

# Cascading Style Sheet Pseudo Classes

Cascading Style Sheet Pseudo Classes are specialized classes that can be applied to some CSS elements. They are used to create special effects. In this document you will discover how to apply the current active pseudo classes. It is expected that when CSS3 is finalized and released it will introduce additional pseudo classes beyond those that are discussed here.

## The Most Common Pseudo Classes

The pseudo classes that are most commonly used are those for links. They allow the designer to specify color and decoration for link states. These can result in “rollover-like” effects without having to use images or JavaScript.

The default appearance for a text link is standard blue and underlined. Using pseudo classes you can specify custom colors, underlines or no underlines, and other changes. There are four states to links, and each can be customized with a pseudo class. Those states are:

- `:link`
- `:visited`
- `:active`
- `:hover`

The `:link` pseudo class specifies the attributes of the link in its normal state.

The `:visited` pseudo class specifies the attributes of the link after it has been visited by the browser.

The `:active` pseudo class specifies the attributes of the link while it is active, such as when the left mouse button is clicked.

The `:hover` pseudo class specifies the attributes of the link when the mouse passes directly over it.

The attributes of a pseudo class may be the same as those of another pseudo class or they can be different. You are allowed to “mix and match” attributes. In some browsers order is important, so to be sure that everything works correctly place the pseudo classes in the CSS document in the order shown above.

The order in which the pseudo classes are listed is important. If they are not in the Style Sheet as noted above one or more may fail. The success or failure of the pseudo class effects are determined by Cascading Style Sheets rules. You may be able to skip the `:active` pseudo class, but you will not be able to re-order it. Be prepared to test your pseudo classes in several browsers to make sure everything works the way you expect. If something does not work check order, spelling, and syntax.

## Specifying Pseudo Classes

Here is a typical HTML snippet with a paragraph featuring a simple sentence and text link:

```
<p>Here is a typical sentence with a link to <a href=http://www.google.com>Google</a> in it.</p>
```

To cause this link to appear underlined and red in the :link state, underlined and green in the :visited state, underlined and yellow in the :active state, and underlined in blue in the :hover state are:

```
a:link { color: #F00; text-decoration: underline; }
a:visited { color: #0F0; text-decoration: underline; }
a:active { color: #FF0; text-decoration: underline; }
a:hover { color: #00F; text-decoration: underline; }
```

You can use standard CSS syntax and customize the various states. Here are some examples:

```
a:link { color: #033;
        text-decoration: none;
        font-family: Verdana, Geneva, sans-serif;
        font-weight: bold; }
a:visited { color: #009;
           text-decoration: none;
           font-family: Verdana, Geneva, sans-serif;
           font-weight: bold; }
a:active { color: #F00;
          text-decoration: underline;
          font-family: Verdana, Geneva, sans-serif;
          font-weight: bold;
          background-color: #CCC; }
a:hover { color: #990;
         text-decoration: underline;
         font-family: Verdana, Geneva, sans-serif;
         font-weight: bold;
         background-color: #9F0; }
```

Notice that in the examples above some have underling and some do not (text-decoration: underling or text-decoration:none), the font family is specified, various colors for the text are specified, and a background color for the :hover pseudo class is established.

Experimenting with the pseudo classes is encouraged as that is an excellent way to become familiar with how they work.