

# WEB DEVELOPMENT

## Advanced Programming I

Author: Eng. Carlos Andrés Sierra, M.Sc.  
cavirguezs@udistrital.edu.co

Computer Engineer  
Lecturer  
Universidad Distrital Francisco José de Caldas

2024-III



UNIVERSIDAD DISTRITAL  
FRANCISCO JOSÉ DE CALDAS

# Outline

1 Sockets and Services

2 Layers Architecture → MUC

3 Web User Interface



# Outline

1 Sockets and Services

2 Layers Architecture

3 Web User Interface

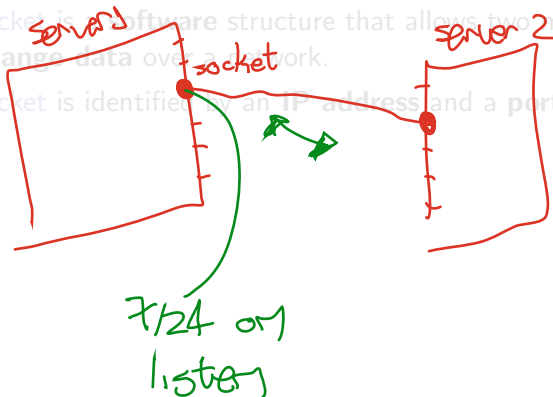


# Sockets

- A socket is an endpoint for communication between two machines over a network.

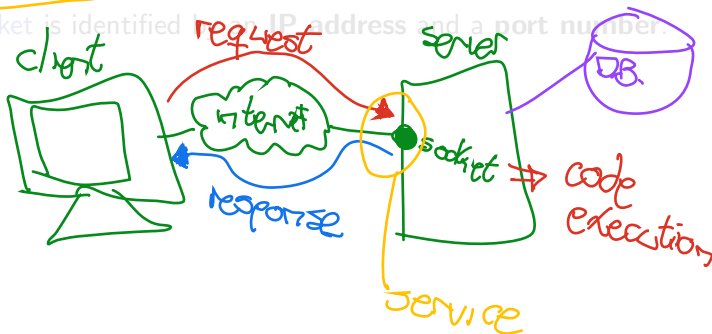
- A socket is a software structure that allows two machines to exchange data over a network.

- A socket is identified by an IP address and a port number.



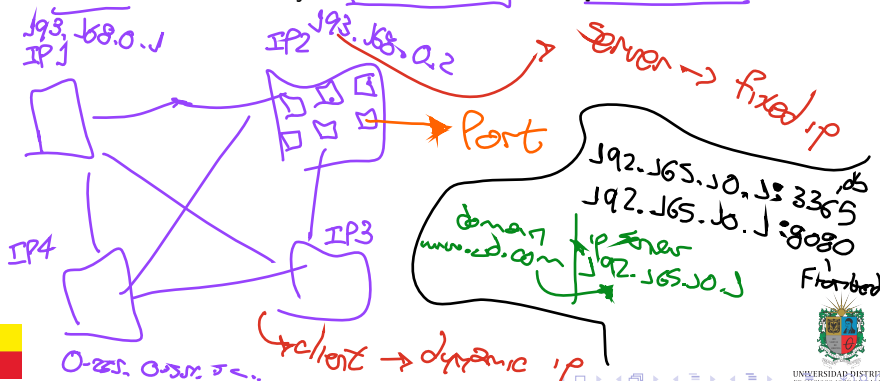
# Sockets

- 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.
- 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.
- A **socket** is identified by an **IP address** and a **port number**.



# Web Services

- A web service is a software system designed to support interoperable machine-to-machine interaction over a network.  
*,code*
- It has an interface described in a machine-processable format (specifically WSDL).  
*client-server + socket*
- 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 mechanism, and other **web-related standards**.  
*no O.S. no Programming language*
- 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). *API REST → HTTP → HTML*
- 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.

