

4.3 Advanced selectors

Universal selector

The **universal selector**, specified using an asterisk character (*), matches *all* elements in the webpage. The universal selector is implied when an element name is not specified. Ex: The CSS selectors `.highlight` and `*.highlight` match exactly the same elements, where the universal selector is implied in `.highlight` and explicit in `*.highlight`.

Multiple selector

The **multiple selector**, specified using a comma (,) to separate selectors, matches all listed elements to apply a style rule. Ex: The figure below shows two CSS examples that apply the same styles to `` and `` elements. The CSS on the left unnecessarily duplicates the styling declarations for `` and `` elements, whereas the CSS on the right uses a single style declaration using a multiple selector.

Figure 4.3.1: Comparing separate and multiple selectors.

Separate selectors	Multiple selector
<pre>ul { background-color: gray; color: white; font-weight: bold; } ol { background-color: gray; color: white; font-weight: bold; }</pre>	<pre>ul, ol { background-color: gray; color: white; font-weight: bold; }</pre>

[Feedback?](#)

Child selector

The **child selector**, specified using a greater than character (>) between two selectors, matches any elements where the second element is a direct child of the first element. The child selector is similar to the descendant selector (space between selectors), but the

matching child element in the child selector must be a direct child of the matching parent element.

PARTICIPATION ACTIVITY

4.3.1: Styling background color with child selector.

**Start** 2x speed

```
p em {  
    background-color: yellow;  
}  
  
p > em {  
    background-color: green;  
}
```

```
<p>  
    The name  
    <strong><em>Neo</em></strong>  
    is an anagram for  
    <em>One</em>.  
</p>
```

The name **Neo** is an anagram for **One**

Captions ^

1. The em elements are descendants of the paragraph element, so the descendant selector matches both elements.
2. The "Neo" em element is a direct child of the strong element, not the p element, so the p > em child selector does not match the "Neo" em.
3. The p > em child selector does match the "One" em element since the em is a direct child of p.

Feedback?**PARTICIPATION ACTIVITY**

4.3.2: List item selectors.



Match the CSS selectors to the best descriptions.

If unable to drag and drop, refresh the page.

li

Match all list items.

The element selector **li** matches all **** elements.

Correct

Match all list items that are descendants of ordered lists.

Correct

ol li

The descendant selector `ol li` matches `` elements contained anywhere within an `` element.

Correct**ol > li**

Match all list items that are direct children of ordered lists.

The child selector matches `` elements that occur as direct children of an `ol` element.

Reset**Feedback?****PARTICIPATION ACTIVITY**

4.3.3: Matching specific elements.



Refer to the HTML below.

```

<p>James Bond is known to drive the <span class="mfg">Aston Martin</span> DB5
Below are a few cars James Bond has driven in recent Bond films according to
<a href="https://en.wikipedia.org/wiki/List_of_James_Bond_vehicles">Wikipedia</a>
</p>

<ol>
  <li><cite>Casino Royale</cite> (2006)
    <ul>
      <li><span class="mfg">Bentley</span> 4.5 Litre</li>
      <li><span class="mfg">Aston Martin</span> DB5</li>
    </ul>
  </li>
  <li><cite>Quantum of Solace</cite> (2008)
    <ul>
      <li><span class="mfg">Aston Martin</span> DBS V12</li>
    </ul>
  </li>
  <li><cite>Skyfall</cite> (2012)
    <ul>
      <li><span class="mfg">Jaguar</span> XJ (X351)</li>
      <li><span class="mfg">Mercedes</span> S400</li>
      <li><span class="mfg">Aston Martin</span> DB5</li>
    </ul>
  </li>
  <li><cite>Spectre</cite> (2015)
    <ul>
      <li><span class="mfg">Aston Martin</span> DB10</li>
    </ul>
  </li>
</ol>

```

- What CSS selector matches all list items?

Correct

li

Check**Show answer**

li

The `li` selector matches all list items.

