ADVANCED FEATURES OF CSS

Visibility

A property called visibility allows you to hide an element from view. You can use this property along with JavaScript to create very complex menu and very complex webpage layouts.

You may choose to use the visibility property to hide error messages that are only displayed if the user needs to see them, or to hide answers to a quiz until the user selects an option.

NOTE – Remember that the source code will still contain whatever is in the invisible paragraph, so you should not use this to hide sensitive information such as credit card details or passwords.

The *visibility* property can take the values listed in the table that follows −

Sr.No.	Value & Description
1	**visible** The box and its contents are shown to the user.
2	**hidden** The box and its content are made invisible, although they still affect the layout of the page.
3	**collapse** This is for use only with dynamic table columns and row effects.

Here is an example −

```
<html>
  <head>
  </head>

  <body>
    <p>
      This paragraph should be visible in normal way.
    </p>

    <p style = "visibility:hidden;">
      This paragraph should not be visible.
    </p>
  </body>
</html>
```

It will produce the following result −

CSS - Positioning

CSS helps you to position your HTML element. You can put any HTML element at whatever location you like. You can specify whether you want the element positioned relative to its natural position in the page or absolute based on its parent element.

Now, we will see all the CSS positioning related properties with examples –

Relative Positioning

Relative positioning changes the position of the HTML element relative to where it normally appears. So "left:20" adds 20 pixels to the element's LEFT position.

You can use two values *top* and *left* along with the *position* property to move an HTML element anywhere in the HTML document.

- Move Left - Use a negative value for *left*.
- Move Right - Use a positive value for *left*.
- Move Up - Use a negative value for *top*.
- Move Down - Use a positive value for *top*.

NOTE – You can use *bottom* or *right* values as well in the same way as *top* and *left*.

Here is the example –

```
<html>
  <head>
  </head>

  <body>
    <div style = "position:relative; left:80px; top:2px; background-color:yellow;">
      This div has relative positioning.
    </div>
  </body>
</html>
```

It will produce the following result –

Absolute Positioning

An element with **position: absolute** is positioned at the specified coordinates relative to your screen top-left corner.

You can use two values *top* and *left* along with the *position* property to move an HTML element anywhere in the HTML document.

- Move Left - Use a negative value for *left*.
- Move Right - Use a positive value for *left*.
- Move Up - Use a negative value for *top*.
- Move Down - Use a positive value for *top*.

NOTE – You can use *bottom* or *right* values as well in the same way as top and left.

Here is an example –

Live Demo

```
<html>
  <head>
  </head>
```

```
  <body>
    <div style = "position:absolute; left:80px; top:20px; background-color:yellow;">
      This div has absolute positioning.
    </div>
  </body>
</html>
```

Fixed Positioning

Fixed positioning allows you to fix the position of an element to a particular spot on the page, regardless of scrolling. Specified coordinates will be relative to the browser window.

You can use two values *top* and *left* along with the *position* property to move an HTML element anywhere in the HTML document.

- Move Left - Use a negative value for *left*.
- Move Right - Use a positive value for *left*.
- Move Up - Use a negative value for *top*.
- Move Down - Use a positive value for *top*.

NOTE − You can use *bottom* or *right* values as well in the same way as *top* and *left*.

Here is an example −

Live Demo

```
<html>
  <head>
  </head>

  <body>
    <div style = "position:fixed; left:80px; top:20px; background-color:yellow;">
      This div has fixed positioning.
    </div>
  </body>
</html>
```

CSS - Layers

CSS gives you opportunity to create layers of various divisions. The CSS layers refer to applying the *z-index* property to elements that overlap with each other.

The z-index property is used along with the *position* property to create an effect of layers. You can specify which element should come on top and which element should come at bottom.

A z-index property can help you to create more complex webpage layouts. Following is the example which shows how to create layers in CSS.

```
<html>
  <head>
  </head>

  <body>
```

```
    <div style = "background-color:red;
      width:300px;
      height:100px;
      position:relative;
      top:10px;
      left:80px;
      z-index:2">
    </div>

    <div style = "background-color:yellow;
      width:300px;
      height:100px;
      position:relative;
      top:-60px;
      left:35px;
      z-index:1;">
    </div>

    <div style = "background-color:green;
      width:300px;
      height:100px;
      position:relative;
      top:-220px;
      left:120px;
      z-index:3;">
    </div>
  </body>
</html>
```

CSS - Pseudo Classes

CSS pseudo-classes are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-classes is as follows −

selector:pseudo-class {property: value}

CSS classes can also be used with pseudo-classes −

selector.class:pseudo-class {property: value}

The most commonly used pseudo-classes are as follows −

Sr.No.	Value & Description
1	**:link** Use this class to add special style to an unvisited link.
2	**:visited**

	Use this class to add special style to a visited link.
3	**:hover** Use this class to add special style to an element when you mouse over it.
4	**:active** Use this class to add special style to an active element.
5	**:focus** Use this class to add special style to an element while the element has focus.
6	**:first-child** Use this class to add special style to an element that is the first child of some other element.
7	**:lang** Use this class to specify a language to use in a specified element.

While defining pseudo-classes in a <style>...</style> block, following points should be noted −

- 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.
- Pseudo-class are different from CSS classes but they can be combined.

The :link pseudo-class

The following example demonstrates how to use the *:link* class to set the link color. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:link {color:#000000}
    </style>
  </head>

  <body>
    <a href = "">Black Link</a>
  </body>
</html>
```

It will produce the following black link −

The :visited pseudo-class

The following is the example which demonstrates how to use the *:visited* class to set the color of visited links. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:visited {color: #006600}
    </style>
  </head>

  <body>
    <a href = "">Click this link</a>
  </body>
</html>
```

This will produce following link. Once you will click this link, it will change its color to green.

