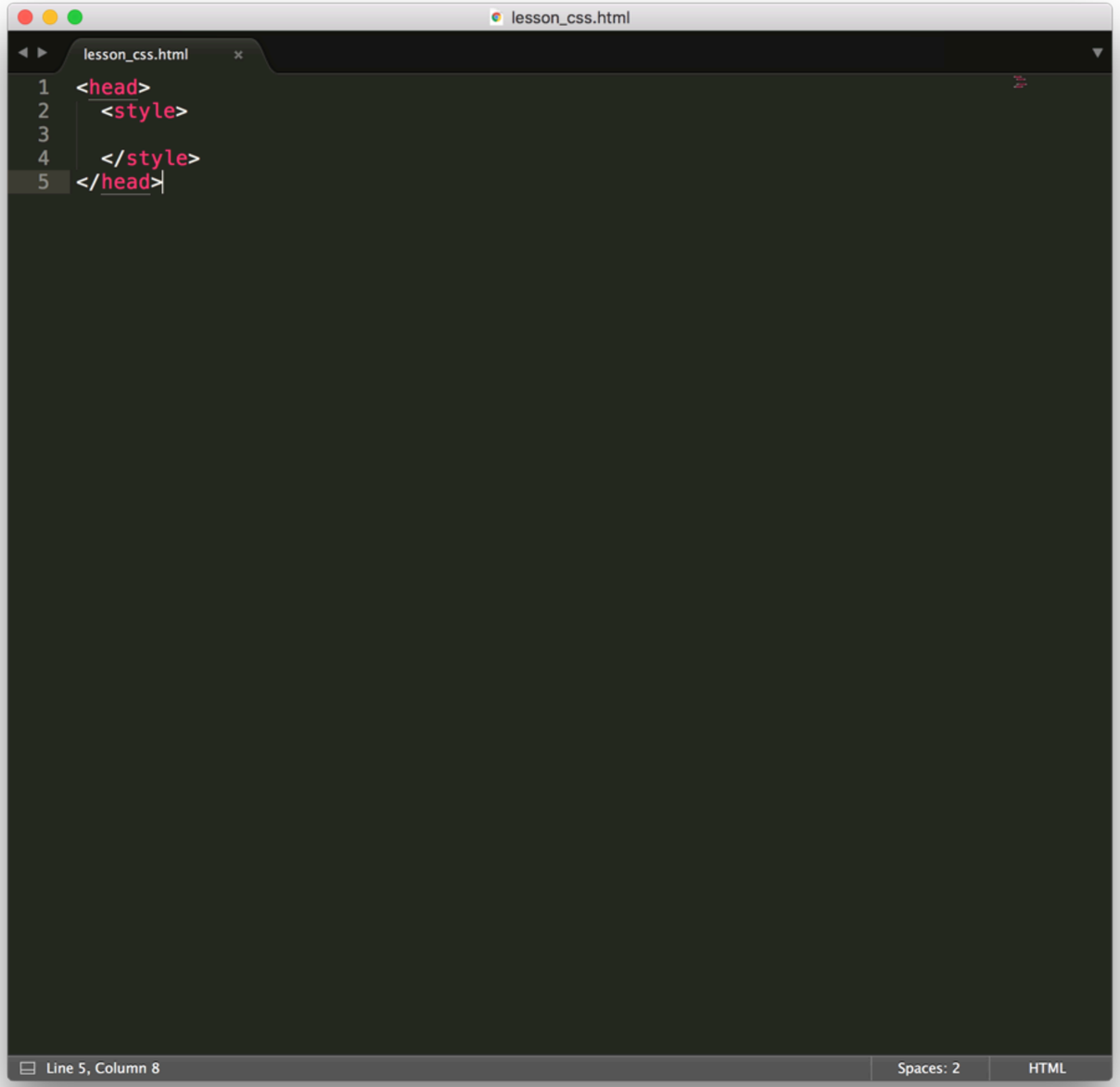
CSS

CSS, or Cascading Style Sheets, is a language that web developers use to *style* the HTML content on a web page. If you're interested in modifying colors, font types, font sizes, shadows, images, element positioning, and more, CSS is the tool to use.



