Frontend Web Development

Introduction

About the instructor

Abdelrahman Abounegm

- Innopolis University graduate of bachelor’s program 2022
- Worked as a frontend developer for 6 years
- Co-developer of Innopoints portal
- Maintainer of a UI components library with 800+ stars

About the course

Content

- 10 lectures + 10 labs
- Last “lab” reserved for project presentations

Assessment

- 6 small individual assignments, based on the lab content
  - Deadline: 1 week
- 1 final group project

Grading

Assignments	50%
Final Project	50%
TA Bonus	up to +10%

A	**85%**
B	**70%**
C	**55%**

Maximum 2 bonus points per lab/assignment, 10 overall

Lectures Outline

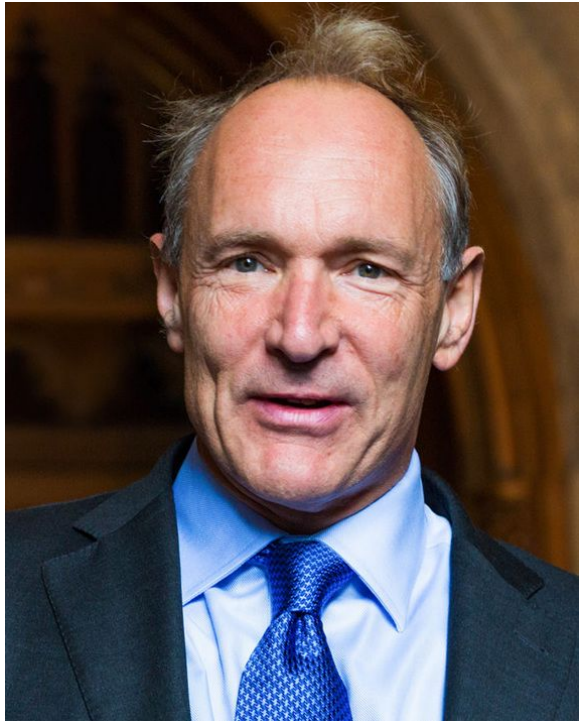
1. Introduction + Basics of HTML, CSS, and JavaScript
2. JavaScript and CSS in more depth
3. JavaScript Ecosystem: NPM, bundlers
4. TypeScript
5. Forms and tables, more CSS
6. Frameworks, Single File Components, Svelte
7. React
8. Code quality, linting, testing, optimizations.
9. Architectures and Patterns
10. Extra topics

Historical Overview of Web Sites

Static Documents

In 1989 Tim Berners-Lee proposed a document where he proposed to create universal linked information system.

Two years later, in 1991, the first ever web-page was published. It described how to create such page with HTML.



World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
: Pointers to the world's online information, subjects , W3 servers, etc.

Help
: on the browser you are using

Software Products
: A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)

Technical
: Details of protocols, formats, program internals etc

Bibliography
: Paper documentation on W3 and references.

People
: A list of some people involved in the project.

History
: A summary of the history of the project.

How can I help ?
: If you would like to support the web..

Getting code
: Getting the code by anonymous FTP , etc.

Dynamic Sites

Dynamic sites appeared after some time. For example, Rasmus Lerdorf wrote a couple of CGI programs to extend his personal homepage. He connected web forms with databases.

Wikipedia

```
<!--include /text/header.html-->

<!--getenv HTTP_USER_AGENT-->
<!--if substr $exec_result Mozilla-->
  Hey, you are using Netscape!<p>
<!--endif-->

<!--sql database select * from table where user='$username'-->
<!--ifless $numentries 1-->
  Sorry, that record does not exist<p>
<!--endif exit-->
  Welcome <!--$user-->!<p>
  You have <!--$index:0--> credits left in your account.<p>

<!--include /text/footer.html-->
```

Example of Early PHP Syntax

Wikipedia

More Dynamicity - JavaScript & AJAX

The next step was in introduction of AJAX technology by Google in 2004. This technology allowed web pages to interact with the server without reloads.

```
var httpRequest;
if (window.XMLHttpRequest) { // Mozilla, Safari, ...
    httpRequest = new XMLHttpRequest();
} else if (window.ActiveXObject) { // IE
    httpRequest = new ActiveXObject("Microsoft.XMLHTTP");
}

httpRequest.overrideMimeType('text/xml');

httpRequest.open('GET', 'http://www.example.org/some.file', true);
httpRequest.send(null);
```

Ajax: A New Approach to Web Applications

Examples of old websites

What is Frontend?

Simply speaking, it is the User Interface of a system that allows users to view and interact with it.

