

WEB DEVELOPMENT

Advanced Programming I

Author: Eng. Carlos Andrés Sierra, M.Sc.
carlos.andres.sierra.v@gmail.com

Computer Engineer
Lecturer
Universidad Distrital Francisco José de Caldas

2024-I



Outline

1 Sockets and Services

2 Layers Architecture

3 Web User Interface



Outline

1 Sockets and Services

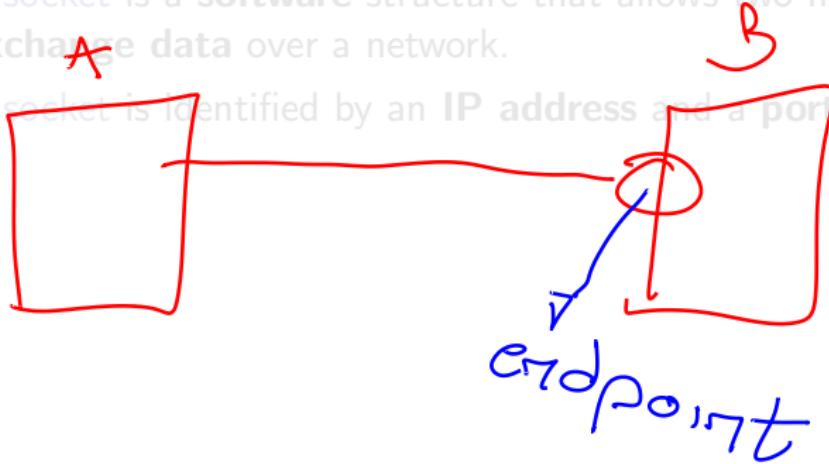
2 Layers Architecture

3 Web User Interface



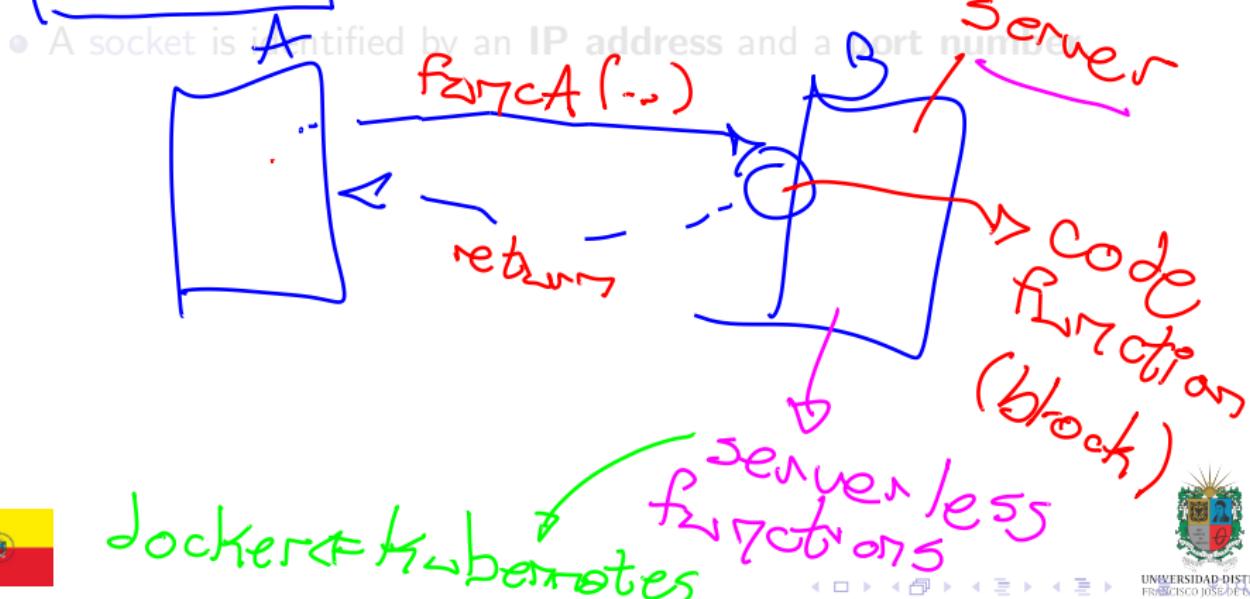
Sockets

- A **socket** is an **endpoint** for communication between two machines over a network.
- A socket is a **software** structure that allows two machines to **exchange data** over a network.
- A socket is identified by an **IP address** and a **port number**.



Sockets

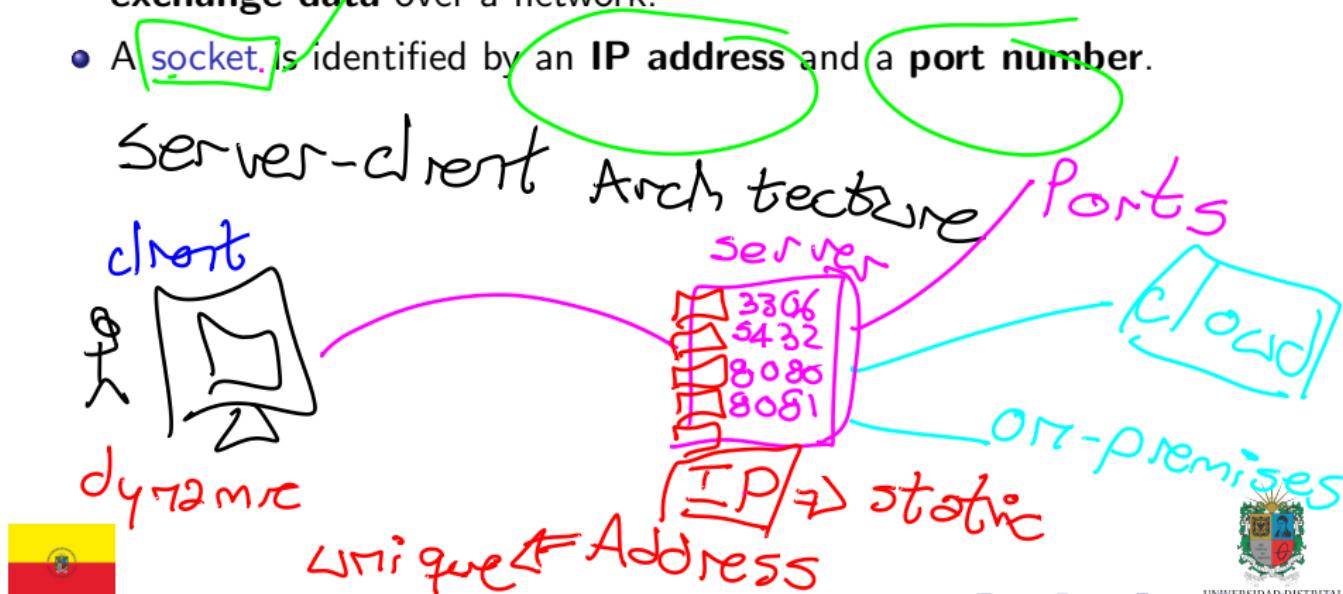
- A **socket** is an **endpoint** for communication between two machines over a network.
- A **socket** is a **software** structure that allows two machines to **exchange data** over a network.