- 2) What selector matches list items that are direct children of ordered lists but not unordered lists?

ol>li

Check**Show answer****Correct**

`ol > li`



The child selector `ol > li` matches list items that are direct children of ordered lists, but will not match the list items of an unordered list.

- 3) What selector matches *all* elements with the `mfg` class attribute?

.mfg

Check**Show answer****Correct**

`*.mfg` or `.mfg`



The `*.mfg` and `.mfg` universal selectors match all elements with the `mfg` class attribute.

- 4) What selector matches ordered and unordered lists?

ol,ul

Check**Show answer****Correct**

`ol, ul`



The multiple selector `ol, ul` matches ordered and unordered lists.

- 5) What selector matches `<cite>`, ``, and `<a>` elements?

cite,span,a

Check**Show answer****Correct**

`cite, span, a`



The multiple selector `cite, span, a` matches all cite, span, and a elements.

Feedback?

PARTICIPATION
ACTIVITY

4.3.4: Child and multiple selectors.



Modify or add a CSS rule for each requirement below to change the `background-color` property as specified.

1. Add a rule to change the background color for `` elements to be `magenta`.
2. Modify the `ol li` selector so the yellow background color applies only to `` elements that are direct children of `` elements.

3. Add a rule to change the background color to **cyan** for **** elements that are descendants of **** elements.
4. Add a rule to change the background color to **cyan** for **** elements that are descendants of **** elements.
5. Combine the previous two rules into a single rule using a multiple selector.

HTML CSS

```
1 <ol>
2 <li>The <em><span>Muppets</span></em> were created in 1955.</li>
3 <li>The first two <span><em>Muppets</em></span> performers were
4   <li>Jim <span>Henson</span></li>
5   <li>Jane <span>Henson</span></li>
6 </ul></li>
7 <li>The <span>Muppets</span> were also in comic strips and movies.
8 </ol>
9
10 <ul>
11 <li>The <em><span>Muppets</span></em> were created in 1955.</li>
12 </ul>
13
14 <p>Every <span>"Muppets"</span> and <span>"Henson"</span> word is colored cyan except in this sentence.
```

Render webpage

Reset code

Your webpage

1. The *Muppets* were created in 1955.
 2. The first two *Muppets* performers were married.
 - Jim Henson
 - Jane Henson
 3. The Muppets were also in comic strips and movies.
- The *Muppets* were created in 1955.
- Every "Muppets" and "Henson" word is colored cyan except in this sentence.

Expected webpage

1. The *Muppets* were created in 1955.
 2. The first two *Muppets* performers were married.
 - Jim Henson
 - Jane Henson
 3. The Muppets were also in comic strips and movies.
- The *Muppets* were created in 1955.
- Every "Muppets" and "Henson" word is colored cyan except in this sentence.

▼ View solution

Explain

--- START FILE: HTML ---

```
<ol>
<li>The <em><span>Muppets</span></em> were created in 1955.
</li>
<li>The first two <span><em>Muppets</em></span> performers
were married.<ul>
    <li>Jim <span>Henson</span></li>
    <li>Jane <span>Henson</span></li>
</ul></li>
<li>The <span>Muppets</span> were also in comic strips and
movies.</li>
</ol>

<ul>
<li>The <em><span>Muppets</span></em> were created in 1955.
</li>
</ul>

<p>Every <span>"Muppets"</span> and <span>"Henson"</span>
word is colored cyan except in this sentence.</p>
```

--- END FILE: HTML ---

--- START FILE: CSS ---

```
ul {
    background: magenta;
}
ol > li {
    background: yellow;
}
ol span, ul span {
    background: cyan;
}
```

--- END FILE: CSS ---

[Feedback?](#)

Sibling selectors

Sibling elements are elements that share the same parent element. The **general sibling selector**, specified using a tilde character (~) between two selectors, matches the second element if the second element occurs after the first element and both elements are siblings. Any number of other elements can be placed between two general sibling elements. In the

example below, the heading and the paragraphs are children of the section element. Consequently, the two paragraphs are both general siblings of the heading element.

