* CSS is composed of rules
* A rule is composed of a “selector” and a declaration block
* The rules cascade

1. Rules created by the browser
2. Rules created by the user
3. Rules created by the document author

* 3 ways to add CSS: Inline, Embedded and External
* CSS measure units are - %percentage units, pixels(1 px is 1 dot in computer), em units, rem units.

**ID and Class attributes:**

* ID must be unique in html
* Class represents a classifier like to represent a module. Its not required to be unique.
* Class name can be reused, can have >1 class

*<table class=”table-stripped table-bordered”>*

*<table class=”table-stripped”>*

**Element Attributes:**

* Lets us consider <header> and <footer> we do not need id or class to style them
* We can also select attribute and style them
* Example: `#’ for id and ‘.’ for classes and footer is element

#homepage-header {

background: skyblue;

}

.grayed {

background: gray;

}

footer {

background: yellow;

}

Input[type=password] {

Background: red;

}

**Relationship Selectors:**

* Descendant Selectors: select elements that are contained within other elements.

section h1 {

color: red;

}

* Child Selectors: used to select an immediate child of an element.

section > h1 {

color: red;

}

* Next Sibling Selectors: used to select an element placed right after another element.

p + a {

color: yellow;

}

**Pseudo Class Selectors:**

a:hover {

color: green;

}

**Tree Proximity**

* Last Rule wins.

HTML:

<section>

<div>

<p> hgkdlkjdpijohi</p>

</div>

</section>

CSS:

div p {

color:red;

}

section p {

color:green;

}

* Here both are targeting to same element, but our CSS make the text “GREEN”
* WIN who comes the last.

**Important rule:**

a {

color: red !important;

}

**CSS Positioning:** How elements are positioned within the webpage.

position takes 4 values: static(default), relative, absolute, fixed.

static: Static elements are not positioned in any way. They are laid down where they occur within the flow of the page. They are unaffected by the css properties – top, left, bottom, right.

relative: The elements can be positioned relatively to the space reserved for them in the flow using css properties - top, left, bottom, right.

absolute: no positioned ancestor. The element is taken out of the flow (No space is allocated for it on its parent).

fixed: The element is fixed. Its position doesn’t change even when scrolling the page. The element overlaps with the content of the page.

**Float and clear**:

Block elements are normally stacked vertically in the page. To stack them horizontally, you can float them, either left or right. Elements coming after will be affected by the floating…

To avoid floating the next elements, apply clear. EX: h1{ clear: both } or clear them by overflow hack.

[www.bestpsdfreebies.com](http://www.bestpsdfreebies.com) //can download themes for our projects.