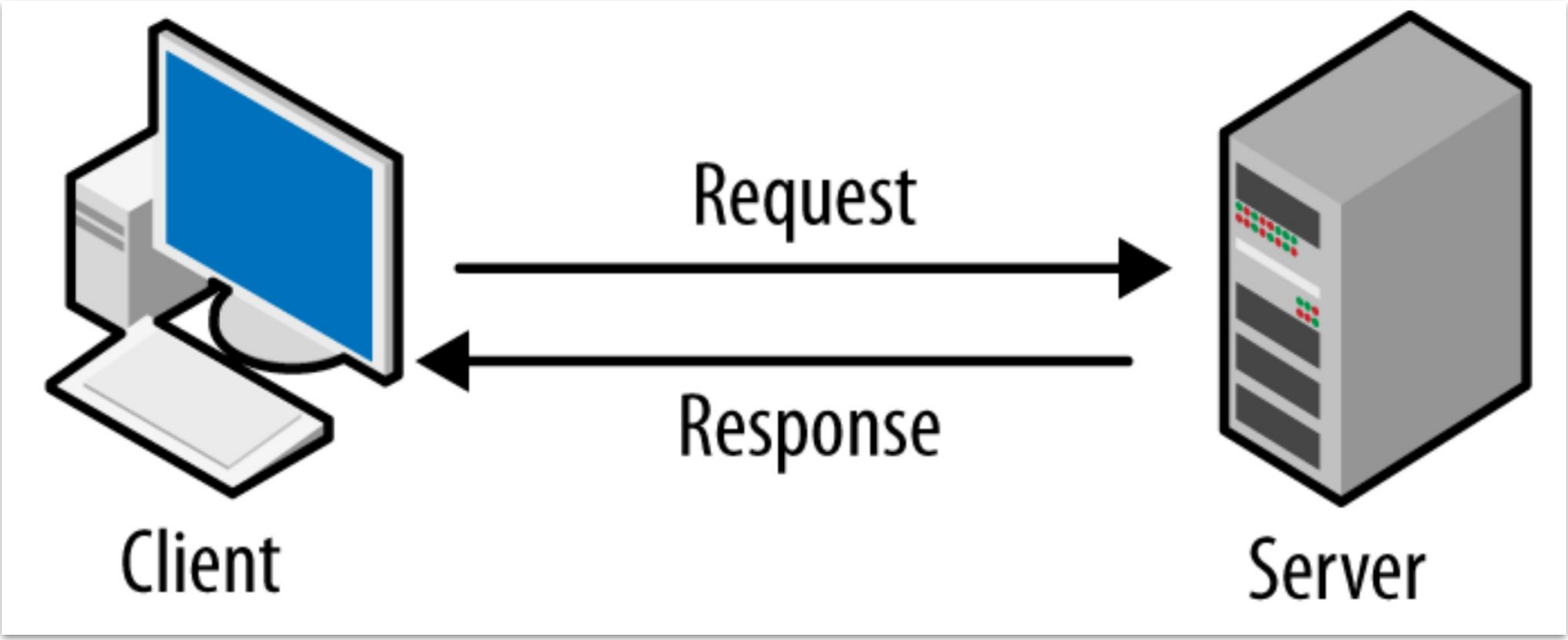
The term “frontend” itself applies to more than just websites, but we are only concerned with web in this course.


500: internal server error
NullPointerException
SQLSyntaxErrorException table does not exist
Frontend
style="border-left: 2px solid green"
STARECAT.COM

Principles / Challenges

- **Accessibility**: making your websites usable by as many people as possible (e.g.: blind people with screen readers)
- **Performance**: Time is money, my friend!
- **Backwards compatibility**: Websites written in 1989 still work today!
- **Runs on almost any device**: PC, mobile, smart watch, smart fridge, ...

Architecture of the internet



Communication protocol: HTTP

HTTP Request

Method URL Protocol Version

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html, */*
Accept-Language: en-us
Accept-Charset: ISO-8859-1,utf-8
Connection: keep-alive
blank line
```

Headers

Body (optional)

HTTP response message

status line (protocol status code status phrase)

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 …...
Content-Length: 6821
Content-Type: text/html

data data data data data ...
```

header lines

requested HTML file

Web protocol standards

The **World Wide Web Consortium** (W3C), founded in 1994, defines all the web standards that web browsers have to follow.

It has 459 members, including Google, Microsoft, Mozilla Foundation, ...

