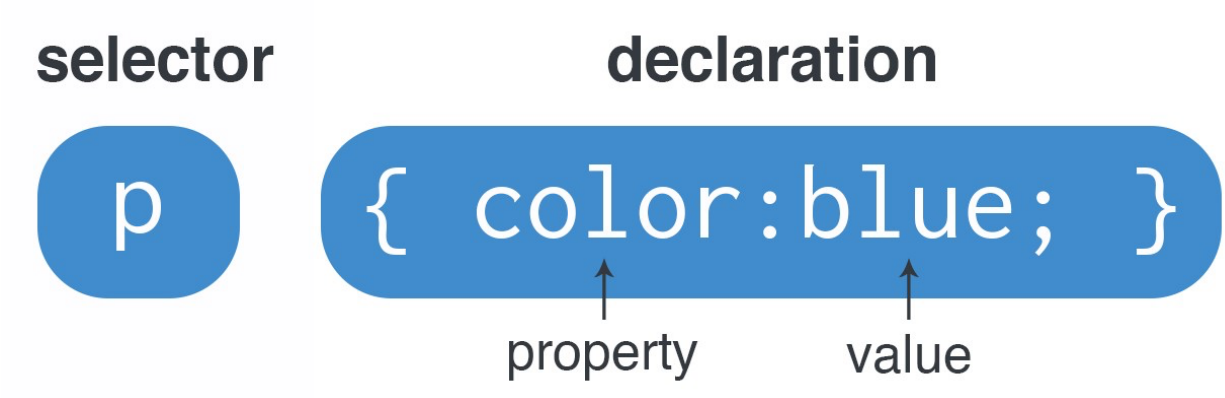
LAB 03 – BASIC CSS

I/. CSS Overview *1/.*

What is CSS?

CSS stands for Cascading Style Sheets. It is the language we use to style a Web page and it describes how HTML elements are to be displayed on screen, paper, or in other media.

2/. What is the CSS syntax like?



There are two parts:

- Selector: points to the HTML element you want to style.
- Declaration: contains one or more declarations separated by semicolons. Each declaration includes:

+ A CSS property name

+ A value

They are separated by a colon (:)

- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

3/. What is CSS Selector? How many categories can CSS selectors be divided into?

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

a/. Simple selectors

(i) Base on name Example:

```
h1 {
        text-align: center;
        color: blue;
}
```

It means that all <h1> elements on the page will be center-aligned, with a blue text color. (ii) Base on id

```
#id01 {
        text-align: left;
        color: green;
}
```

It means that element with id (id01) on the page will be left-aligned, with a green text color.

Note: The id of an element is unique within a page.

(iii) Base on class

```
.class01 {          text-align: right;
        color: yellow;
}
```

It means that all HTML elements with class="class01" will be yellow and rightaligned.

You can also specify that only specific HTML elements should be affected by a class.

```
h2.class01 {        text-align: right;
        color: yellow;
}
```

It means that only <p> elements with class="class01" will be yellow and rightaligned.

HTML elements can also refer to more than one class.

```
<h2 class="class01 class02">Apply 2 classes for h2 tag</h2>
```

It means that the <h2> element will be styled according to class="class01" and to class="class02".

All CSS Simple Selectors:

element	Style for element
#id	Style for id of element
.class	Style for all elements with class
element.class	Style for only element with class
*	Style for all elements
element, element,…	Style for element 1, element 2,…

b/. Combinator selectors

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator. There are four different combinators in CSS:

(i) Descendant selector (space)

The descendant selector matches all elements that are descendants of a specified element.

```
div h2 { color: red;
}
```

It means that selects all <h2> elements inside elements and apply them with red color.

(ii) Child selector (>)

The child selector selects all elements that are the children of a specified element.

```
div > h2 { color: yellow;
}
```

It means that selects all <h2> elements that are children of a element and apply them with yellow color.

(iii) Adjacent sibling selector (+)

The adjacent sibling selector is used to select an element that is directly after another specific element. Sibling elements must have the same parent element, and "adjacent" means "immediately following".

```
div + h2 { color: green;
```

```
}
```

It means that selects the first <h2> element that are placed immediately after elements and apply it with green color.