Sockets

- A **socket** is an **endpoint** for communication between two machines over a network.
 - A **socket** is a **software** structure that allows two machines to **exchange data** over a network.
 - A **socket** is identified by an **IP address** and a **port number**.

localhost:8080

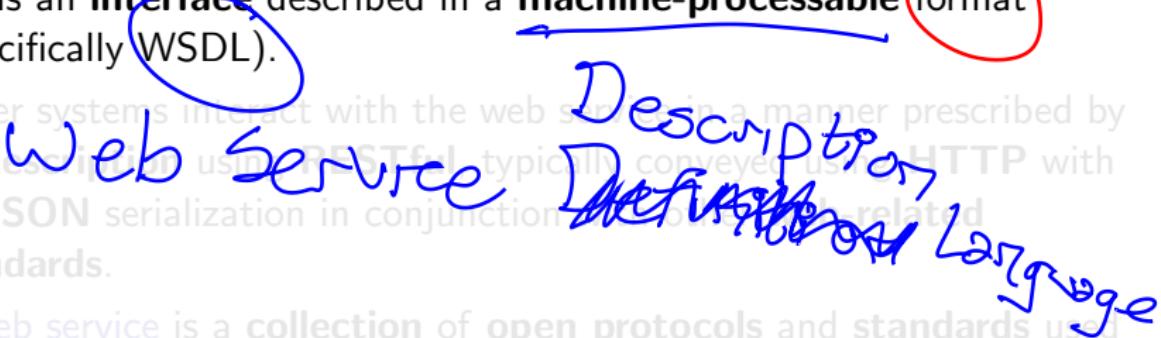


Web Services

- A **web service** is a **software** system designed to support **interoperable** machine-to-machine interaction over a network.
 - It has an **interface** described in a **machine-processable** format (specifically WSDL).
 - Other systems interact with the web service in a manner prescribed by its **description**, using **RESTful**, typically conveyed using **HTTP** with **JSON** serialization in conjunction with other **web-related standards**.
 - A web service is a **collection** of open **protocols** and **standards** used for **exchanging data** between applications or systems.
 - Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the **Internet** in a manner similar to **inter-process communication** on a single computer.
-
- A diagram illustrating Web Services architecture. Two hosts, A and B, are connected via the Internet. Host A is labeled "Client" and host B is labeled "Server". A blue curved arrow points from host A to host B, representing data exchange. A pink circle on host B is labeled "Socket" and "TCP Port". Below the hosts, the word "Sagas" is written in blue.



Web Services

- A **web service** is a **software** system designed to support **interoperable** machine-to-machine interaction over a network.
- It has an **interface** described in a **machine-processable** format (specifically WSDL). 
- Other systems interact with the web service in a manner prescribed by its description using RESTful typically conveyed using HTTP with an JSON serialization in conjunction with other related standards.
- A web service is a **collection** of **open protocols** and standards used for **exchanging data** between applications or systems.
- Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the **Internet** in a manner similar to inter-process communication on a single computer.



Web Services

- A **web service** is a **software** system designed to support **interoperable** machine-to-machine interaction over a network.
- It has an **interface** described in a **machine-processable** format (specifically WSDL).
- Other systems interact with the web service in a manner prescribed by its **description** using **RESTful**, typically conveyed using **HTTP** with an **JSON** serialization in conjunction with other **web-related standards**.
IP
- A web service is a collection of open protocols and standards used for exchanging data between applications or systems.
key : value
- Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the **Internet** in a manner similar to **inter-process communication** on a single computer.

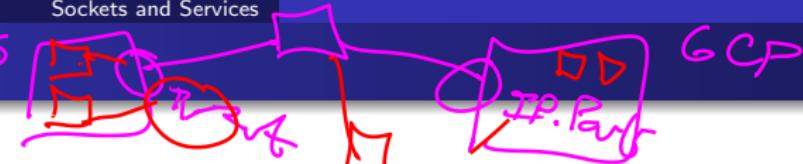


Web Services

- A **web service** is a **software** system designed to support **interoperable** machine-to-machine interaction over a network.
- It has an **interface** described in a **machine-processable** format (specifically WSDL).
- Other systems interact with the web service in a manner prescribed by its **description** using **RESTful**, typically conveyed using **HTTP** with an **JSON** serialization in conjunction with other **web-related standards**.
- A **web service** is a **collection** of **open protocols** and **standards** used for **exchanging data** between applications or systems.
- Software applications written in various programming languages and running on various platforms can use web services to **exchange data** over computer networks like the **Internet** in a manner similar to **inter-process communication** on a single computer.



Web Services



- A **web service** is a **software** system designed to support **interoperable** machine-to-machine interaction over a network.
- It has an **interface** described in a **machine-processable** format (specifically WSDL).
- Other systems interact with the web service in a manner prescribed by its **description** using **RESTful**, typically conveyed using **HTTP** with an **JSON** serialization in conjunction with other **web-related standards**.
- A **web service** is a **collection** of **open protocols** and **standards** used for **exchanging data** between applications or systems.
- **Software applications** **written** in various programming languages and running on various platforms can use web services to **exchange data** over computer networks like the **Internet** in a manner similar to **inter-process communication** on a single computer.



