1. **Z index**

Specifies the stack order of an element.

Syntax:

z-index: auto/number/initial/inherit;

img {

position: absolute;

let:0px;

z-index: -1;

}

1. **Selectors.**

They are used to “find”/”select” HTML elements you want to style.

1. Simple selectors.- select elements based on name, ID, class.

<p> elements will be center aligned with a red text color.

P {

Text-align: center;

Color: red;

}

ID SELECTOR:

Uses ID attribute of an HTML element to select a specific element.

Selects the elements with id=”firstname”

Syntax:

#text {

text-align: center;

color: red;

}

1. Combinatory selectors. select elements based on a specific relationship between them.

Selects only <p> element wih class”intro”

1. Pseudo-class selectors. –select elements based on a certain state
2. Selects all elements with class =”” Selects all<p> elements

Syntax: .center {

Text-align: center;

Color: red;

}

1. Pseudo-element selectors. –select and style a part of an element.

pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s). For example, ::first-line can be used to change the font of the first line of a paragraph.

Pseudo-elements can be used to style an element based on its state. EG:

SYNTAX:

P:: first-line {

Color: blue;

Text-transform: uppercase;

}

Pseudo-elements can be used to style an element based on its state. EG:

selector::pseudo-element {

property: value;

}

You can use only one pseudo-element in a selector. It must appear after the simple selectors in the statement.

Pseudo-elements only use double-colon unlike pseudo-class

1. Attribute selectors. –select elements based on an attribute or attribute value

Universal selectors. (\*)

Slects all elements

Syntax: \*{

text-align: center

}

Element, element div<>p

**3.INLINE, INLINE-BLOCK &BLOCK**

Inline-block- allows to set a width and height on the element. Inline does not.

Inline block- the top and bottom margins/padding are represented. Inline-the top and bottom margins/padding are not represented.

Block adds a line break after elements but inline block does not add a line to-break after elements so, so the elements can sit next to other elements.

**4. PSEUDO-CLASSES.**

Is used to define a special state of an element.

They add style to selectors when a specific condition is met by the selectors.

Can be used to:

* Style an element when the user uses the mouse
* Style visited and unvisited links differently.
* Style an element when it gets focus.

SYNTAX:

Selector: pseudo-class {

Property: value

}

**5. VH & VW IN CSS.**

vh stands for viewport height and vw for viewport width. As you can see, the first unit is based on the viewport height, and 1vh is equivalent to 1% of the viewport height. vw works the same, but for viewport width.

To use vh and vw values, just type “Nvh” or “Nvw” (where “N” represents the percentage of the viewport you'd like to cover) into any width or height field. So to cover 100% of the viewport, you'd set 100% for the width and 100vh for the height. To cover half of the viewport height, you'd set a height of 50vh.

**6. UNITS IN CSS.**

The units in CSS are required to define the measurement such as margin: 20px; in which the px (or pixel) is the CSS unit. They are used to set margin, padding, lengths, and so on. We cannot apply the whitespace between the number and the unit. The unit can be omitted for the value 0.

* ABSOLUTE UNITS:

Absolute units are useful when the responsiveness is not considered in a project. They are less favorable for the responsive sites because they do not scale when the screen changes. Absolute lengths are considered to be the same size always. EG: cm, mm, in, pt, px.

SYNTAX:

<!DOCTYPE html>

<html>

<head>

<style>

body{

text-align: center;

}

</style>

</head>

<body>

<h1> Absolute units </h1>

<p style = "font-size: 20px;" > It has a font-size: 20px; </p>

<p style = "font-size: 1.2cm;" > It has a font-size: 1.2cm; </p>

</body>

</html>

* **RELATIVE UNITS**

Relative units are good to style the responsive site because they scale relative to the window size. They specify the length, which is relative to another length property. EG: VM and VH.

SYNTAX:

<!DOCTYPE html>

<html>

<head>

<style>

body{

text-align: center;

}

p{

line-height: 0.1cm;

color: blue;

}

</style>

</head>

<body>

<h1> Relative units </h1>

<p style = "font-size: 4vw;" > It has a font-size: 4vw; </p>

</body>

</html>

**7. BORDER-BOX & CONTENT-BOX**

Content-box: This is the default value of box-sizing. The dimension of element only includes ‘height’ and ‘width’ and does not include ‘border’ and ‘padding’ given to element. Padding and Border take space outside the element.

Border-box, padding and border are included in the width and height

**8. OPACITY IN CSS3**

 Opacity specifies the transparency of an element.

You can **use an RGB color value with an alpha channel (RGBA)** - which specifies the opacity for a color. An RGBA color value is specified with: rgba (red, green, blue, alpha). The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

The lower the value the more transparent.

 Values for this property range from 0 to 1. If you set the property to 0, the styled element will be completely transparent (ie. invisible)

**9. To center-align a div inside another div**

Use the "inline-block" value of the display property to display the inner <div> as an inline element as well as a block. **Set the text-align property on the outer <div> element to center the inner one**. This property only works on inline elements.

<div class=”parent”>

<div class=”child”>

<p> child div</p>

</div>

</div>

**10. ABSOLUTE AND RELATIVE**

position: relative places an element relative to its current position without changing the layout around it, whereas position: absolute places an element relative to its parent's position and changing the layout around it.

Relative - the element is positioned relative to its normal position. Absolute - the element is positioned absolutely to its first positionedparent..