Figure 4.3.2: General sibling selector.

HTML and CSS	Render
<pre> <style> h1 ~ p { border-top: 1px solid gray; } </style> <section> <h1>Classifieds</h1> <p>Dog for adoption. Call 555-1234. </p> <p>Guard dog wanted. Call 555-4321. </p> </section> </pre>	<p>Classifieds</p> <hr/> <p>Dog for adoption. Call 555-1234.</p> <hr/> <p>Guard dog wanted. Call 555-4321.</p>

[Feedback?](#)

The **adjacent sibling selector**, specified using a plus character (+) between two selectors, matches an element that immediately follows another element, where both elements have the same parent. Ex: The adjacent selector `h1 + p` in the figure below matches the first paragraph immediately following the `<h1>` header element, where both the paragraph and heading share the same section element parent.

Figure 4.3.3: Adjacent sibling selector.

HTML and CSS	Render

```
<style>
h1 + p {
    border-top: 1px solid
orange;
}
</style>

<section>
    <h1>My Book Report</h1>
    <p>I thoroughly enjoyed
this book. The story made me
    laugh and cry and jump for
joy.</p>
    <p>I recommend this book to
anyone who loves to read.</p>
</section>
```

My Book Report

I thoroughly enjoyed this book. The story made me laugh and cry and jump for joy.

I recommend this book to anyone who loves to read.

[Feedback?](#)

Combinators.

Combinators are CSS selectors that match specific relationships between other selectors. The descendant, child, adjacent sibling, and general sibling selectors are all combinators.

PARTICIPATION ACTIVITY

4.3.5: Identify CSS selector names.



Identify which CSS selector is used in each question.

1) li ~ img

- Universal selector
- Multiple selector
- Child selector
- General sibling selector
- Adjacent sibling selector

Correct

li ~ img matches any img elements that follow an li element, where the li and img elements are siblings.



2) *.hide



- Universal selector
- Multiple selector
- Child selector
- General sibling selector
- Adjacent sibling selector

Correct

The universal selector uses an asterisk (*) to match any tag name.

3) h1 + p



- Universal selector
- Multiple selector
- Child selector
- General sibling selector
- Adjacent sibling selector

Correct

The adjacent sibling selector consists of a selector, the plus character (+), and another selector. `h1 + p` matches the first p element following an h1 element, where the p and h1 elements are siblings.

4) p > img



- Universal selector
- Multiple selector
- Child selector
- General sibling selector
- Adjacent sibling selector

Correct

`p > img` matches any img element that is contained directly inside a p element.

Attribute selector

The **attribute selector**, specified with an attribute name and optional value comparison enclosed in square brackets ([and]), matches elements with the specified attribute or the specified attribute and value. Ex: `a[target]` selector matches anchor elements with a `target` attribute specified. The attribute selector can be more specific by matching

[Feedback?](#)

elements with attributes having a specific value. Ex: `a[target="_blank"]` attribute selector matches anchor elements with a target attribute value of `_blank`.

Table 4.3.1: Common attribute selector comparators.

Comparator	Matches element when	CSS example	Matching examples
=	Attribute has exact value	<code>[target="_blank"]</code>	<code></code>
~=	Attribute contains whole word	<code>[alt~="sad"]</code>	<code></code>
^=	Attribute begins with value	<code>[class^="nav"]</code>	<code><ul class="nav-level0"></code> <code><ul class="navigation-item"></code>

[Feedback?](#)

PARTICIPATION ACTIVITY

4.3.6: Attribute selectors.



Complete the CSS attribute selectors to implement the specified behavior.

- 1) Set the text color for input buttons of type submit.

```
<button
  type="submit">Submit</button>
button[ type = "submit" ] {
  color: green;
}
```

[Check](#)

[Show answer](#)

Correct

`type="submit"`

Selects all buttons where the attribute type is submit. The attribute selector can be used to match input elements that have specific type attributes





- 2) Set the input area width for any text input elements to 300 pixels.

```
<input type="text"
      name="firstName">

input [ type="text" ] {
  width: 300px;
}
```

Correct`type="text"`

Selects all input elements where the attribute type is text.