Outline

1 Sockets and Services

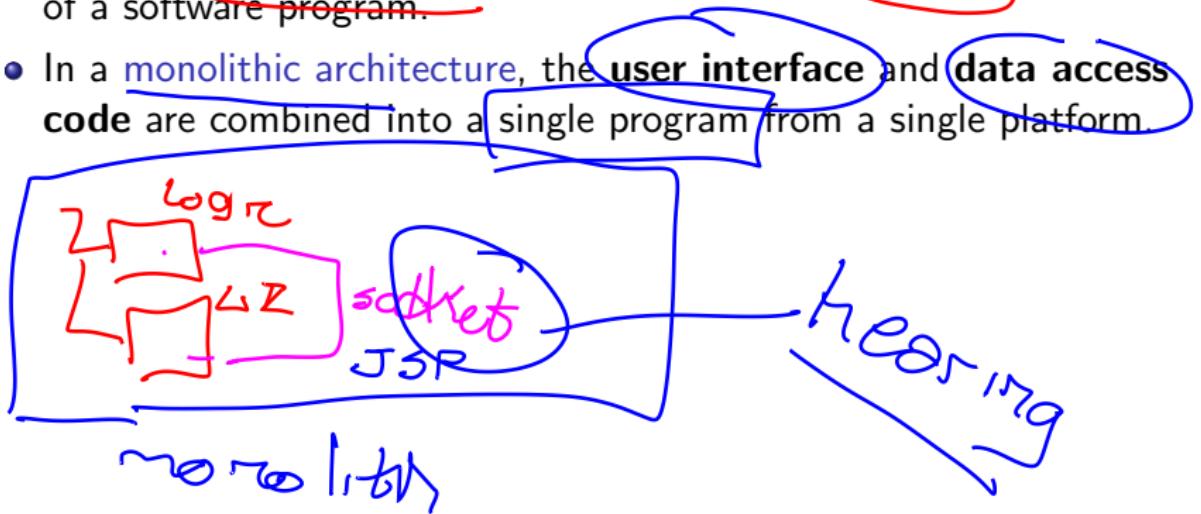
2 Layers Architecture

3 Web User Interface



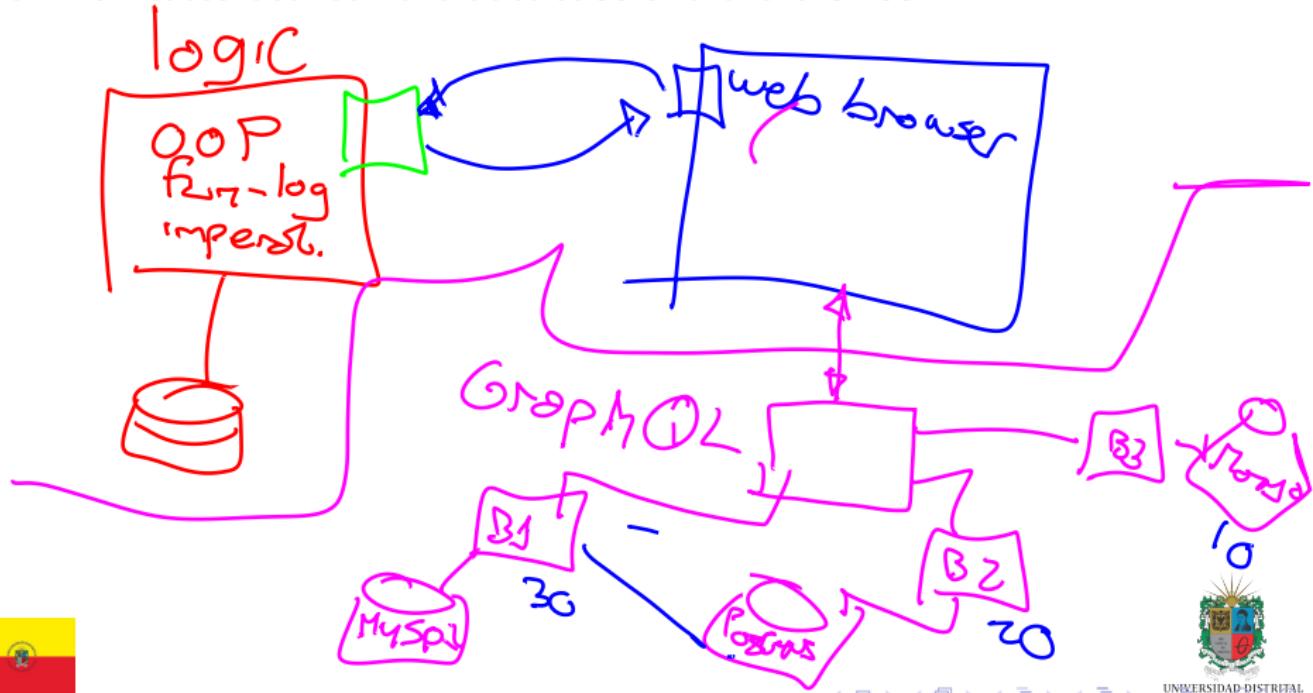
Monolithic Architecture

- A **monolithic architecture** is a **traditional** unified model for the design of a software program.
- In a **monolithic architecture**, the **user interface** and **data access code** are combined into a single program from a single platform



BackEnd Layer

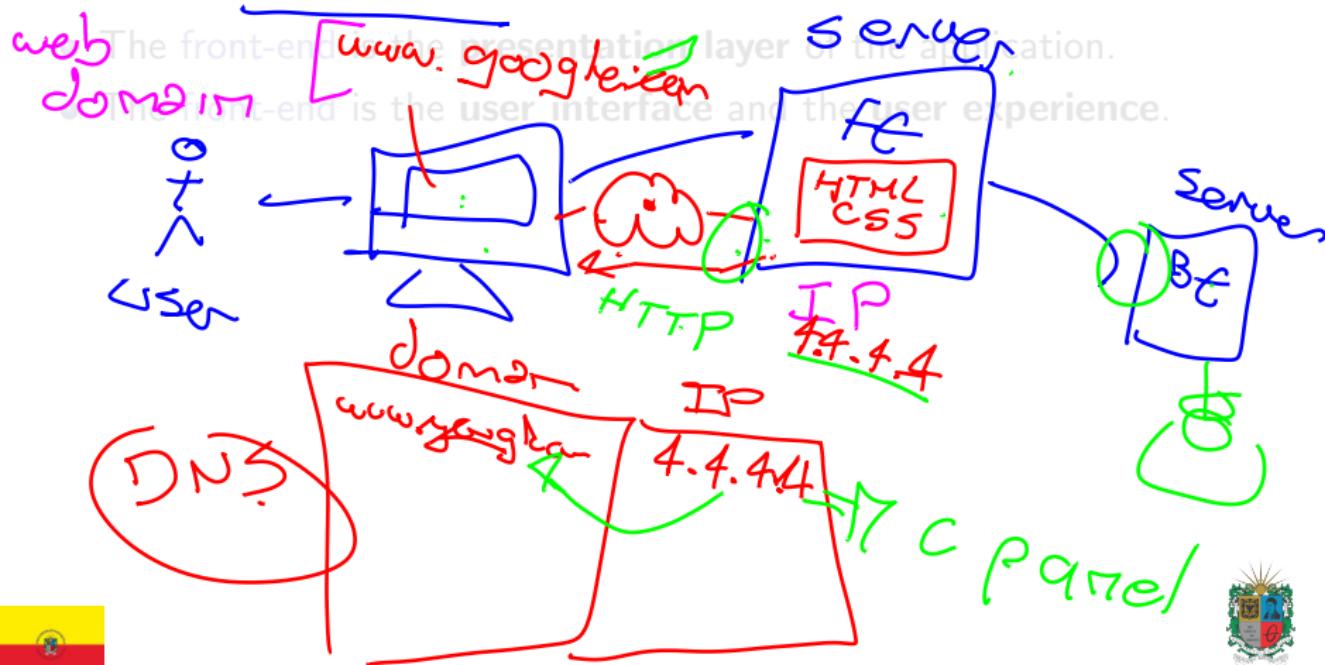
The **back-end** is the **server-side** of the application and everything that communicates between the **database** and the **browser**.