Creates standards through Request For Comments (RFC) documents, such as:

- HTTP
- DOM (Document Object Model, accessing HTML elements in JavaScript)
- WebAssembly (portable Assembly language for the web)
- WebRTC (for peer-to-peer real-time communication)
- CSS, SVG, WAI-ARIA, XML, RDF, SOAP, and a lot more

Frontend technologies

- **HTML** (HyperText Markup Language): page skeleton
- **CSS** (Cascading Style Sheets): aesthetics
- **JavaScript**: Logic

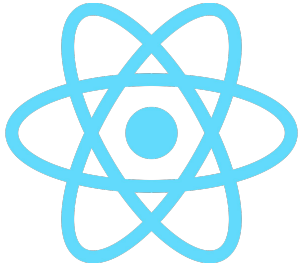
Literally any programmer:

Components frameworks

Provide a way of writing reusable components that encapsulate their markup, styles, and logic.

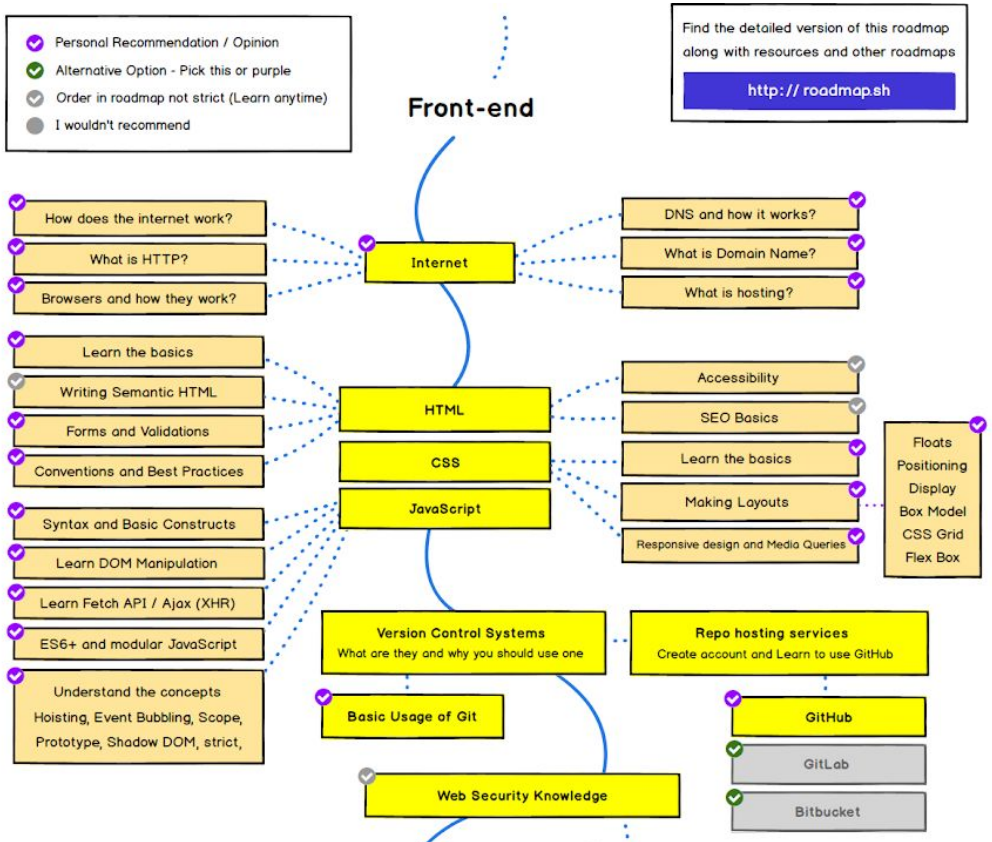
They usually provide advanced state management techniques.

They improve the developer experience and code readability for big-sized projects