*JSON → JavaScript Object Notation*  
*for exchanging data between applications or systems.*

*key-value*

```

{
  "name": "Pepto",
  "age": 20
}
  
```

*Python → Dictionary*

*Java → HashMap*



# 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.

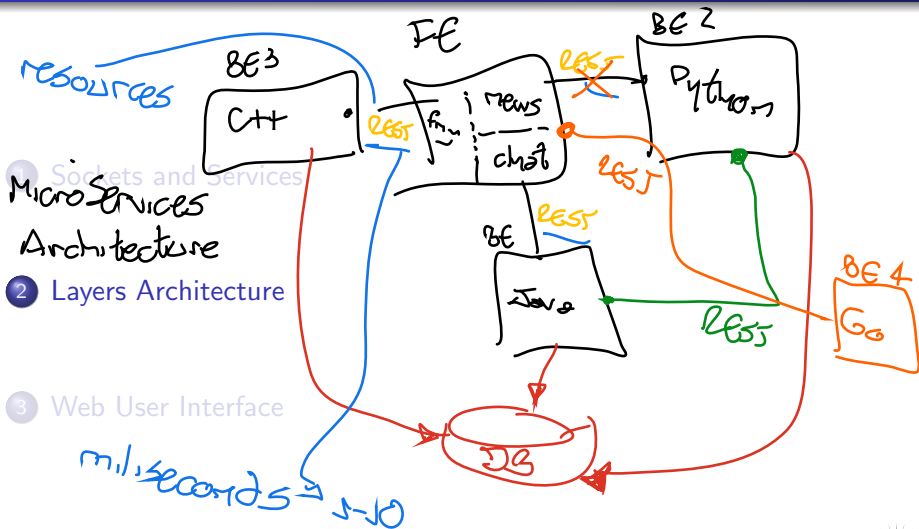


# Web Services

- 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.

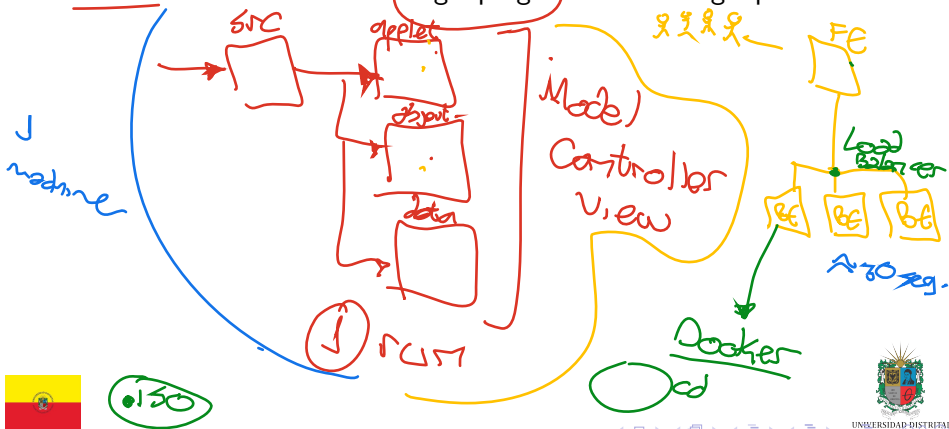


# Outline



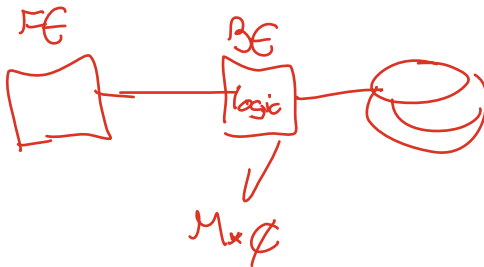
# Monolithic Architecture

- 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.



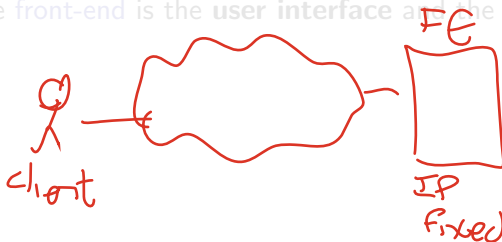
# BackEnd Layer

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



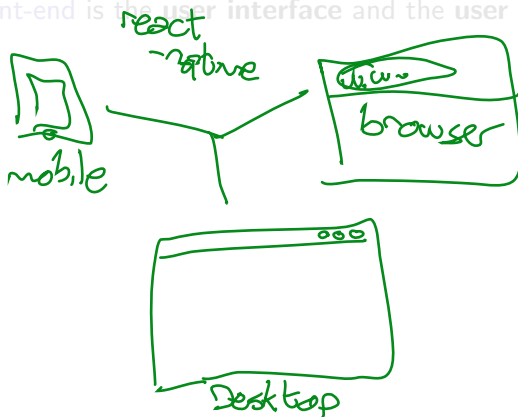
# FrontEnd Layer

- 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**.



