

# Type, class, and ID selectors

English ▼

[< Previous](#)[↑ Overview: Building blocks](#)[Next >](#)

In this lesson we will take a look at the simplest selectors that are available, which you will probably use the most in your work.

**Prerequisites:**

Basic computer literacy, [basic software installed](#), basic knowledge of [working with files](#), HTML basics (study [Introduction to HTML](#)), and an idea of how CSS works (study [CSS first steps](#).)

**Objective:**

To learn about the different CSS selectors we can use to apply CSS to a document.

---

## Type selectors

A **type selector** is sometimes referred to as a *tag name selector* or *element selector*, because it selects an HTML tag/element in your document. In the below example we have used span, em and strong selectors. All instances of `<span>`, `<em>` and `<strong>` elements are therefore styled.

Try adding a CSS rule to select the `<h1>` element and change its color to blue.

# Type selectors

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko **endive** gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga **endive cauliflower** sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale. Celery potato scallion desert raisin horseradish spinach

```
span {  
  background-color: yellow;  
}  
  
strong {  
  color: rebeccapurple;  
}  
  
em {  
  color: rebeccapurple;  
}
```

```
<h1>Type selectors</h1>  
<p>Veggies es bonus vobis, proinde vos postulo essum  
magis <span>kohlrabi welsh onion</span> daikon amaranth  
tatsoi tomatillo  
  melon azuki bean garlic.</p>  
  
<p>Gumbo beet greens corn soko <strong>endive</strong>  
gumbo gourd. Parsley shallot courgette tatsoi pea  
sprouts fava bean collard  
  greens dandelion okra wakame tomato. Dandelion  
cucumber earthnut pea peanut soko zucchini.</p>
```

```
<p>Turnip greens yarrow ricebean rutabaga <em>endive  
cauliflower</em> sea lettuce kohlrabi amaranth water  
spinach avocado
```

Reset

---

## The universal selector

The universal selector is indicated by an asterisk (\*) and selects everything in the document (or inside the parent element if it is being chained together with another element and a descendant combinator, for example). In the following example we have used the universal selector to remove the margins on all elements. This means that instead of the default styling added by the browser, which spaces out headings and paragraphs with margins, everything is close together and we can't see the different paragraphs easily.

# Universal selector

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko **endive** gumbo gourd.

Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato.

Dandelion cucumber earthnut pea peanut soko zucchini.

```
* {  
  margin: 0;  
}
```

```
<h1>Universal selector</h1>  
<p>Veggies es bonus vobis, proinde vos postulo essum  
magis <span>kohlrabi welsh onion</span> daikon amaranth  
tatsoi tomatillo  
  melon azuki bean garlic.</p>  
  
<p>Gumbo beet greens corn soko <strong>endive</strong>  
gumbo gourd. Parsley shallot courgette tatsoi pea  
sprouts fava bean collard  
  greens dandelion okra wakame tomato. Dandelion  
cucumber earthnut pea peanut soko zucchini.</p>
```

Reset

This kind of behavior can sometimes be seen in "reset stylesheets", which strip out all of the browser styling. These were very popular at one point, however stripping out all styling usually meant that you then had to do the job of putting it all back! We tend to use the universal selector carefully therefore, to deal with very specific situations such as the one outlined below.

Using the universal selector to make your selectors easier to read

One use of the universal selector is to make selectors easier to read and more obvious in terms of what they are doing. For example, if I wanted to select the first child of any `<article>` element, no matter what element it was, and make it bold, I could use the `:first-child` selector, which we will learn more about in the lesson on [pseudo-classes and pseudo-elements](#), as a descendant selector along with the `<article>` element selector:

```
1 | article :first-child {  
2 |  
3 | }
```

This could be confused however with `article:first-child`, which will select any `<article>` element that is the first child of another element.

To avoid this confusion we can add the universal selector to the `:first-child` selector, so it is obvious what the selector is doing. It is selecting *any* element which is the first-child of an `<article>` element:

```
1 | article *:first-child {  
2 |  
3 | }
```

---

## Class selectors

The class selector starts with a full stop (.) character and will select everything in the document with that class applied to it. In the live example below we have created a class called `.highlight`, and have applied it to several places in my document. All of the elements that have the class applied are highlighted.

# Class selectors

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko **endive** gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

```
.highlight {  
  background-color: yellow;  
}
```

```
<h1 class="highlight">Class selectors</h1>  
<p>Veggies es bonus vobis, proinde vos postulo essum  
magis <span class="highlight">kohlrabi welsh  
onion</span> daikon amaranth tatsoi tomatillo  
melon azuki bean garlic.</p>  
  
<p class="highlight">Gumbo beet greens corn soko  
<strong>endive</strong> gumbo gourd. Parsley shallot  
courgette tatsoi pea sprouts fava bean collard  
greens dandelion okra wakame tomato. Dandelion
```

Reset

## Targeting classes on particular elements

You can create a selector that will target specific elements with the class applied. In this next example we will highlight a `<span>` with a class of `highlight` differently to an `<h1>` heading with a class of `highlight`. We do this by using the type selector for the element I want to target, with the class appended, with no white space in between.

