

CSS Dropdowns

CSS dropdowns are used to display a list of options or content when a user clicks or hover over an element, like a button or a navigation link.

A CSS dropdown consists of a trigger element (like `<div>`, `<button>`, `<p>`, `<a>`, etc.).

When the trigger element is clicked or hovered over, the dropdown content will be displayed.

The dropdown content is a container element (e.g. `<div>`) that holds the hidden content (can be text, links, images, etc.).

CSS Dropdown Box with Text

Here, we create a dropdown box with some text, that appears when the user mouses over a `<div>` element.

Example

```
<style>
.dropdown {
    position: relative;
}

.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f9f9f9;
    min-width: 130px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    padding: 12px 16px;
}

.dropdown:hover .dropdown-content {
    display: block;
}
```

```
</style>
```

```
<div class="dropdown">Mouse over me!
  <div class="dropdown-content">Hello World!</div>
</div>
```

CSS Dropdown Menu

Create a dropdown menu that allows the user to choose an option from a list:

Dropdown Menu

This example is similar to the previous one, except that we add a button and links inside the dropdown box and style them to fit the dropdown button:

Example

```
<style>
.dropdown {
  position: relative;
}

/* Style the dropdown button */
.dropbtn {
  background-color: #4CAF50;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
  cursor: pointer;
}

/* Dropdown content */
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 200px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
```

```
}

/* Links inside dropdown content */
.dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {
    background-color: #f1f1f1
}

/* Show the dropdown content on hover */
.dropdown:hover .dropdown-content {
    display: block;
}

/* Change background color of dropdown button on hover */
.dropdown:hover .dropbtn {
    background-color: #3e8e41;
}
</style>

<div class="dropdown">
    <button class="dropbtn">Dropdown Menu</button>
    <div class="dropdown-content">
        <a href="#">Link 1</a>
        <a href="#">Link 2</a>
        <a href="#">Link 3</a>
    </div>
</div>
```

CSS Attribute Selectors

CSS attribute selectors are used to select and style HTML elements with a specific attribute or attribute value, or both.

The different types of attribute selectors are described below.

CSS [attribute] Selector

The [attribute] selector is used to select elements with a specific attribute.

The following example selects all <a> elements with a target attribute:

Example

```
a[target] {  
    background-color: yellow;  
}
```

CSS [attribute="value"] Selector

The [attribute="value"] selector is used to select elements with a specific attribute with an exact value.

The following example selects all <a> elements with a target="_blank" attribute:

Example

```
a[target="_blank"] {  
    background-color: yellow;  
}
```

CSS [attribute|= "value"] Selector

The [attribute|= "value"] selector is used to select elements with the specific attribute, whose value can be exactly the specific value, or start with the specific value followed by a hyphen (-).

Note: The value has to be a whole word, either alone, like class="top", or followed by a hyphen (-), like class="top-text".

Example

```
[class|="top"] {  
    background: yellow;  
}
```

CSS [attribute^="value"] Selector

The **[attribute^="value"]** selector is used to select elements with the specific attribute, whose value starts with a specific value.

The following example selects all elements with a class attribute value that starts with "top":

Example

```
[class^="top"] {  
    background: yellow;  
}
```

CSS [attribute\$="value"] Selector

The **[attribute\$="value"]** selector is used to select elements whose attribute value ends with a specific value.

The following example selects all elements with a class attribute value that ends with "test":

Example

```
[class$="test"] {  
    background: yellow;  
}
```

CSS Forms

CSS Styling Forms

CSS is used to style HTML forms. The look of an HTML form can be greatly improved with CSS:

Styling Form Input Fields

With CSS, you can style most of the different input types, like text fields, password fields, checkboxes, radio buttons, and file inputs. You can also style input labels and form buttons.

Some commonly used CSS properties for styling input fields, are:

- width
 - padding
 - margin
 - border
 - border-radius
 - background-color
 - color
 - font-size
-

Style Input Width

The width property is used to set the width of an input field.

Example

```
input {  
    width: 100%;  
}
```

The example above applies to all <input> elements. If you only want to style a specific input type, you can use attribute selectors:

- `input[type=text]` - will only select text fields
 - `input[type=password]` - will only select password fields
 - `input[type=number]` - will only select number fields
-

Style Input Padding

The padding property is used to add some space inside the text field.

Example

```
input[type=text] {  
    width: 100%;  
    padding: 12px;  
    margin: 10px 0;  
    box-sizing: border-box;  
}
```

Style Input Border

The border property is used to change the border size and color, and the border-radius property can be used to add rounded corners:

Example

```
input[type=text] {  
    border: 2px solid red;  
    border-radius: 8px;  
}
```

If you only want a bottom border, use the border-bottom property:

First Name

Example

```
input[type=text] {  
    border: none;  
    border-bottom: 1px solid red;  
}
```

Style Input Background Color and Color

The background-color property is used to add a background color to the input, and the color property is used to change the text color:

Example

```
input[type=text] {  
    background-color: #3CBC8D;  
    color: white;  
}
```

Style Input with Focus

By default, some browsers will add a blue outline around the input when it gets focus (clicked on). You can remove this behavior by adding outline: none; to the input.

Use the :focus selector to do something with the input field when it gets focus:

Example

```
input[type=text]:focus {  
    background-color: lightblue;  
}
```

Example

```
input[type=text]:focus {  
    border: 3px solid #555;  
}
```

Style Input with icon/image

If you want an icon inside the input, use the background-image property and position it with the background-position property. Also notice that we add a large left padding to reserve the space of the icon:

Example

```
input[type=text] {  
    background-color: white;
```

```
background-image: url('searchicon.png');
background-position: 10px 10px;
background-repeat: no-repeat;
padding-left: 40px;
}
```

Style Textarea

By default, a <textarea> can be resized with a "grabber" in the bottom right corner. To remove the grabber, set the resize property to none:

Example

```
textarea {
  width: 100%;
  height: 150px;
  padding: 12px 20px;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 4px;
  background-color: #f8f8f8;
  resize: none;
}
```

Style a Dropdown Menu

Australia Canada USA

Example

```
select {
  width: 100%;
  padding: 16px 20px;
  border: none;
  border-radius: 4px;
  background-color: #f1f1f1;
}
```

Style Form Buttons

Form buttons of type "button", "submit" and "reset" can also be styled with CSS:

Example

```
input[type=button], input[type=submit], input[type=reset] {  
    background-color: #04AA6D;  
    border: none;  
    color: white;  
    padding: 16px 32px;  
    text-decoration: none;  
    margin: 4px 2px;  
    cursor: pointer;  
}
```