The :hover pseudo-class

The following example demonstrates how to use the *:hover* class to change the color of links when we bring a mouse pointer over that link. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:hover {color: #FFCC00}
    </style>
  </head>

  <body>
    <a href = "">Bring Mouse Here</a>
  </body>
</html>
```

It will produce the following link. Now you bring your mouse over this link and you will see that it changes its color to yellow.

The :active pseudo-class

The following example demonstrates how to use the *:active* class to change the color of active links. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:active {color: #FF00CC}
    </style>
  </head>

  <body>
```

```
    <a href = "">Click This Link</a>
  </body>
</html>
```

It will produce the following link. When a user clicks it, the color changes to pink.

The :focus pseudo-class

The following example demonstrates how to use the *:focus* class to change the color of focused links. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:focus {color: #0000FF}
    </style>
  </head>

  <body>
    <a href = "">Click this Link</a>
  </body>
</html>
```

It will produce the following link. When this link gets focused, its color changes to orange. The color changes back when it loses focus.

The :first-child pseudo-class

The *:first-child* pseudo-class matches a specified element that is the first child of another element and adds special style to that element that is the first child of some other element.

To make :first-child work in IE <!DOCTYPE> must be declared at the top of document.

For example, to indent the first paragraph of all elements, you could use this definition –

```
<html>
  <head>
    <style type = "text/css">
      div > p:first-child {
        text-indent: 25px;
      }
    </style>
  </head>

  <body>

    <div>
      <p>First paragraph in div. This paragraph will be indented</p>
      <p>Second paragraph in div. This paragraph will not be indented</p>
    </div>
    <p>But it will not match the paragraph in this HTML:</p>

    <div>
```

```
      <h3>Heading</h3>
      <p>The first paragraph inside the div. This paragraph will not be effected.</p>
    </div>

  </body>
</html>
```

It will produce the following result −

The :lang pseudo-class

The language pseudo-class *:lang*, allows constructing selectors based on the language setting for specific tags.

This class is useful in documents that must appeal to multiple languages that have different conventions for certain language constructs. For example, the French language typically uses angle brackets (< and >) for quoting purposes, while the English language uses quote marks (' and ').

In a document that needs to address this difference, you can use the :lang pseudo-class to change the quote marks appropriately. The following code changes the <blockquote> tag appropriately for the language being used −

```
<html>
  <head>
    <style type = "text/css">

      /* Two levels of quotes for two languages*/
      :lang(en) { quotes: "'" "'" "'" "'"; }
      :lang(fr) { quotes: "<<" ">>" "<" ">"; }
    </style>
  </head>

  <body>
    <p>...<q lang = "fr">A quote in a paragraph</q>...</p>
  </body>
</html>
```

The :lang selectors will apply to all the elements in the document. However, not all elements make use of the quotes property, so the effect will be transparent for most elements.

CSS - Pseudo Elements

CSS pseudo-elements are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-element is as follows −

selector:pseudo-element {property: value}

CSS classes can also be used with pseudo-elements −

selector.class:pseudo-element {property: value}

The most commonly used pseudo-elements are as follows −

Sr.No.	Value & Description
1	**:first-line** Use this element to add special styles to the first line of the text in a selector.
2	**:first-letter** Use this element to add special style to the first letter of the text in a selector.
3	**:before** Use this element to insert some content before an element.
4	**:after** Use this element to insert some content after an element.

The :first-line pseudo-element

The following example demonstrates how to use the *:first-line* element to add special effects to the first line of elements in the document.

```
<html>
  <head>
    <style type = "text/css">
      p:first-line { text-decoration: underline; }
      p.noline:first-line { text-decoration: none; }
    </style>
  </head>

  <body>
    <p class = "noline">
      This line would not have any underline because this belongs to nline class.
    </p>

    <p>
      The first line of this paragraph will be underlined as defined in the
      CSS rule above. Rest of the lines in this paragraph will remain normal.
      This example shows how to use :first-line pseduo element to give effect
      to the first line of any HTML element.
    </p>
  </body>
</html>
```

It will produce the following link –

The :first-letter pseudo-element

The following example demonstrates how to use the *:first-letter* element to add special effects to the first letter of elements in the document.

```
<html>
  <head>
    <style type = "text/css">
      p:first-letter { font-size: 5em; }
      p.normal:first-letter { font-size: 10px; }
    </style>
  </head>

  <body>
    <p class = "normal">
      First character of this paragraph will be normal and will have font size 10 px;
    </p>

    <p>
      The first character of this paragraph will be 5em big as defined in the
      CSS rule above. Rest of the characters in this paragraph will remain
      normal. This example shows how to use :first-letter pseduo element
      to give effect to the first characters  of any HTML element.
    </p>
  </body>
</html>
```

It will produce the following black link –

The :before pseudo-element

The following example demonstrates how to use the *:before* element to add some content before any element.

```
<html>
  <head>
    <style type = "text/css">
      p:before {
        content: url(/images/bullet.gif)
      }
    </style>
  </head>

  <body>
    <p> This line will be preceded by a bullet.</p>
    <p> This line will be preceded by a bullet.</p>
    <p> This line will be preceded by a bullet.</p>
  </body>
</html>
```

It will produce the following black link –

The :after pseudo-element

The following example demonstrates how to use the *:after* element to add some content after any element.

```
<html>
  <head>
    <style type = "text/css">
      p:after {
        content: url(/images/bullet.gif)
      }
    </style>
  </head>

  <body>
    <p> This line will be succeeded by a bullet.</p>
    <p> This line will be succeeded by a bullet.</p>
    <p> This line will be succeeded by a bullet.</p>
  </body>
</html>
```