Assignmenet-2

What happens when we type an URL in the browser and hit enter key?

Ans: When we are hits enter after entering an URL browser will check the cache for DNS record to find corresponding IP address of the URL

DNS(Domain name system) is an database that maintains the name of the website and particular IP address links to it. Every single URL in internet has unique IP address assigned to it.  The IP address belongs to the computer which hosts the server of the website we are requesting to access.

In order to find a DNS record Brower will checks the four caches.

1, First it checks browser cache. Browser maintains the repository of DNS record for a fixed duration for websites you have previously visited.it is the first place to run the DNS query.

2, Second the browser checks the OS Cache. if it is not found in the browser cache browser makes the system call to your OS to fetch the DNS record since OS also maintains the cache of a DNS record.

3, Third, it checks the router cache. If it’s not found on your computer, the browser would communicate with the router that maintains its’ own cache of DNS records.

4, last browser checks the ISP cache if all steps fails then browser will checks the ISP, it can maintains the its own cache of DNS record

In case ISP also not having an DNS record at that time ISP DNS server initiates a DNS query to find the IP address of the server that hosts the websites

These requests are sent using small data packets which contain information such as the content of the request and the IP address it is destined for. These packets travel through multiple networking equipment between the client and the server before it reaches the correct DNS server. This equipment use routing tables to figure out which way is the fastest possible way for the packet to reach its’ destination. If these packets get lost you’ll get a request failed error. Otherwise, they will reach the correct DNS server, grab the correct IP address, and come back to your browser.

**Browser initiates an TCP connections with server**

Once the browser receives the correct IP address it initiates an connections with servers that matches the IP address in order to fetch requested information. Browser use internet protocols to build connections.

**The browser sends an HTTP request to the web server.**

Once the connections established its time to transferred requested data. Browser will send a GET request for asking the requested URL

**The server handles the GET request and sends back a response**

The server contains the web server witch receives the request from browser and passes it to an request handler in order to read and generate a data

**Server sends out an HTTP response That you requested.**

**The Browser will displays the requested content.**

**Accessibility**

Accessibility is essential for developer and organizations that want to create high reach web site and web tools and they not exclude the people from using their products and services

The Web is fundamentally designed to work for all people, whatever their hardware, software, language, location, or ability. When the Web meets this goal, it is accessible to people with a diverse range of hearing, movement, sight, and cognitive ability.

Thus the impact of disability is radically changed on the Web because the Web removes barriers to communication and interaction that many people face in the physical world. However, when web sites, applications, technologies, or tools are badly designed, they can create barriers that exclude people from using the Web.

**Semantics**

**In programming semantics means it is an peace of code if that line of code exist means what happens and what is the role of HTML if that code runs it not at all related to how they look.**

**Semantics in JAVA**

In JavaScript, consider a function that takes a string parameter, and returns an [<li>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li) element with that string as its text Content. Would you need to look at the code to understand what the function did if it was called build

**Semantics in CSS**

**Semantics in CSS, we can style an one of the element means how they will react and how the what part of DOM will select with that element. All this will considered.**

**Semantics in HTML**

In HTML, for example, the [<h1>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h1) element is a semantic element, which gives the text it wraps around the role (or meaning) of "a top level heading on your page."

**<h1> This is a top level heading </h1>**

By default browser will display the all font with big size and make it look like Heading.

**On the other hand we can make an heading by using the other HTML elements as follows**

**<span style=”font-size:32px; margin:30px”>** **This is a top level heading** **</span>**

**On the other hand this will also look like heading only but it will not having an semantic value so it will not get an any extra benefits. So we can proper elements as we want.**

<articale>

<aside>

<details>

<figure>

<footer>

<header>

<main>

<marks>

<nav>

<selection>

<time>

<summary>

**Form Factor**

Form factor is an aspect of hardware [design](https://en.wikipedia.org/wiki/Design) which defines and prescribes the size, shape, and other physical [specifications](https://en.wikipedia.org/wiki/Specification_(technical_standard)) of components, particularly in [consumer electronics](https://en.wikipedia.org/wiki/Consumer_electronics) and [electronic packaging](https://en.wikipedia.org/wiki/Electronic_packaging)

**CSS GRID**

CSS GRID layout is two dimensional layout system for an web development it can allows you to arrange the content with rows and columns and having an many advances.

