CSS – CASCADING STYLE SHEET

CSS:

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

USES OF CSS:

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

ALL CSS SELECTORS:

1. #id
2. .class
3. Element. Class
4. * is a universal selector
5. Element like tags

ADD CSS IN HTML:

When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.

There are 3 ways of inserting a style sheet with cascading oder:

1. Inline style (inside an HTML element)
2. External and internal style sheet (in the section)
3. Brower default

CSS COLORS:

Colors are specified using predefined color names or RGB, HEX, HSL, RGBA, HSLA values.

CSS BACKGROUNDS:

The CSS background properties are used to add background effects for elements.

Property	Description
background	Sets all the background properties in one declaration
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-clip	Specifies the painting area of the background
background-color	Sets the background color of an element
background-image	Sets the background image for an element
background-origin	Specifies where the background image(s) is/are positioned
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated
background-size	Specifies the size of the background image(s)

ALL CSS BORDER:

Property	Description
Border	Sets all the border properties in one declaration
border-color	Sets the color of the four borders
border-style	Sets the style of the four borders
border-width	Sets the width of the four borders
border-radius	Sets all the four border-*-radius properties for rounded corners
border-top	Sets all the top border properties in one declaration
border-top-color	Sets the color of the top border
border-top-style	Sets the style of the top border
border-top-width	Sets the width of the top border
border-right	Sets all the right border properties in one declaration
border-right-color	Sets the color of the right border

border-right-style	Sets the style of the right border
border-right-width	Sets the width of the right border
border-bottom	Sets all the bottom border properties in one declaration
border-bottom-color	Sets the color of the bottom border
0border-bottom-style	Sets the style of the bottom border
border-bottom-width	Sets the width of the bottom border
border-left	Sets all the left border properties in one declaration
border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border

CSS BORDER STYLE:

The **border-style** property specifies what kind of border to display. The **border-style** property can have from one to four values for the

1. **border-top-style**
2. **border-right-style**
3. **border-bottom-style**
4. **border-left-style**

The following values are allowed:

1. **dotted** - Defines a dotted border
2. **dashed** - Defines a dashed border
3. **solid** - Defines a solid border
4. **double** - Defines a double border
5. **groove** - Defines a 3D grooved border. The effect depends on the border-color value
6. **ridge** - Defines a 3D ridged border. The effect depends on the border-color value
7. **inset** - Defines a 3D inset border. The effect depends on the border-color value
8. **outset** - Defines a 3D outset border. The effect depends on the border-color value
9. **none** - Defines no border
10. **hidden** - Defines a hidden border

CSS BORDER WIDTH:

The **border-width** property specifies the width of the four borders. The width can be set as a specific size (in **px, pt., cm, em**, etc.) or by using one of the three pre-defined values: **thin, medium, or thick**. The border-width property can have from one to four values

1. **border-top-width**
2. **border-right-width**
3. **border-bottom-width**
4. **border-left-width**

CSS MARGIN:

The CSS **margin** properties are used to create space around elements, outside of any defined borders. With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (**top, right, bottom, and left**).

Values are used **auto, length (**cm. pt., px**), margin %, inherit**-specifies that the margin should be inherit from the parent element.

Property	Description
margin	A shorthand property for setting all the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element

Shorthand property:

1. If the **margin** property has four values:

   {margin: 25px 50px 75px 100px;}

   1. top margin is 25px
   2. right margin is 50px
   3. bottom margin is 75px
   4. left margin is 100px

2. If the **margin** property has three values:

   {margin: 25px 50px 75px 1;}

   1. top margin is 25px
   2. right and left margin is 50px
   3. bottom margin is 75px

3. If the **margin** property has two values:

{margin: 25px 50px;}

1. top and bottom margin is 25px
2. right and left margin is 50px

4. If the **margin** property has one value:

{margin: 25;}

1. all four margins are 25px

Margin collapse:

Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins. This does not happen on left and right margins! Only top and bottom margins!

