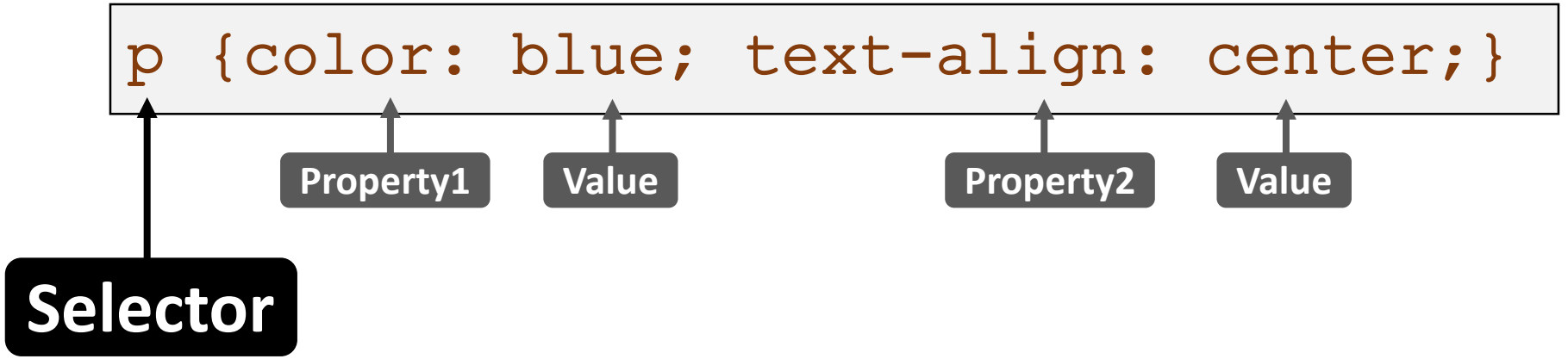
Lesson 2.3

By: Abdullah Abbas

CSS Selectors

How to select elements on which we want to apply styles?

What is a Selector?



It defines on which elements the styles will be applied.

We have already seen the following selectors:

Selector	Example	Example Description
*	*	The **Universal Selector**, selects all elements
element	p	Selects all <p> elements
.class	.intro	Selects all elements with class="intro"
#id	#firstname	Selects the element with id="firstname"

Example

Select All (*)

Example: The borders are applied to all the elements including <html> and <body>.

