**CSS NOTES**

**CSS BASIC SYNTAX:**

Selector

{

Property: value;

Property: value;

}

**CSS COMMENTS:**

/\*

Write something

\*/

**CSS SELECTORS:**

|  |  |
| --- | --- |
| #id | Selects element with Id specified. (Must be unique for every element) |
| .class | Selects elements with class specified. (Can be given to multiple elements) |
| Element.class | Selects element specified with class name specified |
| \* | Selects all elements |
| Element | Selects single element specified |
| Element, element | Selects all specified elements |

**CSS UNITS:**

There are 15 units, few are discussed

|  |  |
| --- | --- |
| px | Pixels |
| em | Relative to original font-size |
| rem | Relative to root font-size |
| % | Relative to parent element |

**KEY POINTS:**

* The current version of CSS is CSS3 which was released in 1999
* <HTML> tag is the root of HTML document.
* All other tags placed in HTML document are descendants of root element.
* If CSS is applied to a tag, then the CSS will also be applied to descendant tags
* The inline CSS always overrides external and internal CSS
* The by-default **font-size** of HTML document is **16px;**
* By-default **color** in CSS is **black.**
* CSS can also be applied for root element i.e <HTML>
* CSS applied to root element will be applied for all elements because all elements are its descendants

**PX, EM, REM AND % EXPLANATION:**

The px unit means pixels.

The em unit is relative to current font-size of same element.

**For example:**

If there is <p> tag and its font-size is set of 2em

Then its font-size will be 32px as by-default size is 16px;

1em=16px,

0.5em = 8px;

Actually, the number with em unit is being multiplied with the current size.

By current size it means if **font-size** is set for a parent or grand-parent element then the same **font-size** is applied to children or grandchildren elements.

See the following example:

<body style="font-size: 20px; background-color: blue;">

    <p style="font-size: 2em;">

        Hello World

        <p style="font-size: 2em;">

            Hello World

        </p>

    </p>

</body>

The **font-size is set to 20px** for the body hence same font-size will also be applied for both p tags.

If font-size for the p tags is set to 2em then their **font-size will be 40px;**

Here 1em = 20px

0.5em = 10px;

If **font-size is set to 2rem** for both p tags then their **font-size will be 32px;**

As no CSS is applied to root element i.e HTML element and its default font-size is 16px.

If **font-size is set to 50%** for both p tags then their **font-size will be 10px.**

The percent unit is relative to parent element.

**Body font-size set to 20px;**

If **font-size is set to 50% for p tag** then it will get the **font-size for body** as body is the parent of **p tag**.

If **font-size is set to 50% for p tag nested inside the p tag** then it will get the **font-size of the p tag in the body** as it is the parent element. The **font-size of parent was 20px** hence 10px will be font-size of child element.

See the following example:

<!DOCTYPE html>

<html lang="en" style="font-size: 20px">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body style="font-size: 20px; background-color: blue;">

    <p style="font-size: 50px;">

        Hello World

        <p style="font-size: 2rem;">

            Hello World

        </p>

    </p>

</body>

</html>

In this example the font-size for the root element is set to 20px;

Hence font-size for all elements will be 20px if font-size property is not set explicitly for elements.

Here font-size for the nested p tag is 2rem which means its font-size will be 40px;

As 2 will be multiplied by the font-size of root element.

**INHERIT, INTIAL, AND UNSET EXPLANATION:**

All three are values in CSS given to different properties.

inherit: Get the property from the parent element.

initial: The default value for the property (the browser default).

unset: Acts as either inherit or initial. It will act as inherit if the parent has a value that matches, or else it will act as initial.

**CSS COLOR SETTING PROPERTIES**

Background-color:

This property specifies the color for background

Usage:

Background-color: color;

Color:

This property specifies the text color.

Usage

Color: color;

**HOW TO DEFINE COLOR?**

Color can be defined by:

1. Name
2. HEX Value
3. RGB Value
4. HSL Value
5. RGBA Value
6. HSLA Value

RGB: Red green blue

HSL: Hue Saturation Lightness

A: Alpha or opacity or transparency

RGB, RGBA, HSL, HSLA are functions they can be used as:

**rgba(225, 255, 133, 0.5);**

RGB are defined by number 0-255

RGB colors can be generated using palette

**hsl(10%, 4%, 6%);**

HSL Values are defined in percentages

Value for alpha is defined as number between 0-1

Where 0 is fully transparent and 1 is fully opaque

**BACKGROUND DESIGNING:**

Background can be set by following properties:

1. Background-color: color;
2. Background-image: url(“image path”);

Sometimes background image is so small that it is repeated on whole page.

