CSS has several options for redefining the style of links.

**LINK PROPERTIES**

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
<th>NS</th>
<th>IE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A:link</td>
<td>&lt;style&gt;</td>
<td>4+</td>
<td>4+</td>
</tr>
<tr>
<td>A:visited</td>
<td>&lt;style&gt;</td>
<td>4+</td>
<td>4+</td>
</tr>
<tr>
<td>A:active</td>
<td>&lt;style&gt;</td>
<td>4+</td>
<td>4+</td>
</tr>
<tr>
<td>A:hover</td>
<td>&lt;style&gt;</td>
<td>6+</td>
<td>4+</td>
</tr>
</tbody>
</table>

**DEFINING STYLES FOR LINKS**

As mentioned in the above table, there are four different selectors with respect to links.

You can specify whatever style you'd like to each of these selectors, just like you'd do with normal text.

The four selectors are:

- **A:link**
  Defines the style for normal unvisited links.

- **A:visited**
  Defines the style for visited links.

- **A:active**
  Defines the style for active links.
  A link becomes active once you click on it.

- **A:hover**
  Defines the style for hovered links.
  A link is hovered when the mouse moves over it.
  Note: Not supported by Netscape browsers prior to version 6.

**PRACTICAL EXAMPLES**

Here you can see a few examples on how CSS can be used to replace the traditional image based mouseover effects for links.
The **hover** style is not supported by Netscape browsers prior to version 6, but since it does no harm, you can still use it for the benefit of the +90% of visitors that arrive using MSIE).

One of the most common uses of CSS with links is to remove the underline. Typically it's done so that the underline appears only when a hover occurs. In the example below, we did just that. In addition we added a red color for hovered links.

**Example: Hover**

```html
<style type="text/css">
A:link {text-decoration: none}
A:visited {text-decoration: none}
A:active {text-decoration: none}
A:hover {text-decoration: underline; color: red;}
</style>
```

The link from the above example would look like this:

```
LOOK HERE
```

Another example would be to create links that are both **underlined** and **overlined**.

**Example: Underline/Overline**

```html
<style type="text/css">
A:link {text-decoration: none}
A:visited {text-decoration: none}
A:active {text-decoration: none}
A:hover {text-decoration: underline overline; color: red;}
</style>
```

The link from the above example would look like this:

```
LOOK HERE
```

A third example would be to create links that change in size when hovered.

**Example: Size changing links**

```html
<style type="text/css">
A:link {text-decoration: none}
</style>
```
A final example would be to create links that have a permanent background color, obviously standing out from the rest.

**Example: Background colored links**

```css
<style type="text/css">
A:link {background: #FFCC00; text-decoration: none}
A:visited {background: #FFCC00; text-decoration: none}
A:active {background: #FFCC00; text-decoration: none}
A:hover {background: #FFCC00; font-weight:bold; color: red;}
</style>
```

The link from the above example would look like this:

**LOOK HERE TO SEE**

---

**MULTIPLE LINKSTYLES ON SAME PAGE**

The final topic deals with how to add multiple link styles that can be used on the same page.

In the above examples we addressed the HTML selector - A:link etc - and thus redefined the overall link style.

How do we define a link style that is only active in a certain area of the page?

The answer is: **context dependent selectors**.

Rather than addressing the A:link selector we will address it while being dependant on a certain outer class that surrounds the area where we'd like our link style to be effective.

For example:
Note how we use the `<span>` to define the context.
This is smart for two reasons:

1) The obvious, that it allows us to use different link styles on
the same page, rather than being limited to using a single
overall link style.

2) We can define entire areas where a certain link style works for all links within
that area. Thus, we don't have to add a style definition to each and every link in
that area.