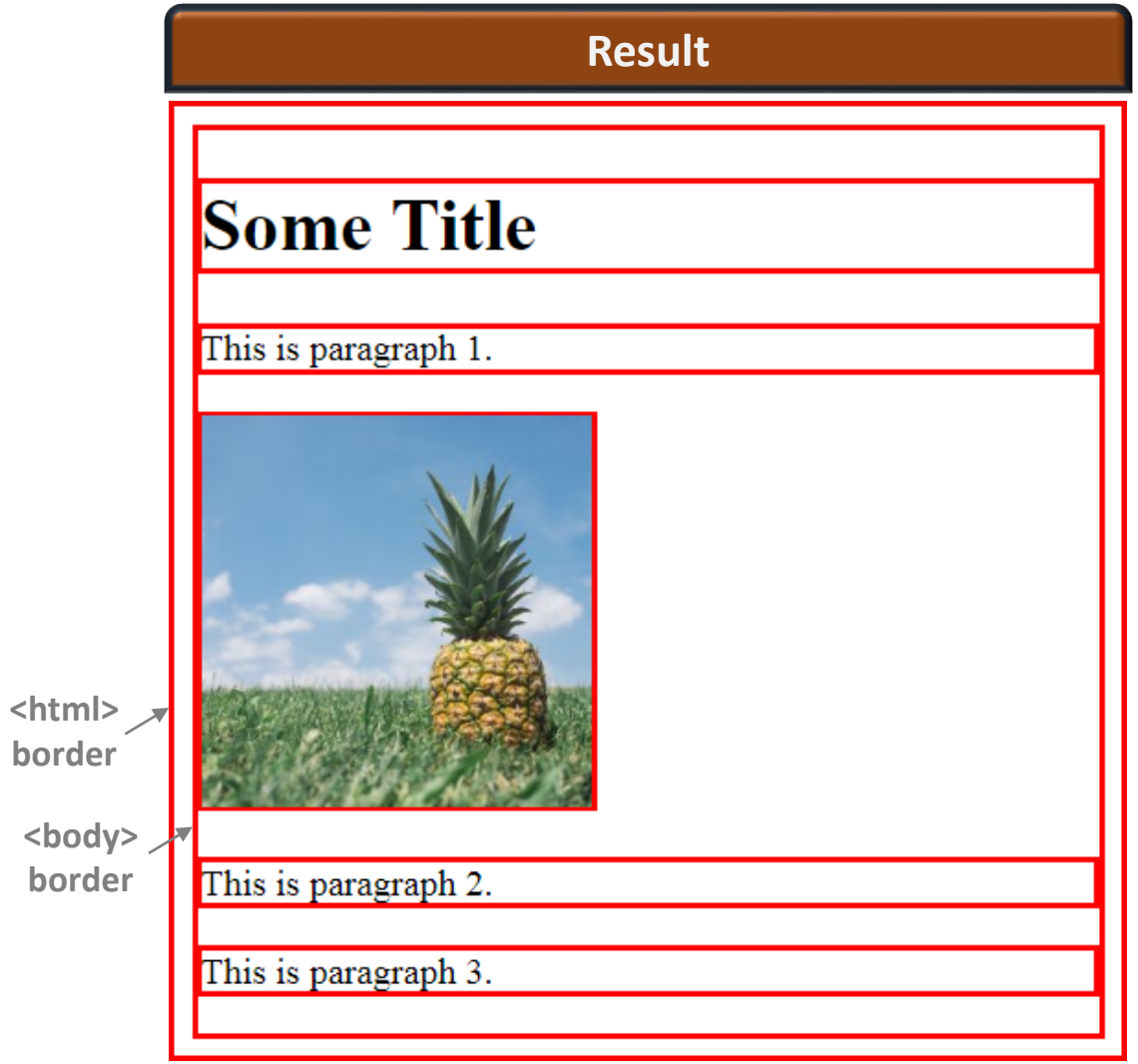
HTML

```
<html>
  <body>
    <h1>Some Title</h1>
    <p>This is paragraph 1.</p>
    <img src="pineapple.jpg">
    <p>This is paragraph 2.</p>
    <p>This is paragraph 3.</p>
  </body>
</html>
```

CSS

```
* {
 border: 2px solid red;
}
```

Result



Note:
The spaces between borders is due to default margins.

Example

Select All (*)

Example: The borders are applied to all the elements including <html> and <body>.

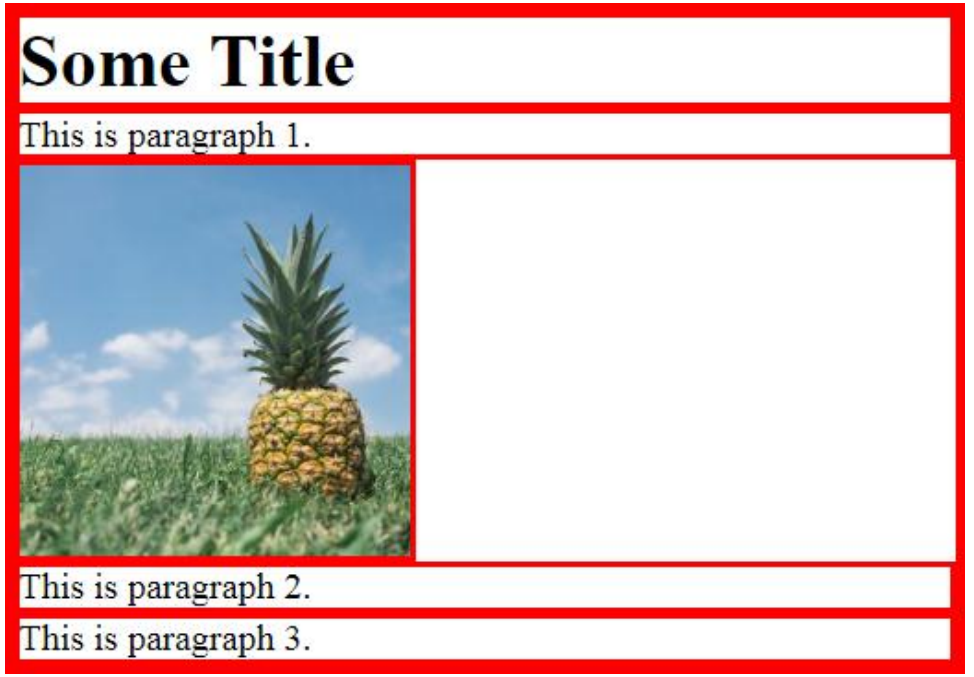
HTML

```
<html>
  <body>
    <h1>Some Title</h1>
    <p>This is paragraph 1.</p>
    <img src="pineapple.jpg">
    <p>This is paragraph 2.</p>
    <p>This is paragraph 3.</p>
  </body>
</html>
```

CSS

```
* {
 border: 2px solid red;
 margin: 0;
}
```

Result



Now we also set margins to 0. The spaces between borders collapsed.