(iv) General sibling selector (~)

The general sibling selector selects all elements that are next siblings of a specified element.

```
div ~ h2 { color: blue;
}
```

It means that selects all <p> elements that are next siblings of elements and apply them with blue color. All CSS Combinator Selectors:

element element	Style for all descendants
element>element	Style for all children
element+element	Style for adjacent sibling
element~element	Style for all siblings

c/. Pseudo-class selectors d/.

Pseudo-elements selectors

(Pseudo-class and Pseudo-elements will be detailed in the following section) e/.

Attribute selectors

It is possible to style HTML elements that have specific attributes or attribute values.

All CSS Attribute Selectors

[attribute]	Selects all elements with an attribute name
[attribute=value]	Selects all elements with an attribute value
[attribute~=value]	Selects all elements with an attribute containing value
[attribute|=value]	Selects all elements with an attribute starting with value
[attribute^=value]	Selects every element with an attribute beginning with value
[attribute$=value]	Selects every element with an attribute ending with value
[attribute*=value]	Selects every element with an attribute containing with substring value

4/. How to apply CSS to HTML elements?

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

There are three ways of inserting a style sheet: a/.

External CSS

External styles are defined within the <link> element, inside the <head> section of an HTML page.

```
.....
<head>
<link rel="stylesheet" href="style-demo.css">
</head>
.....
```

An external style sheet can be written in any text editor, and must be saved with a .css extension.

The external .css file should not contain any HTML tags.

b/. Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.

```
.....
<head> <style>
p { color: blue;
}

h2 {
        color: red;
        background-color: yellow;
}
</style> </head>
.....
```

Internal styles are defined within the <style> element, inside the <head> section of an HTML page. c/. Inline CSS

An inline style may be used to apply a unique style for a single element.

```
.....
<h1 style="color:red;text-align:center;">This is a heading</h1> <p
style="color:green;">This is a paragraph.</p>
.....
```

Inline styles are defined within the "style" attribute of the relevant element.

Note:

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

- Inline style (inside an HTML element)
- External and internal style sheets (in the head section)
- Browser default

So, an ***inline style has the highest priority***, and will override external and internal styles and browser defaults.

II/. Basic CSS

1/. CSS Color

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

(i) Color name

In CSS, a color can be specified by using a predefined color name.

CSS/HTML support 140 standard color names.

Following link: https://www.w3schools.com/colors/colors_names.asp

(ii) RGB, RGBA

In CSS, a color can be specified as an RGB value, using this formula:

```
rgb (red, green, blue)
```

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color. An RGBA color value is specified with:

```
rgb (red, green, blue, alpha)
```

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all).

You can mix RGB color via link:

https://www.w3schools.com/css/css_colors_rgb.asp

(iii) HEX

A hexadecimal color is specified with: #RRGGBB

RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color. (between 00 and ff) You can mix HEX color via link:

https://www.w3schools.com/css/css_colors_hex.asp

(iv) HSL, HSLA

HSL stands for hue, saturation, and lightness.

```
hsl (hue, saturation, lightness)
```

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value. 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage. 0% is black, 50% is neither light or dark, 100% is white.

You can mix HSL color via link:

https://www.w3schools.com/css/css_colors_hsl.asp

HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.

hsl (hue, saturation, lightness, alpha)

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all).

You can apply color for text, background, border.

Example:

I want to design a paragraph like this. How can you do that?

If you want to contact me.
You can send me an email via: hr01@gmail.com

You can try:

```
    <style>
        p {
            color: red;
        }
        a {
            background-color: yellowgreen;
        }
    </style>
</head>
<body>
    <p>
        <em>If you want to contact me. <br>
        You can send me an email via: </em><a href="mailto:hr01@gmail.com">hr01@gmail.com</a>
    </p>
</body>
```

2/. CSS Background

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

(i) background-color

```
background-color: <value>
value can be color name, RGB color, HEX color,…
```

You can set the background color for any HTML elements.

(ii) background-image

```
background-image: url(path-to-image)
```

The background image can also be set for specific elements.

(iii) background-repeat

The background-image property repeats an image both horizontally and vertically.

```
background-repeat: <value>
value can be repeat-x, repeat-y, no-repeat
```

3/. CSS Height and Width

The CSS 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.

CSS Units:

- Relative Units

Relative units are units that are calculated relative to the parent element or other elements. The unit is relatively portable and works better for different devices, screens of different sizes and resolutions.