FrontEnd Layer

HTTP(S) - SSL

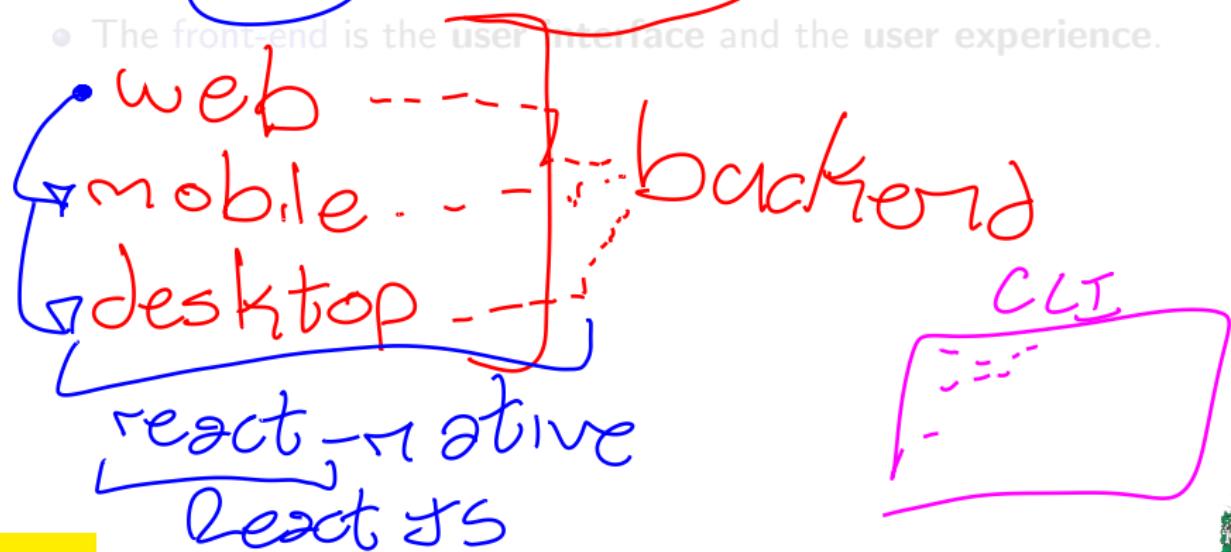
- The front-end is the client-side of the application and everything that the user interacts with.



FrontEnd Layer

- The **front-end** is the **client-side** of the application and everything that the user interacts with.
- The **front-end** is the **presentation layer** of the application.
- The front-end is the **user interface** and the **user experience**.

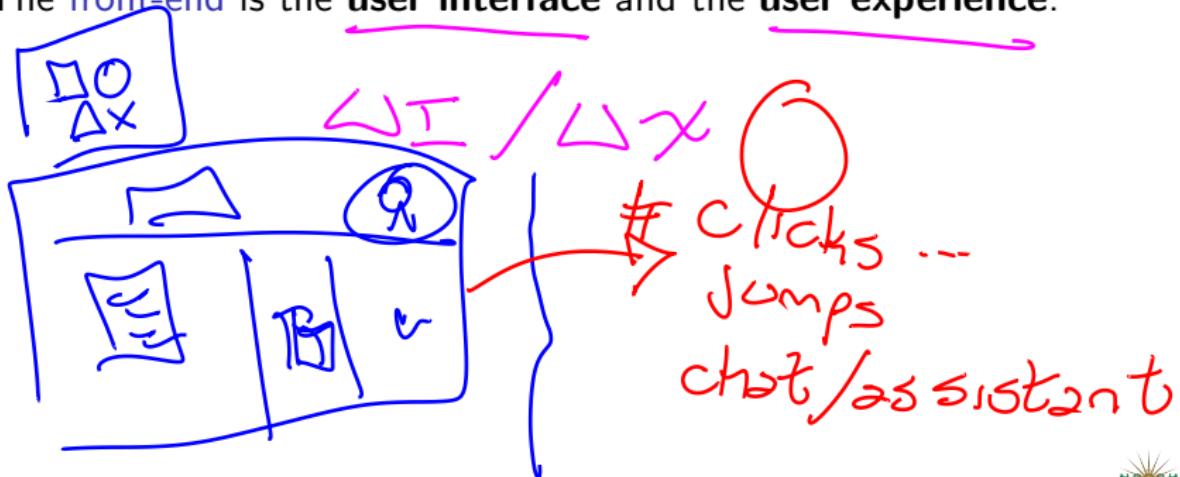
GUI \Rightarrow UI



FrontEnd Layer

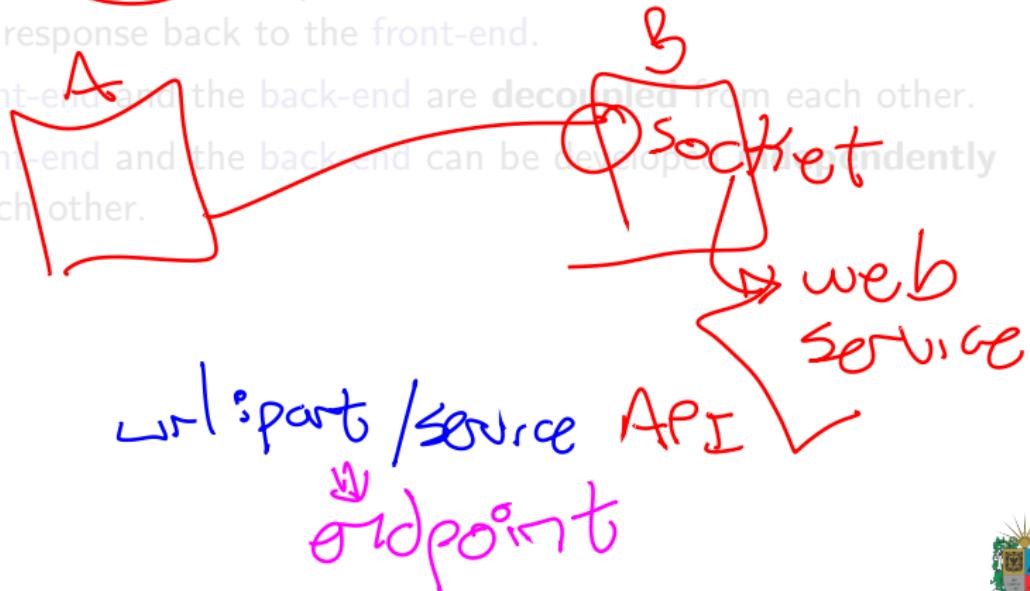
≡

- The **front-end** is the **client-side** of the application and everything that the user interacts with.
- The **front-end** is the **presentation layer** of the application.
- The **front-end** is the **user interface** and the **user experience**.



