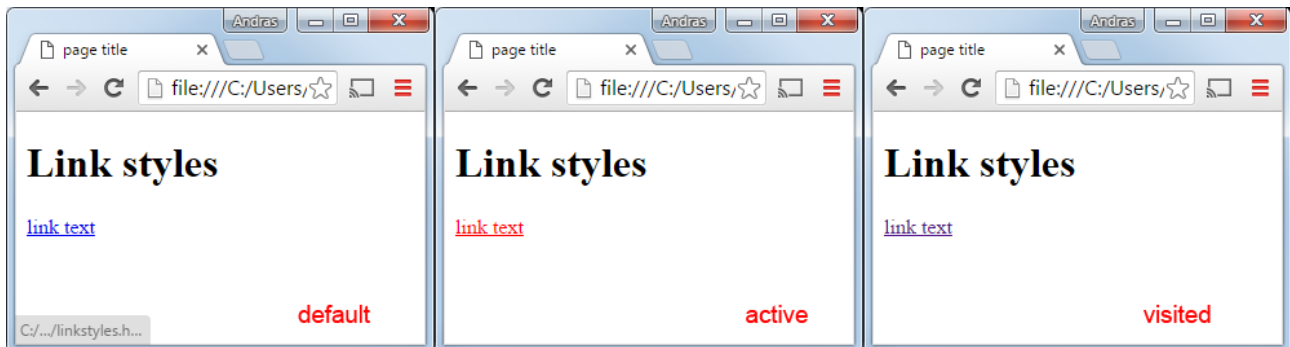


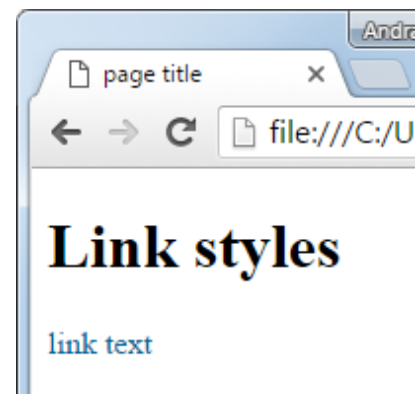
## Hyperlinks and CSS

The appearance of links in the document may be changed by selecting an anchor element (`a`) and redefining its styles. By default, hyperlinks are underlined and have three different states that are identified by **text color**: default (*normal*) state is blue, *active* state is red, and the *visited* state purple. The active state is activated when the link is being clicked or was just clicked. The visited state is active when the linked resource has been visited during the current or a previous browsing session (browser history). The anchors trigger area (where you can activate the link) can be extended with padding.



All these states can be reset by a unified style definition selected with an A selector (anchor element):

```
a {                                /* applies to all links and states */
  color: #006699;
  text-decoration: none;
}
/* deletes underlining, accessibility considerations! */
```



You may define **pseudo selectors** to control link states, **:link**, **:visited**, **:active**, **:hover**

A pseudo selector can be added to an A selector and has a predefined function to control a specific link state. The additional state, **:hover**, is activated when the mouse pointer is over the element's content.

```
a { font-weight: bold; }           /* applies to all states */
a:link { color: #006699; }        /* applies to normal state */
a:visited { color: gray; }        /* applies to visited state */
a:hover {                         /* applies to hover state */
  background-color: #00AAFF;
  color: #006699;
  text-decoration: none;
}
a:active { color: #F00; }          /* applies to active state */
```

When you define the **active** state after the **hover** state, both states will be activated at the same time when you click the link while styles for the hover state are still visible.

You can still add general style rules for your link (rules that apply for the link in all states unless overridden) while you define rules for the separate states.

An anchor point with all 4 states different.

**normal state**

---

```

a { font-weight: bold; } // applies to all states
a:link { color: #006699; } // applies to normal state
a:visited { color: gray; } // applies to visited state
a:hover { // applies to hover state
    background-color: #00AAFF;
    color: #006699;
    text-decoration: none;
}
a:active { color: #F00; } // applies to active state

```

The visited state overrides the color of the normal state if the link's URL was opened earlier...

An anchor point with all 4 states different.