Look at the following example:

{margin: 0 0 50px 0;}

{margin: 20px 0 0 0;}

CSS PADDING:

The CSS **padding** properties are used to generate space around an element's content, inside of any defined borders. With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (**top, right, bottom, and left**).

Values are used **auto, length (**cm. pt., px**), margin %, inherit**-specifies that the margin should be inherit from the parent element.

Property	Description
padding	A shorthand property for setting all the padding properties in one declaration
padding-bottom	Sets the bottom padding of an element
padding-left	Sets the left padding of an element
padding-right	Sets the right padding of an element
padding-top	Sets the top padding of an element

CSS HEIGHT, WIDTH AND MAX-WIDTH:

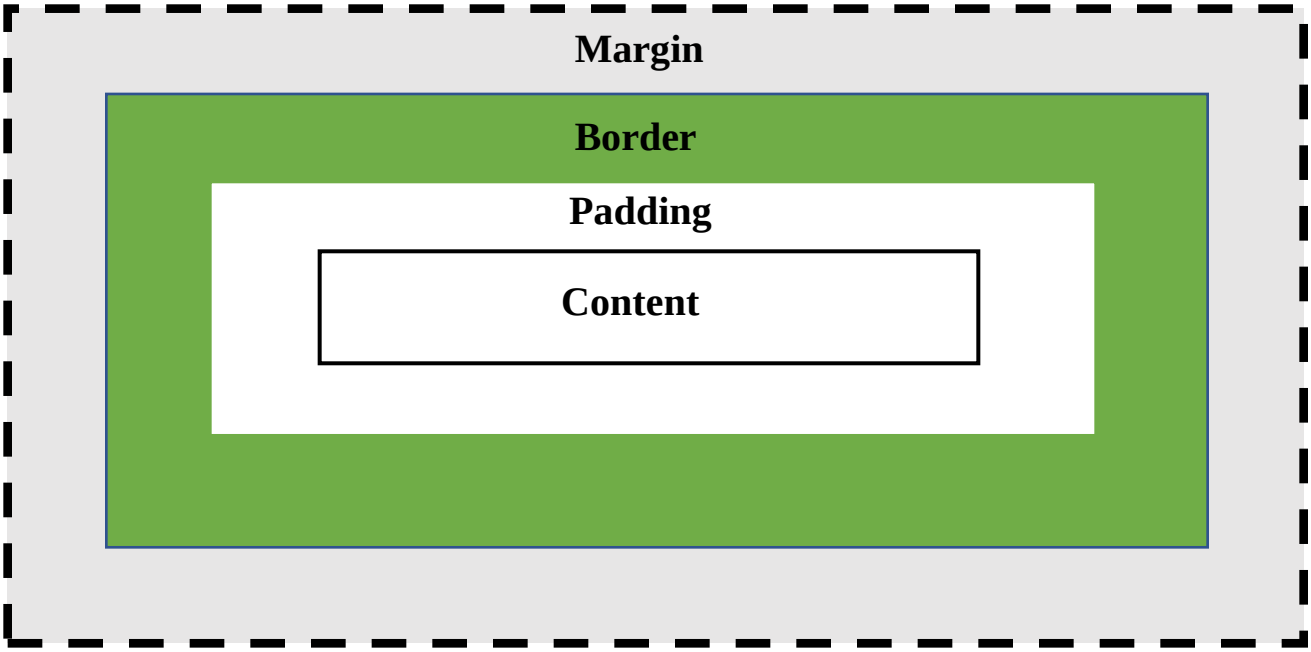
The **height** and **width** properties are used to set the height and width of an element.

