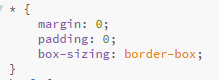
Tips:

* If you mention margin-left and margin-right “auto” content will align in center.
* With the “float” element can be pushed to the left or right, allowing other elements to wrap around it.
* “clear” property is counter part of float.
* To make anything centre make margin-left and margin-right to auto.
* Whenever you start creating website make sure padding and margin to be set to ‘0’.

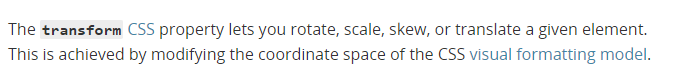


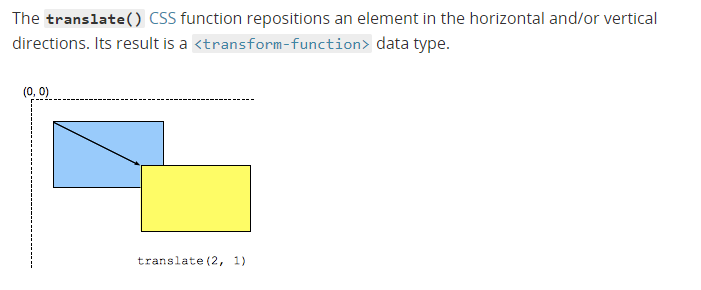
* <header>,<nav>,<footer> these are contains which gives meaning based on their names.
* For setting 100 per view port height and setting background image we use like below



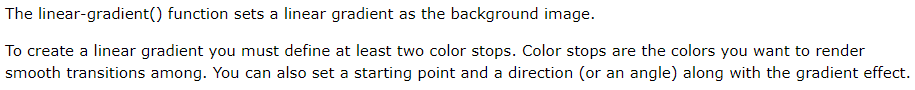
Vh: view port height

* The **background-size** property specifies the size of the background images.
* **Cover**: Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges
* **Contain:** Resize the background image to make sure the image is fully visible
* The background-position property sets the starting position of a background image.





**transform: translate(-50%,-50%):** reposition of the element in a way that here first 50 per represents left and second represents top as they mentioned in negative values.



***background-image:linear-gradient(rgba(0, 0, 0, 0.7),rgba(0, 0, 0, 0.7)),url(img/hero.jpg);***

this is with linear gradient



Without linear gradient:



* When you start styling by assign some values to html like below



* If you want use font-size for any element just mention it in relative way like below.



**Inline-block elements:** It is just like a inline element, but that can have a heading and margin, difference is doesn’t force a line break, but you can attribute it some padding and margin unlike normal inline element.

Ex: Normally <a> is an inline element we want it as a inline -block then we can change that by using one css property “**display:inline-block”**

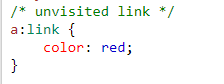
* **Text-decoration:none** removes underline for a elements
* **Border-radius:** makes button or any element to circular or symmetrical shape based on the value.

**Buttons/Links:**

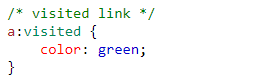
Button or links in general in css can have different states,

* Link: Is just how the elements looks in the normal state

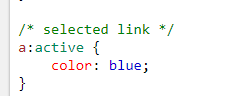
Ex: we have specify pseudo class **btn:link**



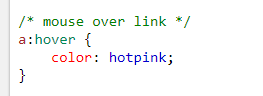
* Visited: how link/button should after visiting the site.