Example

Element Selector

Example: Select all <p> elements.

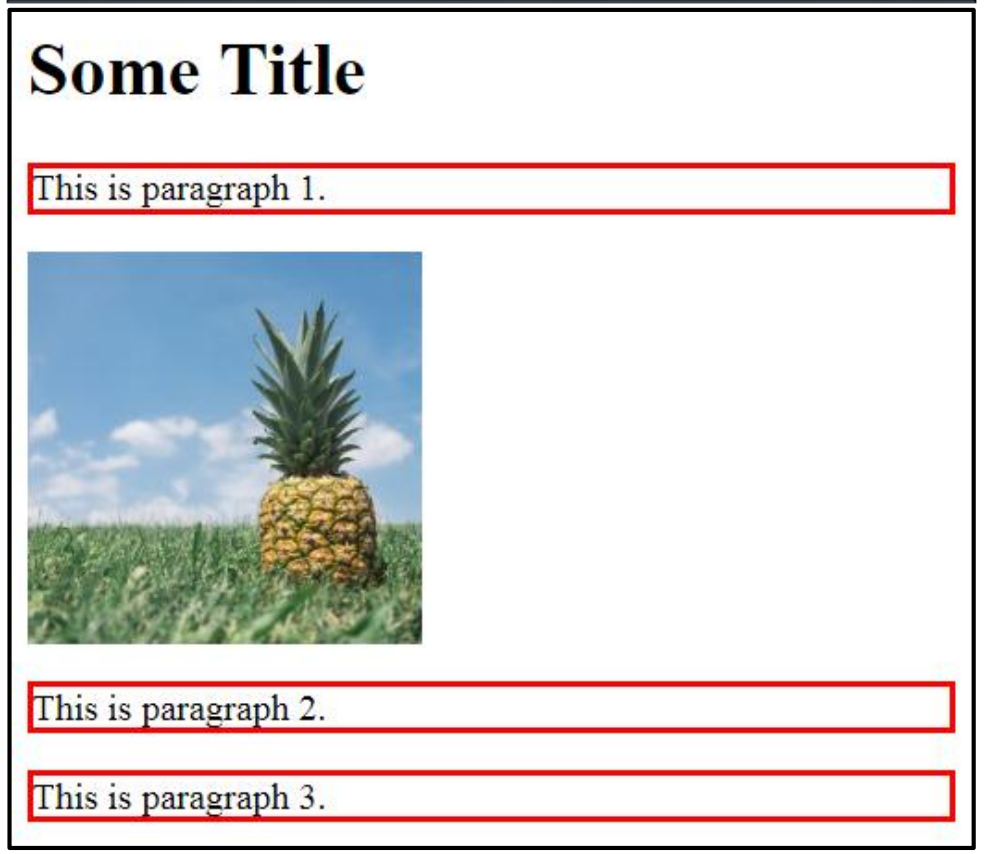
HTML

```
<h1>Some Title</h1>
<p>This is paragraph 1.</p>
<img src="pineapple.jpg">
<p>This is paragraph 2.</p>
<p>This is paragraph 3.</p>
```

CSS

```
p {
 border: 2px solid red;
}
```

Result



Example

Class Selector

Example: We added a **class="top"** attribute to <h1> and first <p>.