# FrontEnd Layer

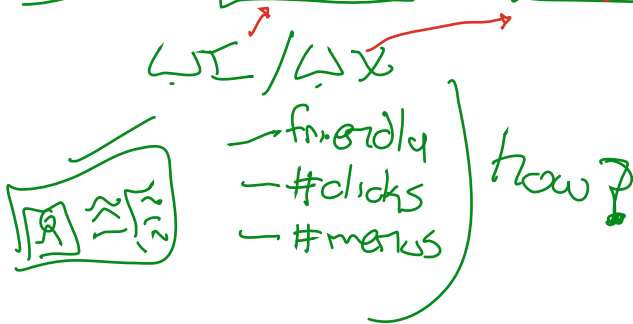
- The **front-end** is the **client-side** of the application and everything that the user interacts with.
- The **front-end** is the ~~presentation layer~~ of the application.
- The front-end is the user interface and the user experience.





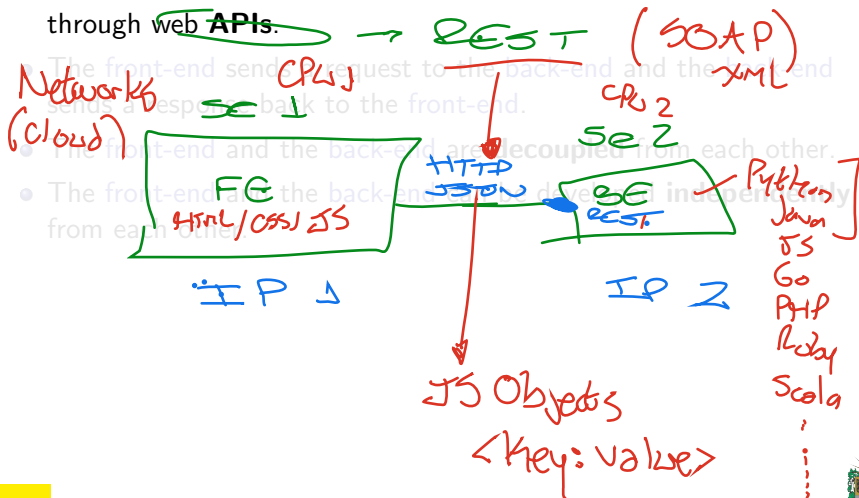
# FrontEnd Layer

- 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

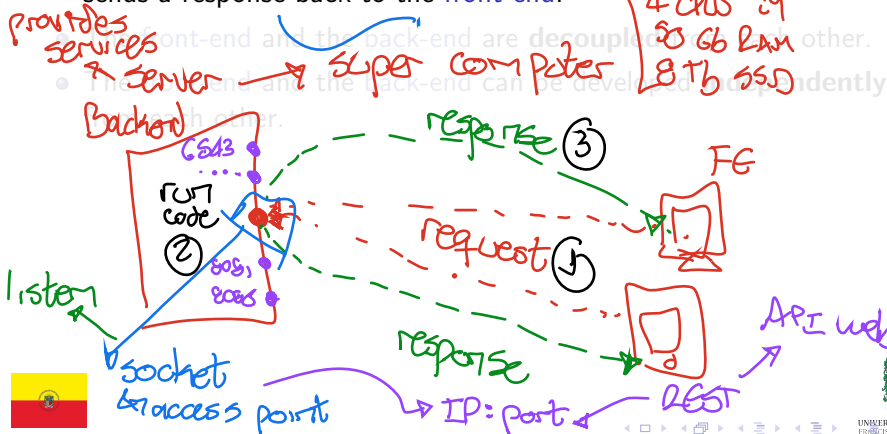
- The front-end and the back-end communicate with each other through web APIs.



# 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**.

(Socket)



# 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.

+ tech choosing  
+ scalability  
+ flexibility

+ maintainability



# Prons and Cons

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

Monolithic

project

↳ src

↳ controller

↳ view

↳ model

→ Industry  
X

→ Main.java

run

exercise

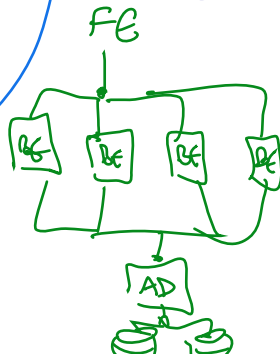


# Prons and Cons

- **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**.

