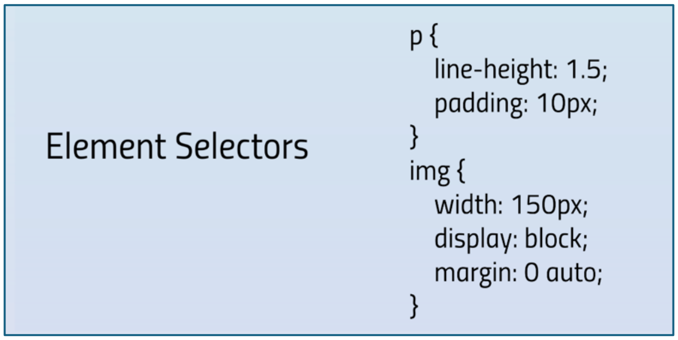
CSS Selectors

In a CSS file, selectors are used to represent parts of your HTML so that you can target specific elements and apply style rules to them.

You can target elements by their **tag names**.

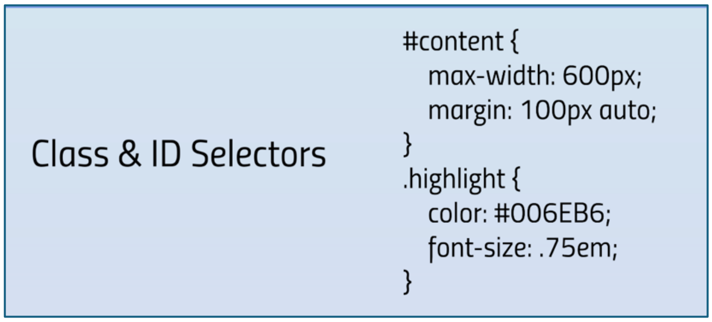


Element Selectors

```
p {
   line-height: 1.5;
   padding: 10px;
}
img {
   width: 150px;
   display: block;
   margin: 0 auto;
}
```

You can also use **class** and **ID selectors** to target specific elements. Only elements with a matching id or class will be styled.

For example, the id of content and the class highlight are being targeted in the example.



Class & ID Selectors

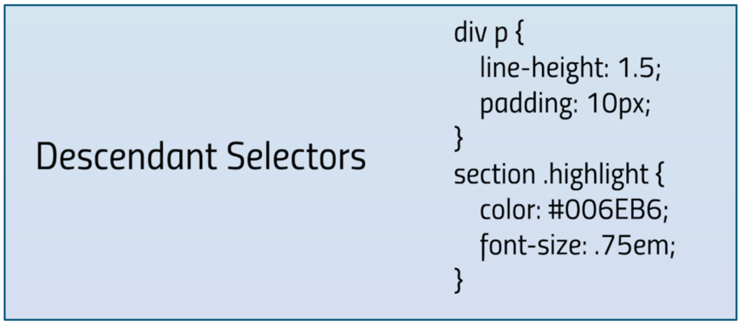
```
#content {
   max-width: 600px;
   margin: 100px auto;
}
.highlight {
   color: #006EB6;
   font-size: .75em;
}
```

Descendant Selectors

Descendant selectors target elements **within** other elements.
For instance:

- Paragraphs inside a <div>
- Elements with the class highlight inside a <section>

Only those nested elements will be affected.

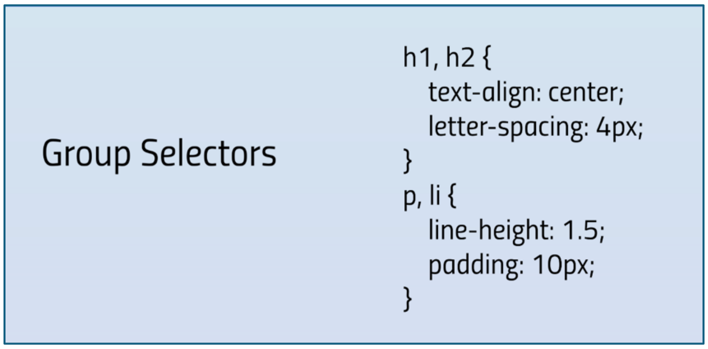


Descendant Selectors

```
div p {
   line-height: 1.5;
   padding: 10px;
}
section .highlight {
   color: #006EB6;
   font-size: .75em;
}
```

Group Selectors

With **group selectors**, you can target multiple elements at once by separating them with commas.
For example, all <h1>, <h2>, <p>, and <li> elements can be grouped and styled together.