**visited state**

---

```

a { font-weight: bold; } // applies to all states
a:link { color: #006699; } // applies to normal state
a:visited { color: gray; } // applies to visited state
a:hover { // applies to hover state
    background-color: #00AAFF;
    color: #006699;
    text-decoration: none;
}
a:active { color: #F00; } // applies to active state

```

The hover state is triggered if the mouse pointer is over the content of the anchor element. Here, it also replaces the color of text of the visited state if applicable (same color is set as in the normal state).

An anchor point with all 4 states different.

**hover state**

---

```

a { font-weight: bold; } // applies to all states
a:link { color: #006699; } // applies to normal state
a:visited { color: gray; } // applies to visited state
a:hover { // applies to hover state
    background-color: #00AAFF;
    color: #006699;
    text-decoration: none;
}
a:active { color: #F00; } // applies to active state

```

When the mouse pointer is over the link as you click it, both hover and active states are visible:

An anchor point **with all 4 states** different. **hover & active states**

```
a { font-weight: bold; } // applies to all states
a:link { color: #006699; } // applies to normal state
a:visited { color: gray; } // applies to visited state
a:hover { // applies to hover state
    background-color: #00AAFF;
    color: #006699;
    text-decoration: none;
}
a:active { color: #F00; } // applies to active state
```

However if only the active state is visible, without the mouse pointer over the content of the anchor:

An anchor point **with all 4 states** different. **only active state**

Link states are generally presented in the following order **L V H A** (link, visited, hover, active) or **LVAH** (link, visited, active, hover) depending what layout features are set and how they may overlap. For example, a link in its active state may keep its properties while the mouse-pointer may hover over the link showing additional layout features.

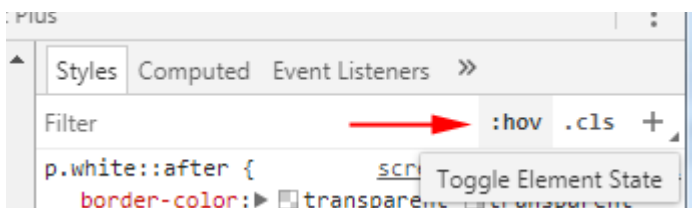
The pseudo selectors are more specific, have more weight, than the simple selector for the anchor element!

```
a:link { color: #006699; }
a { color: red; }
```