What is grid layout?

A grid layout is an collection of vertical and horizontal lines creating a pattern to design and place an elements. They help us to arrange the elements with proper manner and it can’t shift any where, when we move one page to another.

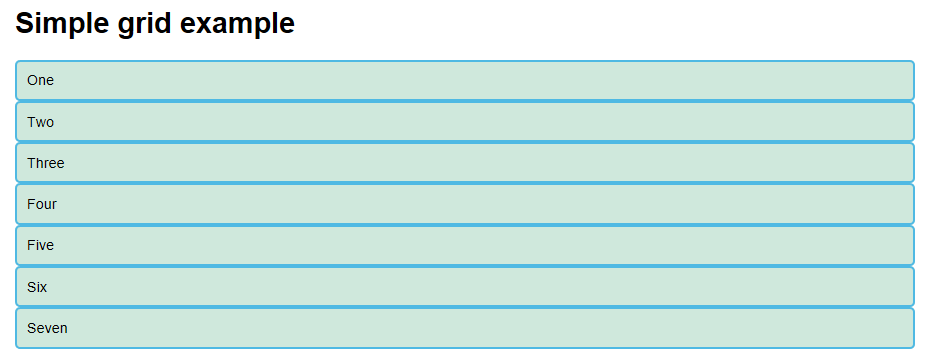
Here Is an example of grid



**Creating your grid system.**

Defining of Grid

As stating point we can download and open stating point file in your browser and text editor, for example it can see as bellow, by default this grids can be display one below the other and we arrange them as our requirement.



To define a grid we use the grid value of the display property. As with Flexbox, this switches on Grid Layout, and all of the direct children of the container become grid items. Add this peace of code to your the CSS file:

.container {

display: grid;

}

Unlike flexbox, the items will not immediately look any different. Declaring display: grid gives you a one column grid, so your items will continue to display one below the other as they do in normal flow.

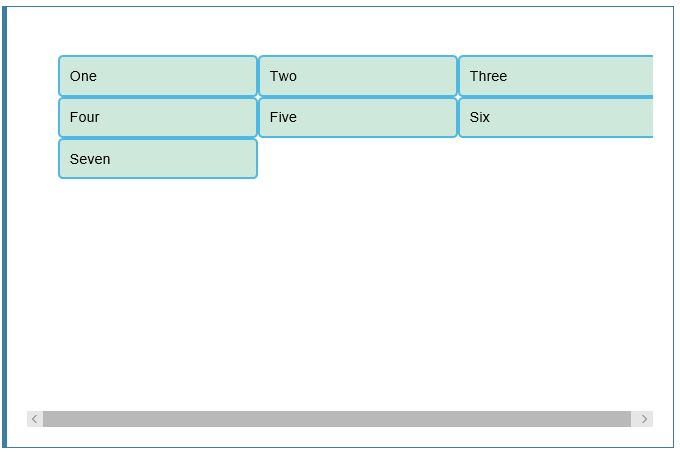
To see the more grids you want to add more column to CSS grids as follows.

.container {

display: grid;

drid-template-columns: 200px 200px 200px

}

If we add this seconds to your CSS file and reload your HTML page you will get an following display

**Flexible grid with the fr unit**

In addition to create grid by mentioning rows and column we can mention fr unit to your rows and column. This means one fraction of available space in grid layout.

For example

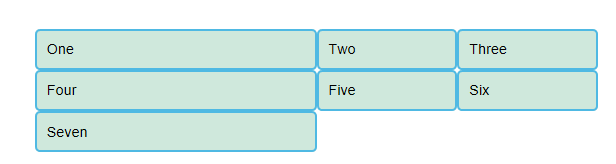
.container {

display: grid;

drid-template-columns: 1fr 1fr 1fr;

}

You can give deferent values to grid system as you want The fr unit distributes space in proportion, therefore you can give different positive values to your tracks, for example if you change the definition like so:



**Gaps between grids**

To create gaps between tracks we use the properties grid-column-gap for gaps between columns grid-row-gap for gaps between rows, and grid-gap to set both at once.