The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

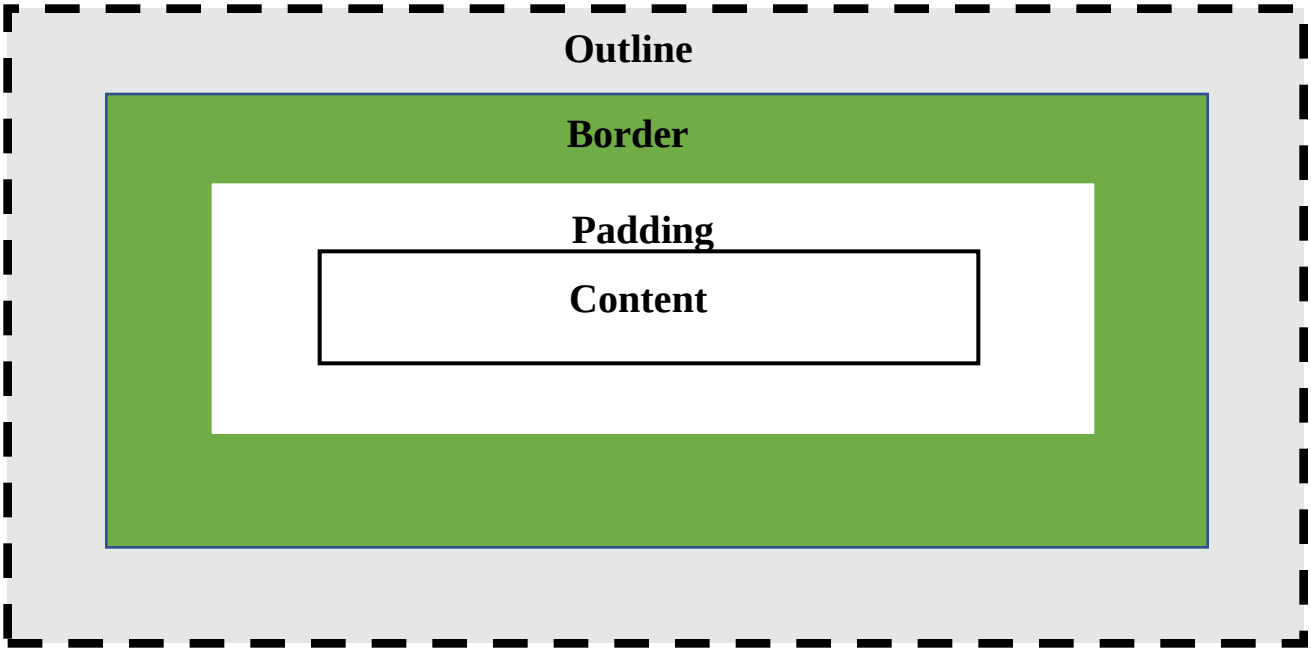
If you for some reason use both the width property and the **max-width** property on the same element, and the value of the **width** property is larger than the **max-width** property; the **max-width** property will be used (and the **width** property will be ignored).

Property	Description
height	Sets the height of an element
max-height	Sets the maximum height of an element
min-height	Sets the minimum height of an element
width	Sets the width of an element
min-width	Sets the minimum width of an element
max-width	Sets the maximum width of an element

CSS BOX MODEL:



CSS OUTLINE:



Property	Description
outline	A shorthand property for setting outline-width, outline-style, and outline-color in one declaration
outline-color	Sets the color of an outline
outline-offset	Specifies the space between an outline and the edge or border of an element
outline-style	Sets the style of an outline
outline-width	Sets the width of an outline

CSS TEXT:

Property	Description
color	Specifies the color of text
letter-spacing	Specifies the space between characters in a text
line-height	Specifies the line height
text-indent	Specifies the indentation of the first line in a text-block
white-space	Specifies hoe to handle white-space inside an element
word-spacing	Specifies the space between words in a text
direction	Specifies the text direction/writing direction
text-align	Specifies the horizontal alignment of text
text-align-last	Specifies how to align the last line of a text
unicode-bidi	Used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document
vertical-align	Sets the vertical alignment of an element

text-decoration	Set all the text-decoration properties in one declaration
text-decoration-color	Specifies the color of the text-decoration
text-decoration-line	Specifies the kind of text decoration to be used (underline, overline, line through, etc.)
text-decoration-style	Specifies the style of the text decoration (Solid, dotted, dashed, etc.)
text-decoration-thickness	Specifies the thickness of the text decoration line
text-transformation	Controls the capitalization of text
text-shadow	Specifies the shadow effect added to text

CSS FONTS:

In CSS, we use the **font-family** property to specify the font of a text there are five generic font families:

1. **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
2. **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
3. **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
4. **Cursive** fonts imitate human handwriting.
5. **Fantasy** fonts are decorative/playful fonts.

