDEEN
62782	HUGO SIMÕES FONTAN
62704	MARK GRADY GRESHAM JR
57964	MARTIM REIS XAVIER
62931	NIKLAS IDRICA FISCHER
62894	RITIKA DHARAMRAJ SHARMA
53520	TAÍSSA DOS SANTOS

Pool C (not enrolled in EBUSI):

62567	FRANCESCO CASAGRANDE
63408	JAN HRUŠÁK
64471	LAIHANG GUO
64430	MALENA AZUL ASTIGUETA
64384	NICOLAS MAUTE

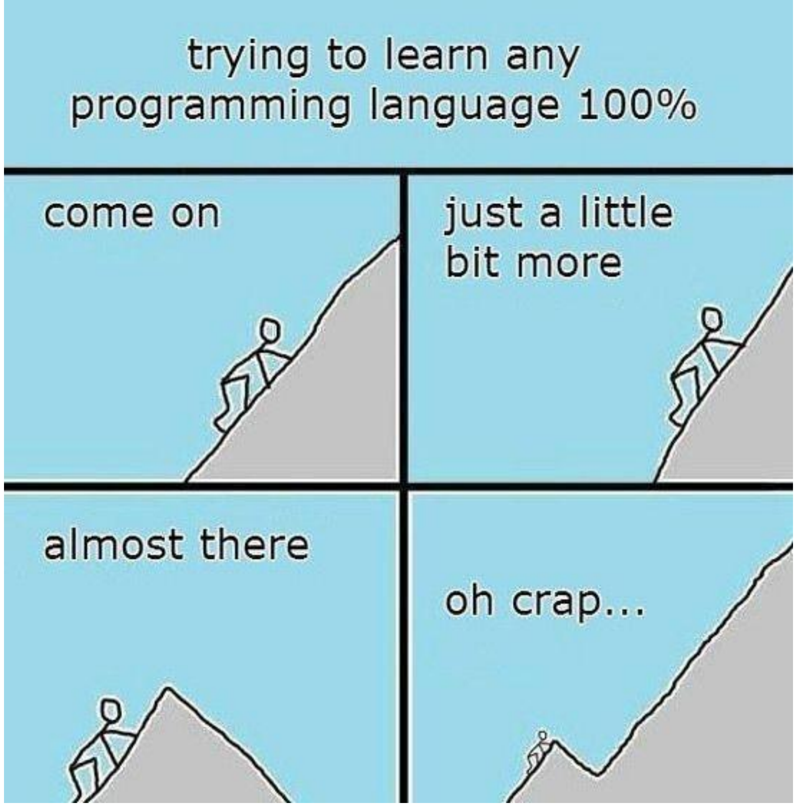
Lecture Topic

Web Development:

- Programming languages
- Programmer’s profiles

Webgraphy:

- Learn Web Development: 7 Basic Steps for Beginners
  https://bootcamp.berkeley.edu/resources/coding/learn-web-development/
- Low Code vs No Code Explained
  https://www.bmc.com/blogs/low-code-vs-no-code/
- OutSystems: OutSystems Overview
  https://www.outsystems.com/training/courses/173/outsystems-overview/?LearningPathId=18
- OutSystems: Service Studio Overview
  https://www.outsystems.com/training/courses/174/service-studio-overview/?LearningPathId=18
- OutSystems: Intro to OutSystems Development
  https://www.outsystems.com/training/courses/122/intro-to-outsystems-development/?LearningPathId=18