This is by default in CSS to disable this we need to set **Background-repeat** property.

**Background-repeat** can be set by following values:

1. No-repeat (image is not repeated)
2. Repeat-y (image is repeated vertically)
3. Repeat-x (image is repeated horizontally)

If background is small then its size can also be altered by using **background-size** property

**Background-size** property can be set to:

1. Width height – Can be set in percentage or pixel
2. Contain – Resize background image to make it fully visible
3. Cover – Resize background image to cover whole container

**BORDER:**

Border can be defined using border property.

There are 10 types of border styles, few are discussed here:

1. Solid 2. Dashed 3. Dotted
2. Double 5. Inset 6. Outset

Border can either be defined for one side by using

1. Border-top
2. Border-right
3. Border-bottom
4. Border-left

Properties

Or for all sides by using border shorthand:

**Border: width style (required) color;**

**Rounded borders:**

Rounded borders can be defined by **border-radius** property.

**Usage:**

Border-radius: length;

**Border Collapse:**

When border of two elements touch each other, they form a single border. This is property of table tag.

It can have 4 values. Two are discussed:

1. Separate (To stop borders forming a single border, this is by default behavior)
2. Collapse (Borders are merged into single one)

**OUTLINE:**

An outline is a line that is drawn around elements, OUTSIDE the borders.

There are 10 types of outline styles, few are discussed here:

1. Solid 2. Dashed 3. Dotted
2. Double 5. Inset 6. Outset

Outline has 4 properties:

1. Width
2. Style
3. Color
4. Offset

Offset property defines how far will be the outline from border.

Width property have following values:

* thin (typically 1px)
* medium (typically 3px)
* thick (typically 5px)
* A specific size (in px, pt, cm, em, etc)

**Outline Shorthand:**

Outline shorthand:

Outline: width style(required) color

**Usage:**

**Outline: 2px solid red;**

**EXAMPLE CODE OF BORDER AND OUTLINE**

<!DOCTYPE html>

<html lang="en" style="font-size: 20px">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        div

        {

            height: 200px;

            width: 200px;

            border: 5px solid red;

            margin: 20px;

            outline: 5px solid blue;

            outline-offset: 2px;

            /\*

            Outline-offset defines how far will be the outline from border

            \*/

        }

    </style>

</head>

<body>

    <div>

    </div>

    <div>

    </div>

</body>

</html>

**MARGIN AND PADDING:**

**Margin** property gives the element space from outside and **padding** property gives space from inside.

Both padding and margin properties can be set separately for sides by using top, right, bottom, left properties.

Both padding and margin can be set by shorthand properties.

**Usage:**

Padding: top right bottom left; (Four lengths)

Padding: top [right and left] bottom; (Three lengths: single length for right and left)

Padding: single length; (Sets for all sides)

Margin: top right bottom left; (Four lengths)

Margin: top [right and left] bottom; (Three lengths: single length for right and left)

Margin: single length; (Sets for all sides)

Padding: 10px 20px 30px 40px;

10px from top

20px from right

30px from bottom

40px from left

Padding: 10px 40px 10px;

10px from top

40px from right and left

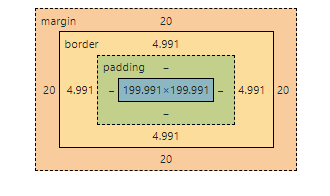
10px from bottom

Padding: 10px;

10px space inside element from all sides

Margin: 10px;

10px space outside element from all sides



* In above picture it shows the margin is outside area of element. Padding is the inside space between element and its content. The blue area shows the content area. The border area depends on the size of border. The greater the size of border the more area it will cover. Although outline property does not cover the area.

**SIZE SETTING PROPERTIES:**

For setting size there are 6 properties available in CSS:

1. Height (Sets fixed height)
2. Width (Sets fixed width)
3. Max-height (Height does not increases above given value)
4. Max-width (Width does not increases above given value)
5. Min-height (Height does not decreases below given value)
6. Min-Width (Width does not decreases above given value)

**Values:**

auto - This is default. The browser calculates the height and width

length - Defines the height/width in px, cm, em etc.

% - Defines the height/width in percent of the containing block

initial - Sets the height/width to its default value

inherit - The height/width will be inherited from its parent value