**The position Property**

The position property specifies the type of positioning method used for an element.

There are five different position values:

* static
* relative
* fixed
* absolute
* sticky

**Position – Static**

This is the default value. The element is positioned according to the normal flow of the document.

**Example:**

.p-static {position: static; }

**Position – Relative**

The element with position: relative is positioned relative to its normal position. We can adjust the element position by setting the top, left, bottom, right values

**Example:**

.p-relative {  
  position: relative;top: 30px;   left: 30px;  
}

## position: fixed;

An element with position: fixed; will be fixed at the same place even if we scroll the page. Top, Right, Left, Bottom values are used to position the element

**Example**

Position-fixed {  
  position: fixed;  bottom: 0;right: 0; width: 300px;  
}

## position: absolute

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

**Example**

Position-absolute {position: absolute;  bottom: 0;right: 0; }

## position: sticky

An element with position: sticky; is positioned based on the user's scroll position.

**Example:**

div.sticky { position: sticky; top: 0; padding: 5px;background-color: #cae8ca;border: 2px solid #4CAF50;}

<h3>Scroll the page</h3>

<div class="sticky">I will stick to the screen when you reach my scroll position</div>

## Overflow Properties

The overflow property is used to create the scroll in the web page.

The overflow property has the following values:

* visible -Overflow visible is default value for the HTML Elements. The content renders outside the element's box
* hidden - The overflow is hidden, and the rest of the content will be invisible
* scroll – the scroll will be added to the elements.
* auto - Similar to scroll, but it adds scrollbars only when necessary

**Example**

div {width: 200px; height: 50px;  backgroundcolor: #eee;overflow: visible;}

**Canter Image or Div**

img {display: block; margin-left: auto; margin-right: auto; width: 40%;}

# **CSS Combinators**

A combinators used to specify the relationship between the selectors.

There are four different combinators in CSS:

* descendant selector (space)
* child selector (>)
* adjacent sibling selector (+)
* general sibling selector (~)

## Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside <div> elements:

div p {  
  background-color: yellow;  
}

## Child Selector (>)

The child selector selects all elements that are the children of a specified element.

The following example selects all <p> elements that are children of a <div> element:

## Adjacent Sibling Selector (+)

## The + selector is used to select an element that is directly after another specific element

<div>

<p>Paragraph 1 in the div.</p>

<p>Paragraph 2 in the div.</p>

</div>

<p>Paragraph 3. After a div.</p>

<p>Paragraph 4. After a div.</p>

<div>

<p>Paragraph 5 in the div.</p>

<p>Paragraph 6 in the div.</p>

</div>

<p>Paragraph 7. After a div.</p>

<p>Paragraph 8. After a div.</p>

## General Sibling Selector (~)

The general sibling selector selects all elements that are siblings of a specified element.

The following example selects all <p> elements that are siblings of <div> elements:

### **Example**

div ~ p {  
  background-color: yellow;  
}

<h2>General Sibling Selector</h2>

<p>The general sibling selector (~) selects all elements that are siblings of a specified element.</p>

<p>Paragraph 1.</p>

<div>

<p>Paragraph 2.</p>

</div>

<p>Paragraph 3.</p>

<code>Some code.</code>

<p>Paragraph 4.</p>

# **CSS Pseudo-classes**

## What are 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
* Syntax
* The syntax of pseudo-classes:

selector:pseudo-class {  
  property: value;  
}

## Anchor Pseudo-classes

Links can be displayed in different ways:

### **Example**

/\* unvisited link \*/  
a:link {color: #FF0000; }  
/\* visited link \*/  
a:visited {color: #00FF00; }  
/\* mouse over link \*/  
a:hover {color: #FF00FF;}  
/\* selected link \*/

a:active {color: #0000FF;}

## Hover on Div

## An example of using the :hover pseudo-class on a <div> element:

### **Example**

div:hover {background-color: blue;}

# **CSS Pseudo-elements**

# **What are Pseudo-Elements?**

# A CSS pseudo-element is used to style specified parts of an element.

# For example, it can be used to:

* Style the first letter, or line, of an element
* Insert content before, or after, the content of an element

**Syntax**

The syntax of pseudo-elements:

selector::pseudo-element {  
  property: value;  
}

## The ::first-line Pseudo-element

The ::first-line pseudo-element is used to add a special style to the first line of a text.

### **Example**

p::first-line {  
  color: #ff0000;  
  font-variant: small-caps;  
}

# **CSS Dropdowns**

<!DOCTYPE html>

<html>

<head>

<style>

.dropdown {

position: relative;

display: inline-block;

}

.dropdown-content {

display: none;

position: absolute;

background-color: #f9f9f9;

min-width: 160px;

box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

padding: 12px 16px;

z-index: 1;

}

.dropdown:hover .dropdown-content {

display: block;

}

</style>

</head>

<body>

<h2>Hoverable Dropdown</h2>

<p>Move the mouse over the text below to open the dropdown content.</p>

<div class="dropdown">

<span>Mouse over me</span>

<div class="dropdown-content">

<p>Hello World!</p>

</div>

</div>

</body>

</html>