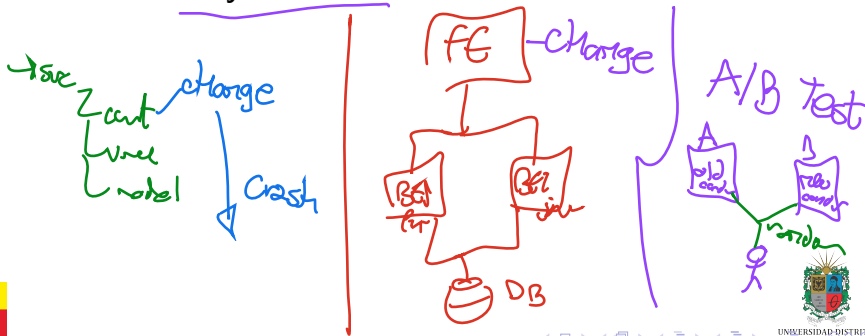
- **Monolithic Architecture** is **difficult** to **maintain**. **Layers architecture** is **easy** to **maintain**.

expensive



# 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 **graphical 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**.

Graphical User Interface



buttons

labels

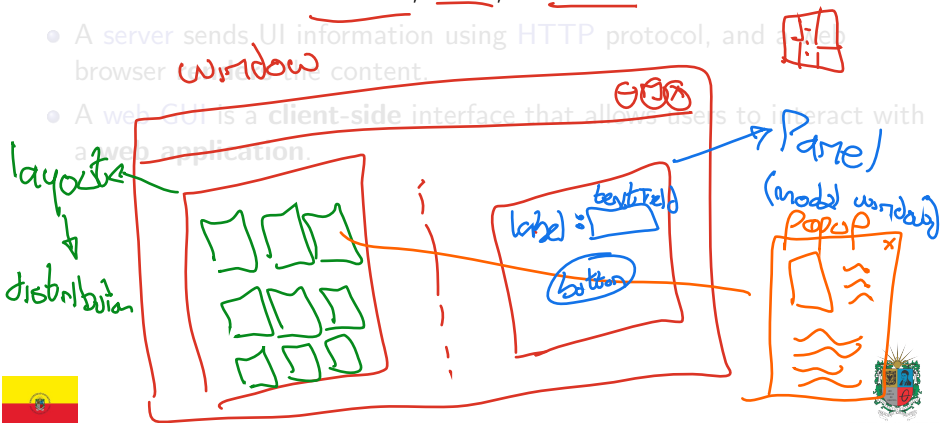
menus

render

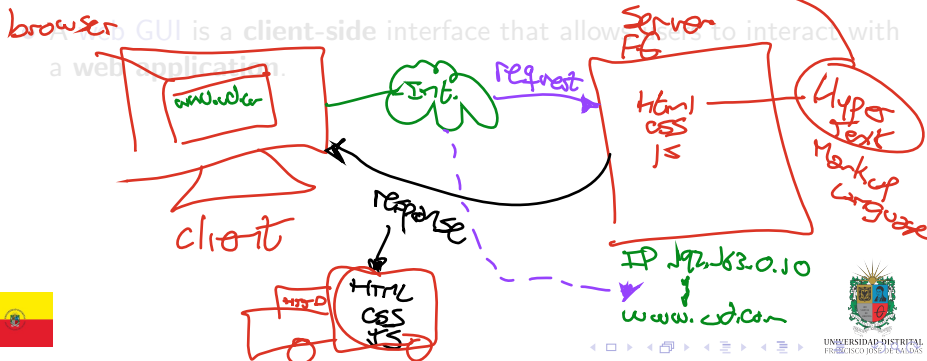


# What is a web GUI?

- 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 **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.



# What is a web GUI?

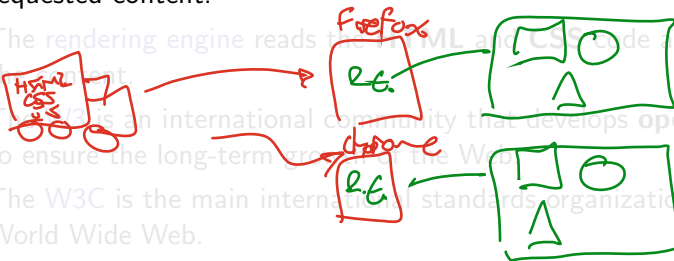
- A **web GUI** is a **graphical user interface** that is displayed in a **web browser**.
- Also, a **GUI** in general is a **user interface** that includes **graphical elements** such as **windows**, **icons**, and **buttons**.
- A **server** sends UI information using **HTTP** protocol, and a web browser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

- Handwritten notes at top: 404, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000*
- 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.