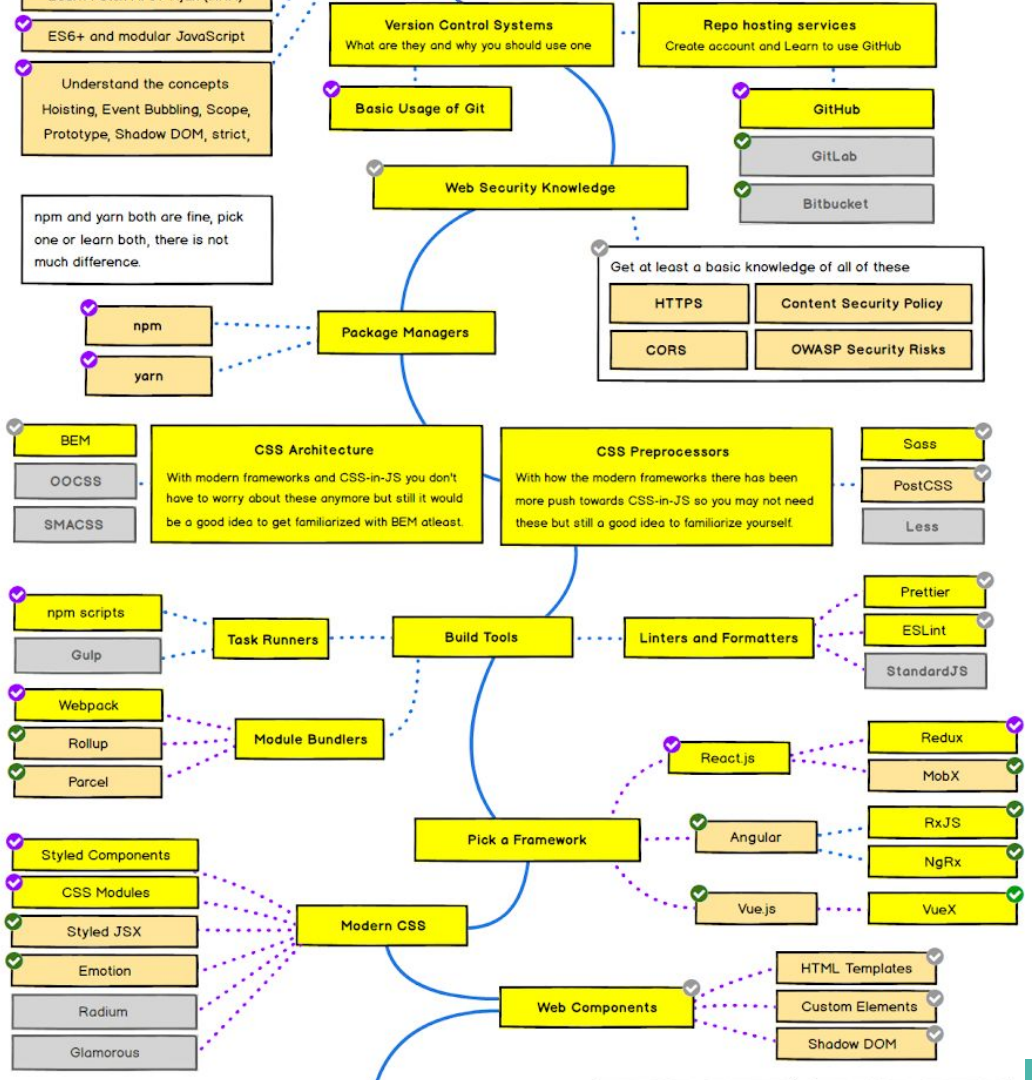
BACKBONE.JS

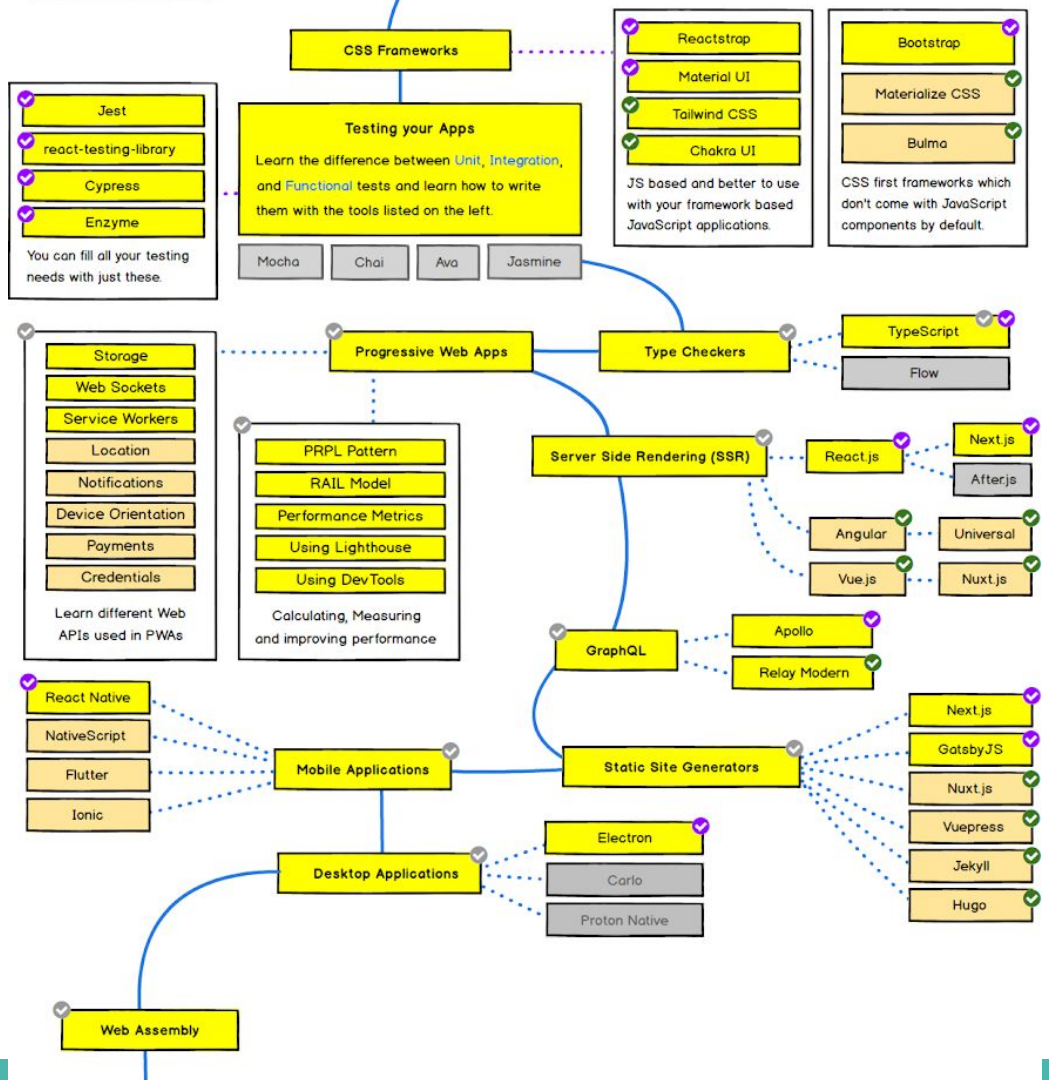
Web-Developer Roadmap

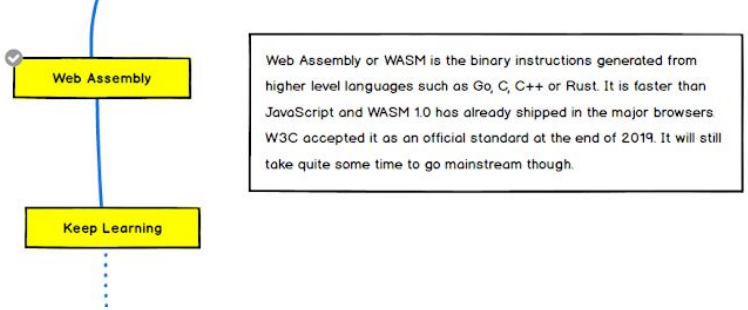


roadmap.sh

Джуность


ES6+ and modular JavaScript
Understand the concepts
Hoisting, Event Bubbling, Scope, Prototype, Shadow DOM, strict,
Version Control Systems
What are they and why you should use one
Repo hosting services
Create account and Learn to use GitHub
Basic Usage of Git
GitHub
GitLab
Bitbucket
Web Security Knowledge
npm and yarn both are fine, pick one or learn both, there is not much difference.
Get at least a basic knowledge of all of these
HTTPS
Content Security Policy
CORS
OWASP Security Risks
npm
yarn
Package Managers
BEM
OOCSS
SMACSS
CSS Architecture
With modern frameworks and CSS-in-JS you don't have to worry about these anymore but still it would be a good idea to get familiarized with BEM atleast.
CSS Preprocessors
With how the modern frameworks there has been more push towards CSS-in-JS so you may not need these but still a good idea to familiarize yourself.
Sass
PostCSS
Less
npm scripts
Gulp
Task Runners
Build Tools
Linters and Formatters
Prettier
ESLint
StandardJS
Webpack
Rollup
Parcel
Module Bundlers
React.js
Redux
MobX
Pick a Framework
Angular
RxJS
NgRx
Vue.js
VueX
Styled Components
CSS Modules