ISEG Lisbon School of Economics & Management Universidade de Lisboa

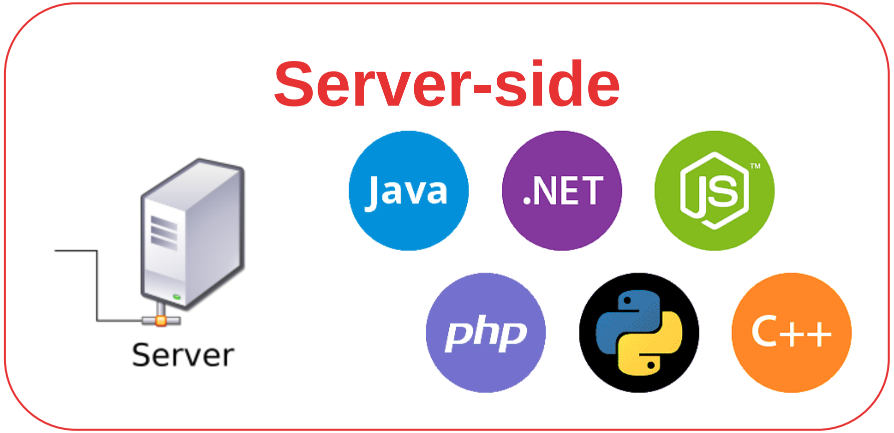
U LISBOA | UNIVERSIDADE DE LISBOA

Web Development

Server-side languages: Java, .Net, Node.js, PHP, Python, Ruby, C++, …

Databases: MS SQL, Oracle DB, PostgreSQL, mySQL, …

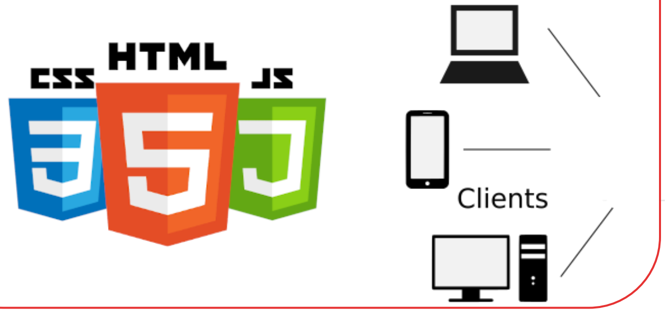
Server-side programming involves managing web servers, interacting with databases and using data analysis — all of which are functions the user doesn't see when interacting with the site



SQL (Structured Query Language) is a query language popular among data scientists and back-end (server-side) developers. It is used to create databases, add new data to existing databases, query and modify data within databases

Web Development

Client-side



Client-side languages: HTML, CSS, JavaScript

Client-side libraries: jQuery, React JS

Responsive Design: Think of all the devices you use in your day-to-day life — laptops, desktops, tablets and smartphones. Every website you create needs to function across every device a user could own. Responsive design is the idea that websites should respond to a user's behaviour and device.

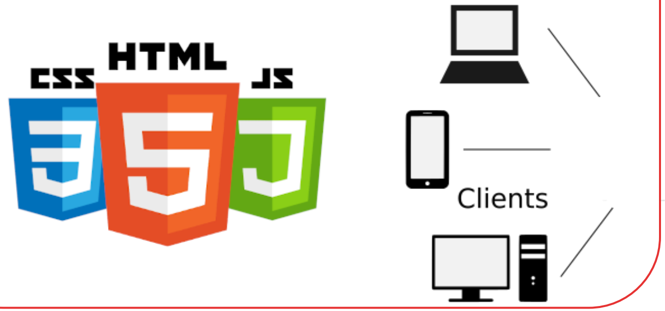
HTML (HyperText Markup Language) allows coders to define a website's basic structure and design. An HTML file tells a browser **what** to display on a device's screen and how elements like paragraphs, lists and images are arranged

CSS (Cascading Style Sheets) changes **how** HTML elements are displayed on a screen. Understanding CSS allows you to create great-looking web pages across all major browsers. You can change a page's layout, colours and fonts, as well as add effects to page elements

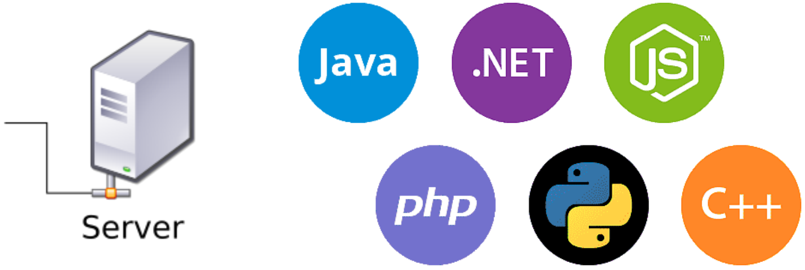
JavaScript is a client-side language that is used alongside HTML and CSS to create dynamic, responsive websites

