**Web development Bootcamp**

**What is web development?**

Web development is building websites.

**What is HTTP/HTTPS?**

It is the standard protocol used to communicate in web.

**What is URL?**

URL stands for Uniform resource locator.

[https://**academind.com**/tutorials](https://academind.com/tutorials)

***Domain*** (The human-readable address (“identifier”) of a website)

**What is IP address?**

IP stands for Internet protocol is a unique identifier (“address”) of network devices. Ex 102.59.62.31

**What is relation between domain and IP address?**

When ever a browser sends a request with domain (amazon.com), this domain is registered globally so, browser first sends the request to DNS (Domain Name System) server which has several mappings between domains and their IP addresses. So, DNS server gives the IP address of amazon.com, then the browser uses that IP address and calls the remote servers to get the response. Every computer has an IP address, our personal computers also have Ip address, but our computers won’t allow incoming requests, so we are safe. But remote servers configured to allow incoming requests.

**What about “www”?**

A lot of websites can be visited via different domains. For example, the Amazon website can be visited via amazon.com but also via www.amazon.com. Both addresses can be entered in the browser and you will reach the same website.

*So, what's the difference?*

Especially in the earlier days of the internet, it was also often called the "world wide web" (or: "www"). You still hear that term quite a bit and it's this term, that explains why some websites support such a "www" prefix in the domain.

Technically, www.amazon.com is called a "subdomain" of the "root domain" amazon.com. As the operator of a website and the owner of a domain, you can register any subdomains you want to. You could also register mysite.amazon.com, if you were the owner of the amazon.com domain. And you can then configure your web server such, that your website is served, if users enter mysite.amazon.com.

That's what many (but not all) websites do with the "www" subdomain: They register it and configure it such, that it also points at the main website.

It's not required and not all websites have it. Most users also don't type "www.somewebsite.com" manually anymore - instead just the root domain is entered in many cases.

**Why use HTML elements?**

We use them to describe your content and to provide the correct semantic around your content.

**What are void elements in HTML?**

Link and img tags are considered as void because they are self-closing tags and don’t have content.

**Why text-align: center CSS property don’t work for img tag?**

Because img is void tag which doesn’t have content. Text-align center is applied to content only not for whole HTML element. The solution is we can wrap img tag inside a div and we can apply the text-align center to that div, it works because now img tag is content for div tag.

***Understanding the HTML ELEMENTS***

Graphical user interface, diagram

Description automatically generated

Global CSS selectors:

Diagram

Description automatically generated

**What is Inheritance between HTML elements?**

Inheritance simply means (selected) container rules apply to descendants.

**What is Cascading?**

Multiple rules can be applied to the same element.

**What is Specificity?**

More specific selector’s rule wins over less specific selector.

**What is Padding?**

Padding is the space between content and borders of a box.

**What id Margin?**

Margin is space between one element to another element.

**Box Model**

Graphical user interface, application

Description automatically generated

Selectors and combinators:

Graphical user interface

Description automatically generated

**Block and Inline Elements :**

Block elements will occupy full line and complete width. Inline elements just occupy the width that is necessary.

Graphical user interface

Description automatically generated

Graphical user interface, website

Description automatically generated

**Differences between Block and Inline elements?**

For Inline elements we can’t add top and bottom margin and top and bottom padding and element won’t push the next line elements top or down, to resolve this problem we can use property ***display: inline-block.***

But there is an exception on replaced inline elements like img tag.

**What is Margin collapsing?**

Margin collapsing happens for block elements and higher margin elements wins. When a H1 tag with 12px and p tag with 8px are present. There margins will collapse, and we can use top margin between this element is 12px. But if there is a p tag and a tag elements up and down we can see total margin is 20px.

**What is Hosting or Deployment?**

Deployment : Moving the website code onto a remote machine (“server”) that serves the site to visitors.

Hosting : The remote machine (“server”) hosts (stores) the website code and serves it to visitors.

**Relative and Absolute paths**

Graphical user interface

Description automatically generated

**Netlify for easy deployments :**

<https://app.netlify.com/drop>

**What is Flexbox concept ?**

When we have a two block elements and we want them next to each other, in that case we give the parent of two elements a property called display: flex then the items will be next to each other because of flex-direction: row by default. Some cases the alignment between two properties are not correct then we can use align-items: center for parent. If we want them to be right and left, we can use justify-content: space-between.