Styled JSX
Emotion
Radium
Glamorous
Modern CSS
Web Components
HTML Templates
Custom Elements
Shadow DOM


CSS Frameworks
Reactstrap
Material UI
Tailwind CSS
Chakra UI
JS based and better to use with your framework based JavaScript applications.
Bootstrap
Materialize CSS
Bulma
CSS first frameworks which don't come with JavaScript components by default.
Jest
react-testing-library
Cypress
Enzyme
You can fill all your testing needs with just these.
Testing your Apps
Learn the difference between Unit, Integration, and Functional tests and learn how to write them with the tools listed on the left.
Mocha
Chai
Ava
Jasmine
Storage
Web Sockets
Service Workers
Location
Notifications
Device Orientation
Payments
Credentials
Learn different Web APIs used in PWAs
Progressive Web Apps
Type Checkers
TypeScript
Flow
PRPL Pattern
RAIL Model
Performance Metrics
Using Lighthouse
Using DevTools
Calculating, Measuring and improving performance
Server Side Rendering (SSR)
React.js
Next.js
After.js
Angular
Universal
Vue.js
Nuxt.js
GraphQL
Apollo
Relay Modern
React Native
NativeScript
Flutter
Ionic
Mobile Applications
Static Site Generators
Next.js
GatsbyJS
Nuxt.js
Vuepress
Jekyll
Hugo
Desktop Applications
Electron
Carlo
Proton Native
Web Assembly



Each point is a science that is worth its own bachelor degree.