# Class selectors

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko **endive** gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

```
span.highlight {  
  background-color: yellow;  
}
```

```
h1.highlight {  
  background-color: pink;  
}
```

```
<h1 class="highlight">Class selectors</h1>  
<p>Veggies es bonus vobis, proinde vos postulo essum  
magis <span class="highlight">kohlrabi welsh  
onion</span> daikon amaranth tatsoi tomatillo  
melon azuki bean garlic.</p>  
  
<p class="highlight">Gumbo beet greens corn soko  
<strong>endive</strong> gumbo gourd. Parsley shallot  
courgette tatsoi pea sprouts fava bean collard  
greens dandelion okra wakame tomato. Dandelion
```

Reset

This approach does make the CSS less reusable as the class will now only apply to that particular element, and you would need to add another selector if you decided that the rules should apply to other elements too.

Target an element if it has more than one class applied

You can apply multiple classes to an element and target them individually, or only select the element when all of the classes in the selector are present. This can be helpful when building up components that can be combined in different ways on your site.

In the example below we have a `<div>` that contains a note. The grey border is applied when the box has a class of `notebox`. If it also has a class of `warning` or `danger`, we change the `border-color`.

We can tell the browser that we only want to match the element if it has all of these classes by chaining them together with no white space between them.



This is an informational note.

**This note shows a warning.**

**This note shows danger!**

```
.notebox {  
  border: 4px solid #666;  
  padding: .5em;  
}  
  
.notebox.warning {  
  border: 4px solid orange;  
  font-weight: bold;  
}  
  
.notebox.danger {  
  border: 4px solid red;  
  font-weight: bold;  
}
```

```
<div class="notebox">  
  This is an informational note.  
</div>  
  
<div class="notebox warning">  
  This note shows a warning.  
</div>  
  
<div class="notebox danger">  
  This note shows danger!  
</div>
```

Reset

---

## ID Selectors

An ID selector begins with a `#` rather than a full stop character, but is basically used in the same way as a class selector. An ID however can be used only once per document. It can select an element that has the `id` set on it, and you can precede the ID with a type selector to only target the element if both the element and ID match. You can see both of these uses in the following example:

## ID selector

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko **endive** gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

```
#one {  
  background-color: yellow;  
}  
  
h1#heading {  
  color: rebeccapurple;  
}
```

```
<h1 id="heading">ID selector</h1>  
<p>Veggies es bonus vobis, proinde vos postulo essum  
magis kohlrabi welsh onion daikon amaranth tatsoi  
tomatillo  
  melon azuki bean garlic.</p>  
  
<p id="one">Gumbo beet greens corn soko  
<strong>endive</strong> gumbo gourd. Parsley shallot  
courgette tatsoi pea sprouts fava bean collard  
  greens dandelion okra wakame tomato. Dandelion
```

Reset



**Note:** As we learned in the lesson on specificity, an ID has high specificity and will overrule most other selectors. This can make them difficult to deal with. In most cases it is preferable to add a class to the element rather than use an ID, however if using the ID is the only way to target the element — perhaps because you do not have access to the markup and so cannot edit it — this will work.

---

## In the next article

We'll continue exploring selectors by looking at [attribute selectors](#).

[← Previous](#)

[↑ Overview: Building blocks](#)

[Next →](#)

---

## In this module

1. [Cascade and inheritance](#)
2. [CSS selectors](#)
  - [Type, class, and ID selectors](#)
  - [Attribute selectors](#)
  - [Pseudo-classes and pseudo-elements](#)
  - [Combinators](#)
3. [The box model](#)
4. [Backgrounds and borders](#)
5. [Handling different text directions](#)
6. [Overflowing content](#)
7. [Values and units](#)
8. [Sizing items in CSS](#)
9. [Images, media, and form elements](#)
10. [Styling tables](#)
11. [Debugging CSS](#)
12. [Organizing your CSS](#)

[Type selectors](#)

[The universal selector](#)

[Class selectors](#)

[ID Selectors](#)

[In the next article](#)

[In this module](#)

## Related Topics

**Complete beginners start here!**

- ▶ [Getting started with the Web](#)

### HTML — Structuring the Web

- ▶ [Introduction to HTML](#)
- ▶ [Multimedia and embedding](#)
- ▶ [HTML tables](#)
- ▶ [HTML forms](#)

### CSS — Styling the Web

- ▶ [CSS first steps](#)
- ▼ [CSS building blocks](#)
  - [CSS building blocks overview](#)
  - [Cascade and inheritance](#)
  - [CSS selectors](#)
  - [The box model](#)
  - [Backgrounds and borders](#)
  - [Handling different text directions](#)
  - [Overflowing content](#)
  - [Values and units](#)

Sizing items in CSS

Images, media, and form elements

Styling tables

Debugging CSS

Organizing your CSS

- ▶ Styling text

- ▶ CSS layout

## **JavaScript — Dynamic client-side scripting**

- ▶ JavaScript first steps

- ▶ JavaScript building blocks

- ▶ Introducing JavaScript objects

- ▶ Asynchronous JavaScript

- ▶ Client-side web APIs

## **Accessibility — Make the web usable by everyone**

- ▶ Accessibility guides

- ▶ Accessibility assessment

## **Tools and testing**

- ▶ Cross browser testing

## **Server-side website programming**

- ▶ First steps

- ▶ Django web framework (Python)

- ▶ Express Web Framework (node.js/JavaScript)

## Further resources

► [Common questions](#)

[How to contribute](#)



# Learn the best of web development

Get the latest and greatest from MDN delivered straight to your inbox.

**Sign up now**