**Mention of some properties in CSS related to flexbox ?**

display: flex;

flex-direction: row;

justify-content: center;

flex-wrap: nowrap;

align-items: center; (mostly used when using flex, to align vertically center)

**What is CSS grid ?**

CSS grid is used 2d layouts with rows and columns, while flexbox is used for 1d layout only. But we can use container concept from flexbox. Some of the properties mostly used in grid are below:

**ul** {

display: grid

grid-template-columns: 1fr 1fr; (Tells how many columns and column width)

gap: 100px 200px; (100px-> gap between rows, 200px->gap between columns)

}

**li**: nth-of-type(3){

grid-column:1 / 3; (way of telling the item 3 in list should act as a block element and occupy 2 column)

}

**Difference between units?**

Graphical user interface

Description automatically generated

Default font-size of browser settings is 16px. That is if in a blank html document, you have a h2 element and styled to ***font-size : 100%*** , the font-size will be 16px because parent is root which is browser settings. In the same way if we styled h2 with font-style: 1em , the font size will be 16px because of root parent browser settings. Suppose you wrapped the h2 element around the div tag and applied font-size to 200%, now the font-size of h2 will be 32px, because for div tag the root is browser which has default of 16px but in div tag you said 200% so 16px \* 2 = 32px. This behavior is common for %, em units.

**Rem** : rem, root ephemeral unit, means that the font size, is always relative to the root element's font size. With no default value being set for the root element, the size always is relative to the browser settings of the user, which gives us a very high level of flexibility.

***Note*** : So, to quickly summarize what we learned here, the percentage value and the em unit, lead two similar results in a font size context. *Font size in an em unit, it depends on the font size of the parent*. With the rem unit, we can calculate our font size, always relative to the root element.

Font size in an em/rem unit, it depends on the font size of the parent, which is totally true, but using the em unit on an element for a property different than the font size, then the calculation result of this em unit will be based on the actual element's font size.

Percentage always refers to parent elements. em and rem are always related to the font size, no matter which property the em or rem unit is actually applied to.

Common breakpoints for media queries

Graphical user interface

Description automatically generated

**max-width & min-width** :

***max-width*** is used for desktop to mobile development when max-width is used in media query like

@media (max-width: 768px) it means 768px or less then trigger inside CSS selectors and apply the tags.

***min-width*** is used for mobile to desktop development. @media (min-width: 768px) it means 768px or more then trigger inside CSS selectors and apply the tags.

**Some best designs standards to follow :**

* One thing to keep in mind is add some space to the left and to the right to keep the content centered a bit enough to load the entire screen with content.
* Choose font family by filtering with sans-serif and opting family with more styles available in google font.
* Choosing colors for website is important. A grey color for text is preferred than black. A primary and accent color should be choose properly.