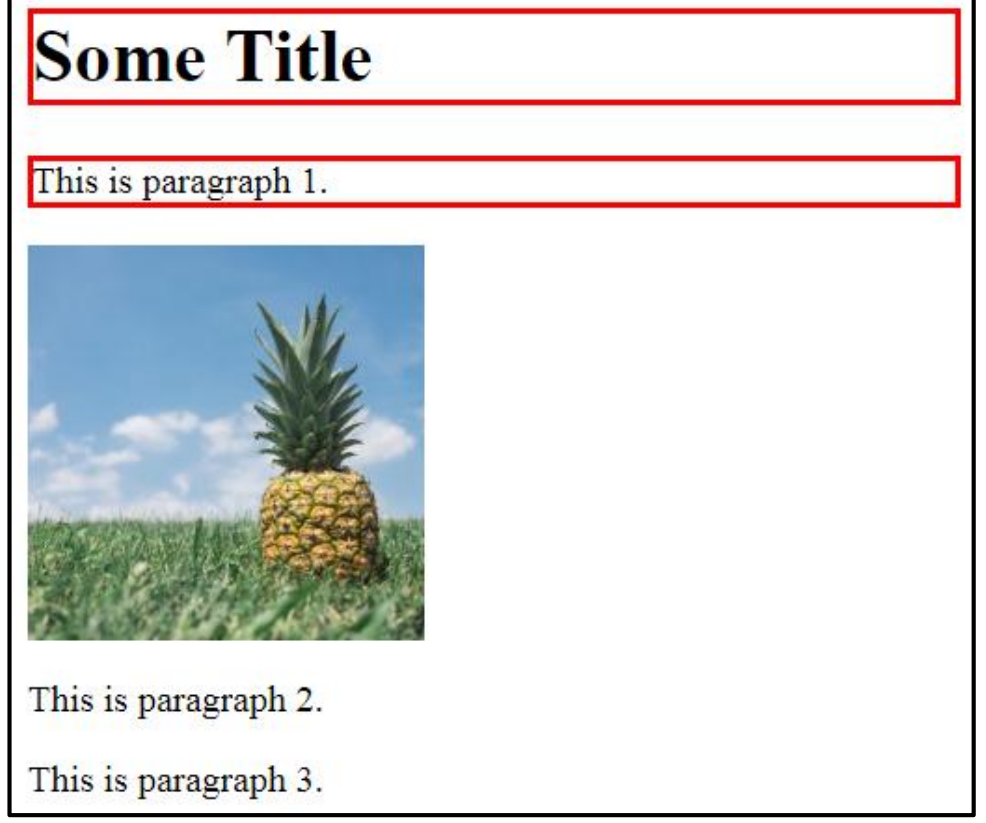
HTML

```
<h1 class="top">Some Title</h1>
<p class="top">This is paragraph 1.</p>
<img src="pineapple.jpg">
<p>This is paragraph 2.</p>
<p>This is paragraph 3.</p>
```

CSS

```
.top {
 border: 2px solid red;
}
```

Result



Example

ID Selector

Example: We added an **id="important"** attribute to one of the <p> elements. It uniquely identifies an element.

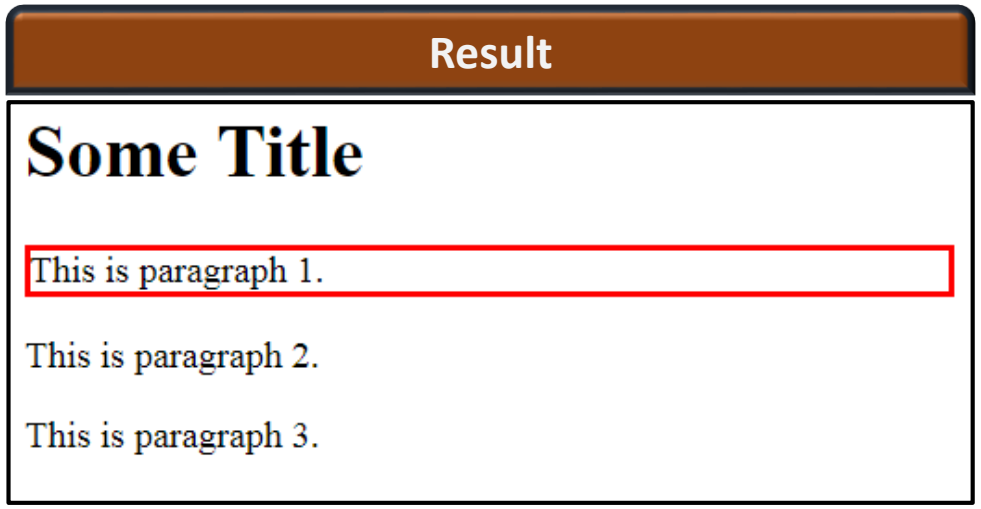
HTML

```
<h1>Some Title</h1>
<p id="important">This is paragraph 1.</p>
<p>This is paragraph 2.</p>
<p>This is paragraph 3.</p>
```

CSS

```
#important {
 border: 2px solid red;
}
```

Result



Comma Separated Selectors

Selector	Description
selector1, selector2, selector3, …	Selects all elements from **selector1**, and all elements from **selector2**, and all elements from **selector3** …

It is used to write the same CSS style for different selectors.