Web Development

Client-side



Server-side

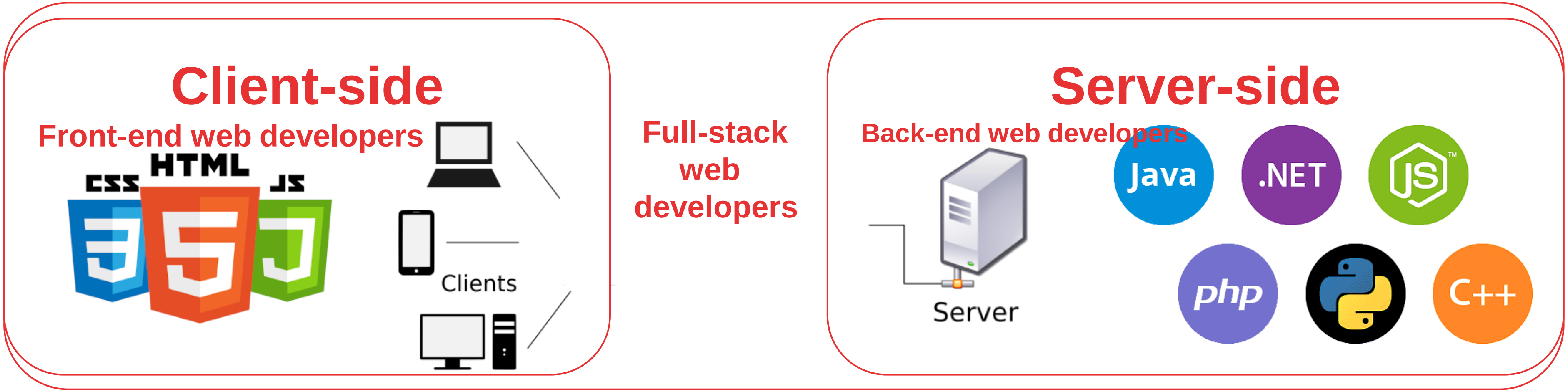


Web Development includes the programming of server-side code which needs to be **compiled** (with some exceptions), **uploaded**, **deployed** (or **published**) into the server to **run** and **debug**.

These server-side programs run in the server to interact with databases, implement business logic, and **reply** to the client with a Web Page

This Web Page is **interpreted** and displayed by the Web Browser

Web Development



Client-side

Front-end web developers

Full-stack web developers

Server-side

Back-end web developers

Front-end web developers focus on a website's client-side functions. They determine how a **website looks** when loaded on the client side. They're responsible for **creating and designing all user-facing website elements** such as menus, buttons and animations that can execute on a client's machine.

Back-end web developers focus on a website's server-side development and **provide the functionality** that supports the frontend. This includes **server-side scripting, database management, and server-side APIs** that provide data to the front end. Programming cycle: build the code, compile it (with some exceptions), upload, deploy (or publish), run and debug.

Full-stack developers work with just that — the "full stack" of development technologies — and have mastery of both front and back end technologies. With enough time, full-stack engineers can create websites and applications entirely on their own. Generally, these software developers use a mix of both front-end and back-end languages.

Types of Programming Languages

- A programming language **is** system of notation for writing computer programs
- A programming language **has** two components: syntax (form) and semantics (meaning)

There are many different ways of classifying programming languages. One way relevant to this course is to distinguish between text-based and graphical (visual) programming languages:

- Text-based formal languages: use textual symbols and written commands to represent code
  - Examples: JavaScript, PHP, Python, Java, C, C++, …
- Visual programming languages: use graphical symbols and visual elements to represent code. Users can drag and drop pre-built programming blocks, or "code blocks," to create a program
  - Examples: OutSystems, Scratch, Apple Shortcuts, Node-RED, …

Text-based programming languages require knowledge of coding syntax, whereas visual programming languages can be easier to learn and use for beginners.

Types of Development Platforms

To program efficiently, a programmer needs an integrated development environment.

An **integrated development environment (IDE)** is a software application that provides comprehensive facilities for computer programmers to develop, test, and debug software.

An IDE for a text-based programming language typically includes a source code editor, a debugger, a build automation tool, and other tools to make the process of writing and testing code, as well as uploading and deploying applications, more efficient

Types of Development Platforms

The integrated development environment (IDE) of a visual programming language is provided by the language vendor. It typically fits into one of the following types of development platforms:

- **Low-code development platform**: provides a development environment for creating application software through a graphical user interface. It can produce fully functional applications, or require additional coding for specific situations (typically no more than 10% of the coding process)
- **No-code development platform**: provides a development environment for creating application software through graphical user interfaces and configuration. Requires no code to be written and usually provides pre-built templates from which programmers can build applications.