(i) % (percentages):

Relative unit for the size of the child element relative to the parent element (parent).

Example: If parent element has width:500px attribute, child element has width:50% attribute, then child element has width:250px attribute.

(ii) em:

A relative unit based on the parent element's font size property.

Example: If parent element has font-size:16px property, child element has font size:2em attribute, then child element has font size:32px property.

(iii) rem:

A relative unit based on the font-size property of the html element (root element).

Example:

If the html element has the font size:12px attribute, the parent element has the font size:16px attribute, and the child element has the font size:2rem attribute, it means the child element has the font size:24px attribute; That is, the rem unit does not depend on the parent element.

- Absolute Units

Absolute units will have fixed values that do not change. Units are absolutely not recommended for monitors, as there are many different screen sizes available. It should only be used for cases where the size of the interface is known for sure.

Some absolute units are used in CSS: px, pt, cm, mm, in Example:

I want to design like this. How can you do that?

You can try:

```
    <style>
        p {
            color: chartreuse;
            text-align: center;
            height: 340px;
            width: 525px;
            background-image: url('./images/background01.jpg');
        }
    </style>
</head>
<body>
    <p>
        <br>
        <strong>You will succeed if you always try every day!</strong>
    </p>
</body>
```

4/. CSS Border, Margin, Padding a/.

CSS Border

The CSS border properties allow you to specify the style, width, and color of an element's border.

(i) 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 (for the top border, right border, bottom border, and the left border).

(ii) CSS Border Color

The border-color property is used to set the color of the four borders.

The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).

(iii) CSS Border Sides

It is possible to specify a different border for each side.

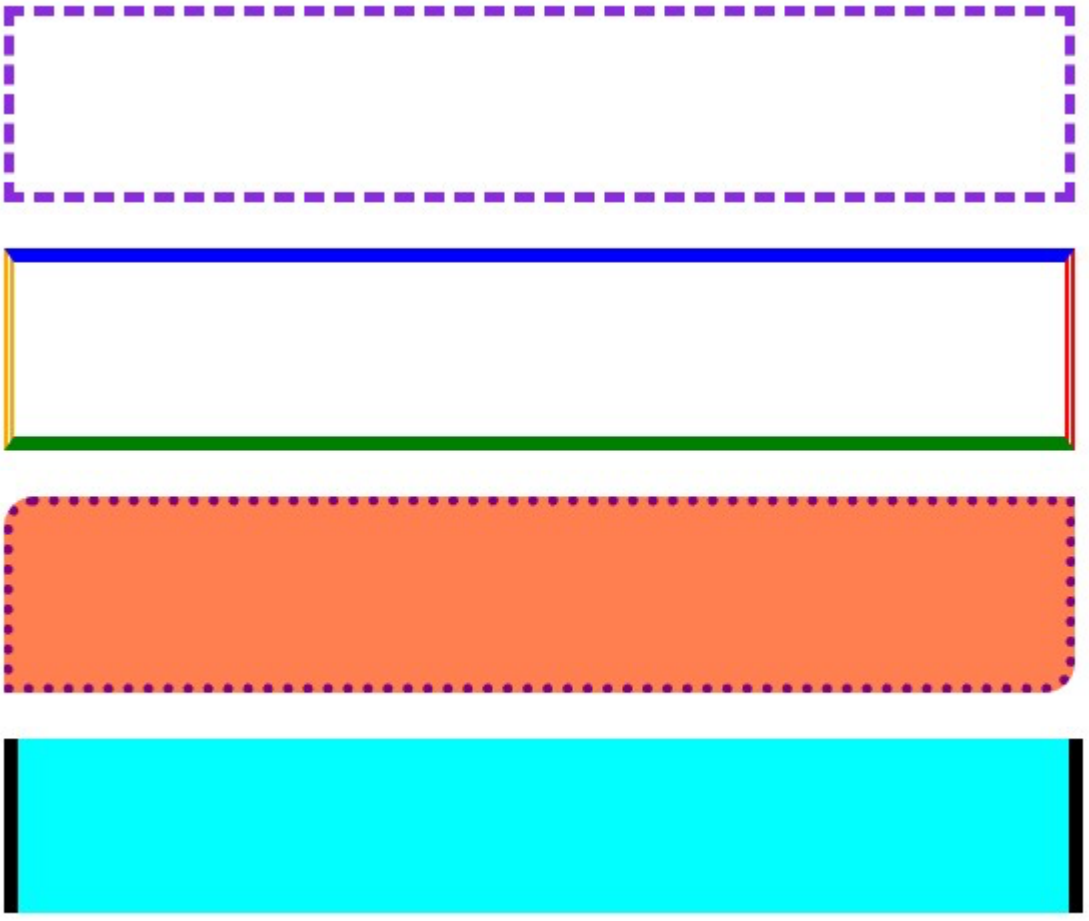
In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left). They are: border-top-style, border-right-style, border-bottomstyle, border-left-style.

