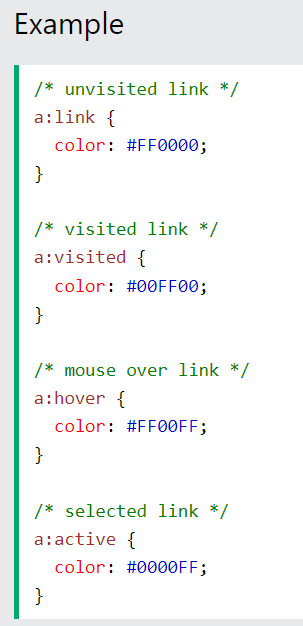
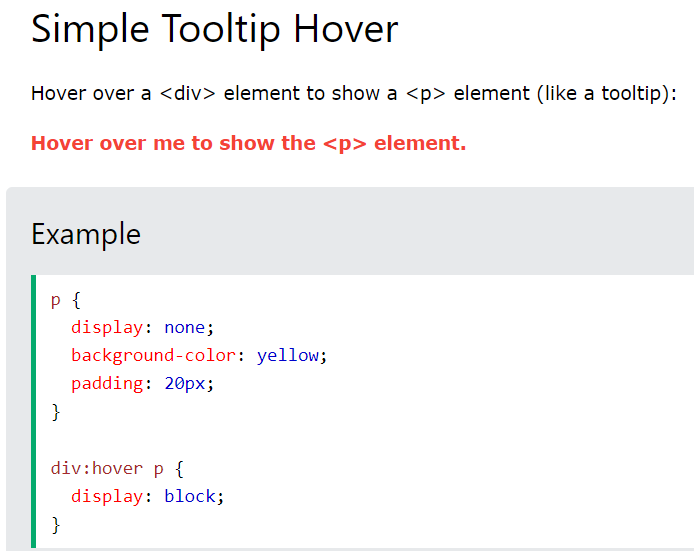
CSS Selectors:

* Simple selectors (selecting elements by name, id, class)
* Combination Selectors : There are four different combinators in CSS:
  + descendant selector (space)
    - div p {  
        background-color: yellow;  
      }
  + child selector (>)
    - div > p {  
        background-color: yellow;  
      }
  + adjacent sibling selector (+) (only one sibling)
    - div + p {  
        background-color: yellow;  
      }
  + general sibling selector (~) (all siblings)
    - div ~ p {  
        background-color: yellow;  
      }
* Pseudo class: A Pseudo class defines special statement of an element.

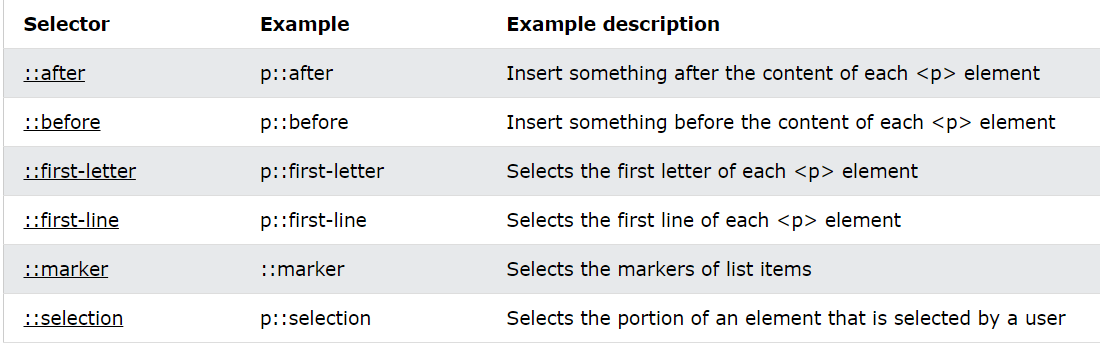








* Pseudo Element Selector: It is used to style a specified parts of an element



**CSS Position:**

CSS Position can be of 5 types. Elements are positioned using top, bottom, right, left properties.

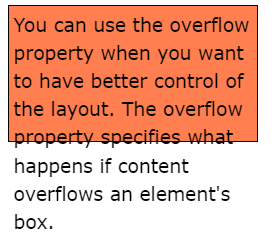
* Static 🡪 All HTML elements are static by default i.e. specifying *position:static* means it is always positioned normal to the page. Properties **Top, Bottom, Left And Right** don’t have any effect on static positioned elements
* Relative 🡪 Elements with *position:relative* are positioned relative to the normal position of the element. Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position
* Fixed 🡪 These elements will be positioned relative to the viewport so hence they will fixed even if the page is scrolled. We can change the position of this element by using top, bottom , left and right properties.
* Absolute 🡪 These elements will be positioned relative to their nearest positioned ancestor , if it has no positioned ancestors it uses document body and moves along with the page scrolling.
* Sticky 🡪 This element toggles between ‘relative’ and ‘fixed’ positions depending on scroll position. It’s position is relative until a given “offset” position is met in viewport then it “sticks” in place.

**CSS Z-Index:**

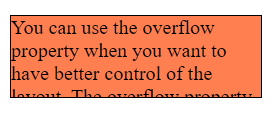
* It only works on positioned elements
* It defines the stack order for positioned elements

**CSS Overflow:**

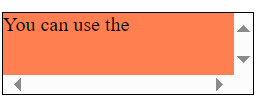
* It specifies whether to clip the content or add scrollbars when the content of an element is too big to fit in the specified area.
* It only works on block level elements with specified height
* It has these values:
  + Visible 🡪 Default one. The content will overflow and will fall outside the element.



* + Hidden 🡪 The content will cropped if it is going outside the size of the element

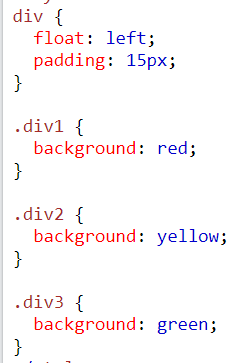


* + Scroll 🡪 A scroll bar is always added even if the content is not going outside the size of the element



* + Auto 🡪 This is same like ‘scroll’ but it will have scroll bars only when it is necessary.

**CSS Float:**

* It is used to position an element
* It has 4 properties : left, right, none & inherit (gets the float value from parent).
* 

**CSS Clear:**

* It specifies what should happen to the element that is next to floating element.
* It can have following values :
  + None 🡪 Default value. The element is not pushed below left or right to floated elements.
  + Left 🡪 This specifies that the element is pushed below left to floated elements.
  + Right 🡪 This specifies that the element is pushed below right to floated elements.
  + Both 🡪 This specifies that the element is pushed below left and right floated elements.
  + Inherit 🡪 The element inherits the clear value from its parent.

**CSS Inline/Inline-block/Block:**

* Inline 🡪 This sets the style of element ignoring width, height , padding & margin.
* Inline-Block 🡪 This sets the style of element with width, height , padding & margin. It also doesn’t add any line breaks after the element so that elements can sit next to each other.
* Block 🡪 Gives line breaks after the element

**CSS Align:**

* To horizontally center a block level element (like <div>) we can use **margin:auto.** Margin-left, margin-right, margin:auto works only if the element has width specified.
* The element will take up the specified with and remaining space will be equally split into equal margins.
* To center a text we can use text-align:center and to make an image centered we can use below: 