Although CSS is a different language than HTML, it's possible to write CSS code directly within an HTML file. This is possible because of the <style> element.

lesson_css.html

```html
<head>
  <style>
    h2 {
      font-family: Arial;
    }
  </style>
</head>
```

Line 7, Column 8; Saved ~/Desktop/lesson_css.html (UTF-8) | Spaces: 2 | HTML

Once **<style>** is placed in the web page's head, we can begin writing CSS code.

See an example on the left.

Although the <style> element allows you to write CSS code within HTML files, this mixture of HTML and CSS can result in code that is difficult to read and maintain.

It's common for developers to add substantial amounts of custom CSS styling to a web page. When all of that CSS code is placed within a <style> element in an HTML file, you risk the following two things:

1. Creating a large HTML file that is difficult to read and maintain (by you and other developers). Overall, this can result in an inefficient workflow.

2. Maintaining a clear distinction between web page structure (HTML) and web page styling (CSS).

Fortunately, the following solution will help you avoid creating large HTML files that mix in CSS code: a CSS file!

HTML files are meant to contain only HTML code. Similarly, CSS files are meant to contain only CSS code. You can create a CSS file by using the .css file name extension, like so: style.css

With a CSS file, you can write all the CSS code needed to style a page without having to sacrifice the readability and maintainability of your HTML file.

lesson_css.html

```html
<head>
  <style>
    * {
      font-family: 'Georgia', 'Times', serif;
    }

    a {
      color: SeaGreen;
      text-decoration: none;
    }

    img {
      border-radius: 100%;
    }
  </style>
</head>
```

Line 16, Column 9 | Spaces: 2 | HTML

Trying making an index.html file and a style.css file.

In the html file add an anchor and image. Then in your css file type the code on the left.

Be sure to only take the css code!

Notice that nothing looks any different when you refresh the page. Why?

When HTML and CSS code are in separate files, the HTML file must know exactly where the CSS code is kept, otherwise, the styling can't be applied to the web page. In order to apply the styling to the web page, we'll have to *link* the HTML file and the CSS file together.

You can use the <link> element to link the HTML and CSS files together. The <link> element must be placed within the head of the HTML file. It is a self-closing tag and requires the following three attributes:

href – like the anchor element, the value of this attribute must be the address, or path, to the CSS file.

type – this attribute describes the type of document that you are linking to (in this case, a CSS file). The value of this attribute should be set to “text/css”.

rel – this attribute describes the relationship between the HTML file and the CSS file. Because you are linking to a stylesheet, the value should be set to “stylesheet”.