*Handwritten notes on the right:*

UTF-8

Unicode

→ emojis

→ asian language

han

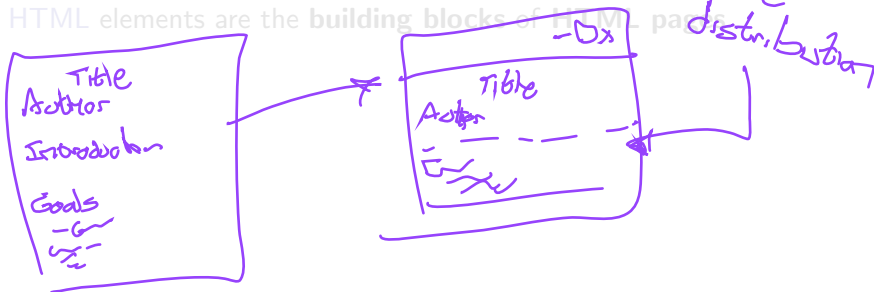


# Hypertext Markup Language (HTML)

- 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.

FORM

<code>&lt;label&gt;</code> Name:	<code>&lt;text&gt;</code> <input type="text"/>
<code>&lt;label&gt;</code> Last Name:	<code>&lt;text&gt;</code> <input type="text"/>
<code>&lt;label&gt;</code> Email:	<code>&lt;text&gt;</code> <input type="text"/>
<code>&lt;button&gt;</code> <input type="button" value="OK"/>	



# 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.

<html>

```

<body>
  <table>
    <tr>
      <td> A </td>
      <td> B </td>
    </tr>
  </table>
</body>

```

A	B

</html>

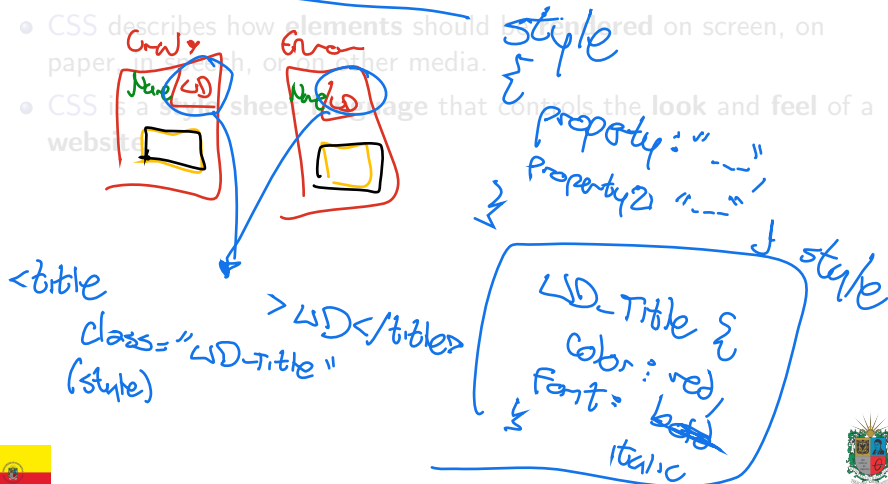


# Cascading Style Sheets (CSS)

- **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.

HTML  $\Rightarrow$  what  
CSS  $\Rightarrow$  how



# 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**.

Handwritten notes and diagrams:

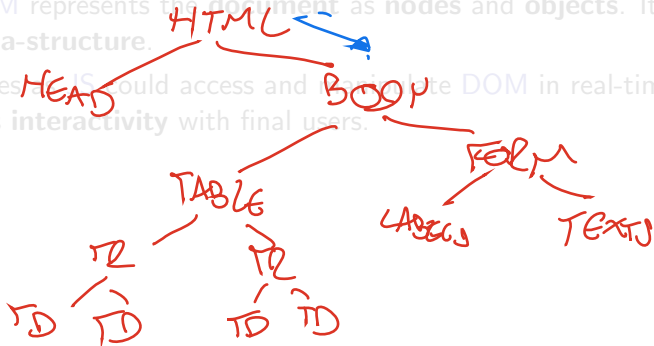
- A yellow oval is drawn around the title "Cascading Style Sheets (CSS)".
- A blue box is drawn around the phrase "look and feel" in the third bullet point.
- A blue arrow points from "look and feel" to the word "website".
- A yellow arrow points from "look and feel" to the handwritten text "Color Theory".
- Handwritten text includes "Color Theory", "time spent", and "tired?".
- Handwritten symbols include "4x", a downward arrow, and "5x".