Useful Resources

- Mozilla Developer Network (MDN): Best reference/guides for web developers
- Roadmap.sh: study plans, paths, and resources for learning the different frontend technologies
- Stack Overflow: Any question you might have is probably already answered there (multiple times 😁)
- Awesome list - frontend: A curated collection of everything that is awesome

Lecture 1

Introduction to HTML/CSS/JS

Frontend Web Development

HTML and CSS

Hyper**T**ext **M**arkup **L**anguage defines the way how data is structured when displayed in web browser.

Cascade **S**tyle **S**heets help change the visual appearance of an HTML element inside of the document.

Courses

codeacademy HTML Course

HTML Academy

References

htmlbook.ru

MDN Web Docs

w3schools HTML Tutorial

Learn HTML

Learn CSS Layout

HTML


Me and the boys going to go get hired at Google after learning HTML:

HTML Tags

```
<tag attribute="value">content</tag>
```

```
<tag attribute="value" />
```

tag - name of an HTML element, e.g. *img*

attribute - name of attribute available for the tag, e.g. *src*

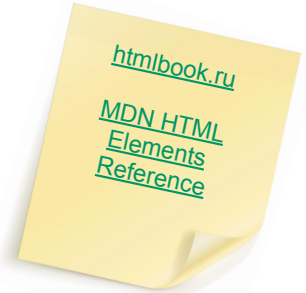
value - value assigned to the attribute, e.g. *logo.svg*

content - optional child content of the tag, can be either text, or another tag

```
<img src="logo.svg" alt="Site Logo" />
```

Examples of Tags

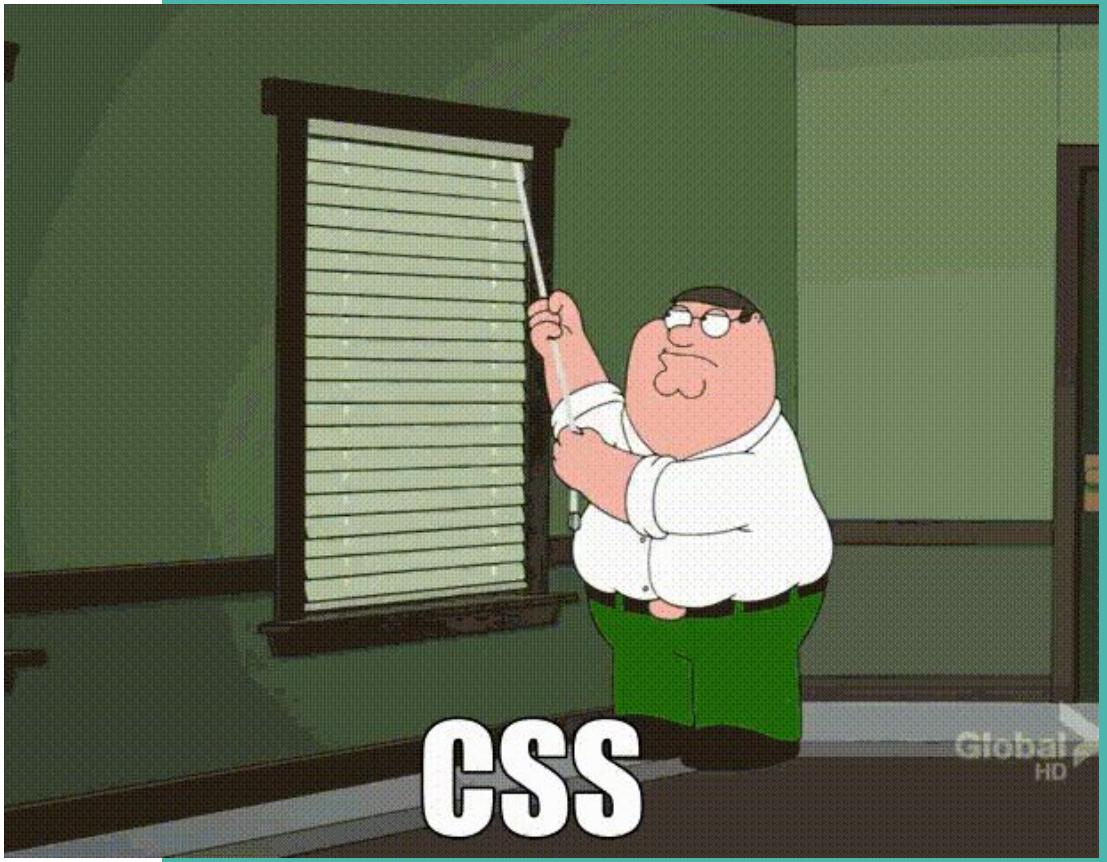
html, head, body, meta, title, link,
h1, h2, h3, h4, h5, h6, p, span, div, img, hr, table, tr, td, th,
header, base, style, main, nav, section,
li, ol, ul, pre, a, q, iframe, script...