For example

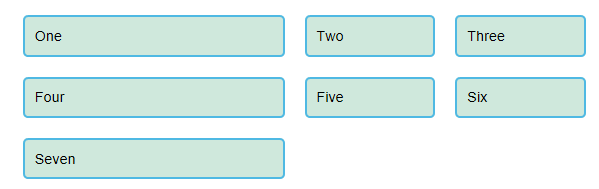
.container {

display: grid;

grid-template-columns: 1fr 1fr 1fr;

grid-gap:20px;

}



For example

.container {

display: grid;

grid-template-columns: 1fr 1fr 1fr;

grid-gap:20px;

grid:20px;

}

**Repeating track listening**

You can repeat all, or a section of, your track listing using repeat notation. Change your track listing to the following:

For example

.container {

display: grid;

grid-template-columns:reapet(3. 1fr);

grid-gap:20px;

}

You will now get 3 1fr tracks just as before. The first value passed to the repeat function is how many times you want the listing to repeat, while the second value is a track listing, which may be one or more tracks that you want to repeat

**The implicit and explicit**

We have only specified column tracks so far, and yet rows are being created to hold our content. This is an example of the explicit versus the implicit grid. The explicit grid is the one that you create using grid-template-columns or grid-template-rows. The implicit grid is created when content is placed outside of that grid — such as into our rows. The explicit and implicit grids are analogous to the main and cross flexbox axes.

By default, tracks created in the implicit grid are auto sized, which in general means that they are large enough to fit their content. If you wish to give implicit grid tracks a size you can use the [grid-auto-rows](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-auto-rows) and [grid-auto-columns](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-auto-columns) properties. If you add grid-auto-rows with a value of 100px to your CSS, you will see that those created rows are now 100 pixels tall.

For example

.container {

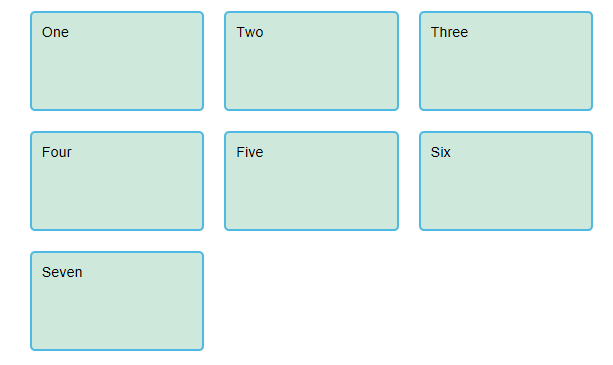
display: grid;

grid-template-columns:reapet(3. 1fr);

grid-auto-rows:100px;

grid-gap:20px;

}

****

**The minmax() Function**

Our 100-pixel tall tracks won’t be very useful if we add content into those tracks that is taller than 100 pixels, in which case it would cause an overflow. It might be better to have tracks that are *at least* 100 pixels tall and can still expand if more content gets into them. A fairly basic fact about the web is that you never really know how tall something is going to be; additional content or larger font sizes can cause problems with designs that attempt to be pixel perfect in every dimension.

The minmax function lets us set a minimum and maximum size for a track, for example minmax(100px, auto). The minimum size is 100 pixels, but the maximum is auto, which will expand to fit the content. Try changing grid-auto-rows to use a minmax value:

For example

.container {

display: grid;

grid-template-columns:reapet(3. 1fr);

grid-auto-rows :minmax(100px , auto);

grid-gap:20px;

}

**As many column as will fit**

Sometimes it is helpful to be able to ask grid to create as many columns as will fit into the container. We do this by setting the value of grid-template-columns using repeat() notation, but instead of passing in a number, pass in the keyword auto-fill. For the second parameter of the function we use minmax(), with a minimum value equal to the minimum track size that we would like to have, and a maximum of 1fr.

Try this in your file now, using the below CSS:

For example

.container {

display: grid;

grid-template-columns: reapet(auto-fill, minmax(100px , 1fr));

grid-auto-rows :minmax(100px , auto);

grid-gap:20px;

}

This works because grid is creating as many 200 pixel columns as will fit into the container, then sharing whatever space is leftover between all of the columns — the maximum is 1fr which, as we already know, distributes space evenly between tracks.