Examples

- **p, h1**
  Applies to all <p> and <h1> elements.
- **p, .class1**
  Applies to all <p> elements, and to all elements with class "class1".
- **.class1, .class2**
  Applies to all elements with class "class1", and to all elements with class "class2".

Example

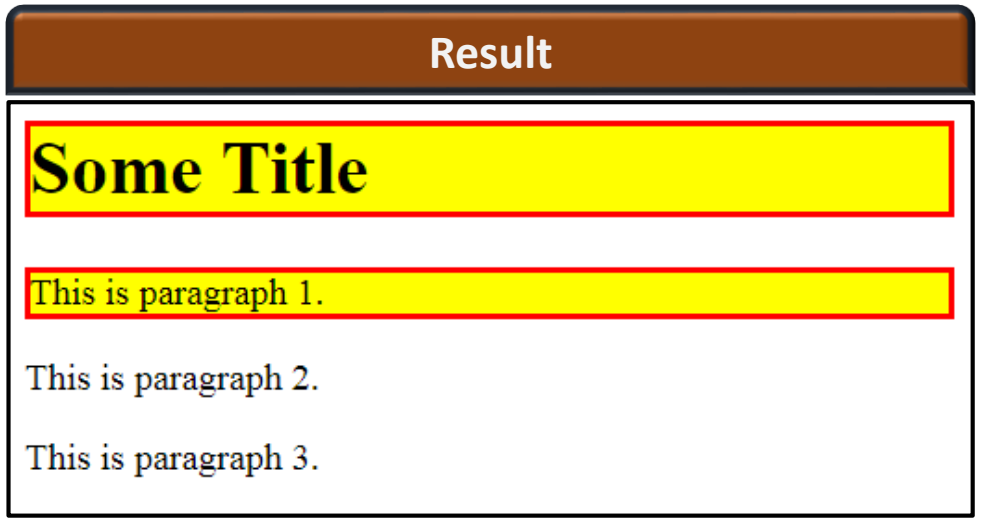
HTML

```
<h1>Some Title</h1>
<p id="important">This is paragraph 1.</p>
<p>This is paragraph 2.</p>
<p>This is paragraph 3.</p>
```

CSS

```
h1, #important {
 border: 2px solid red;
 background-color: yellow;
}
```

Result



Classified Element

Selector	Example	Example Description
element.class	**p.intro**	Selects all <p> elements with **intro** class

Example

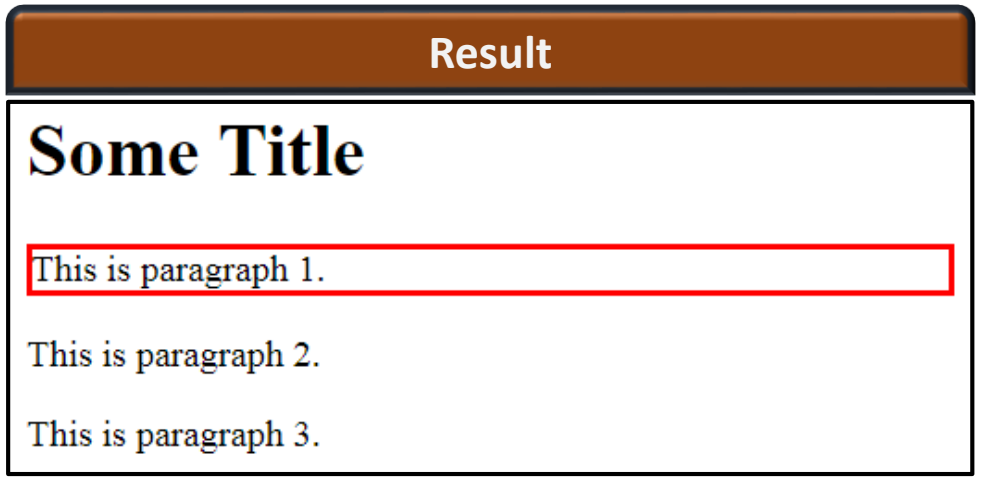
HTML

```
<h1 class="top">Some Title</h1>
<p class="top">This is paragraph 1.</p>
<p>This is paragraph 2.</p>
<p>This is paragraph 3.</p>
```