**window & document objects** :

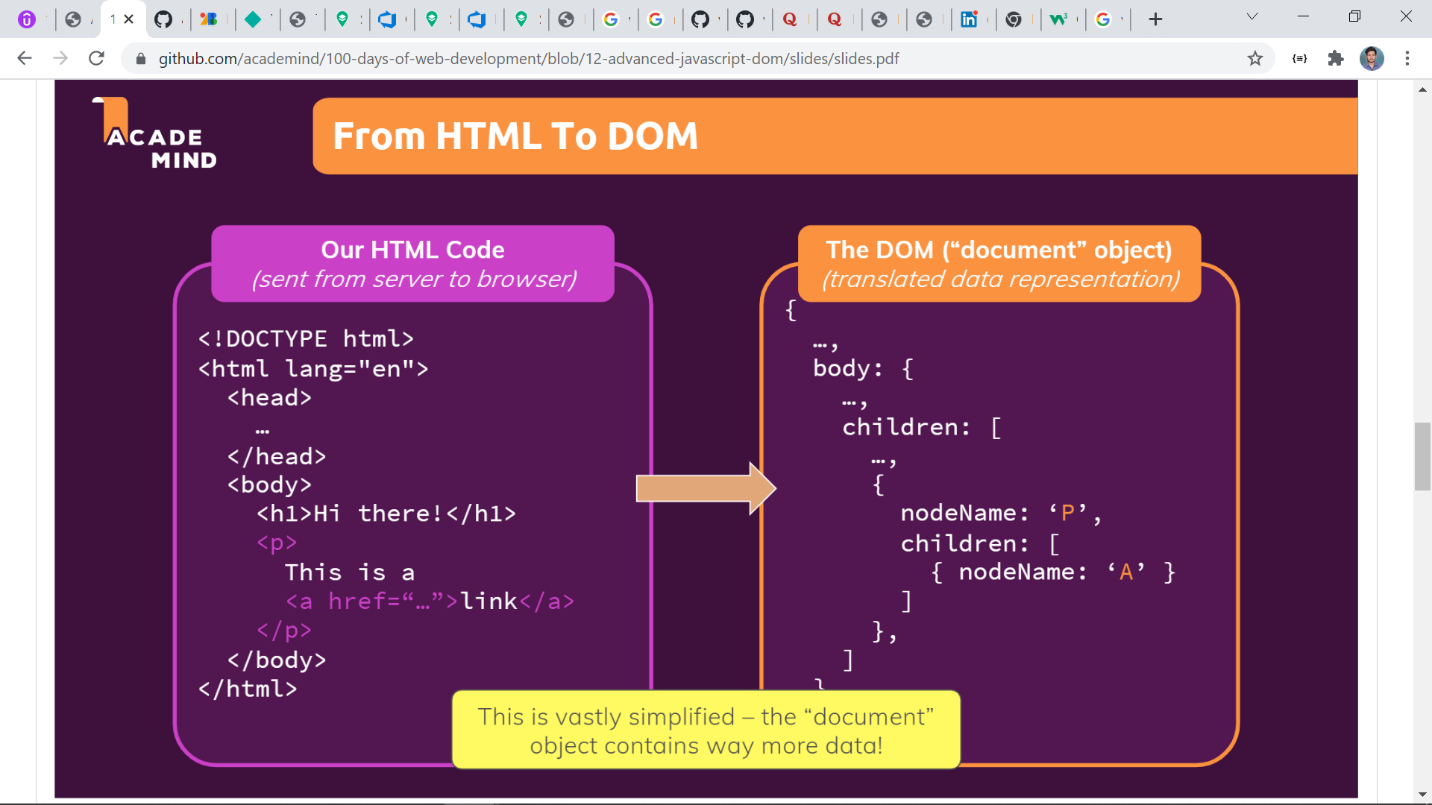
window is a global object and document is present inside window object. document object is door to start and manipulate the DOM.

Graphical user interface, diagram

Description automatically generated

**What is DOM ?**

DOM is a document object model which is the data representation (“internal representation”) of the parsed HTML code. Dom is like a tree. The browser parses ours HTML code and saves all elements as JavaScript objects. Since this HTML structure was translated into a structure of JavaScript objects, our JavaScript code is able to interact with the DOM.



**What is node in DOM?**

In the DOM, all parts of the document, such as elements, attributes, text, etc. are organized in a hierarchical tree-like structure; consisting of parents and children. These individual parts of the document are known as nodes.

**What is Nodejs ?**

Nodejs is “JavaScript for the server-side”. Nodejs is JavaScript run time environment built on chrome v8 engine that executes JavaScript code.

**To create a Unique Id in JavaScript?**

uuid, Date.now() =>{gives a number in milli seconds}

**What is difference between Promises and async/await?**

Promise is an object representing intermediate state of operation which is guaranteed to complete its execution at some point in future. Async/Await is a syntactic sugar for promises, a wrapper making the code execute more synchronously.

Promises are used to decrease the complexity involved around callback functions. A Promise is something which promises you to give back some response as data in future and on that promise, we have to apply then keyword to get the response as data. Promise chains can become difficult to understand sometimes.

**Security Threats:**

***CSRF attack*** : **C**ross **S**ite **R**equest **F**orgery . The idea behind this is the bad guy creates a request and that send to your backend that causes an action that shouldn’t be performed.

Think suppose we logged into our correct site with address localhost:3000. Then we receive a mail link which directs us to bad guy page, but we can’t see the difference because page looks exact same. When we are in a bad guy page example localhost:8000 and when you are submitting a form (the input request is modified but we don’t know that) which have an action path to localhost:3000, if then suppose we submit the form what will happen is browser see to what address(localhost:3000) the form request has to go, it sees it is localhost:3000 then it checks browser cookies that any belongs to localhost:3000 then they will attach to that modified request.

**Defense for CSRF attacks** :

**SameSite**: Yes, cookies can be attached to requests that come from request site but only if you visited that site from the main site. By default, some browsers like chrome has support for samesite which is ‘LAX’.

**CSRF token**: These tokens are random looking string values are generated on the server side which are only known by the server and which are short lived, they are life cycle is one request-response and which are generated by server.

**XSS attack** : XSS stands for Cross site scripts. It is all about injecting JavaScript code into our website.

**SQL Injection attack** :