OutSystems Online Training: Becoming a Web Developer

Short review and Q&A: OutSystems platform

Learning goals

Familiarisation with the OutSystems platform and development environment

To understand the main Web Applications' related jargon:

- Server-side and client-side programing
- Front-end languages: HTML, CSS, JavaScript
- Back-end languages: SQL, Java, .Net, Node.js, PHP, Python, Ruby, C++

To understand the programming cycle: build the code, compile it (if compilable language), upload, deploy (or publish), run and debug.

Types of Programming Languages: text-based and visual programming languages

Types of Development Platforms: low-code and no-code

Homework YouTube

Alternative 1 (Portuguese 🇵🇹 [Aprenda a programar]
https://www.youtube.com/playlist?list=PLY-9oEzuBhdfzxLxLeTr56YG8Vqu-Y0dl

- Aprenda a Programar - #06 Estruturas de Decisão I: 21 minutes ↗
- Aprenda a Programar - #07 Estruturas de Decisão II: 15 minutes ↗
- Aprenda a Programar - #08 Estrutura Escolha Caso (switch case): 16 minutes ↗

Homework YouTube

Alternative 2 (English 🇬🇧 CodeWithPraveen [Fundamentals of Programming Languages]

https://www.youtube.com/watch?v=F7CWjuaC6gw&list=PLb__S-rkKhexdiJomXSGeqQ46c_MUTPaj

- Fundamentals of Programming Languages #4 | Understanding Conditional Statements (24 minutes) ↗

Homework outsystems

OutSystems Online Training: Becoming a Web Developer
https://learn.outsystems.com/training/journeys/web-developer-662

UI Development 101 (45 minutes) ↗
https://learn.outsystems.com/training/journeys/web-developer-662/ui-development/o11/561

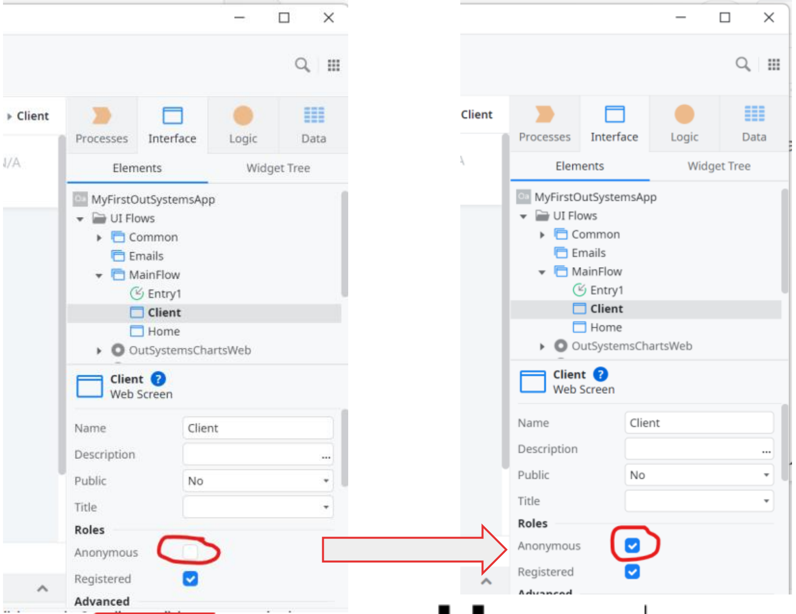
1. UI Development
2. Widgets
3. Exercise (simple calculator)

⚠ To avoid name collision with your colleagues, **always start the name of your apps** with your student nr: l58460_

Homework outsystems

After having implemented your Web App as described in the OutSystems exercise (item #03), do the following:

1. Add the “anonymous” role to all your Web Screens:
    a. Click on the Web Screen (widget tree on the right side)
    b. The attributes area will open. Click on “Anonymous”
    c. Repeat (a) and (b) for each Web Screen you created
    d. Publish and test
2. Submit the address of your Web Application by following up on the email



- MyFirstOutSystemsApp
  - UI Flows
    - Common
    - Emails
    - MainFlow
      - Entry1
      - Client
      - Home
    - OutSystemsChartsWeb

Client
Web Screen

Name	Client
Description	
Public	No
Title	

Roles

	Before	After
Anonymous	□	☑
Registered	☑	☑

Advanced

Expected total effort: 90 to 120 minutes