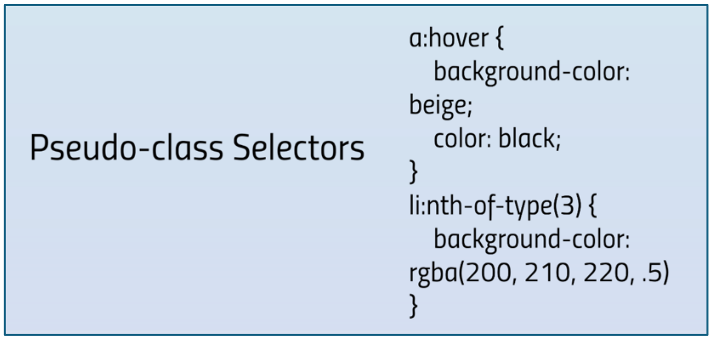
Group Selectors

```
h1, h2 {
   text-align: center;
   letter-spacing: 4px;
}
p, li {
   line-height: 1.5;
   padding: 10px;
}
```

Pseudo-Class Selectors

Pseudo-class selectors let you target elements in specific states or positions.
For example:

- a:hover targets only anchor tags that are being hovered over.
- li:nth-of-type(3) targets the **third** list item of its type.



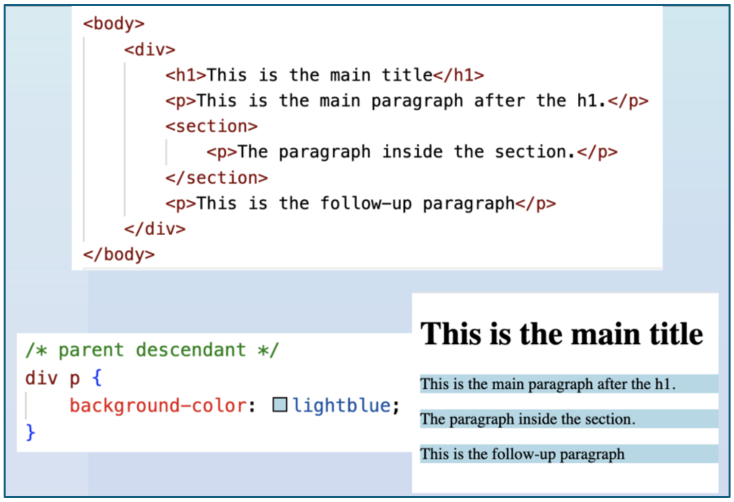
Pseudo-class Selectors

```
a:hover {
   background-color:
beige;
   color: black;
}
li:nth-of-type(3) {
   background-color:
rgba(200, 210, 220, .5)
}
```

Descendant Combinators: A Closer Look

Let’s take a closer look at **descendant combinators**.

div p



```
<body>
    <div>
        <h1>This is the main title</h1>
        <p>This is the main paragraph after the h1.</p>
        <section>
            <p>The paragraph inside the section.</p>
        </section>
        <p>This is the follow-up paragraph</p>
    </div>
</body>
```

```
/* parent descendant */
div p {
    background-color: lightblue;
}
```

This is the main title

This is the main paragraph after the h1.

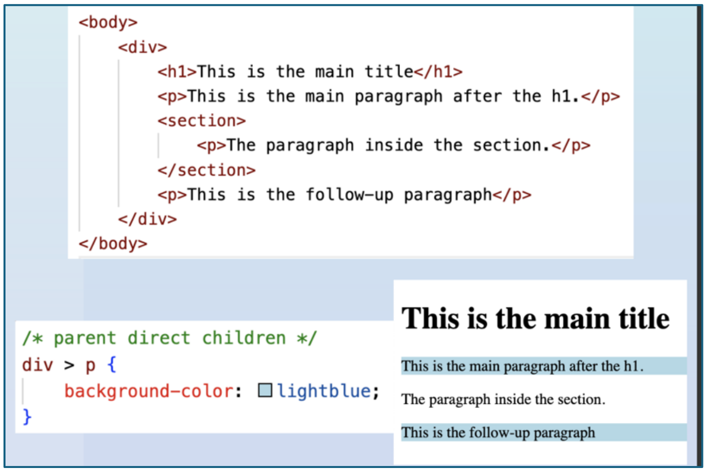
The paragraph inside the section.

This is the follow-up paragraph

This targets all `<p>` elements **inside** a `<div>` (not just direct children—any level of nesting).

Let’s look at some you many have not seen. Here we see the selector:

div > p



```
<body>
    <div>
        <h1>This is the main title</h1>
        <p>This is the main paragraph after the h1.</p>
        <section>
            <p>The paragraph inside the section.</p>
        </section>
        <p>This is the follow-up paragraph</p>
    </div>
</body>
```

```
/* parent direct children */
div > p {
    background-color: lightblue;
}
```

This is the main title

This is the main paragraph after the h1.