CSS

```
p.top {
 border: 2px solid red;
}
```

Result



Multiple Classes

Selector	Example	Example Description
.class1.class2	**.foo.bar**	Selects elements each of which belong to both **foo** and **bar** classes

In order to assign an element multiple classes we separate them with spaces:

```
<p class="foo bar abc">Some Text.</p>
```

In this example the element <p> belongs to three classes: **foo**, **bar**, and **abc**. Thus it will be selected by **.foo.bar** since it has these two classes.

Example

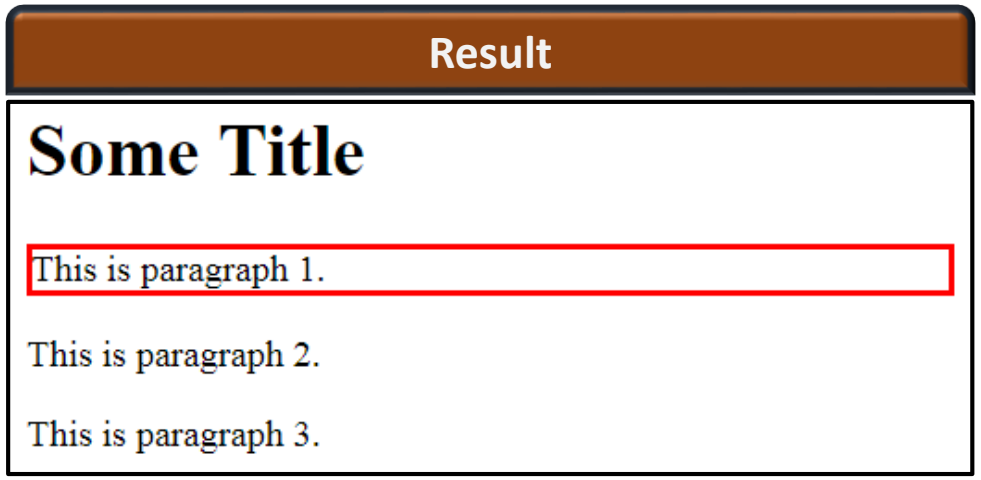
HTML

```
<h1 class="red">Some Title</h1>
<p class="bordered red">This is paragraph 1.</p>
<p>This is paragraph 2.</p>
<p>This is paragraph 3.</p>
```