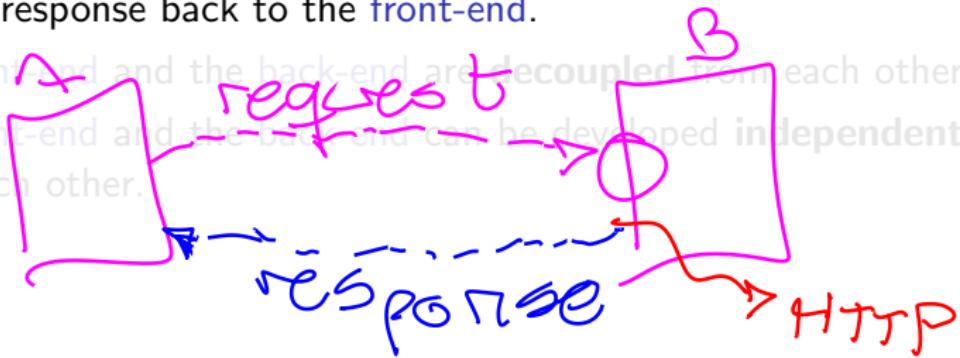
Communication

- The front-end and the back-end communicate with each other through web APIs.
- The front-end sends a request to the back-end and the back-end sends a response back to the front-end.
- The front-end and the back-end are decoupled from each other.
- The front-end and the back-end can be developed independently from each other.



Communication

- The front-end and the back-end communicate with each other through web APIs.
- The front-end sends a request to the back-end and the back-end sends a response back to the front-end.
- The front-end and the back-end are decoupled from each other.
- The front-end and the back-end can be developed independently from each other.



Communication

- The front-end and the back-end communicate with each other through web APIs.
- The front-end sends a request to the back-end and the back-end sends a response back to the front-end.
- The front-end and the back-end are decoupled from each other.
- The front-end and the back-end can be developed independently from each other.



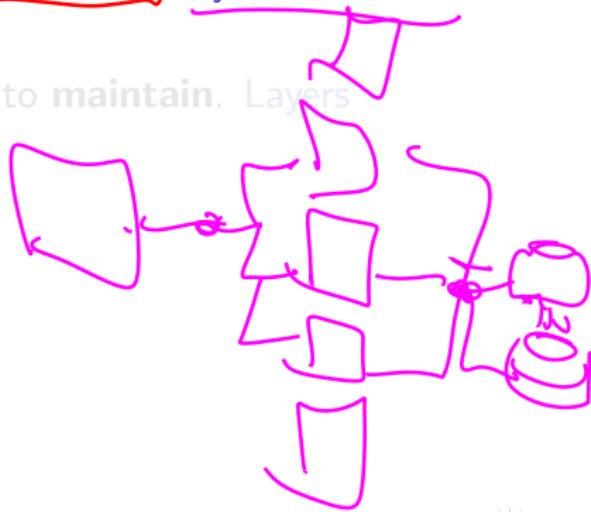
Prons and Cons

- **Monolithic Architecture** is **simple** and **easy** to develop. Layers architecture is **complex** and **difficult** to develop.
- Monolithic Architecture is difficult to scale. Layers architecture is easy to scale.
web server - LGSS - HTTP
- Monolithic Architecture is difficult to maintain. Layers architecture is easy to maintain.
Open - Docker



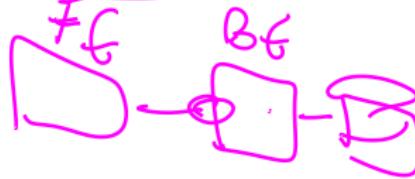
Prons and Cons

- **Monolithic Architecture** is **simple** and **easy** to develop. Layers architecture is **complex** and **difficult** to develop.
- **Monolithic Architecture** is **difficult to scale**. Layers architecture is **easy to scale**.
- Monolithic Architecture is difficult to maintain. Layers



Prons and Cons

- **Monolithic Architecture** is **simple** and **easy** to develop. Layers architecture is **complex** and **difficult** to develop.
- **Monolithic Architecture** is **difficult** to **scale**. Layers architecture is **easy** to **scale**.
- **Monolithic Architecture** is **difficult** to **maintain**. Layers architecture is **easy** to **maintain**.



Outline

1 Sockets and Services

2 Layers Architecture

3 Web User Interface



What is a web GUI?

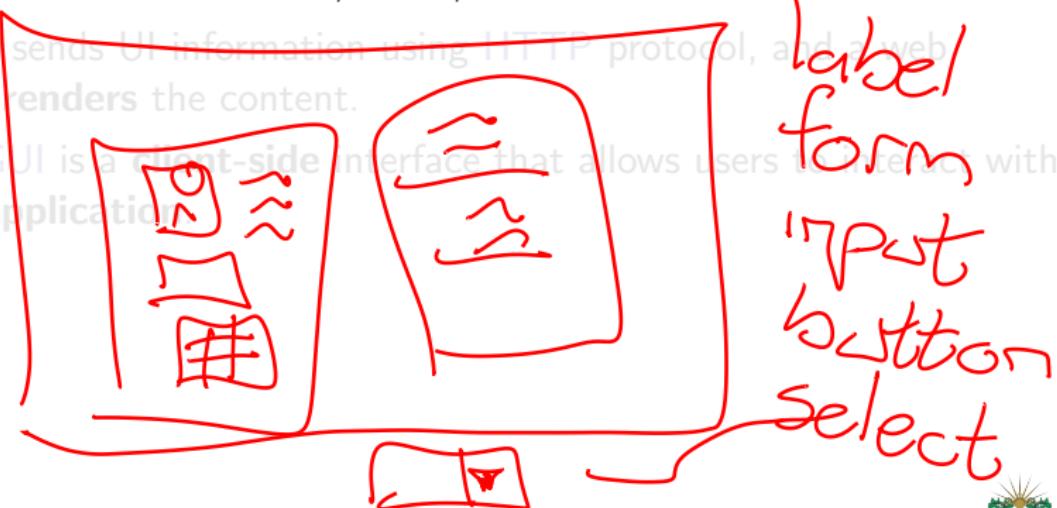
- A web GUI is a **graphical user interface** that is displayed in a **web browser**.
- Also, a GUI in general is a **user interface** that includes **graphical elements** such as **windows**, **icons**, and **buttons**.
- A server sends UI information using **HTTP** protocol, and a web browser **renders** the content.
- A web GUI is a **client-side** interface that allows users to interact with a **web application**.



