

```

tagname.class {
  property: value;
  ...
}

```

Example 3.40 Syntax template for tag/class selector

```

h2.urgent {
  text-align: center;
}

```

Example 3.41 Tag/class selector

Some web programmers have trouble remembering which of **id** and **class** uses the hash sign # for its CSS selector and which uses the dot. We use a mnemonic device to help remember this. Our students have learned Java before learning CSS; compiling a Java program produces a **.class** ("dot-class") file. So we tell them to remember "dot-class" because the dot goes with **class** here as well.

The names you give for your IDs and classes are largely up to you, but some names are better than others. Many programmers name them poorly, choosing names based on how they want the content to look. Examples are **redtext**, **boldheading**, **centered**, and **bigfont**. These are poor names because they focus on appearance rather than on the meaning of content. If the web designer changes the appearance, such as making **redtext** items blue, the names won't make sense.

Instead, focus on the semantics and meaning of the content. Why is the text red? If the text is red to alert the user about an important warning, name the class **warning** instead. What text is being shown in the **biggerfont**? If the **biggerfont** is used for emphasis, maybe you should just use an **em** HTML tag instead and then style **em** tags to use a larger non-italic font. Semantic names are better for those designing the page and anyone reading it later to try to understand its content.

#### Don't Be a Newb

Give your IDs and classes semantic names



### 3.3.4 Pseudo-class Selectors

A CSS pseudo-class is a special kind of selector that targets an element only under certain specific conditions, generally based on the state of that element. For example, using a pseudo-class selector you can apply a style only to links that have already been clicked, or only to every other row of a table, or only to elements that are currently underneath the mouse cursor, and so on.

A pseudo-class selector begins with a colon, followed by its name. Some pseudo-classes have additional information provided in parentheses after the name of the pseudo-class. A pseudo-class can be written by itself, in which case it applies to all elements, or preceded by any other selector, in which case it applies only to the specific element(s) targeted by that selector. Example 3.42 shows the general syntax.

#### pseudo-class selector

A CSS rule beginning with a colon (:) that applies only to particular element(s) under specific conditions.

```

selector:pseudoclassname {
  property: value;
  ...
}
selector:pseudoclassname(parameters) {
  property: value;
  ...
}

```

Example 3.42 Syntax template for pseudo-class selector

CSS pseudo-class	Description
:visited, :link	links that have, or have not, been clicked previously
:active	elements that are being interacted with by the user (e.g. while mouse is being pressed on the element)
:checked	checkboxes that are currently checked
:enabled, :disabled	elements that are currently enabled or disabled on screen
:focus	elements that currently have the user input focus
:hover	element that has the mouse pointer positioned on top of it
:nth-child( <b>expr</b> ), :nth-of-type( <b>expr</b> ), :first-child, :last-child,	targets specific children of a given element
:not( <b>selector</b> )	all elements that do <i>not</i> match the given CSS selector
::first-letter	the first non-space character of text within the element
::first-line	the first line of text within the element

If a pseudo-class selector is written by itself, it applies to all elements, but it can be preceded by an element, class, or id. A common usage of a pseudo-class is to highlight links in a different color or style when the user hovers the mouse over them. Example 3.43 shows such a technique. Notice that the pseudo-class is preceded by **a** to indicate that it should apply only to links, **a** elements. Without this change, every single element on the page would become cyan and italic when the user moved the mouse over it.

```
a: hover {
  background-color: cyan;
  font-style: italic;
}
```

This is [a link](#); click it!

This is **a link**; click it!

**Example 3.43** Link that highlights on hover

CSS3 contains a useful new pseudo-class named `nth-child` that allows you to apply a style to, say, every second or third element in a list or table or other area of the page. Before the existence of this pseudo-class, the web developer was forced to manually apply the style to every second or third child, or to apply it programmatically using code on a web server. The `nth-child` pseudo-class is followed by a set of parentheses containing an expression representing which children to target. The expression can be the word `odd` or `even`, to select alternating elements starting with the first or second element respectively. Example 3.44 demonstrates using this pseudo-class to highlight alternating items in a list, a technique sometimes called "zebra striping."

```
<ul>
  <li>Monday</li> <li>Tuesday</li> <li>Wednesday</li> <li>Thursday</li>
  <li>Friday</li> <li>Saturday</li> <li>Sunday</li>
</ul>
```

```
li:nth-child(odd) {  
  background-color: gray;  
  font-weight: bold;  
}
```

- **Monday**
- Tuesday
- **Wednesday**
- Thursday
- **Friday**
- Saturday
- **Sunday**

**Example 3.44 Using nth-child pseudo-class for "zebra striping"**

The expression provided to `nth-child` can also be a mathematical expression of the form  $an+b$ , indicating that every  $a^{\text{th}}$  element should be targeted, starting from the  $b^{\text{th}}$  element. Example 3.45 uses an expression to select every third element of the list, starting with the first.

```
li:nth-child(3n+1) {  
  background-color: gray;  
  font-weight: bold;  
}
```

- **Monday**
- Tuesday
- Wednesday
- **Thursday**
- Friday
- Saturday
- **Sunday**

**Example 3.45 Using nth-child pseudo-class with complex expression**

The `:first-child` and `:last-child` pseudo-classes are shorthand to select only the very first or last child element within a specific part of the page. The `:nth-of-type` pseudo-class is very similar to `:nth-child` except that it targets only children with the same element type as each other. If our selector already constrains us to one element type, as with the `li` selector used in our example, there is no difference between the two pseudo-classes. Note that `:nth-child` and its relatives are all new properties introduced in CSS3 and may not display properly in old browsers.

### 3.3.5 The W3C CSS Validator

Because browsers don't show any error messages when CSS is incorrect, it can be difficult to tell whether your CSS code is valid. Sometimes you'll see an unusual or incorrect appearance on the page that will alert you to a problem in your styles, but other mistakes are more subtle.

The W3C provides a validation service for CSS that you can use to be sure your CSS syntax is correct. The W3C CSS validator, found at <http://jigsaw.w3.org/css-validator/>, is a web site that checks your CSS code to be sure it meets the official W3C specifications.

If your webpage is accessible on the Internet, another way to send your code to the validator is to add a link on any page you create that, when clicked, will run the W3C CSS validator on that page. This link should target <http://jigsaw.w3.org/css-validator/check/referer>. Many documents use a standard W3C image for their CSS validator link, as shown in Example 3.46.