lesson_css.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>CSS Lesson</title>
    <link href="style.css" type="text/css" rel="stylesheet">
  </head>
  <body>
    <a href="https://www.google.com">Test link</a>
    <img src="test_image.jpg" />
  </body>
</html>
```

Line 9, Column 25 | Spaces: 2 | HTML

When linking an HTML file and a CSS file together, the `<link>` element will look like this.

Let's review what you've learned so far:

1. HTML and CSS are kept in separate files in order to keep code maintainable and readable, as well as keep structure separate from styling.

2. The <style> element allows you to write CSS code within an HTML file.

3. A CSS stylesheet can be linked to an HTML file using the <link> element, which requires three attributes:

href - set equal to the path of the CSS file.
type - set equal to “text/css”.
rel - set equal to “stylesheet”.

You've learned how to separate HTML and CSS into two files and link them together, but you haven't learned how CSS syntax is used to style elements on a web page.

style.css

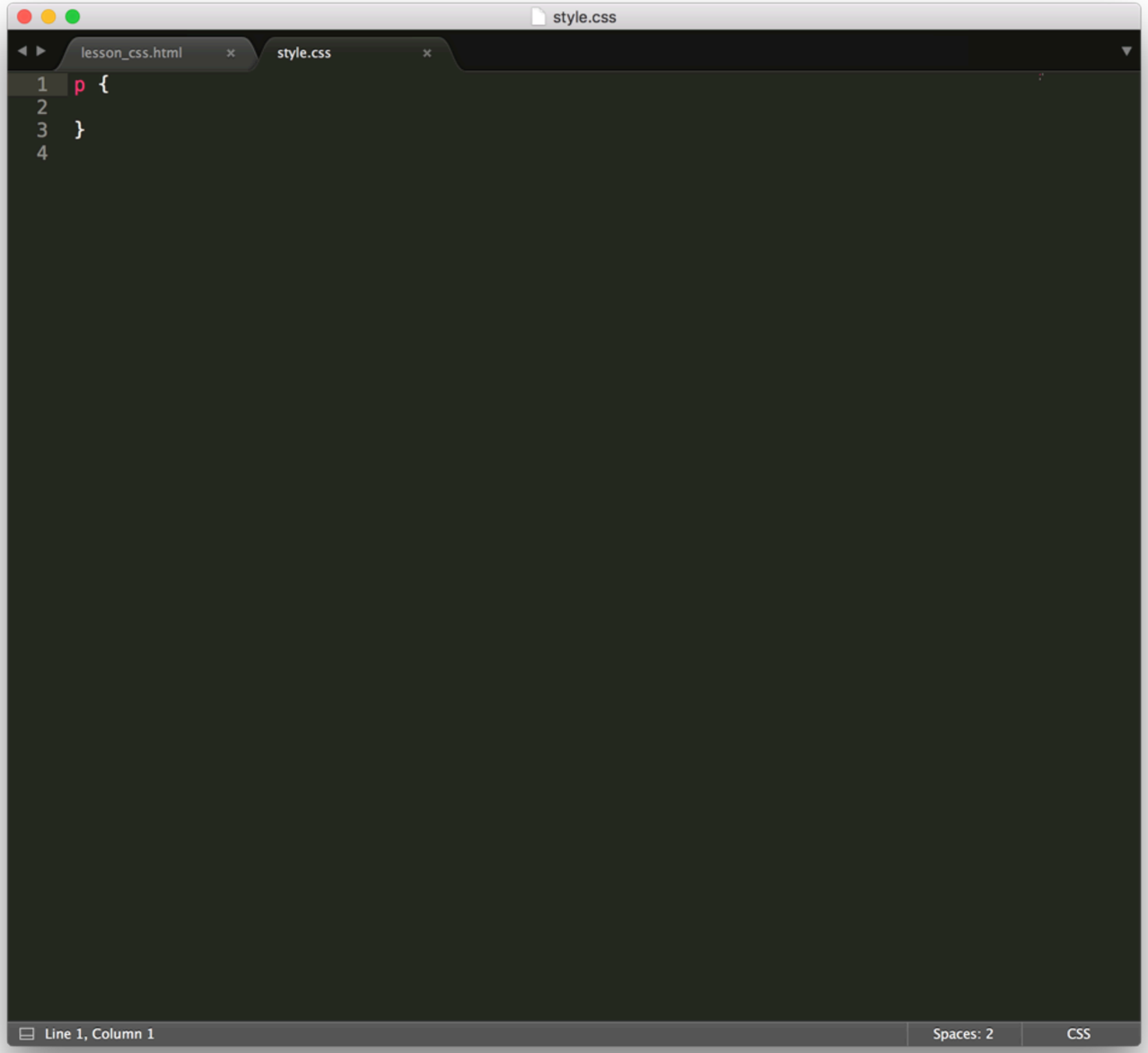
lesson_css.html | style.css