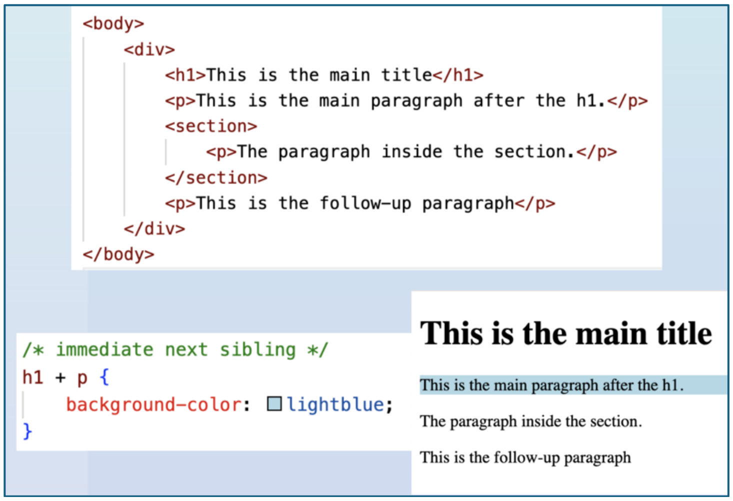
The paragraph inside the section.

This is the follow-up paragraph

This targets only <p> elements that are **direct children** of a <div>.
Notice that a <p> inside a nested <section> wouldn't be selected because it's a **grandchild**, not a direct child.

Here we see the selector:

h1 + p



```
<body>
    <div>
        <h1>This is the main title</h1>
        <p>This is the main paragraph after the h1.</p>
        <section>
            <p>The paragraph inside the section.</p>
        </section>
        <p>This is the follow-up paragraph</p>
    </div>
</body>
```

```
/* immediate next sibling */
h1 + p {
    background-color: □lightblue;
}
```

This is the main title

This is the main paragraph after the h1.

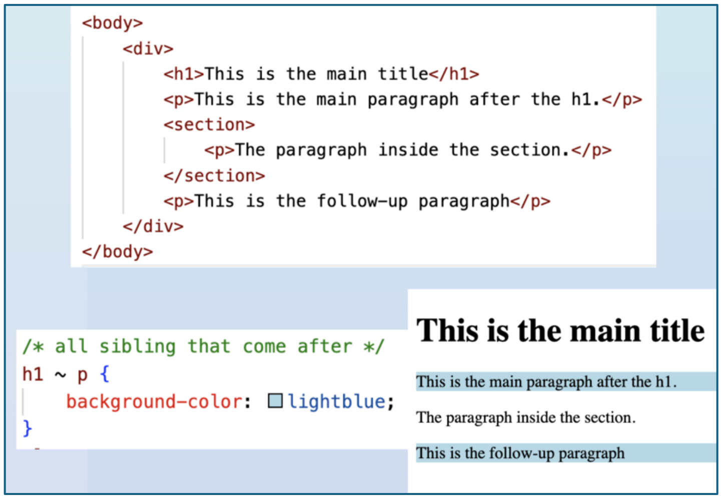
The paragraph inside the section.

This is the follow-up paragraph

This targets the **immediate sibling**—the first <p> that comes **right after** an <h1>.

Here we see the selector:

h1 ~ p



```
<body>
    <div>
        <h1>This is the main title</h1>
        <p>This is the main paragraph after the h1.</p>
        <section>
            <p>The paragraph inside the section.</p>
        </section>
        <p>This is the follow-up paragraph</p>
    </div>
</body>
```

```
/* all sibling that come after */
h1 ~ p {
    background-color: □lightblue;
}
```

This is the main title

This is the main paragraph after the h1.

The paragraph inside the section.

This is the follow-up paragraph

This targets **all sibling <p> elements** that come **after** an <h1>.

More Pseudo-Class Selectors

Here are some additional pseudo-class selectors:

- p:first-child — targets a <p> only if it is the **first child** of its parent.
- p:last-child — targets a <p> only if it is the **last child**.
- li:nth-child(odd) — targets all **odd-numbered** <li> elements (1st, 3rd, 5th, etc).
- input:focus — targets an <input> element **only when it is focused**.

There are many more!
For a full reference, check out:
🔗 https://www.w3schools.com/cssref/css_ref_pseudo_classes.php

CSS Attribute Selectors

Attribute selectors let you target elements based on their attributes in the HTML.

Examples:

- a[target="_blank"] — targets all <a> tags with a target attribute set to _blank.
- input[type="text"] — targets only <input> elements where the type attribute is "text".