Structure of HTML Document

```
<!DOCTYPE html>
<html>
▼ <head>
    <title>Test Page</title>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
  </head>
▼ <body>
    <h1>Test Page</h1>
    <p>Hello, World!</p>
  </body>
</html>
```

CSS



All style properties on w3school, on quackit, on htmlbook

Styles in a Document

HTML elements can be styled in many ways. Developers can change width, height, margins and paddings, colors of background and included text, borders, display type (block, inline, flex, etc.) and other properties.

Styles can be added either as inline, (using the *style* attribute) or as separate style sheets.

Style attribute

```html
<!DOCTYPE html>
<html>
<body>
  <h1 style="color: blue; text-align: center;">
      This is a header
  </h1>
  <p style="color: green;">
      This is a paragraph.
  </p>
</body>
</html>
```

Cascading Style Sheets

- CSS is a set of rules
- Each rule consists of a selector that defines the element(s) to be styled and block of declarations between curly brackets
- Each declaration includes property name and value, that should be assigned to that property

```
body {
    background-color: green;
    min-height: 500px;
}
```

CSS Selectors

Selectors can be defined quite flexibly. W3school defines 5 main groups of possible selectors:

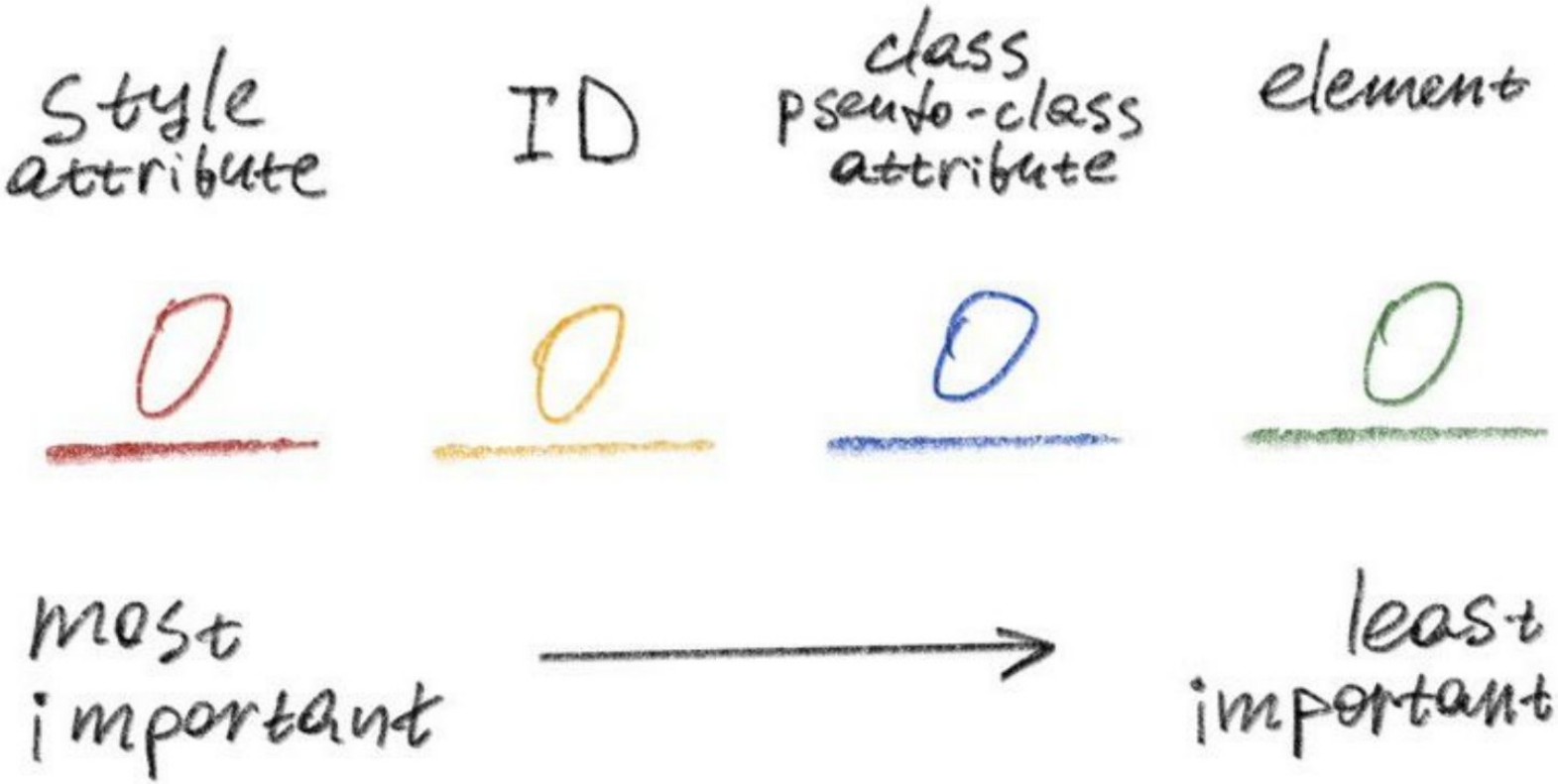
- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

