1. Z- index

Specifies the stack order of an element.

It only works on positioned elements (position:asolute/relative/sticky) and flex items (display:flex elements)

An element with greater stack order is always in the front of an element with a lower stack

Syntax:

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

Auto –sets the stack order equal to its parents.(this is default)

Number - sets the stack order 0fthe elements. Negative numbers are allowed

Initial - Sets this property to its default value.

Inherit - inherits this property from its parents.

img {

position: absolute;

left:0px;

top: 0px;

Z-index: -1;

}

**2. SELECTORS.**

They are patterns used to select the element you want to style.

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

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

Syntax:

#first name {

text-align: center;

color: red;

}

Combinatory selectors. Selects elements based on a specific relationship between them.

Selects only <p> element with class”intro”

.intro p{}

element selectors. –select and style a part of an element.

P{}

Universal selectors. (\*)

Selects all elements

Syntax: \*{

text-align: center

}

Element, element

Div p

Selects all<p> elements inside <div> elements.

::AFTER

Inserts something after the content of an element

P::after

::Before

Inserts something before the content of an element

P::before

**3. VH AND VW**

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.

**4. INLINE, INLINE-BLOCK &BLOCK**

**BLOCK**

A block element always starts on a new line and the browsers automatically adds some space before and after the element (margin)

Commonly used elements are <p. and ,div>

<p>Hello World</p>

**INLINE**

Does not start on a new line.

Only takes up as much width as necessary

<span> Hello world</span>

**INLINE-BLOCK**

inline-block allows to set a width and height on the element.

Also, with display: inline-block, the top and bottom margins/paddings are respected, but with display: inline they are not.

Compared to display: block, the major difference is that display: inline-block does not add a line-break after the element, so the element can sit next to other elements.

One common use for display: inline-block is to display list items horizontally instead of vertical  
  
span.a {  
  display: inline; /\* the default for span \*/  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid blue;  
  background-color: yellow;  
}  
  
span.b {  
  display: inline-block;  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid blue;  
  background-color: yellow;  
}  
  
span.c {  
  display: block;  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid blue;  
  background-color: yellow;

**5. pseudo classes**

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

* Style an element when a user mouses over it
* Style visited and unvisited links differently
* Style an element when it gets focus

selector:pseudo-class {  
  property: value;  
}

**Note:** a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective! a:active MUST come after a:hover in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.

The opacity property specifies the opacity/transparency of an element.

The opacity property can take a value from 0.0 - 1.0. The lower the value, the more transparent:

img {  
  opacity: 0.5;  
}

Transparent Hover Effect

The opacity property is often used together with the :hover selector to change the opacity on mouse-over:

img {  
  opacity: 0.5;  
}  
  
img:hover {  
  opacity: 1.0;  
}

**6. CSS UNIT**

They are required to define the measurement. Px is the css unit.

CSS has several different units for expressing a length.

Many CSS properties take "length" values, such as width, margin, padding, font-size, etc.

**Length** is a number followed by a length unit, such as 10px, 2em, etc.

Types include relative and absolute

h1 {  
  font-size: 60px;  
}  
  
p {  
  font-size: 25px;  
  line-height: 50px;  
}

**7. DIFF BETWEEN BORDER BOX AND 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.

box-sizing: content-box;

**border-box:**In this value, not only width and height properties are included but you will find padding and border inside of the box for example .box {width: 200px; border: 10px solid black;} renders a box that is 200px wide.

**8. SPECIFYING OPACITY**

The opacity property sets the opacity level for an element.

The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent.

div {  
  opacity: 0.5;  
}

To not apply opacity to child elements (like in the example above) use **RGBA** color values instead. The following example sets the opacity for the background color, but not for the text:

div.first {  
  background: rgba(76, 175, 80, 0.1);  
}  
  
div.second {  
  background: rgba(76, 175, 80, 0.3);  
}  
  
div.third {  
  background: rgba(76, 175, 80, 0.6);  
}

**9. CENTER ALIGNING 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.

Parent div display:flex; justify-content: center; align-items: center

**10. ABSOLUTE AND RELATIVE**

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

The element is removed from the normal document flow and no more space is created for the element in the page layout

Relative length units specify a length relative to another length property. the element is positioned absolutely relative to its normal position