What is a web GUI?

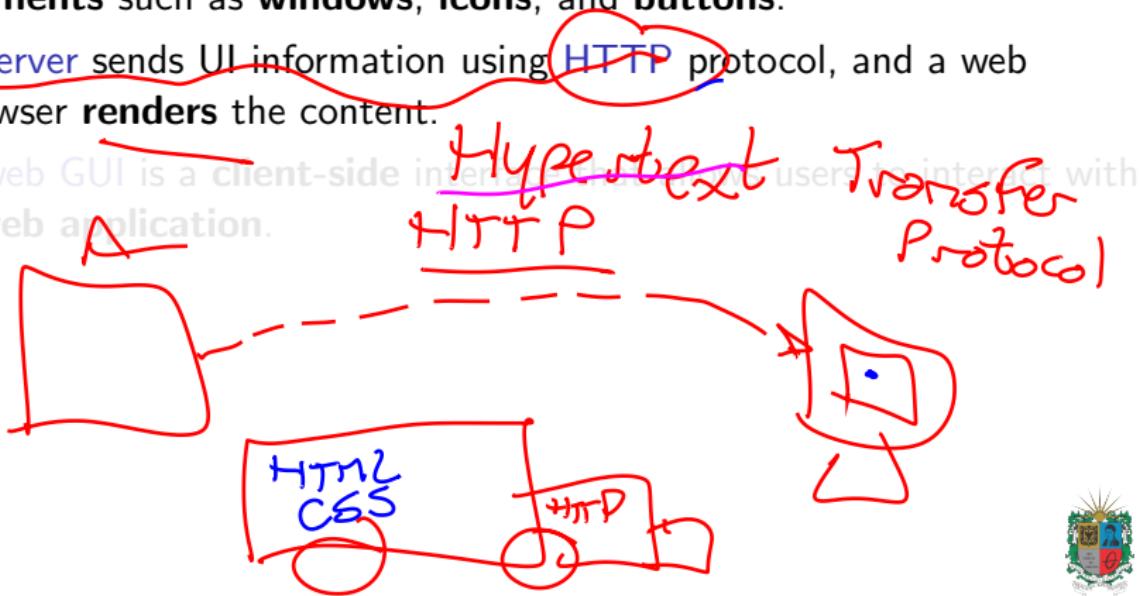
- A **web GUI** is a **graphical user interface** that is displayed in a **web browser**.
- Also, a **GUI** in general is a **user interface** that includes **graphical elements** such as **windows**, **icons**, and **buttons**.

- A server sends UI information using HTTP protocol, and a web browser **renders** the content.
- A web GUI is a client-side interface that allows users to interact with a web application.



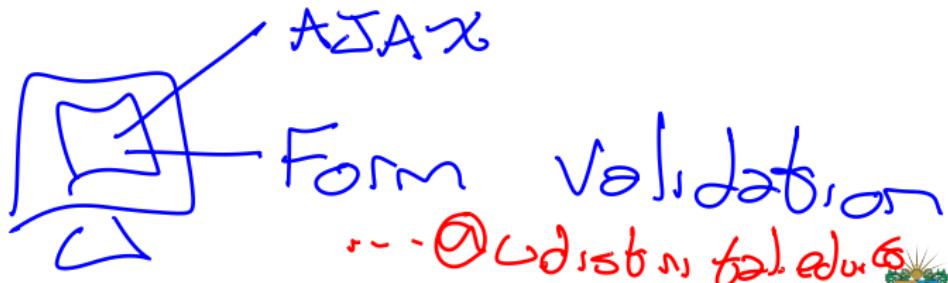
What is a web GUI?

- A **web GUI** is a **graphical user interface** that is displayed in a **web browser**.
- Also, a **GUI** in general is a **user interface** that includes **graphical elements** such as **windows, icons, and buttons**.
- A **server** sends UI information using **HTTP** protocol, and a web browser **renders** the content.
- A web GUI is a client-side interface that interacts with a user interface with a web application.



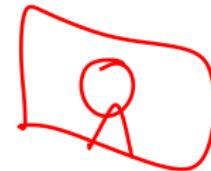
What is a web GUI?

- A **web GUI** is a **graphical user interface** that is displayed in a **web browser**.
- Also, a **GUI** in general is a **user interface** that includes **graphical elements** such as **windows, icons, and buttons**.
- A **server** sends UI information using **HTTP** protocol, and a web browser **renders** the content.
- A **web GUI** is a **client-side** interface that allows users to interact with a **web application**.



Render and Web Standards

- The rendering engine is a **software component** that displays the requested content.
- The rendering engine reads the **HTML** and **CSS** code and renders the content.
- The W3C is an international community that develops **open standards** to ensure the long-term growth of the Web.
- The W3C is the main international standards organization for the World Wide Web.



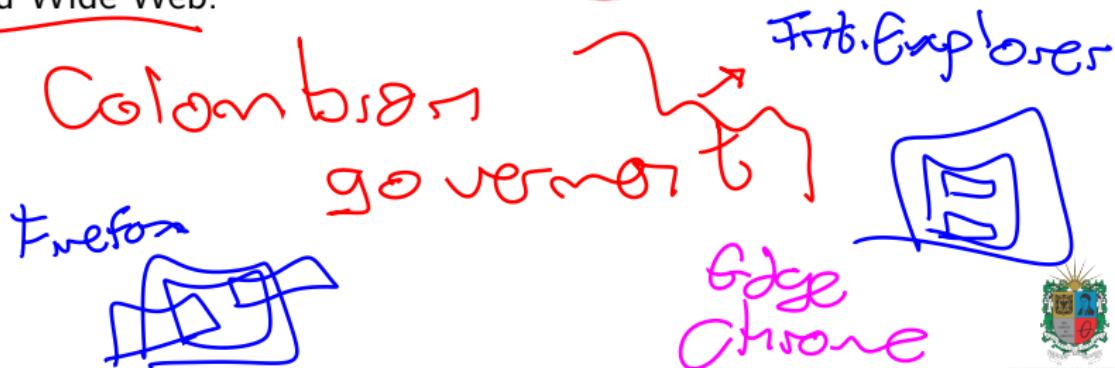
Render and Web Standards

- The **rendering engine** is a **software component** that displays the requested content.
- The **rendering engine** reads the **HTML** and **CSS** code and **renders** the content.
- The **W3** is an international community that develops **open standards** to ensure the long-term growth of the Web.
- The **W3C** is the main international standards organization for the World Wide Web.