**Line based Placement**

We now move on from creating a grid, to placing things on the grid. Our grid always has lines, these lines start at 1 and relate to the Writing Mode of the document. Therefore in English, column line 1 is on the left hand side of the grid and row line 1 at the top

We can place things according to these lines by specifying the start and end line. We do this using the following properties:

grid-column-start

grid-column-end

grid-row-start

grid-row-end

These properties can all have a line number as the value. You can also use the shorthand properties:

grid-column

grid-row

**What are the previous version of HTML5? Why we are in HTML5?**

The main difference between HTML and HTML 5 is that video and audio are not part of HTMLwhile these both are integral parts of HTML5specifications. The primary thing you ought to know is that, maybe interestingly, the improvement of a dialect standard is recognizing this present reality. So as to keep document similarity with the present standard – which is, in fact, HTML 4.01 – the bold choice was made to particular the way the web program renders records from the route we, as engineers, must think of them. So the program, or “client specialist”, should even now handle HTML4 develops like the inside component, on the grounds that there will, in any case, a large number of records on the Internet that happen to utilize it. In any case, we won’t compose any more HTML with focus; it’s essentially being dropped from the dialect (use CSS). This similarity goes both ways: more established programs can (and will) basically overlook HTML5 code without botching things.

As cool as it is to see what HTML 5 can accomplish for you, it hasn’t been institutionalized like HTML4. You don’t need to stress over overhauling pages assembled utilizing HTML 4. It’s over ten years of age and it’s a set standard obviously, this is needy the amount you rely on upon rich components, however, it’s absolutely a danger you should mull over when utilizing a liquid dialect. HTML5 is utilized to build up the capacity of the program to be an application stage, by means of HTML, CSS, and Javascript. Numerous components have been added specifically to the dialect that is as of now (in HTML4) Flash or JS-based hacks, for example, <canvas>, <video>, and <audio>.

However, HTML4 is still a W3C standard for program applications, it doesn’t completely oblige the changing patterns of the registering business. In this way, HTML5 has been created with the goal to cop-up with these new difficulties in the web industry. HTML5 is more adaptable, powerful and progressed as a contrast with its more established partner.

**Why we need HTML? Deferent ways to make a grid layout in HTML?**

**Why we need HTML?**

HTML is a markup language, heavily utilized for creating web pages and web applications. HTML, when combined with JavaScript and CSS, has become a more advantages for development. One of the useful aspects of HTML is, it can embed programs [written in a scripting language](https://www.educba.com/programming-languages-vs-scripting-languages/) like JavaScript, which is responsible for affecting the behaviour and content of web pages. CSS inclusion would affect the layout and appearance of the content. The basic building blocks of any HTML pages are HTML elements. A structured document can be created with the help of structural-semantic text like heading, paragraph, list, link, and other items. Browser indeed does not display the HTML tags but utilize them to interpret the content of the page. One needs to study various tags and then understand their behaviour.

There are five different ways to create multicolumn layouts. Each way has its pros and cons:

HTML tables

CSS float property

CSS flexbox

CSS framework

CSS grid

HTML tables

The <table> element was not designed to be a layout tool! The purpose of the <table> element is to display tabular data. So, do not use tables for your page layout! They will bring a mess into your code. And imagine how hard it will be to redesign your site after a couple of months.

**CSS Float property :**

It is common to do entire web layouts using the CSS float property. Float is easy to learn - you just need to remember how the float and clear properties work.

**Disadvantages:** Floating elements are tied to the document flow, which may harm the flexibility. Learn more about float in our [CSS Float and Clear](https://www.w3schools.com/css/css_float.asp) chapter.

**Flexbox**

Flexbox is a new layout mode in CSS3.

Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.

