**CSS Notes**

CSS Intro

CSS stands for Cascading Style Sheet it describes how HTML elements and document layout are to be presented on screen. that includes font, color, size, and more, of the elements.

Syntax:

selector {

property: value ;

}



Types of CSS and their Priority

Types of CSS

Adding CSS can be done three different ways i.e.

* Inline CSS: Inline CSS has directly applied a unique style to a single element of the element of HTML document using the style attribute.
* Internal CSS: Internal CSS holds style code for the entire web page using selectors, Internal is defined inside of the HTML file using the style tag and the style tag should be inside the head tag.
* External CSS: It is the different CSS file where we define the HTML element's style using some selectors which we discuss later. To use external CSS, we need to link the CSS file to the HTML file using a link tag i.e.

<link rel= “stylesheet” type=” text/css” href=” ./style.css”

Where the rel(required) attribute defines the relationship of css file with html, the type attribute defines the document type and the href attribute gives the reference of the file.

Priority of CSS types

Cascading order priority: Inline > (internal ≅ external) > browser default

Selectors

Selectors

Selectors point to the HTML element which we want to style. We use selectors in internal and external stylesheets. There are three types of selectors that are used to apply styles:

* Element Selector:

The element selector selects all elements with the specified element name. This will select all the elements in the HTML document. This method is not used so commonly. So, to apply styles to only some elements, we need to use some restrictions.

Syntax: element {css declarations;}

* Class Selector

The class selector selects multiple elements which have the same class with a specific class attribute. To select elements with a specific class, write a period(.) character, followed by the name of the class.

Syntax: .class-name {css declarations;}

* ID Selector

The id selector selects only one element with a specific id attribute. To select an element with a specific id, write a hash (#) character, followed by the name of the id.

Syntax: #class-name {css declarations;}

* Grouping Selector

Sometimes we need to apply specific styles to different elements, and to reduce the redundancy in the code, we use grouping. Separate the selectors with a comma followed by the CSS declarations.

* Universal Selector

The universal selector (\*) selects all HTML elements on the page

Priority of CSS Selectors

Universal > ID > Class > Element

Note: If the same property is defined inside the same type of selector, then the property defined at the last will be used by the browser.

Hierarchical Selector and Specificity

Hierarchical Selector

We can also select the elements in different ways with more specific i.e., by using a parent selector we can target the child element which makes the selection too specific. We can target the child by traversing parent-to-child separated by space.

Example: #parent .child .sub-child sub-child-element

Specificity of Hierarchical Selectors

Reference: [Link of Specificity](https://polypane.app/css-specificity-calculator/)

Fonts, Color, and Background

Text in CSS

We can style and align the text of HTML documents by using font-size, font-family, font-weight, color, and text-align, text-indent, text-decoration, line-height, letter-spacing, word-spacing properties.

The font properties are used to style the Text of an HTML document which can have some value and the important properties are:

* font-family: Roboto, sans-serif;
* font-size: 10em;
* font-weight: 700;
* font-style: italic;

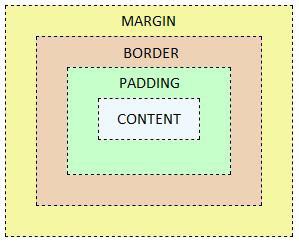
The color property is as simple as that used for the color of the Text of HTML documents. And there is only one property i.e., color: red; and the value may be hex, hsl, or rgb format.

Box Model

Box Model

The box model describes the layout of the elements. The HTML elements are considered as boxes.

The CSS box model is essentially a box that wraps around every HTML element. It consists of margins, borders, padding, and content. The image below illustrates the box model.



* 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.

The element’s total width is equal to the total of the horizontal borders, padding, and content width of the element.

The total height of the element is equal to the total of the vertical borders, padding, and content height of the element.

What is Box Sizing Property?