Render and Web Standards

- The **rendering engine** is a **software component** that displays the requested content.
- The **rendering engine** reads the **HTML** and **CSS** code and **renders** the content.
- The **W3C** is an international community that **develops open standards** to ensure the long-term **growth of the Web**.
- The **W3C** is the main international **standards organization** for the **World Wide Web**.



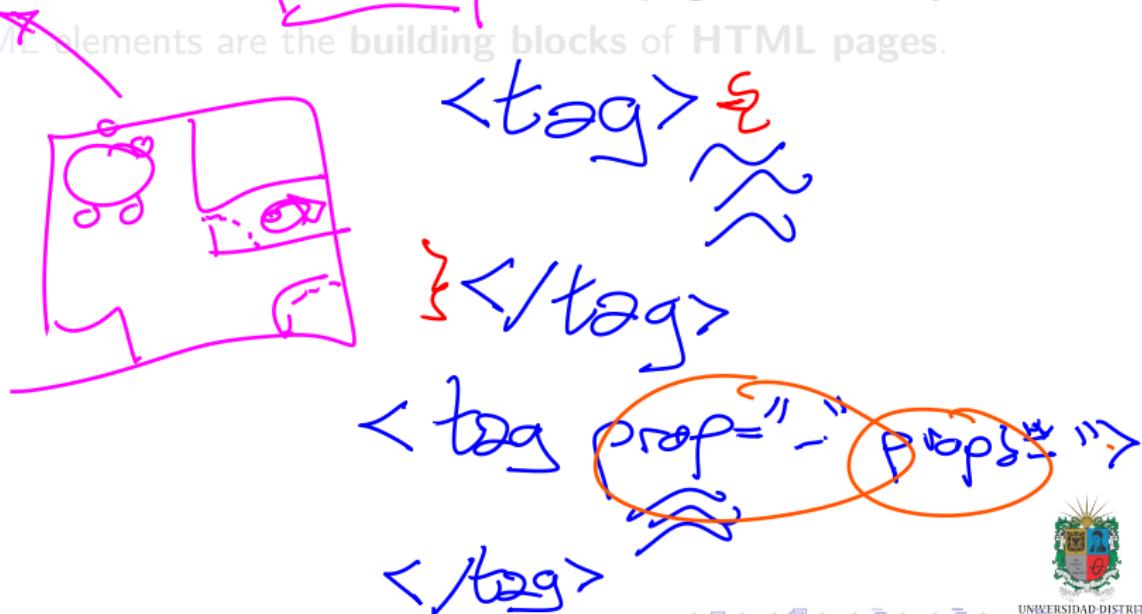
Hypertext Markup Language (HTML)

- **HTML** is the **standard** markup language for **documents** designed to be displayed in a **web browser**.
- HTML describes the **structure** of a web page semantically.
- HTML elements are the **building blocks** of HTML pages.



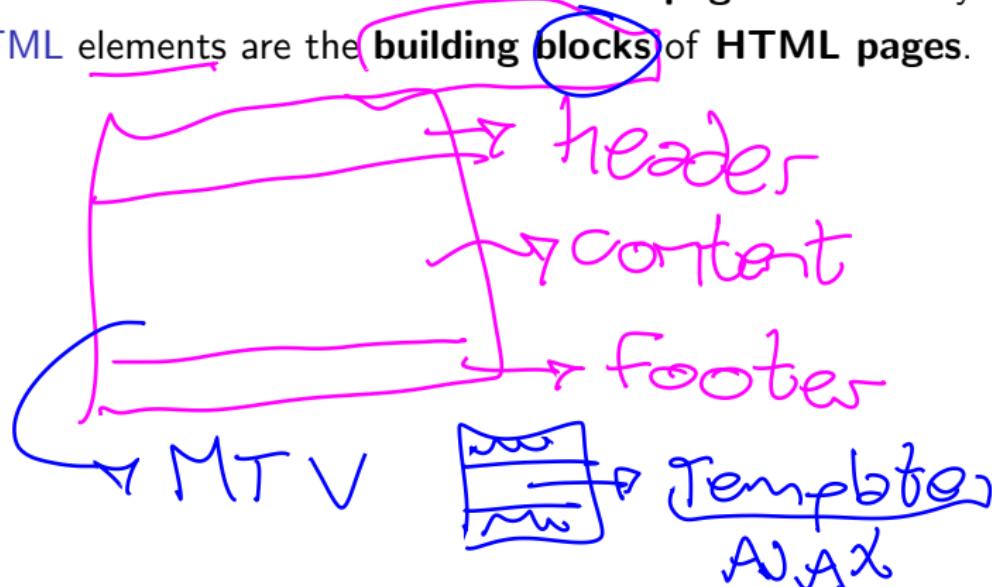
Hypertext Markup Language (HTML)

- HTML is the **standard** markup language for **documents** designed to be displayed in a **web browser**.
- HTML describes the **structure** of a **web page** semantically.
- HTML elements are the **building blocks** of HTML pages.



Hypertext Markup Language (HTML)

- HTML is the **standard** markup language for **documents** designed to be displayed in a **web browser**.
- HTML describes the **structure** of a **web page** semantically.
- HTML elements are the **building blocks** of HTML pages.



Cascading Style Sheets (CSS)

- CSS is a **style sheet language** used for describing the **presentation** of a **document** written in **HTML**.
- CSS describes how **elements** should be rendered on screen, on paper, in speech or on other media.
- CSS is a **style sheet language** that controls the **look** and **feel** of a website.



Cascading Style Sheets (CSS)

- CSS is a **style sheet language** used for describing the **presentation** of a **document** written in **HTML**.
- CSS describes how **elements** should be **rendered** on screen, on paper, in speech, or on other media.
- CSS is a **style sheet language** that controls the **look** and **feel** of a website.