All the different font names belong to one of the generic font families.

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	Brush Script MT Lucida Handwriting
Fantasy	COPPERPLATE Papyrus

Note: 16px = 1em

FONT PAIRINGS:

1. Georgia and verdana
2. Helvetica and Garamond
3. Merriweather and open sans
4. Cinzel and fauna one
5. Spectral and Rubik

PROPERTIES:

Property	Description
font	Sets all the font properties in one declaration
font-family	Specifies the font family for text
font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-weight	Specifies the weight of a font

CSS LINKS:

The four links states are:

- **a: link** - a normal, unvisited link
- **a: visited** - a link the user has visited
- **a: hover** - a link when the user mouse over it
- **a: active** - a link the moment it is clicked.

When setting style for several link states, there are some order rules:

- **a: hover** must come after **a: link** and **a: visited.**
- **a: active** must come after **a: hover**

The **text-decoration** property is mostly used to remove underlines from links, which is **text-decoration: none;**

CSS LIST:

The **list-style-type** property specifies the type of list item marker.

Links selectors are **ol, ul** and to specifies the particular list element use **ol li, ul li.**

Property	Description
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker
list-style-position	Specifies the position of the list-item markers (bullet points)
list-style-type	Specifies the type of list-item marker

CSS TABLES:

Selector used to the select the tables are **id, class or table, th, tr, td.** Normally, **width** and **height** properties are adjusting the table size. The **text-align** property sets the horizontal alignment like **left, center, right** of the content in table of **th, or td.**

As same, the **vertical-align** sets the vertical alignment like **top, bottom or middle** of the content in table of **th or td.**

For zebra-striped tabled, use the **nth-chid()** selector and add **background-color** to all **even or odd** table rows.

Use the**: hover** selector on **tr** to highlight table rows on mouse over.

To control the space between the border and the content in a table, use the **padding** property on **td** and **th** elements.

A responsive table will display a horizontal scroll bar if the screen is too small to display the full content. **Overflow-x:auto** property use around the **table** element to make it responsive.

Property	Description
border	Sets all the border properties in one declaration
border-collapse	Specifies whether or not table borders should be collapsed
border-spacing	Specifies the distance between the borders of adjacent cells
Border-bottom	Specifies the line bottom of each content
caption-side	Specifies the placement of a table caption
empty-cells	Specifies whether or not to display borders and background on empty cells in a table
table-layout	Sets the layout algorithm to be used for a table

CSS DISPLAY/VISIBILITY:

1. **display: none;**
2. **display: block;**
3. **display: inline;**
4. **visibility: hidden;**

Property	Description
display	Specifies how an element should be displayed
visibility	Specifies whether or not an element should be visible

CSS LAYOUT:

Using **width, max-width** and **margin: auto**; a block level element always takes up the full width available stretches out to the left and right as far as it can.

The **position** property specifies the type of positioning method used for an element like **static, relative, fixed, absolute or sicky.**

Property	Description
top	Sets the top margin edge for a positioned box
bottom	Sets the bottom margin edge for a positioned box
left	Sets the left margin edge for a positioned box
right	Sets the right margin edge for a positioned box
clip	Clips an absolutely positioned element

POSITION: VALUES

Value	Description
Static	Is not positioned in any special way, positioned according to the flow of the page. Not affect by the top, right, bottom, left properties.
Relative	Is positioned to its normal position. Setting the top, right, bottom, left properties of relatively-positioned element will cause it to be adjust away from its normal position.
Fixed	Is positioned to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to positioned the element.
Absolute	Is positioned relative to the nearest positioned ancestor instead of positioned to the viewport, like fixed. Absolute positioned elements are removed from the normal flow, and can overlap elements.
sticky	Is positioned based on the user scroll position. A sticky element toggles between relative and fixe, depending upon the scroll position.