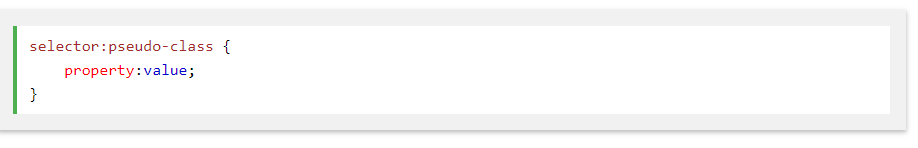


* Active: selects the active link

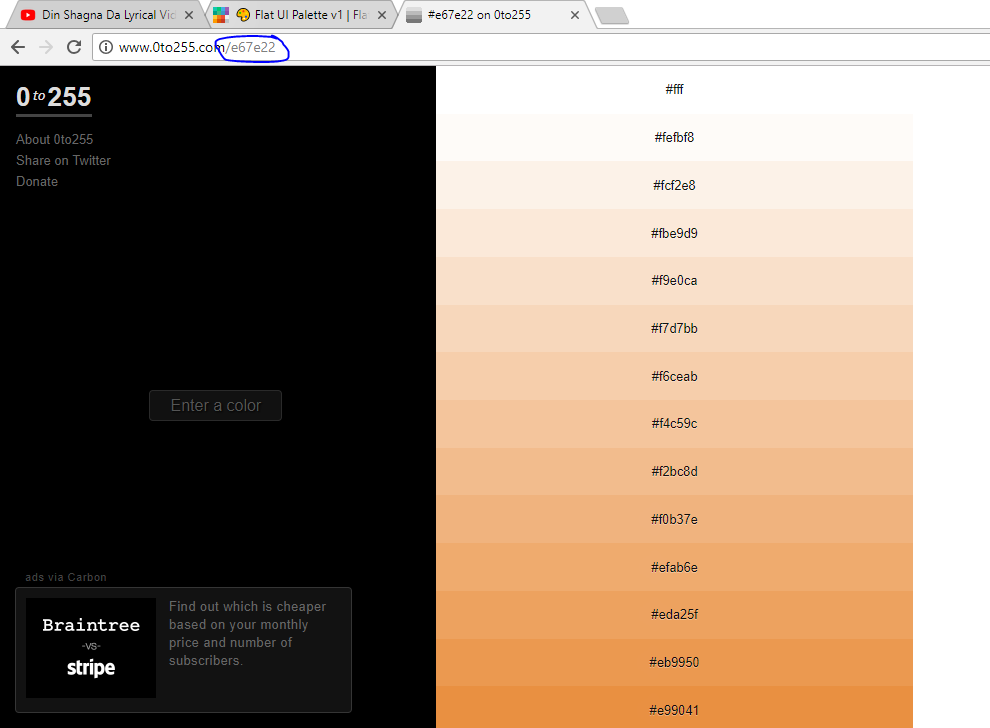


* Hover: We can specify what happens we put mouse over it





[*http://www.0to255.com/*](http://www.0to255.com/) *: 0to255 is a color tool for finding lighter and darker colors based on any color.*



**Transitions:**

CSS transitions allows you to change property values smoothly (from one value to another), over a given duration.



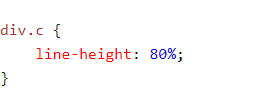
**List-style:none** removes the style for the list elements

::after - The ::after selector inserts something after the content of each selected element(s).

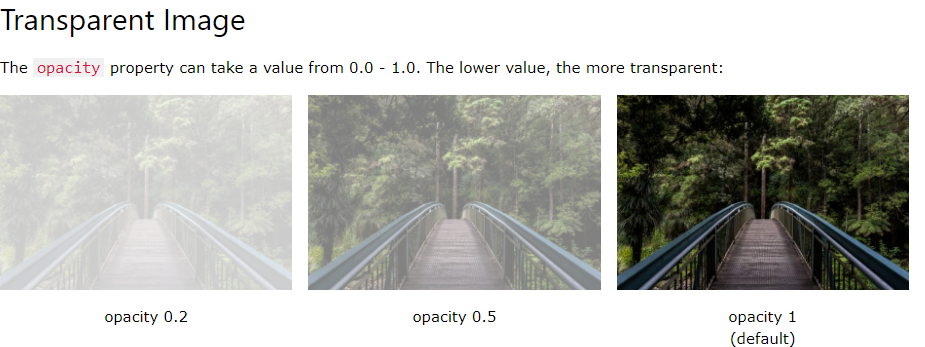


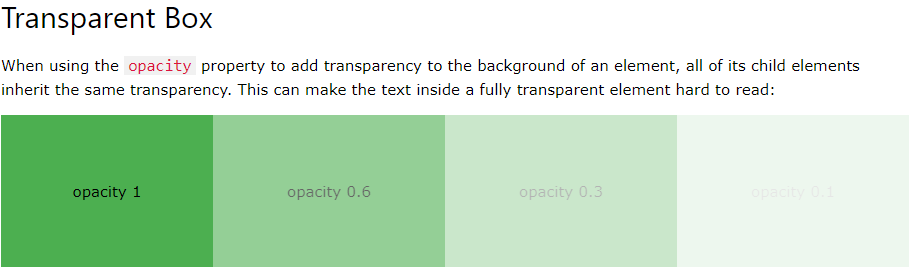


The line-height property specifies the height of a line.



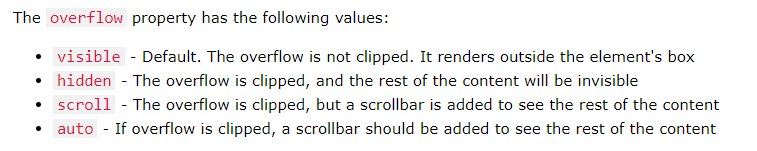
Opacity: The opacity property specifies the opacity/transparency of an element.





Overflow:

The CSS overflow property controls what happens to content that is too big to fit into an area.

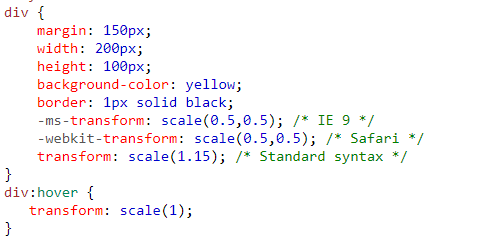


Display:

The display property specifies the display behavior (the type of rendering box) of an element.

Transform: Scale (1.15):

The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).



* Text-align doesn’t only works for text it works for inline , inline block elements and images.
* ***When we define some floats you then must clear those floats***

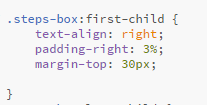


From above screen shot what we can say is whenever we use **clearfix** class, In after pseudo class

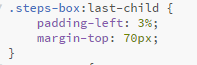
Always make an element as **block** element, as you all know we must to mention content and this **content** can be empty or anything, but we must to hide using **visibility** those content.

Important thing is using “clear:both” everywhere

**The :first-child** selector is used to select the specified selector, only if it is the first child of its parent.



The **:last-child** selector matches every element that is the last child of its parent.



The **:last-of-type** selector matches every element that is the last child, of a **particular type**, of its parent.