(iv) CSS Rounded Borders

The border-radius property is used to add rounded borders to an element.

Example:

I want to design like this. How can you do that?



You can try:

```
<style>
    #div01 {
        height: 70px;
        width: 420px;
        border: 4px dashed;
        border-color: blueviolet;
    }
    #div02 {
        height: 70px;
        width: 420px;
        border-width: 6px 4px 6px 4px;
        border-style: solid double;
        border-color: blue red green orange;
    }
    #div03 {
        height: 70px;
        width: 420px;
        border: 4px dotted;
        border-radius: 12px 0px;
        border-color: purple;
        background-color: coral;
    }
    #div04 {
        height: 70px;
        width: 420px;
        border-width: 6px;
        background-color: aqua;
        border-style: none solid none solid;
    }
</style>
```

b/.

CSS Margin

Margins 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).

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! c/.

CSS Padding

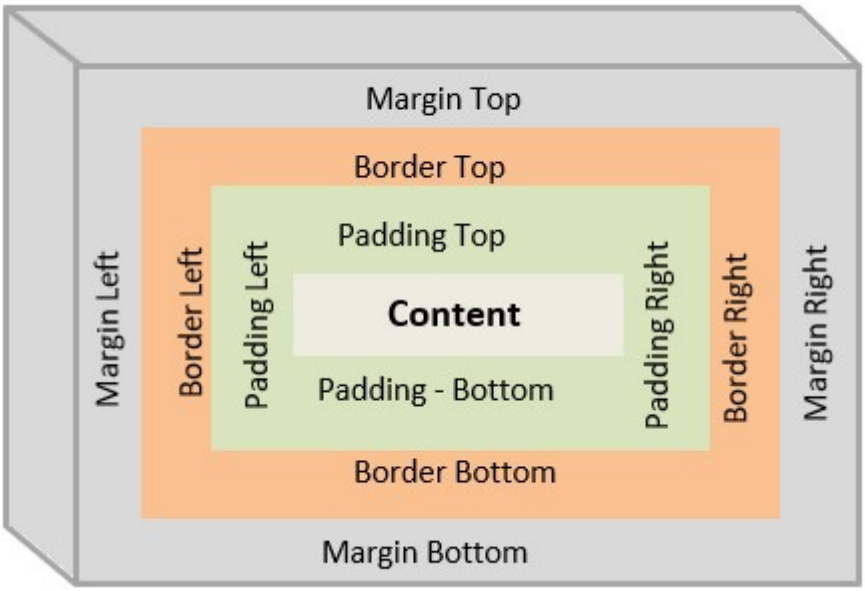
Padding is used to create space around an element's content, inside of any defined borders.

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).

5/. CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

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

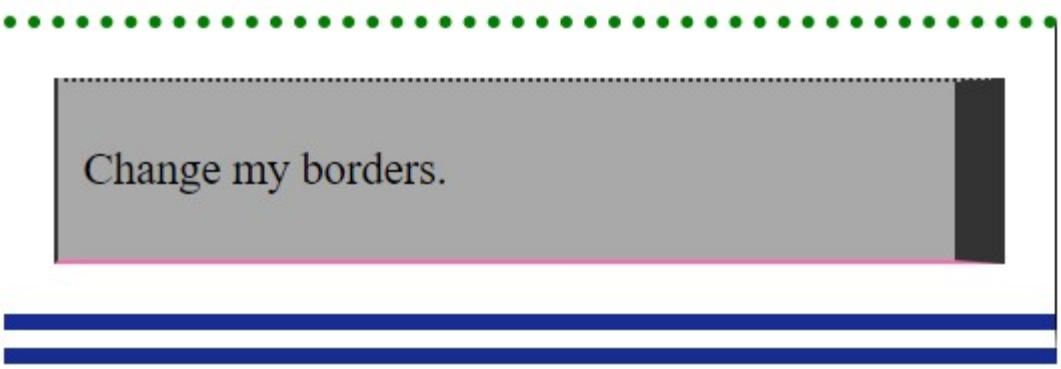


Total element width = width + left padding + right padding + left border + right border + left margin + right margin

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

Example:

I want to design like this. How can you do that?