OVERFLOW:

The **overflow** property has the following values:

Overflow-x and **overflow-y**

- **visible**
- **hidden**
- **scroll**
- **auto**

Property	Description
overflow	Specifies what happens if content overflows an element's box
overflow-wrap	Specifies whether or not the browser can break lines with long words, if they overflow its container
overflow-x	Specifies what to do with the left/right edges of the content if it overflows the element's content area
overflow-y	Specifies what to do with the top/bottom edges of the content if it overflows the element's content area

CSS FLOAT:

Property	**Description**
box-sizing	Defines how the width and height of an element are calculated: should they include padding and borders, or not
clear	Specifies what should happen with the element that is next to a floating element
float	Specifies whether an element should float to the left, right, or not at all
overflow	Specifies what happens if content overflows an element's box

overflow-x	Specifies what to do with the left/right edges of the content if it overflows the element's content area
overflow-y	Specifies what to do with the top/bottom edges of the content if it overflows the element's content area

BOX SIZING:

You can easily create three floating boxes side by side. However, when you add something that enlarges width of each box (e.g. padding or borders), the box will break. The **box-sizing** property allows us to padding and border in the box's total width and height, making sure that the padding says inside of the box and that it does not break.

ALL PSEUDO CLASSES:

Selector	**Example**	**Example description**
:active	**a:active**	Selects the active link
:checked	**input:checked**	Selects every checked <input> element
:disabled	**input:disabled**	Selects every disabled <input> element
:empty	**p:empty**	Selects every <p> element that has no children
:enabled	**input:enabled**	Selects every enabled <input> element

:first-child	**p:first-child**	Selects every <p> elements that is the first child of its parent
:first-of-type	**p:first-of-type**	Selects every <p> element that is the first <p> element of its parent
:focus	**input:focus**	Selects the <input> element that has focus
:hover	**a:hover**	Selects links on mouse over
:in-range	**input:in-range**	Selects <input> elements with a value within a specified range
:invalid	**input:invalid**	Selects all <input> elements with an invalid value
:lang(*language*)	**p:lang(it)**	Selects every <p> element with a lang attribute value starting with "it"
:last-child	**p:last-child**	Selects every <p> elements that is the last child of its parent
:last-of-type	**p:last-of-type**	Selects every <p> element that is the last <p> element of its parent
:link	**a:link**	Selects all unvisited links

:not(selector)	**:not(p)**	Selects every element that is not a <p> element
:nth-child(n)	**p:nth-child(2)**	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	**p:nth-last-child(2)**	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	**p:nth-last-of-type(2)**	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	**p:nth-of-type(2)**	Selects every <p> element that is the second <p> element of its parent
:only-of-type	**p:only-of-type**	Selects every <p> element that is the only <p> element of its parent
:only-child	**p:only-child**	Selects every <p> element that is the only child of its parent
:optional	**input:optional**	Selects <input> elements with no "required" attribute
:out-of-range	**input:out-of-range**	Selects <input> elements with a value outside a specified range

:read-only	**input:read-only**	Selects <input> elements with a "readonly" attribute specified
:read-write	**input:read-write**	Selects <input> elements with no "readonly" attribute
:required	**input:required**	Selects <input> elements with a "required" attribute specified
:root	**root**	Selects the document's root element
:target	**#news:target**	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	**input:valid**	Selects all <input> elements with a valid value
:visited	**a:visited**	Selects all visited links

CSS PSEUDO ELEMENT:

Selector	Example	Example description
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
::first-letter	p::first-letter	Selects the first letter of each <p> element
::first-line	p::first-line	Selects the first line of each <p> element
::marker	::marker	Selects the markers of list items
::selection	p::selection	Selects the portion of an element that is selected by a user