Cascading Style Sheets (CSS)

- CSS is a **style sheet language** used for describing the **presentation** of a **document** written in **HTML**.
- CSS describes how **elements** should be **rendered** on screen, on paper, in speech, or on other media.
- CSS is a **style sheet language** that controls the **look** and **feel** of a **website**.

look \Rightarrow WI

feel \Rightarrow UX

color palette

Theory of
the Colors



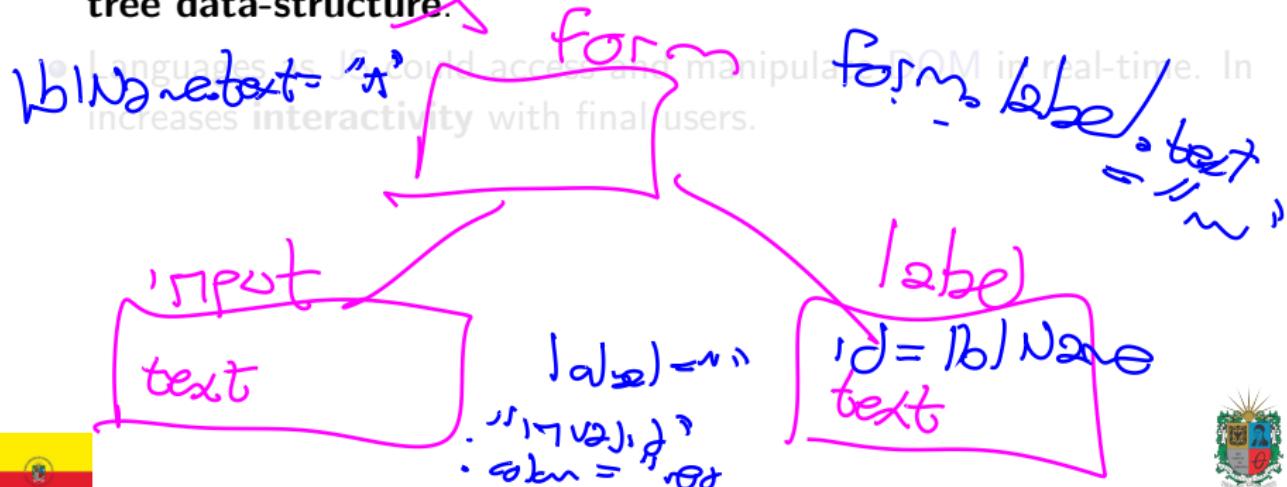
Document Object Model (DOM)

- The Document Object Model is a **cross-platform** and **language-independent** interface that treats an **XML** or **HTML** document as a **tree structure** where each **node** is an **object** representing a part of the document.
- The DOM represents the document as nodes and objects. It is a tree structure.
- Languages as JS could access and manipulate DOM in real-time. It increases interactivity with final users.



Document Object Model (DOM)

- The Document Object Model is a **cross-platform** and **language-independent** interface that treats an **XML** or **HTML** document as a **tree structure** where each **node** is an **object** representing a part of the document.
- The **DOM** represents the **document** as **nodes** and **objects**. It is a **tree data-structure**.



Document Object Model (DOM)

- The Document Object Model is a **cross-platform** and **language-independent** interface that treats an **XML** or **HTML** document as a **tree structure** where each **node** is an **object** representing a part of the document.
- The **DOM** represents the **document** as **nodes** and **objects**. It is a **tree data-structure**.
- Languages as **JS** could access and manipulate **DOM** in real-time. It increases **interactivity** with final users.

HTML \Rightarrow structure
CSS \Rightarrow styles/colors
JS \Rightarrow Interactivity

JavaScript
AngularJS
ReactJS
NodeJS



JavaScript (JS)

- JavaScript is a **high-level** programming language that conforms to the **ECMAScript** specification.

- It is a **multi-paradigm** language, supporting **object-oriented**, **imperative**, and **declarative** styles.

- JavaScript is a **client-side** language that allows **interactivity** with the **web page**.

- JavaScript is a **scripting** language that allows you to **create** **dynamically** **updates**, **content**, **control multimedia**, **animate** **images**, and much more.

↓
Object → **functions, besides**
methods
key: value → **JSON**



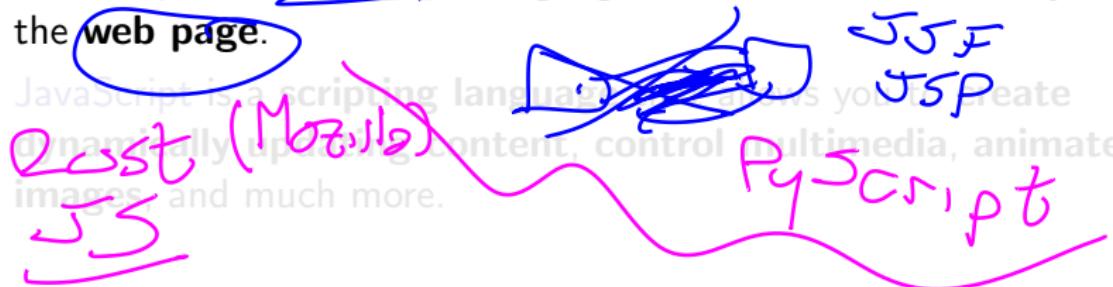
JavaScript (JS)

- JavaScript is a **high-level** programming language that conforms to the **ECMAScript** specification.
- It is a **multi-paradigm** language, supporting **object-oriented**, **imperative**, and **declarative** styles.
- JavaScript is a client-side language that allows interactivity with the ~~client~~ page.
Fundamentos
- JavaScript is a scripting language that allows you to create dynamically updating content, control multimedia, animate images, and much more.



JavaScript (JS)

- JavaScript is a **high-level** programming language that conforms to the **ECMAScript** specification.
- It is a **multi-paradigm** language, supporting **object-oriented**, **imperative**, and **declarative** styles.
- JavaScript is a **client-side** language that allows **interactivity** with the **web page**.
- JavaScript is a scripting language that allows you to create dynamically updating content, control multimedia, animate images, and much more.



JavaScript (JS)

- JavaScript is a **high-level** programming language that conforms to the **ECMAScript** specification.
- It is a **multi-paradigm** language, supporting **object-oriented**, **imperative**, and **declarative** styles.
- JavaScript is a **client-side** language that allows **interactivity** with the **web page**.
- JavaScript is a scripting language that allows you to create dynamically updating content, control multimedia, animate images, and much more.

JQuery



Outline

1 Sockets and Services

2 Layers Architecture

3 Web User Interface



Thanks!

Questions?



Repo:

 github.com/engandres/ud-public/courses/advanced-programming

