

Web pages don't look very interesting when they only contain HTML elements. That's why we use CSS to create appealing websites that captivate people. So how can we add style to a webpage? Using **contextual selectors** is a great option. This topic will teach you how to utilize contextual selectors when applying design to your HTML elements. Let's get started!

## Contextual selectors in CSS

As explained in the introduction, we can use contextual selectors to add style/design to our web pages — they allow us to apply specific properties to our HTML elements.

There are four different types of contextual selectors:

1. The **indirect descendant selector**.
2. The **direct descendant selector**.
3. The **adjacent sibling selector**.
4. And the **general sibling selector**.

In the following sections, you will learn about the differences between these selector types and when to use them.

### Indirect descendant selector

When we want to apply a style to an element that is inside another element, we use an indirect descendant selector. Take a look at the code snippet below:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>Example</title>
</head>
<body>
  <h2>The heading</h2>
  <p>The element b is used for bold text. Here is an example
<b>Bold</b></p>
</body>
</html>
```

Let's try to change the font color of the `<b>` element. Since the `<b>` element is inside the `<p>` element, we can access it by combining the two elements with a space. The way to do this is shown in the following example:

```
p b {
  color: red;
}
```

Now your code should output text like this:

## The heading

The element `b` is used for bold text. Here is an example **Bold**

### Direct descendant selector

If you look at the example near the start of the previous section again, you'll see that the `<h2>` element is a direct child of the `<body>` element. In this case, if we want to change the `<h2>` element, we can use the `>` selector (also known as a **combinator**) to combine the parent and child elements. This type of selector is called a direct descendant selector:

```
body > h2 {  
    color: red;  
}
```

The change made by applying this selector is shown below:

## The heading

The element `b` is used for bold text. Here is an example **Bold**

### Adjacent sibling selector

The `+` combinator is used to apply a change to an element that is right next to another element — it's known as the adjacent sibling selector. You can see how to use it in the following example:

```
h2 + p {  
    font-weight: bold;  
    background-color: aquamarine;  
}
```

And this is the output:

# The heading

**The element b is used for bold text. Here is an example Bold**

[ALERT-warning] If we had `<h2>` and `<p>` sibling combinations in several places, the style we have specified would be applied in each location. In addition, please be aware that the style is also applied to any elements inside the `<p>` element.[/ALERT]

## General sibling selector

Lastly, the `~` combinator is used when we want to apply a design to all the elements that are the next siblings of a specified element. It's called the general sibling selector. Looking at a longer piece of code will make it easier to understand how it can be applied:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>Example</title>
</head>
<body>
  <h2>First paragraph</h2>
  <p>Paragraph</p>
  <span>This is not a paragraph.</span>
  <span>This is also not a paragraph.</span>
  <h2>Second paragraph</h2>
  <p>Paragraph</p>
  <span>This is also not a paragraph.</span>
  <h2>End</h2>
</body>
</html>
```

The following snippet shows how to use the general sibling selector:

```
p ~ span {
  color: red;
}
```

You can see the output below:

# First paragraph

Paragraph

This is not a paragraph. This is also not a paragraph.

# Second paragraph

Paragraph

This is also not a paragraph.

# End

## Conclusion

You learned about contextual selectors in this topic. And you now know that there are four different types: the indirect descendant selector, the direct descendant selector, the adjacent sibling selector, and the general sibling selector. You've also seen examples of how they can be used. So let's put this newly acquired knowledge to the test!