CSS

```
.bordered.red {
 border: 2px solid red;
}
```

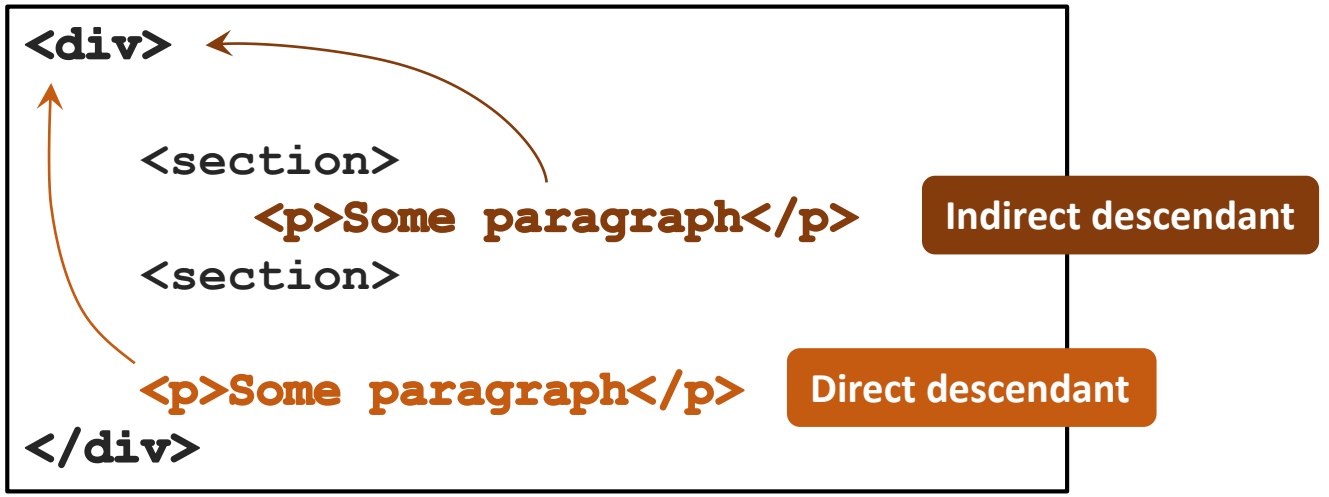
Result



Descendant Elements

Selector	Example	Example Description
element element	**div p**	Selects all <p> elements inside elements

It selects both direct and indirect descendants.



Example

HTML

```
<h1>Title</h1>
<p>This is paragraph 1.</p>

<div>
  <h2>Sub-title</h2>
  <p>This is paragraph 2.</p>
</div>
```

CSS

```
div p {
 border: 2px solid red;
 background-color: yellow;
}
```

Result

Title

This is paragraph 1.

Sub-title

This is paragraph 2.

Example

CSS selectors can be combined in several ways.
In the following example we select two different types of descendants.

HTML

```
<article>
  <h1>Article Title</h1>
  <p>This is a paragraph.</p>
</article>

<footer>
  <h1>Footer Title</h1>
  <p>This is a footer.</p>
</footer>
```

CSS

```
article h1,
footer p
{
 border: 2px solid red;
}
```

Result

This CSS example selects both:

- **The <h1> descendants of the <article>**
- **The <p> descendants of the <footer>**

CSS Pseudo-Classes

a:link, a:visited, a:hover, a:active, …

What is a Pseudo-Class?

A pseudo-class is used to define a special state of an element, and then we can give styles to this particular state.

Example

For example we can define a style for a link **<a>Google** if it is **unvisited**, and another style for the same link when it becomes **visited**.

You probably have seen the default behavior in most browsers.
Unvisited: Google
Visited: Google

We can change this behavior!

How is It Used?

If we write the following CSS code:

```
a { color: red; }
```

The style will be applied on all the states of the link.

To style a specific state we put a pseudo-class after the selector:

```
a:link { color: red; }
a:visited { color: black; }
```

Link Pseudo-Classes

Selector	Example	Example Description
:link	**a:link**	Selects all unvisited links
:visited	**a:visited**	Selects all visited links
:hover	**a:hover**	Selects links on mouse over
:active	**a:active**	Selects the active link (i.e. during mouse click)

Example

HTML

```
<a href="http://google.com">Google</a>
```

CSS

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

a:visited { color: green; }

a:hover { color: pink; }

a:active { color: blue; }
```

Result: (Unvisited State)

Google

Result: (Hover State)

Google

Result: (Active State)

Google

Result: (Visited State)

Google

Order of Pseudo-Classes

Order sometimes matters:

a:hover MUST come after **a:link** and **a:visited** in the CSS definition in order to be effective.

a:active MUST come after **a:hover** in the CSS definition in order to be effective.

Why?

Because multiple states can occur at the same time. For example when we are clicking on a link, the mouse would be hovering over it at the same time. In this case, which ever is written last overrides the previous one.

Can You Click on Elements?

Pseudo-classes are not only for links. For example :active and :hover can be used with any element to change its style at these states.

Examples

- **div:hover**
  Applies to all elements on mouse over.
- **.foo:hover**
  Applies to all elements with class "foo" on mouse over.
- **a.foo:hover**
  Applies to all <a> elements with class "foo" on mouse over.
- **#important:active**
  Applies to the element with id "important" in its active state (i.e. during mouse click).

Example

When you click the box it will change style !

HTML

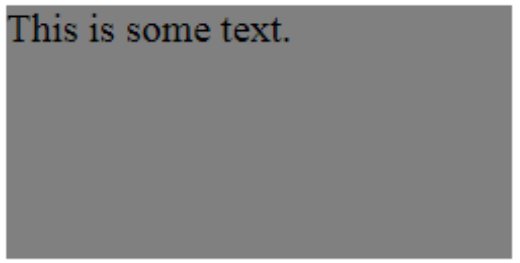
```
<div>This is some text.</div>
```

CSS