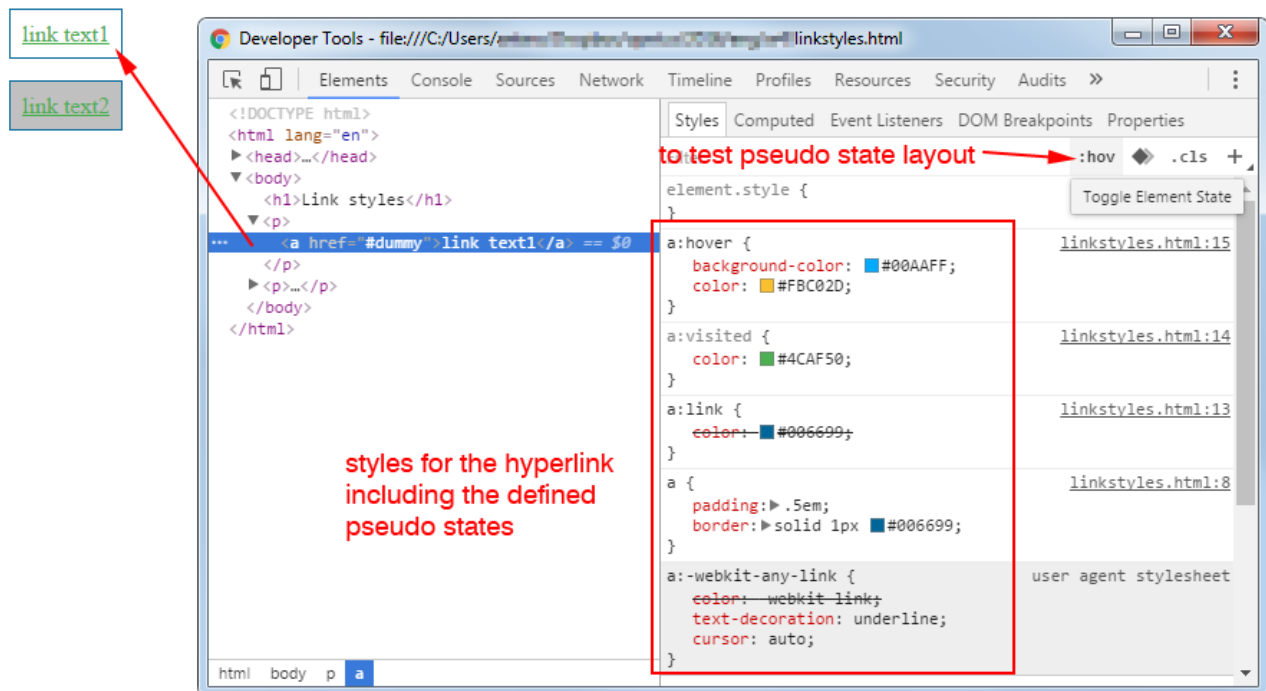


<- The link text is blue, not red...

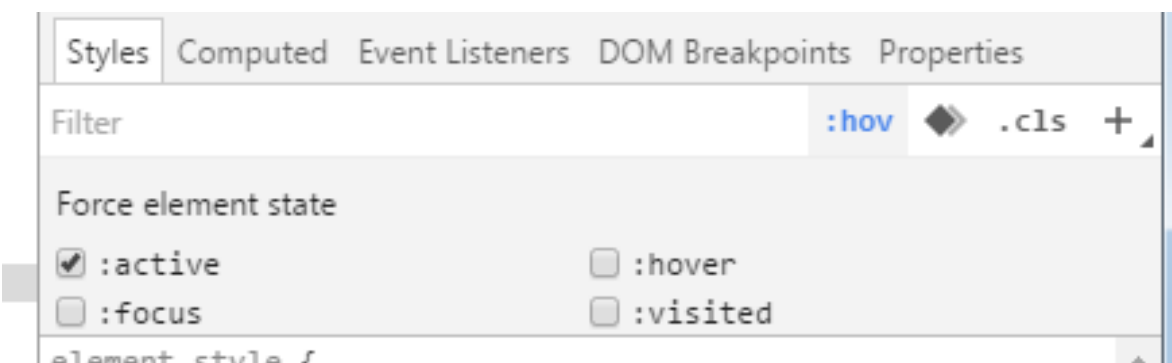
You can test all link states in the Developer tool. Select an anchor element to see the link styles in the Styles tab. At the top of the Styles tab you can toggle the states – Filter -> **:hov**. See images below:



## Link styles

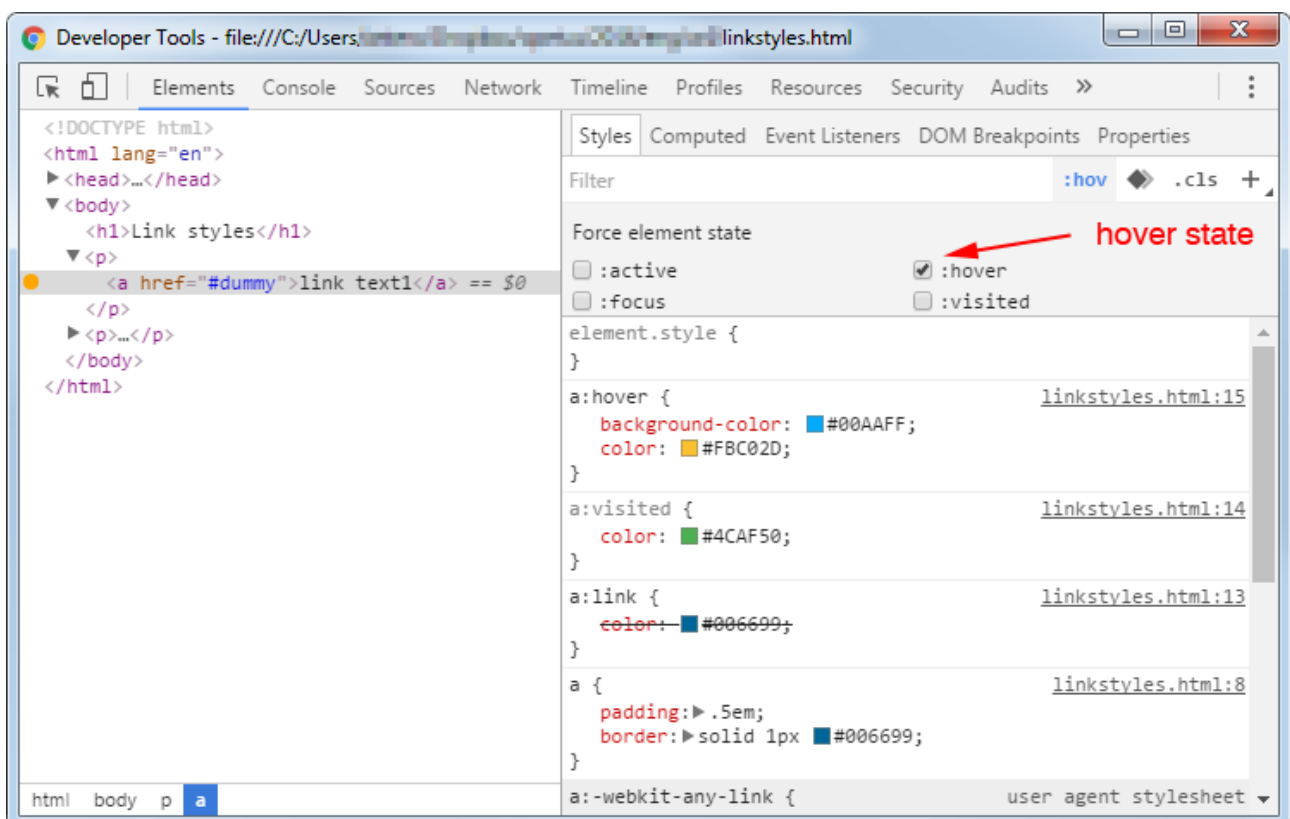
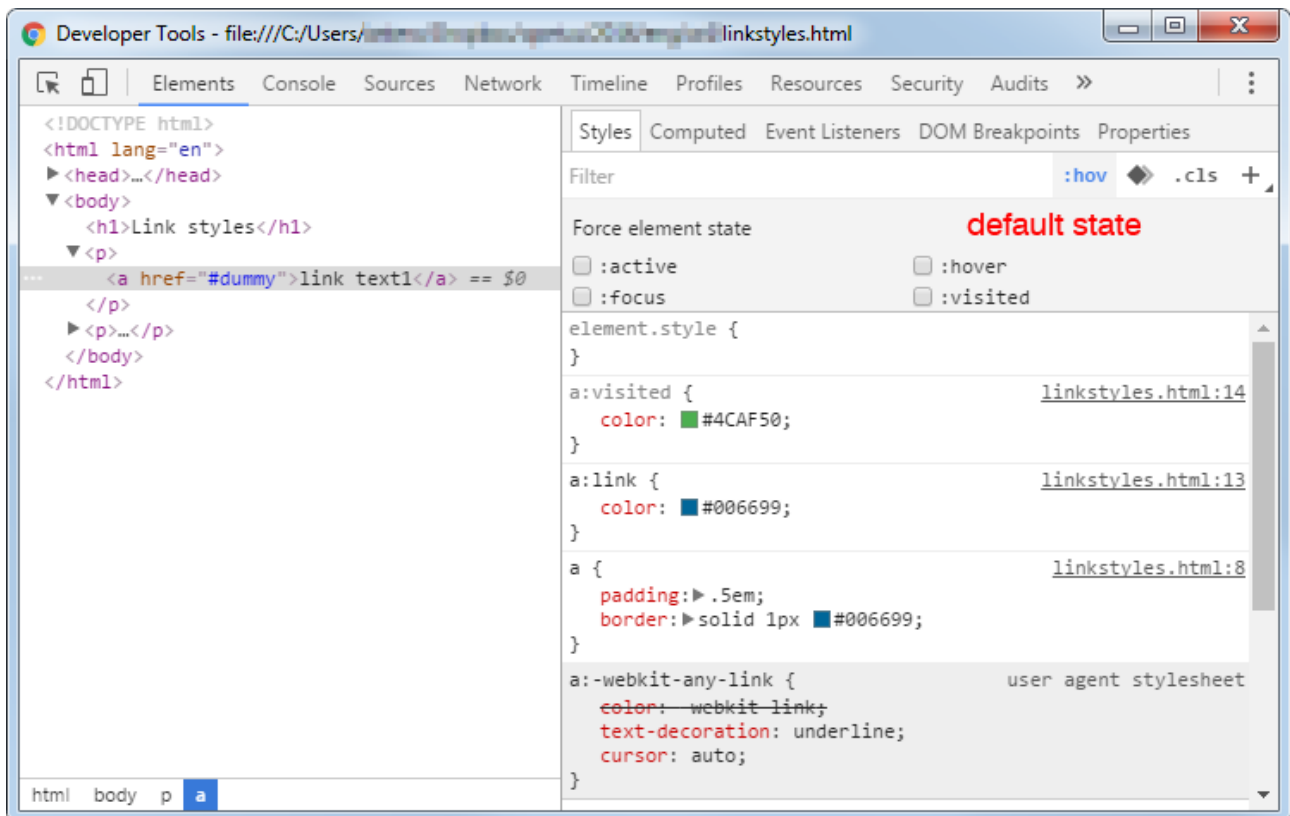