You can try:

```
    <style>
        .container {
            border-top: 5px dotted green;
            border-right: 1px solid black;
            border-bottom: 20px double rgb(23,45,145);
            width: 420px;
        }

        .box {
            border: 2px solid #333333;
            border-top-style: dotted;
            border-right-width: 20px;
            border-bottom-color: hotpink;
            padding-top: 25px;
            padding-bottom: 25px;
            padding-left: 10px;
            margin: 20px;
            font-size: large;
            background-color: darkgrey;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="box">Change my borders.</div>
    </div>
```

6/. CSS Display a/.

CSS Display

The display property is the most important CSS property for controlling layout.

The display property specifies if/how an element is displayed. It consists of 3 basic types that we need to pay attention to: none, block, inline, inline-block.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards.

Note:

Setting the display property of an element only changes how the element is displayed, NOT what kind of element it is. So, an inline element with display: block; is not allowed to have other block elements inside it.

b/. CSS Float

The CSS float property specifies how an element should float.

The CSS clear property specifies what elements can float beside the cleared element and on which side.

The float property is used for positioning and formatting content.

The float property can have one of the following values:

- left - The element floats to the left of its container
- right - The element floats to the right of its container
- none - The element does not float (will be displayed just where it occurs in the text). This is default.
- inherit - The element inherits the float value of its parent Example 01:

I have a simple HTML page like this:

After that, I want to style this page like this:

How can you do that?

You can try:

```
<body>
    <header>
        <h1>WELCOME TO MY WEBSITE</h1>
        <ul style="list-style-type: none;">
            <li><a href="#" target="_blank">Home</a></li>
            <li><a href="#" target="_blank">Schedule</a></li>
            <li><a href="#" target="_blank">Contact</a></li>
        </ul>
    </header>
</body>
```

```
<style>
    * {
        box-sizing: border-box;
    }
    header {
        border: 2px solid blue;
        border-radius: 20px;
        background-color: blanchedalmond;
    }
    h1 {
        text-align: center;
        font-weight: bold;
        color: red;
    }
    ul {
        text-align: center;
    }
    li {
        display: inline-block;
        border: 2px solid orangered;
        width: 90px;
        height: 45px;
        text-align: center;
        padding: 10px;
        background-color: cyan;
        border-radius: 10px 0px;
    }
    a {
        text-decoration: none;
        font-size: large;
        font-weight: bold;
    }
</style>
```

Example 02:

I have a design for images like this. How can you do that?

Rose *1.5$*

Tulip *2.0$*

Daisy *1.0$*

You can try:

```html
<div class="ProductContainer">
    <div class="Product">
        <img src="./images/rose.jpg" />
        <div class="ProductDetails">
            <span class="ProductName">Rose</span>
            <span class="ProductPrice">1.5$</span>
        </div>
    </div>
    <div class="Product">
        <img src="./images/tulip.jpg" />
        <div class="ProductDetails">
            <span class="ProductName">Tulip</span>
            <span class="ProductPrice">2.0$</span>
        </div>
    </div>
    <div class="Product">
        <img src="./images/daisy.jpg" />
        <div class="ProductDetails">
            <span class="ProductName">Daisy</span>
            <span class="ProductPrice">1.0$</span>
        </div>
    </div>
</div>
```

```html
<style>
    .Product img {
        width: 150px;
        height: 150px;
        border-radius: 20px;
    }
    .ProductName {
        font-weight: bold;
    }
    .ProductPrice {
        font-style: italic;
    }
    .Product {
        text-align: center;
        width: 150px;
        margin-top: 0px;
        padding: 10px;
        display: block;
    }
</style>
```