```
div {
  background-color: gray;
  width: 200px;
  height: 100px;
}

div:active {
  color: red;
  background-color: yellow;
  border: 5px solid red;
}
```

Result (Default State)



Result (Active State)

CSS Attribute Selector

Attribute Selector

Selector	Example	Example Description
[attribute=value]	**[**name=gender**]**	Selects all elements with **name="gender"** attribute
	input[name=gender**]**	Selects <input> elements with **name="gender"** attribute

Example

The following example styles text fields and submit buttons based on the type attribute.

HTML

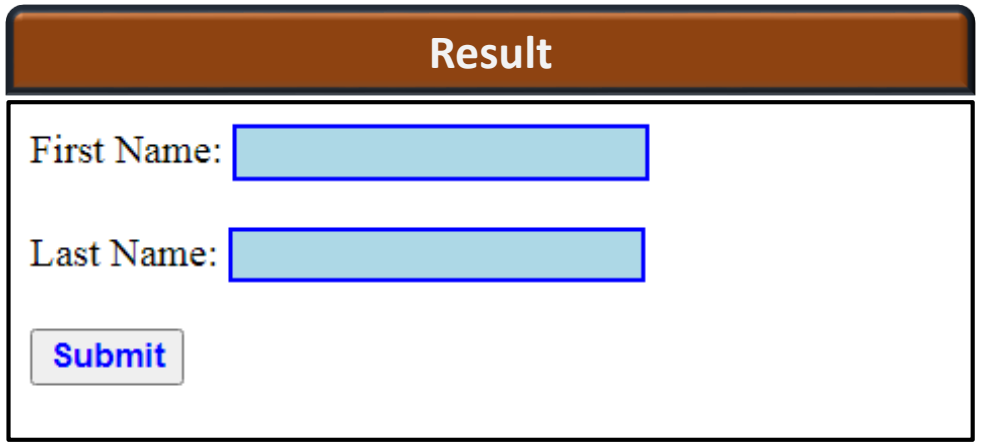
```
<form action="page.php">
  First Name: <input type="text"> <br><br>
  Last Name: <input type="text"> <br><br>
  <input type="submit">
</form>
```

CSS

```
input[type=text] {
  border: 2px solid blue;
  background-color: lightblue;
}

input[type=submit] {
  color: blue;
  font-weight: bold;
}
```

Result



Appendix

Other Selectors

Selector	Example	Example Description
element element	**div p**	Selects all <p> elements inside elements
element>element	**div > p**	Selects all <p> elements which are direct descendants of elements
element+element	**div + p**	Selects every <p> element that is placed immediately after a element
element1~element2	**div ~ p**	Selects every <p> element that is placed after a element

Other Pseudo-Classes

Selector	Example	Example Description
:checked	**input:checked**	Selects every checked <input> element
:disabled	**input:disabled**	Selects every disabled <input> element
:empty	**p:empty**	Selects every <p> element that has no content (neither a space)
:enabled	**input:enabled**	Selects every enabled <input> element
:focus	**input:focus**	Selects the <input> element that has focus
:invalid	**input:invalid**	Selects all <input> elements with an invalid value
:not(selector)	**:not(p)**	Selects every element that is not a <p> element
:optional	**input:optional**	Selects <input> elements with no "required" attribute
:read-only	**input:read-only**	Selects <input> elements with a "readonly" attribute specified
:read-write	**input:read-write**	Selects <input> elements with no "readonly" attribute
:required	**input:required**	Selects <input> elements with a "required" attribute specified
:target	**#news:target**	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	**input:valid**	Selects all <input> elements with a valid value

Other Attribute Selectors

Selector	Example	Example Description
[attribute]	**[checked]**	Selects all elements with a **checked** attribute
[attribute=value]	**[target=_blank]**	Selects all elements with **target="_blank"** attribute
[attribute~=value]	**[title~=flower]**	Selects all elements with a **title** attribute containing the word **"flower"**
[attribute|=value]	**[lang|=en]**	Selects all elements with a **lang** attribute value starting with **"en"**, optionally followed by a hyphen and another string like **"en-us"**
[attribute^=value]	**a[href^="https"]**	Selects every <a> element whose **href** attribute value begins with **"https"**
[attribute$=value]	**a[href$=".pdf"]**	Selects every <a> element whose **href** attribute value ends with **".pdf"**
[attribute*=value]	**a[href*="google"]**	Selects every <a> element whose **href** attribute value contains the substring **"google"**