The box-sizing CSS property sets how the total width and height of an element are calculated. i.e.,

* Content-box: The default width and height values apply to the element's content only. The padding and border are added to the outside of the box.
* Padding-box: Width and height values apply to the element's content and its padding. The border is added to the outside of the box. Currently, only Firefox supports the padding-box value.
* Border-box: Width and height values apply to the content, padding, and border

Background in CSS

Background

The background properties are used to define the background effects for elements. The background of an element is the total size of the element. It includes padding and borders but not the margin.

Backgrounds can be filled with a color or image, clipped or resized, and otherwise modified. CSS background properties:

* background-color
* background-image
* background-repeat
* background-size
* background-position
* background-attachment

The background-color property sets the background color of an element. It has the same value as that of the color property.

The background-image property is used to specify an image to use as the background of an element. This can set one or more background images for an element. url value is used to set the background image.

❖ url('URL') - specifies the URL of the image. You can specify more than one image by separating the URLs with a comma

NOTE: By default, a background image is placed at the top-left corner of an element and is repeated, covering the entire element both vertically and horizontally.

The background-repeat property is used to specify how/if a background image will be repeated. By default, a background image repeats both vertically and horizontally, so background-repeat will how the image repetition works.

The values this property can take are:

* repeat - This is the default value. The background image is repeated both vertically and horizontally. The last image will be clipped if it does not fit
* repeat-x - The image is repeated only horizontally
* repeat-y - The image is repeated only vertically
* no-repeat - The image will only be shown once
* space - the background image is repeated without clipping. The space remaining is distributed evenly between images, with the first and last images pinned to the sides of the element
* round - the image is repeated and shrunk or stretched to fill the space

The background-size property is used to specify the size of the background images. The values it can take are:

* auto - This is a default value. The image is displayed in its original size
* length - sets the width and height of the background image. The first value sets the width, the second value sets the height
* percentage - sets the width and height of the background image in percent. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto"
* cover - resizes the background image to cover the horizontal width of the container
* contain - resizes the background image to make sure the image is fully visible

The background-position property is used to specify the initial position of a background image. By default, a background image is placed at the top-left corner of an element, and you can change the position with the background-position property.

The values this property can take are (X represents horizontal position and Y represents vertical position):

* X Y - they both can each take value from one of the following - left, right, top, bottom, center. If one value is specified, the other value will be "center."
* Xpos Ypos - specifies the horizontal and vertical position relative to the viewport. Units can be any of the CSS units. If you only specify one value, the other value will be 50%.

The background-attachment property is used to specify whether a background image scrolls with the rest of the page or is fixed. The values it can take are:

* scroll - This is the default value. The background image will scroll with the page
* fixed - The background image will not scroll with the page
* local - The background image will scroll with the element's contents

Overflow

Overflow content

The overflow property is used to specify what happens if the content of an element overflows, i.e., the content’s height or width is larger than the element’s height or width. This property adds a scroll bar or clips the content when an element's content is too big to fit in a specified area.

The values that the overflow property can take are:

* visible - This is the default value. The content overflows and is seen outside the box
* hidden - Only the content that fits inside the box is visible, and the overflow is clipped
* scroll - All the content is visible through a scroll bar added to the box. NOTE: This adds a scrollbar both horizontally and vertically.
* auto - a scroll bar gets added if content overflows visible, auto is similar to scroll, but it only adds the scrollbar if needed.
* overflow-x and overflow-y - The overflow-x and overflow-y are used to handle the overflow horizontally or vertically i.e.,

**CSS :**

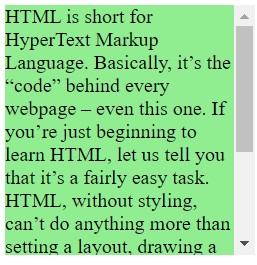
div {

overflow-x: hidden; /\* Hide horizontal scrollbar \*/

overflow-y: scroll; /\* Add vertical scrollbar \*/

}

**Browser :**



NOTE: The overflow property only works for block elements with a specified height.

Height and Width

Absolute Height/Width

By using the height/width properties of the css we define the fixed size of the block element i.e., it does not depend on any parent element, and if we want to make the element responsive then css also facilitates to use of max/min with height/width i.e.,

Relative max/min with Height/Width

This specifies the maximum or minimum height/width that an element can have. If the browser window's height/width becomes smaller or larger than the element’s height/width, the element's height/width adjusts with the browser's height/width.