# 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 like JavaScript should access and manipulate the 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**.
- Languages as JS could access and manipulate DOM in real-time. In increases **interactivity** with final users.

HTML → Tree



# 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. In increases **interactivity** with final users.

JavaScript

(Rest)





# JavaScript (JS)

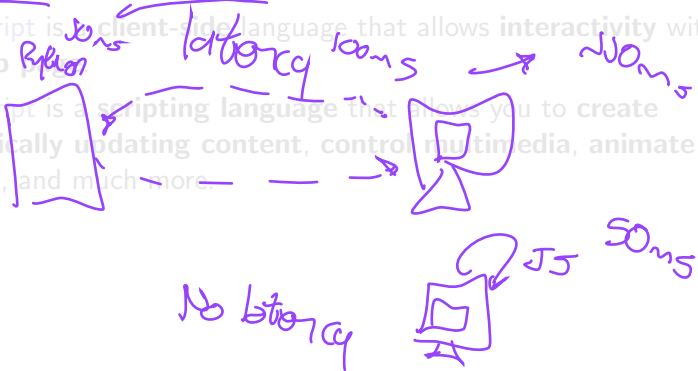
- **JavaScript** is a **high-level** programming language that conforms to the **ECMAScript** specification.
- It is a **multi-paradigm** language, supporting **object-oriented**, **imperative**, and **declarative** styles.
- JavaScript is a **client-side** language that allows **interactivity** with 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.**



# 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.

web browser



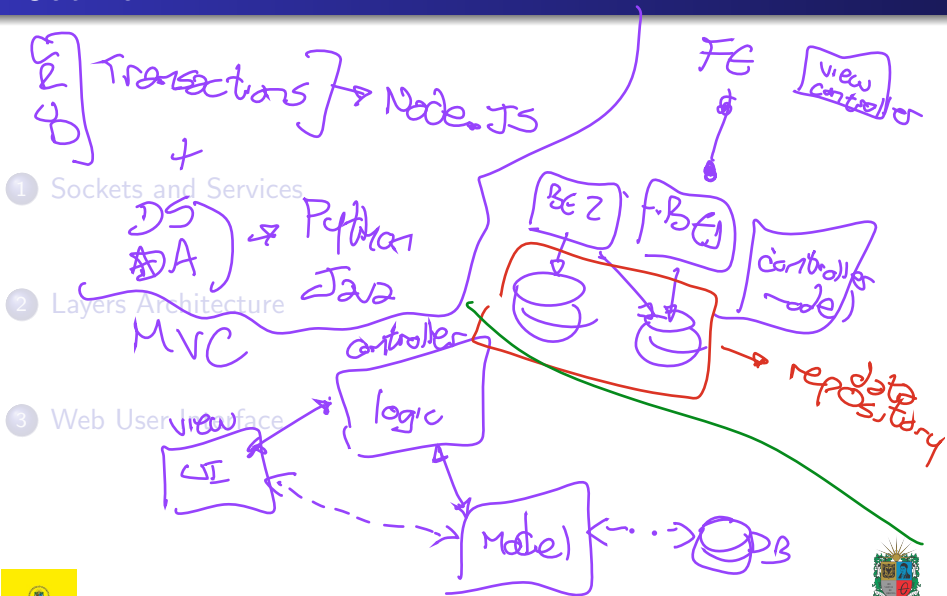
# JavaScript (JS)

- **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.

multimedia

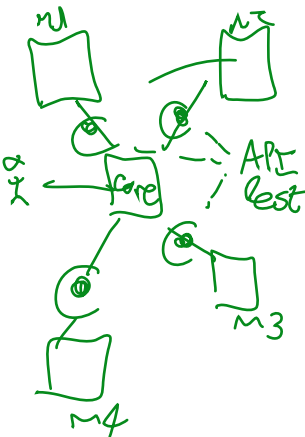


# Outline



# Thanks!

## Questions?



Repo:

[github.com/engandres/ud-public/tree/main/courses/  
advanced-programming](https://github.com/engandres/ud-public/tree/main/courses/advanced-programming)