**Disadvantages:** Does not work in IE10 and earlier.

CSS Grid View

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

**Disadvantages:** Does not work in IE nor in Edge 15 and earlier.

**CSS Grid View**

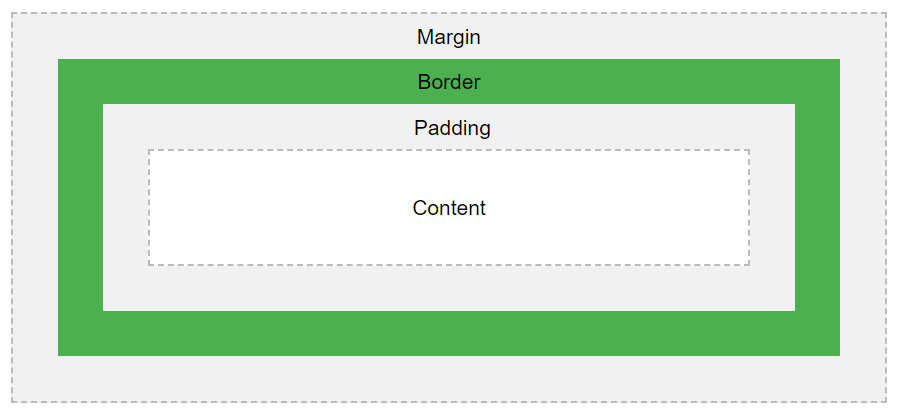
The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

**Disadvantages:** Does not work in IE nor in Edge 15 and earlier.

**BOX Model**

All HTML elements can be consider as boxes. In css BOX model word can use under the design and layout process.

CSS box model is nothing but box only it can contain padding, margin, boards and actual content means the how the the HTML page can display the content in this box model.as shown below.



Explanation of the different parts:

* **Content** - The content of the box, where text and images appear
* **Padding** - Clears an area around the content. The padding is transparent
* **Border** - A border that goes around the padding and content
* **Margin** - Clears an area outside the border. The margin is transparent

**DOM (Document Object Model)**

When the HTML page Loaded the browser will creates the document object model.HTML DOM model is constructed as tree of objects as Shown below



What Is DOM?

The DOM is a W3C (World Wide Web Consortium) standard.

The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.

The W3C DOM standard is separated into 3 different parts:

* Core DOM - standard model for all document types
* XML DOM - standard model for XML documents
* HTML DOM - standard model for HTML documents

What is HTML DOM

The HTML DOM is a standard object model and programming interface for HTML. It defines:

* The HTML elements as **objects**
* The properties of all HTML elements
* The methods to access all HTML elements
* The events for all HTML elements

**The box model**

**Block and Inline blocks**

In CSS we broadly have two types of boxes **block boxes** and **inline boxes**. These characteristics refer to how the box behaves in terms of page flow, and in relation to other boxes on the page:

If a box is defined as a block, it will behave in the following ways:

The box will extend in the inline direction to fill the space available in its container. In most cases this means that the box will become as wide as its container, filling up 100% of the space available.

The box will break onto a new line.

The [width](https://developer.mozilla.org/en-US/docs/Web/CSS/width) and [height](https://developer.mozilla.org/en-US/docs/Web/CSS/height) properties are respected.

Padding, margin and border will cause other elements to be pushed away from the box

Unless we decide to change the display type to inline, elements such as headings (e.g. <h1>) and <p>s all use block as their outer display type by default.

If a box has an outer display type of inline, then:

The box will not break onto a new line.

The [width](https://developer.mozilla.org/en-US/docs/Web/CSS/width) and [height](https://developer.mozilla.org/en-US/docs/Web/CSS/height) properties will not apply.

Padding, margin and borders will apply but will not cause other inline boxes to move away from the box.

The <a> element, used for links, <span>, <em> and <strong> are all examples of elements that will display inline by default.

The type of box applied to an element is defined by [display](https://developer.mozilla.org/en-US/docs/Web/CSS/display) property values such as block and inline, and relates to the outer value of display.

**Inner and outer display types**

As mentioned above display type have an inner and outer display.

Boxes also have an inner display type, however, which dictates how elements inside that box are laid out. By default the elements inside a box are laid out in [normal flow](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Normal_Flow), which means that they behave just like any other block

We can however, change the inner display type by using display values like flex. If we set display: flex; on an element, the outer display type is block, but the inner display type is changed to flex. Any direct children of this box will become flex items, and will be elements will be displayed according to the css rules. Block and inline layout is an default way that the our web pages will works.