# Simple selectors

Previous Overview: Introduction to CSS Next

In this, our first selectors article, we'll learn about "simple" selectors. They directly match one or more elements of a document, based on the type of element, `class`, or `id`.

## Type selectors aka element selectors

This selector is just a case-insensitive match between the selector name and a given HTML element name. This is the simplest way to target all elements of a given type. Let's take a look

at an example:

Here is some HTML:

```
<p>What color do you like?</p>
<div>I like blue.</div>
<p>I prefer BLACK!</p>
```

A simple style sheet:

```
/* All p elements are red */
p {
  color: red;
}

/* All div elements are blue */
div {
  color: blue;
}
```

And we get this:

What color do you like?

I like blue.

I prefer BLACK!

## Active learning: Selecting different elements

For this active learning, we'd like you to try changing the selector on the single CSS rule, so that different elements in the example are styled. Do you know how to write a selector to apply the ruleset to multiple elements, at once?

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the *Show solution* button to see a potential answer.

### HTML Input

```
<h1>Hello World!</h1>

<p>This is a paragraph.</p>

<ul>
  <li>This is</li>
  <li>A list</li>
</ul>
```

### CSS Input

```
h1 {
  color: red;
  text-shadow: 1px 1px 1px black;
  background: linear-gradient(to bottom, rgba(0,0,0,0.25),
rgba(0,0,0,0.1));
  padding: 3px;
  text-align: center;
  box-shadow: inset 2px 2px 5px rgba(0,0,0,0.5), inset -2px -2px 5px
rgba(255,255,255,0.5);
}
```

## Output


## Class selectors

The class selector consists of a dot, '`.`', followed by a class name. A class name is any value, without spaces, placed within an HTML `class` attribute. It is up to you to choose a name for

the class. It is also noteworthy that multiple elements in a document can have the same class value, and a single element can have multiple class names separated by white space. Here's a quick example:

## Simple example

Here is some HTML:

```
<ul>
  <li class="first done">Create an HTML document</li>
  <li class="second done">Create a CSS style sheet</li>
  <li class="third">Link them all together</li>
</ul>
```

A simple style sheet that styles two of these classes:

```
/* The element with the class "first" is bolded */
.first {
  font-weight: bold;
}

/* All the elements with the class "done" are strikethrough */
.done {
  text-decoration: line-through;
}
```

And we get this result:

- ~~**Create an HTML document**~~
- ~~Create a CSS style sheet~~
- Link them all together

## Active learning: Handling multiple classes

For this active learning, we'd like you to try changing the class attributes on the paragraph elements, so you can apply multiple separate effects. Try applying a `base-box` class plus a role class (`editor-note` or `warning`), and optionally `important,` if you want to give the box strong importance. Think about how this kind of ruleset allows you to build up a modular system of styles.

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the *Show solution* button to see a potential answer.

### HTML Input

```
<p class="">My first paragraph.</p>
<p class="">My second paragraph.</p>
<p class="">My third paragraph</p>
```

### CSS Input

```
.base-box {
  background-image: linear-gradient(to bottom, rgba(0,0,0,0.1), rgba(0,0,0,0.3));
  padding: 3px 3px 3px 7px;
}

.important {
  font-weight: bold;
}

.editor-note {
  background-color: #9999ff;
```

## Output

My first paragraph.

My second paragraph.

My third paragraph

Reset Show solution

## ID selectors

The ID selector consists of a hash/pound symbol (`#`), followed by the ID name of a given element. Any element can have a unique ID name set with the `id` attribute. It is up to you to choose an ID name. It's the most efficient way to select a single element.

> **Important**: An ID name must be unique in the document. Behaviors regarding duplicated IDs are unpredictable. For example, in some browsers, only the first instance is counted, and the rest are ignored.

Let's look at a quick example — here is some HTML:

```
<p id="polite"> – "Good morning."</p>
<p id="rude"> – "Go away!"</p>
```

A simple style sheet:

```
#polite {
  font-family: cursive;
}

#rude {
  font-family: monospace;
  text-transform: uppercase;
}
```

And we get this as an output:

— "GOOD MORNING."

— "GO AWAY!"

## Active learning: assigning the winner's colors with IDs

For this active learning, we'd like you to give the first, second and third place in the competition an appropriate gold, silver, and bronze color. You will do this by giving CSS rules 2–4 appropriate selectors, which select the relevant elements based on their ID.

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the *Show solution* button to see a potential answer.

### HTML Input

```
<p id="first"><strong>Winner</strong>: Velma Victory</p>
<p id="second"><strong>2nd</strong>: Colin Contender</p>
<p id="third"><strong>3rd</strong>: Phoebe Player</p>
```

### CSS Input

```
p {
  text-transform: uppercase;
```

```
  text-transform: uppercase;
  padding: 5px;
}

 {
  background-color: goldenrod;
}

 {
```

## Output

**WINNER**: VELMA VICTORY

**2ND**: COLIN CONTENDER

**3RD**: PHOEBE PLAYER

Reset Show solution

---

## Universal selector

The universal selector (`*`) is the ultimate joker. It allows selecting all elements on a page. As it is rarely used to apply a style to every element on a page, it is often used in combination with other selectors (see Combinators.)

**Important**: Take care when using the universal selector. As it applies to all elements, using it in large web pages can have a perceptible impact on performance: web pages display slower than expected. There are not many instances where you'd use this selector.

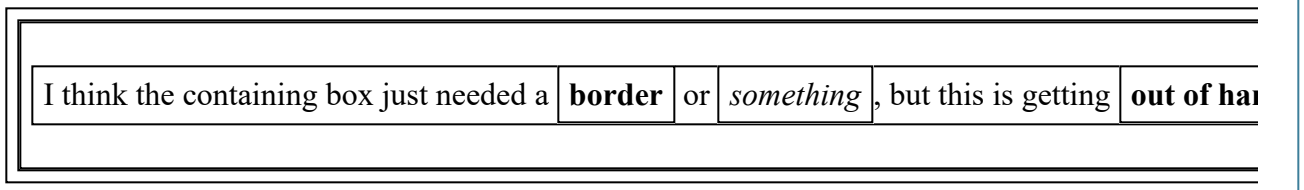
Now for an example; first some HTML:

```
<div>
  <p>I think the containing box just needed
  a <strong>border</strong> or <em>something</em>,
  but this is getting <strong>out of hand</strong>!</p>
</div>
```

And a simple style sheet:

```
* {
  padding: 5px;
  border: 1px solid black;
  background: rgba(255,0,0,0.25)
}
```

Together, these give the following result:




## In the next article

Well done for reaching the end of our first selectors tutorial! We hope you found your first play with selectors fun. We've now looked at the simple core selectors we'll commonly use. Let's level-up to more advanced features, starting with Attribute selectors.

Previous | Overview: Introduction to CSS | Next

## In this module

- CSS syntax
- Selectors
- Simple selectors
- Attribute selectors
- Pseudo-classes and pseudo-elements
- Combinators and multiple selectors
- CSS values and units
- Cascade and inheritance
- The box model
- Debugging CSS
- Fundamental CSS comprehension