CSS Selectors Examples

```
div
#some-block
div#some-block
div h1
.title
.button
div > .title
h1 + button
```

```
<div id="some-block">
    <h1 class="title">Test</h1>
    <button class="button">
        123
    </button>
</div>
```

Selectors Specificity



Selectors Specificity Examples

`div`	0, 0, 0, 1
`#some-block`	0, 1, 0, 0
`div#some-block`	0, 1, 0, 1
`div h1`	0, 0, 0, 2
`.title`	0, 0, 1, 0
`.button`	0, 0, 1, 0
`div > .title`	0, 0, 1, 1
`h1 + button`	0, 0, 0, 2

CSS Specificity: Things You Should Know

Specifics on CSS Specificity

```
.Cat{
    width: 100%;
    height: 100%;
}
```

cat.css

JavaScript

JavaScript

It is a lightweight, interpreted*, programming (scripting) language with first-class functions. It makes HTML pages more dynamic and interactive. It is also classified as:

- single-threaded
- dynamically typed
- prototype-based OOP
- multi-paradigm
  - object-oriented
  - functional
  - event-driven

Syntax

Variables and Literal

```
var a;                          // variable declaration
var b = "init";                 // string
var c = "Hi" + " " + "Joe";     // = "Hi Joe"
var d = 1 + 2 + "3";            // = "33"
var e = [2, 3, 5, 8];           // array
var f = false;                  // boolean
var g = /()/;                   // RegEx
var h = function(){};           // function object
const PI = 3.14;                // constant
var a = 1, b = 2, c = a + b;    // one line
let z = 'zzz';                  // block scope local variable
let map = { abc: 5 };           // object literal
```

Syntax

```
function sum(x, y) {
    return x + y
}

const greet = function(name) {
  if (name == null) name = ‘human’
  return `Hello, ${name}!`
}

greet(‘John’);          // Hello John!
greet();                // Hello human!
greet(“John”, “Smith”); // Hello John!
```

```
for (let i = 0; i < 10; i++) {
    console.log(i);
}

const food = 'salad';
switch (food) {
  case 'oyster':
    console.log('The taste of the sea');
    break;
  case 'pizza':
    console.log('A delicious pie');
    break;
  default:
    console.log('Enjoy your meal');
}
```

Syntax

Primitive Types

1. string
2. number
3. bigint
4. boolean
5. symbol
6. null
7. undefined

0

null

undefined

Non-primitive Types

- Object

Anything that is constructed with **new** or an object/array/function literal

Non-primitives are compared/passed by reference!

Global Built-in Objects

`window` and everything inside it, including:

- `document` - contains useful properties for DOM manipulation
  - `getElementByID`
  - `querySelector`
  - `createElement`
- `Object` - contains utilities for dealing with any object (getting keys, values, copying, freezing, ... and many advanced usages)
- `navigator` - gives scripts info about (and some control on) the user agent
- `console` - allows you to print stuff to the console for debugging purposes
- ...

These can all be accessed without the `window.` prefix

Basic usage

Its main use is to manipulate the DOM and styles and to react to events.

JS can be written in some HTML attributes or in the special `<script>` tag

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Homework</h1>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<script>
function myFunction() {
    document.getElementById("demo").style.fontSize = "50px";
    document.getElementById("demo").style.color = "blue";
}
</script>

<button type="button" onclick="myFunction()">Click Me!</button>

</body>
</html>
```