Adding Images

**Why Images Matter**

This entire course you've focused on adding content that you've authored, like headings, paragraphs, lists, and more. Much of this content is text-heavy, by nature. As you've no doubt seen, most web pages are not text-heavy, they make extensive use of great looking images to improve the user's experience on the site.

In the very beginning of the course, we taught you how to add images to your web page using the <img> element. While this technique is still valid, it's possible to also add images (and style them) to a website and style them using CSS techniques. The aim of this unit will be to teach you how to add images to your web page by using a variety of techniques.

Let's begin!

**Width x Height**

As with any other element, the dimensions of an image can be set with the height and width properties.

<img src="#" alt="A red leaf" class="leaf" />

img.leaf {

width: 350px;

height: 200px;

}

Specifying the dimensions of an image helps the browser determine how much space should be reserved for the image.

**Note:** Images should be saved at the dimensions they will be displayed in on the web page. Using dimensions for an image that exceed the original dimensions will distort the image.

# Centering Images

Images can also be centered, but first, their default behavior must be changed. By default, images are inline elements. For images to center properly, they must behave as block-level elements.

img.leaf {

width: 300px;

height: 200px;

display: block;

}

In the example above, the image's display property has been set to block. Now the image can be aligned as a block-level element.

img.leaf {

width: 300px;

height: 200px;

display: block;

margin: 0px auto;

}

In the example above, the image is aligned using the margin property. The top and bottom margins of the image's box are set to 0 margin. The left and right margins are set to auto, which automatically sets the exact amount of margin needed on the left and right sides in order to center the image.

**Note:** To align images to the left or right side of a page, you can use the float property you learned about earlier.

# Background: Images

Images can also be added to the backgrounds of HTML elements with the background-image property.

body {

background-image: url("https://www.example.com/leaf.jpg");

}

The background-image can be set to a value of url().

1. url() - contains the URL of the image, enclosed in double quotes.

In the example above, the background of the body will be set to the image specified in double quotes. This technique can be used on nearly any HTML element.

**Background: Repeat**

In the last exercise, you probably noticed that the background image was repeated, creating a tiled background (this is the default behavior).

You can control how a background image *repeats* with the background-repeat property. This property can take one of four values:

1. repeat - the default value — the image will repeat horizontally and vertically.
2. repeat-x - the background image will be repeated only along the x-axis (horizontally).
3. repeat-y - the background image will be repeated only along the y-axis (vertically).
4. no-repeat - the background image will not be repeated at all and will appear only once.

p {

background-image: url("#");

background-repeat: repeat-x;

}

In the example above, the paragraph's background image will repeat horizontally.

**Background: Position**

When a background image is not repeated, its position can be modified with the background-position property.

p {

background-image: url("#");

background-repeat: no-repeat;

background-position: right center;

}

A background image is positioned using a 3 by 3 grid (three rows, three columns), meaning there are 9 total possible positions for the image:

1. left top - top left corner of the element's box.
2. center top - top center of the element's box.
3. right top - top right corner of the element's box.
4. left center - left column, center row.
5. center center - the center of the element's box.
6. right center - right column, center row.
7. left bottom - bottom left corner of the element's box.
8. center bottom - bottom center of the element's box.
9. right bottom - bottom right corner of the element's box.

Note that the values are in pairs.

p {

background-image: url("#");

background-repeat: no-repeat;

background-position: right top;

}

In the example above, the background image is not repeated. It's positioned in the top right corner of the element's box.

**Note:** When setting this property, if only one value is specified, the second value will default to center.

# Background: Shorthand

In the last few exercises, you learned how to set a background image, its repetition pattern, and its position using three different properties. CSS allows you to set all three properties at once using a shorthand property: background.

Previously, we set these three properties as follows:

p {

background-image: url("#");

background-repeat: no-repeat;

background-position: right center;

}

The code in the example above can be shortened using the background property.

p {

background: url("#") no-repeat right center;

}

Note that the background property includes all of the properties that we previously styled individually: background image, repetition, and position (in that order). It's considered best practice to follow this order of values when setting the background property.

# Background: Size

To modify a background image's size, you can use the background-size property.

This exercise will focus on two of the most common values of the background-size property:

1. cover - expands the image as large as possible to cover the full width or height of a container. If the dimensions of the container (say, a div) are larger than the dimensions of the image, the image will become distorted. This value is best for images that don't communicate important content to the user, like background images.
2. contain - expands the image as large as possible, but the image will be [letterboxed](https://www.codecademy.com/courses/learn-html-css/lessons/adding-images/exercises/background-size?action=lesson_resume), which means it won't cover the full width or height of a container.

div.header {

height: 400px;

width: 100%;

background: url("#") no-repeat right center;

background-size: cover;

}

In the example above, the background image will expand to cover the full size of the div.

**Background: Attachment**

With the background-attachment property, you can specify whether or not a background image should remain at one position on the web page or whether it should move up and down as the user scrolls through a web page.

The background-attachment property can take one of two values:

1. scroll - this value allows the image to move up and down as a user scrolls on the web page (this is the default value).
2. fixed - this value pins the image's position on the page.

p {

background: url("#") no-repeat right center;

background-attachment: fixed;

}

In the example above, the background image will remained fixed as a user scrolls through the web page.

**Background: Color**

A wide variety of different background styles are used across many modern websites today. Some sites use large images in their layout, while others use creative [color gradients](https://en.wikipedia.org/wiki/Color_gradient) as their background.

The background-image property you learned about earlier allows you to do more than simply set the background image of an element. It also lets you use color gradients in your web page.

Gradients are planned to be a part of the newest revision of CSS, CSS3. Due to the many browsers available, there is no standard way of defining a gradient using CSS (different browsers require different syntax). In this exercise, we'll look at one value supported on a couple of major browsers.

The background-image property can be set to the following value:

1. -webkit-linear-gradient() - this value accepts two arguments: the two colors the linear gradient will transition to and from. The colors are usually specified as hex color codes.

div.header {

height: 400px;

width: 400px;

background-image: -webkit-linear-gradient(#666CCC, #BC1324);

}

In a later course, you'll learn more about the -webkit-linear-gradient syntax, as well as other similar CSS values. For now, it's important to understand how to create a linear gradient using the value in the example above.

# Image vs Div Background

You've learned how to add images to a web page using the <img> element and the background property. What's the difference between these two methods? When should one method be used over another?

The method used depends on the type of image.

Some images are part of the content of a web page (icons, logos, album photos, etc.) and they communicate important information to a user. Other images are intended simply to style a web page (header backgrounds, etc.), not to communicate important information.

When an image communicates important information, you can use the <img> element and style the image using CSS, if needed.

When an image is intended to style a web page, you can use the background property and further style it with CSS.