Check**Show answer**

- 3) Set the text color to blue for any links where the `rel` attribute contains `nofollow`.

```
<a rel="abstract nofollow"
     href="https://example.com">

a [ rel~="nofollow" ] {
  color: blue;
}
```

Correct`rel~="nofollow"`

Selects `a` elements if the word `nofollow` appears anywhere within the `rel` attribute.

Check**Show answer**

- 4) Create a border for images where the `alt` attribute starts with `test`.

```


img [ alt ^= "test" ] {
  border: red 1px
  solid;
}
```

Correct`alt^="test"`

Selects `img` elements if the word `test` appears at the beginning of the `alt` attribute.

Check**Show answer****Feedback?**

Pseudo-element selector

The **pseudo-element selector**, specified with two colon characters (`::`) followed by a pseudo-element, matches parts of elements. The pseudo-element selectors allow styles to apply to the first line or first letter of text of an element or to text that is selected by the user, or allow additional content to be inserted before or after an element.

Table 4.3.2: Common pseudo-element selectors.

Pseudo-element	Explanation	CSS example
::after	Add content after the matched element.	<code>li::after { content: "<" }</code>
::before	Add content before the matched element.	<code>li::before { content: "***" }</code>
::first-line	Match the first line of text in a block element.	<code>p::first-line { color: red }</code>
::first-letter	Match the first letter of text in a block element.	<code>p::first-letter { font-size: 200% }</code>
::selection	Matches the text selected by user.	<code>::selection { background: yellow }</code>

[Feedback?](#)**PARTICIPATION ACTIVITY****4.3.7: Pseudo-element selectors.**

Complete the CSS pseudo-element selectors to implement the specified behavior.

- 1) Change the text color to green for any text that is currently selected using the mouse.

```
::selection // {
    color: green;
}
```

Correct`::selection`

Matches all text that the user has selected.

Check**Show answer**

- 2) Make the first letter uppercase in all list items.

Correct

```
<li>Bread</li>
```

```
li::first-letter {
    text-transform: uppercase;
}
```

Check**Show answer**

```
li::first-letter
```

`::first-letter` matches the first letter in the block element `li`.

- 3) Underline the first line in each blockquote.

```
<blockquote>To be or
not to be...
</blockquote>
```

```
blockquote::first-line {
    text-decoration: underline;
}
```

Correct

```
blockquote::first-line
```

`::first-line` matches the first line in the block element `blockquote`.

Check**Show answer**

- 4) Add a red asterisk at the end of each paragraph.

```
<p>Turn the switch on.
</p>
```

```
p::after {
    content: "*";
    color: red;
}
```

Correct

```
p::after
```

`::after` adds content after the element `p`.

Check**Show answer****Feedback?**
CHALLENGE ACTIVITY
4.3.1: Advanced selectors.


530096.4000608.qx3zqy7

[Jump to level 1](#)

1



2



3



4

Use attribute selection to apply the below rules to all `` tags with an alt containing the word "one". **SHOW EXPECTED**

CSS**HTML**

```
1 /* Your solution goes here */
2 img[alt~="one"]{
```

3 }

1

2

3

4

Check

Next



Testing opacity of first tag

Yours

0 . 3

Testing opacity of second tag

Yours

0 . 3

Testing opacity of third tag

Yours

1

Testing opacity of fourth tag

Yours

1

Testing opacity of fifth tag

Yours

1

Testing opacity of sixth tag

Yours

1

Your webpage

View your last submission 

```
/* Your solution goes here */  
img[alt~="be"]
```

[Feedback?](#)

Exploring further:

- [CSS selector reference](#) from W3 Consortium.

How was
this
section?



[Provide section feedback](#)