Units

Units in CSS

Units are used to measure or define the size of the elements i.e., length, width, padding, margin, and font size there are two types of measurement units in CSS:

Absolute Units

The absolute units are a fixed size/length of the element. Absolute length units are not recommended for use on-screen because screen sizes vary so much. The absolute units consist of the following:

● cm - centimeters

● mm - millimeters in inches (1in = 96px = 2.54cm)

● px - pixels (1px = 1/96th of 1in)

● pt - points (1pt = 1/72 of 1in)

● pc - picas (1pc = 12 pt)

Relative Units

Relative length units specify a length relative to another length property. Some of the relative units are the following:

● em - Relative to the font size of the element (2em means 2 times the size of the current font)

● rem - Relative to the font size of the root element

● vw - Relative to 1% of the width of the browser window size

● vh - Relative to 1% of the height of the browser window size

● % - Relative to the parent element

Display

Position

z-index

Float and Clear

Flex Box

Flex Box

The flex is a value for the display property. It has to be provided in the container for the flex to work. Only if it is defined inside the container, flex properties will work. Flex properties are defined on the child elements.

The Flexible Box Layout Module makes designing a flexible, responsive layout structure easier without using float or positioning.

To make the container to be flex, add this property to the container:

● display: flex;

● display: inline-flex; for the inline variation.

Flex-Direction

The flex-direction property defines in which direction the container lays out the flex-items. It may take 4 values:

● flex-direction: row (Default)

The row value stacks the flex items horizontally, from left to right.

● flex-direction: column

The column value stacks the flex items vertically, from top to bottom.

● flex-direction: row-reverse

The row-reverse value stacks the flex items horizontally, from right to left.

● flex-direction: column-reverse

The column-reverse value stacks the flex items vertically, from bottom to top.

Order of Flex items

We can define the order of the flex item using order property i.e., order: 1;

Order is manipulated by value in ascending order by default order of items is 0.

Wrap in Flex Box

The flex-wrap property specifies whether the flex items should wrap or not. By default, flex items try to fit into one line.

This property allows you to change that and allow the items to flow into multiple lines as needed with this property.

● nowrap (Default)

● wrap

● wrap-reverse

Flex Grow and Shrink

The flex-grow property specifies the ratio with which a flex item grows relative to the other flex items when there is some extra space available. This controls the extent how much a flex item grows, with respect to other flex items. This takes up any value from 0(zero) to any positive number.

0(zero) means that the flex items' width would not change.

The flex-shrink property specifies the ratio with which a flex item shrinks relative to the other flex items when there is some extra space available. This is just the opposite of flex-grow. Here, also the range of value is from 0(zero) to any positive number. Zero means the width would not change.

Justify-Content

The justify-content property is used to align the flex items along the main axis. This defines the alignment along the main axis. This property distributes extra free space inside the layout between the elements. The justify-content property takes on any of the values below:

* flex-start (Default)
* flex-end
* center
* space-between
* space-around
* space-evenly

Align-Item

The align-items property is used to align the flex items along the cross-axis. The cross-axis is perpendicular to the main axis. If the main axis is horizontal, then the cross-axis is vertical. If the main axis is vertical, then the cross axis is horizontal. Align-items can be set to any of these values:

* flex-start
* flex-end
* center
* stretch (Default)
* baseline

Align-Content

Align-content property is used to align the flex lines, i.e., multiple lines of flex items. This property aligns flex lines when there is extra space in the cross-axis, similar to how justify-content aligns individual items within the main-axis. Align-content can be set to any of these values:

* stretch (Default)
* flex-start
* flex-end
* center
* space-between
* space-around

Align-Self

The align-self property is used for the alignment of selected flex items inside the flex container. This overrides the default alignment set by the flex container. Align-self can be set to any of these values:

* auto (Default)
* flex-start
* flex-end
* center
* stretch
* baseline

Grid

Media Query

Pseudo-Class and Pseudo-Element