When you open the element state toggle view, you can test the different states for the link states by forcing different states to be visible (as a preview in the browser window):



The active state is forced to show in the above image.

The `:focus` state (visible in the image) is a state that is associated with the focus selection of interactive elements, that could be links and form elements. The user may use the tabulator key on the keyboard to hop through interactive elements on the page (the focus state is then activated).



Here is another style example:

<p>A paragraph with text and a <a href="example.html">hyperlink</a> in a document.</p>

```
a {                                /* general rules */
font-weight: bold;
text-decoration: none;
border-bottom: solid 2px blue;
padding: 5px 2px;
}

a:link { color: black; }          /* rules for all states */
a:visited { color: blue; }
a:hover { background: #ddd; }
a:active { border-bottom: solid 5px red; }
```

A paragraph with text and a [hyperlink](#) in a document.

```
a {
font-weight: bold;
text-decoration: none;
border-bottom: solid 2px blue;
padding: 5px;
}

a:link { color: black; }
a:visited { color: blue; }
a:hover { background: #ddd; }
a:active { border-bottom: solid 5px red; }
```

**normal** state:

A paragraph with text and a [hyperlink](#) in a document.

**visited** state visible:

A paragraph with text and a [hyperlink](#) in a document.

**hover** state:

A paragraph with text and a [hyperlink](#) in a document.

**active** state:

A paragraph with text and a [hyperlink](#) in a document.

**active** and **hover** states usually apply together

A paragraph with text and a [hyperlink](#) in a document.

**visited** and **hover** states may apply together as well

A paragraph with text and a [hyperlink](#) in a document.

also **active**, **visited** and **hover** states may all apply together...

A paragraph with text and a [hyperlink](#) in a document.

All these variations can be tested with the developer tool and refined as necessary...

## Defining separate sets of links within a document

If there is a need for differently styled sets of links, you can create a class or use an id for different sections of a document. You may also use existing hierarchy, here is an example of a conditional rule, that can be used to select an element based on its position in the element order – descendant selector:

```
<div id="nav">Text content with <a href="URL">links</a></div>
<div id="content"> Text content with <a href="URL">links</a></div>
<div class="box"> Text content with <a href="URL">links</a></div>
<div id="footer"> Text content with <a href="URL">links</a></div>
```

`#nav a { ... }` descendant selector -> selects A element if it is a descendant (child) of an element with the ID "nav". And with different states: `#nav a:link { ... }` `#nav a:visited { ... }` `#nav a:hover { ... }` `#nav a:active { ... }`  
So all content parts may use a different link style:

```
#nav a { ... }
#content a { ... }
.box a { ... }
#footer a { ... }
```

Alternatively, link styles may be simply added with separate classes:

```
<a class="rednav">...</a>
a.rednav { color: red; }
a.rednav:hover { background: orange; }
```