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 −

Value	Description
:link	Use this class to add special style to an unvisited link.
:visited	Use this class to add special style to a visited link.
:hover	Use this class to add special style to an element when you mouse over it.
:active	Use this class to add special style to an active element.
:focus	Use this class to add special style to an element while the element has focus.
:first-child	Use this class to add special style to an element that is the first child of some other element.
: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 −

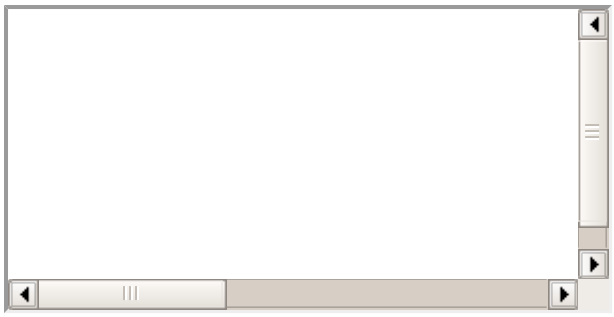
- a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.
- a:active MUST come after a:hover in the CSS definition in order to be effective.
- Pseudo-class names are not case-sensitive.
- Pseudo-class are different from CSS classes but they can be combined.

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 −

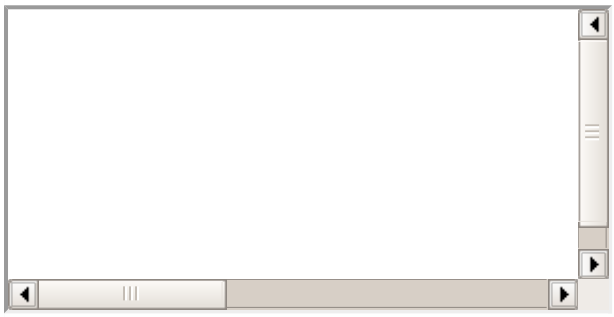


The :visited pseudo-class

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.

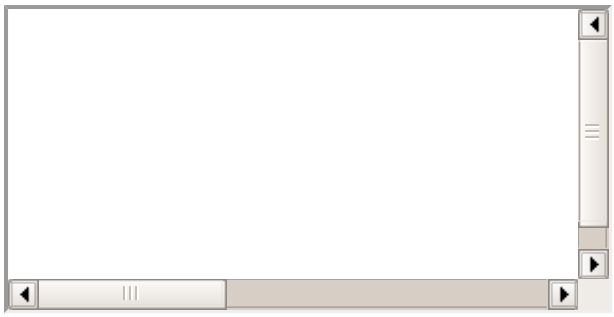


The :hover pseudo-class

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.

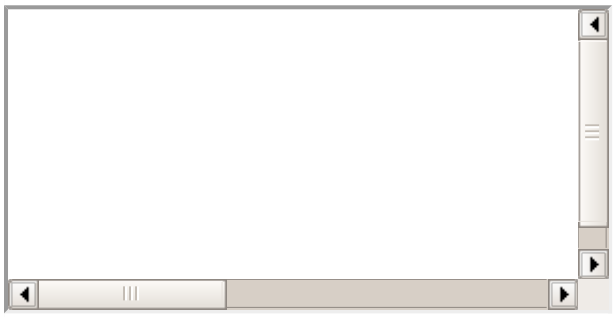


The :active pseudo-class

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.

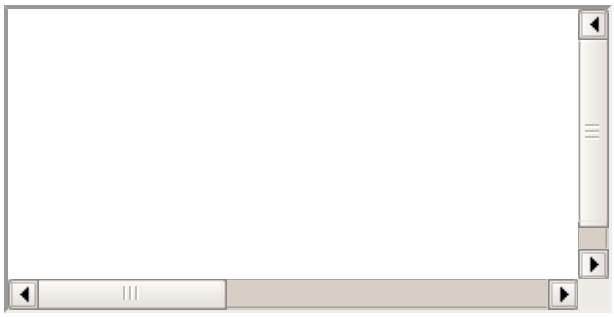


The :focus pseudo-class

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

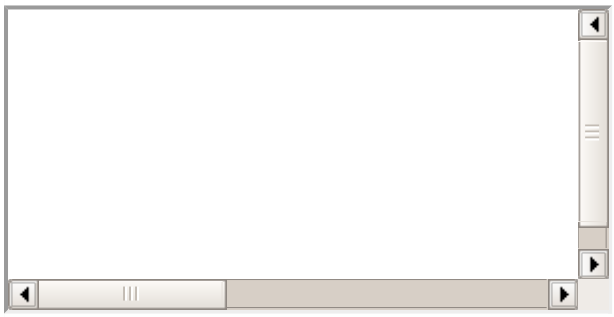
The *:first-child* pseudo-class matches a specified element that is the first child of another element and adds special style to that element that is the first child of some other element.

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.

It will produce the following result −