learn.freecodecamp.org/

Basic HTML and HTML5

font-family: FAMILY\_NAME, GENERIC\_NAME;

The GENERIC\_NAME is optional, and is a fallback font in case the other specified font is not available.

[Google Fonts](https://fonts.google.com/) is a free library of web fonts that you can use in your CSS by referencing the font's URL. To import a Google Font, you can copy the font(s) URL from the Google Fonts library and then paste it in your HTML.

<link href="https://fonts.googleapis.com/css?family=Lobster" rel="stylesheet" type="text/css">

To tell the browser which version of HTML your page is using add <!DOCTYPE ...>tag on the first line. For HTML5, you use <!DOCTYPE html>.

The !and uppercase DOCTYPEis important, especially for older browsers. The htmlis not case sensitive.

## **Checkboxes**

Each of your checkboxes can be nested within its own labelelement. By wrapping an inputelement inside of a labelelement it will automatically associate the checkbox input with the label element surrounding it.

It is considered best practice to explicitly define the relationship between a checkbox inputand its corresponding labelby setting the forattribute on the labelelement to match the idattribute of the associated inputelement.

<label for="loving"><input id="loving" type="checkbox" name="personality"> Loving</label>

You can set a **checkbox or radio** button to be checked by default using the checkedattribute.

To do this, just add the word "checked" to the inside of an input element. For example:

<input type="radio" name="test-name" checked>

## **Radio Buttons**

<label for="indoor">  
  <input id="indoor" type="radio" name="indoor-outdoor">Indoor  
</label>

<input type="radio" name="test-name" checked>

## **Rounded Corners with border-radius**

.thick-green-border {

border-color: green;

border-width: 10px;

border-style: solid;

border-radius: 10px;

}

a href="#"><img class="smaller-image thick-green-border" src="https://bit.ly/fcc-relaxing-cat" alt="A cute orange cat lying on its back."></a>



Circle

<style>

:root {

--red-color: red;

}

.red-box {

border-radius: 50%;

background: var(--red-color);

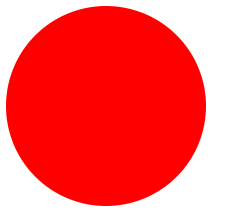
height: 200px;

width:200px;

}

</style>

<div class="red-box"></div>



## **Use the em Tag to Italicize Text**

To emphasize text, you can use the emtag. This displays text as italicized, as the browser applies the CSS of font-style: italic;to the element.

CSS treats each HTML element as its own box, which is usually referred to as the CSS Box Model.

## **Push Elements Left or Right with the float Property**

floatproperty of an element. Floating elements are removed from the normal flow of a document and pushed to either the leftor rightof their containing parent element. It's commonly used with the widthproperty to specify how much horizontal space the floated element requires.

#left {

float: left;

width: 50%;

}

#right {

float: right;

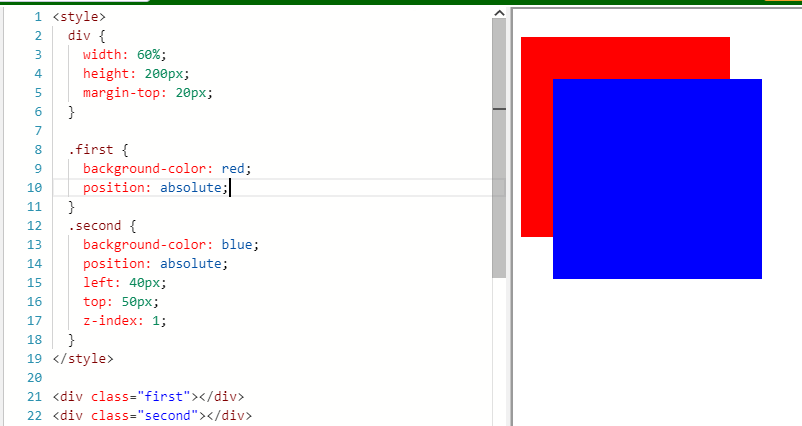
width: 40%;

}

## **Change the Position of Overlapping Elements with the z-index Property**

When elements are positioned to overlap, the element coming later in the HTML markup will, by default, appear on the top of the other elements. However, the z-indexproperty can specify the order of how elements are stacked on top of one another. It must be an integer (i.e. a whole number and not a decimal), and higher values for the z-indexproperty of an element move it higher in the stack than those with lower values.

In simple words, the element with higher z-index value sits upon the elements with lower z-index value.





Display a list horizontally then: display: inline

li { display: inline }

## **Adjust the Tone of a Color**

The hsl()option in CSS also makes it easy to adjust the tone of a color.

header {

background-color: hsl(180, 90%, 35%);

color: #FFFFFF;

}

## **Create a Gradual CSS Linear Gradient**

CSS provides the ability to use color transitions, otherwise known as gradients, on elements.

This is accessed through the backgroundproperty's linear-gradient()function.

background: linear-gradient(gradient\_direction, color 1, color 2, color 3, ...);

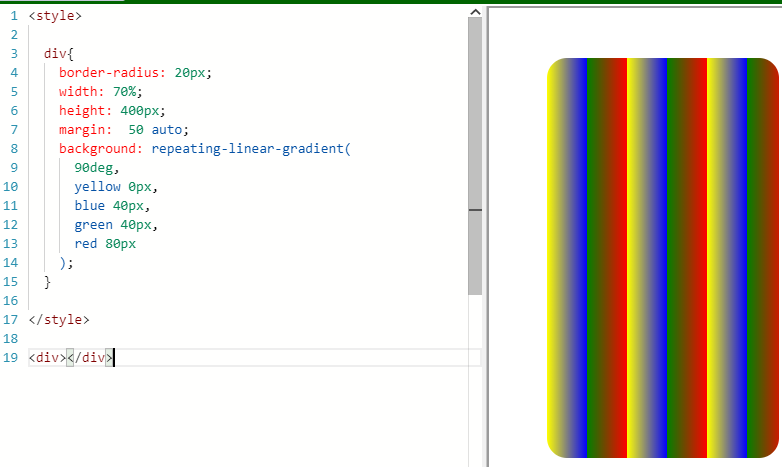
The first argument specifies the direction from which color transition starts - it can be stated as a degree, where 90deg makes a vertical gradient and 45deg is angled like a backslash. The following arguments specify the order of colors used in the gradient.

Example:

background: linear-gradient(90deg, red, yellow, rgb(204, 204, 255));



The repeating-linear-gradient()function is very similar to linear-gradient()with the major difference that it repeats the specified gradient pattern.



## **Learn How the CSS @keyframes and animation Properties Work**

There are properties and keyframes used to bring animation. There are eight animation properties in total.

1. animation-namesets the name of the animation, which is later used by @keyframesto tell CSS which rules go with which animations.
2. animation-durationsets the length of time for the animation.
3. @keyframesis how to specify exactly what happens within the animation over the duration.

#rect {

animation-name: rainbow;

animation-duration: 4s;

}

@keyframes rainbow {

0% { background-color: blue; }

50% { background-color: green; }

100% { background-color: yellow; }

}

</style>

<div id="rect"></div>

In this example, the rectangle’s color will change from blue to green and then yellow in duration of 4sec.