```
1  html, body {
2    margin: 0;
3  }
4
5  h1, h2, a {
6    font-family: 'Oswald', sans-serif;
7    text-transform: uppercase;
8  }
9
10 p {
11   font-family: Helvetica, Arial, sans-serif;
12 }
13
14 .container {
15   max-width: 940px;
16   margin: 0 auto;
17 }
18
19 /* Main */
20 .main {
21   text-align: left;
22   background: url("bg.jpg") no-repeat center center;
23   background-size: cover;
24   height: 600px;
25 }
26
27 .main .container {
28   position: relative;
29   top: 100px;
30 }
31
32 .main h1 {
33   color: #fff;
34   margin: 0;
35   font-size: 150px;
36 }
```

Line 22, Column 20 | Spaces: 2 | CSS

Take a look... this is what css looks like.

To style an HTML element using CSS, you must first *select* that element in the CSS file.



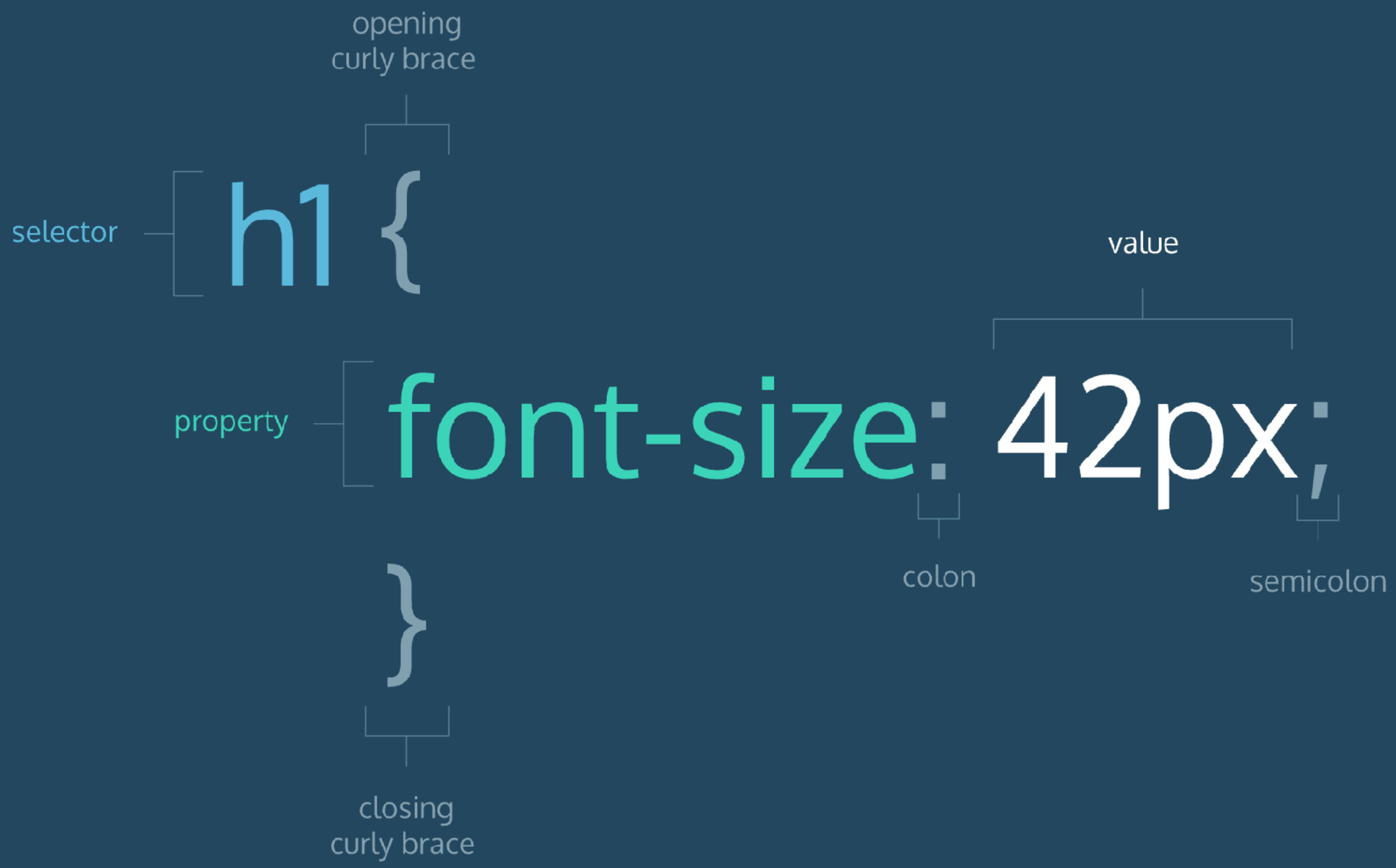
In the example above, all paragraph elements are selected using a CSS *selector*. The selector (in this case) is p. Note that the CSS selector essentially matches the HTML tag for that element, but without the angle brackets.

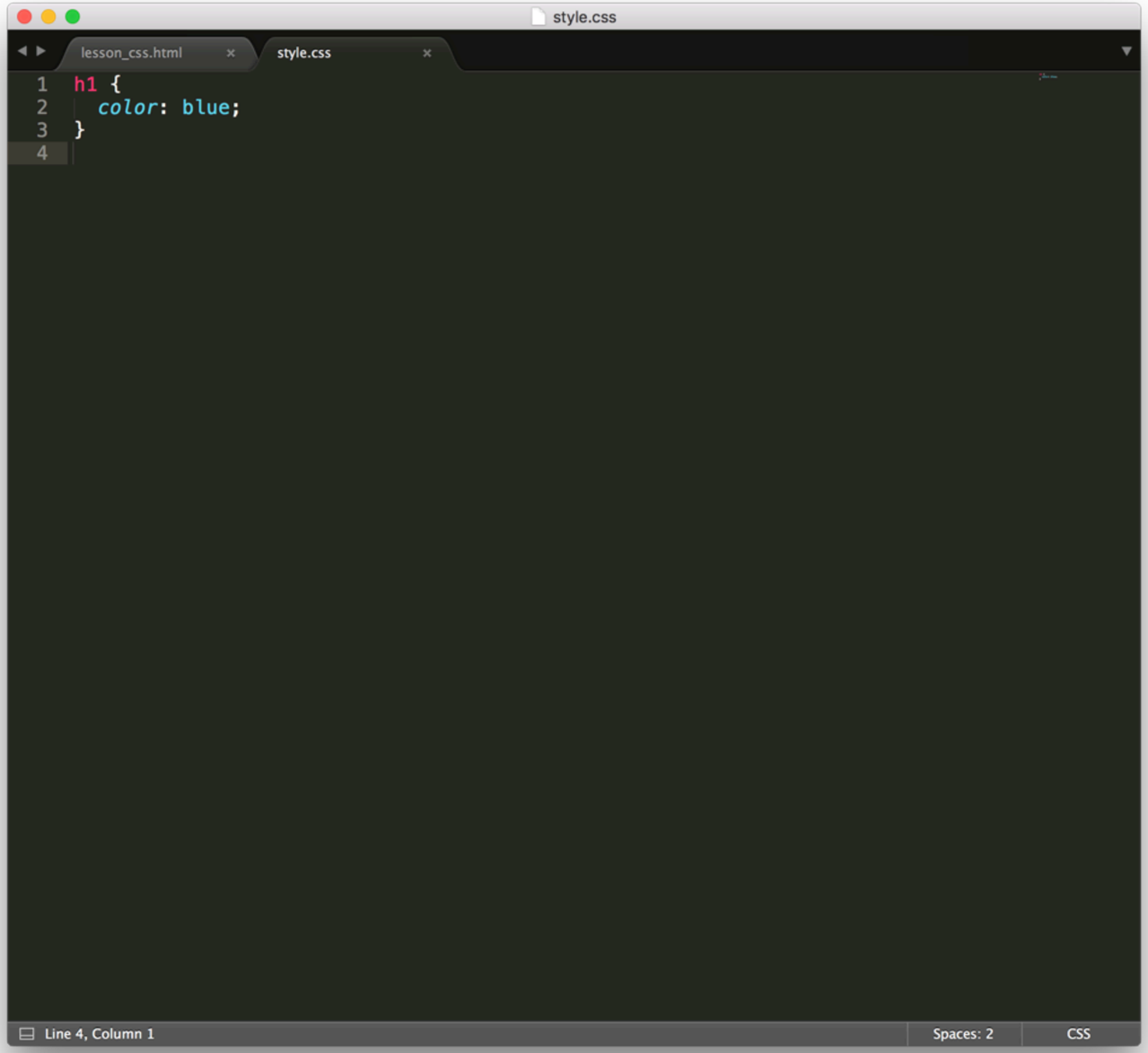
Note: The p selector in the example above will select *all* <p> elements on the web page.

It's not enough to simply select an HTML element in a CSS file. To actually style the element, you need to specify two things inside the body of the selector:

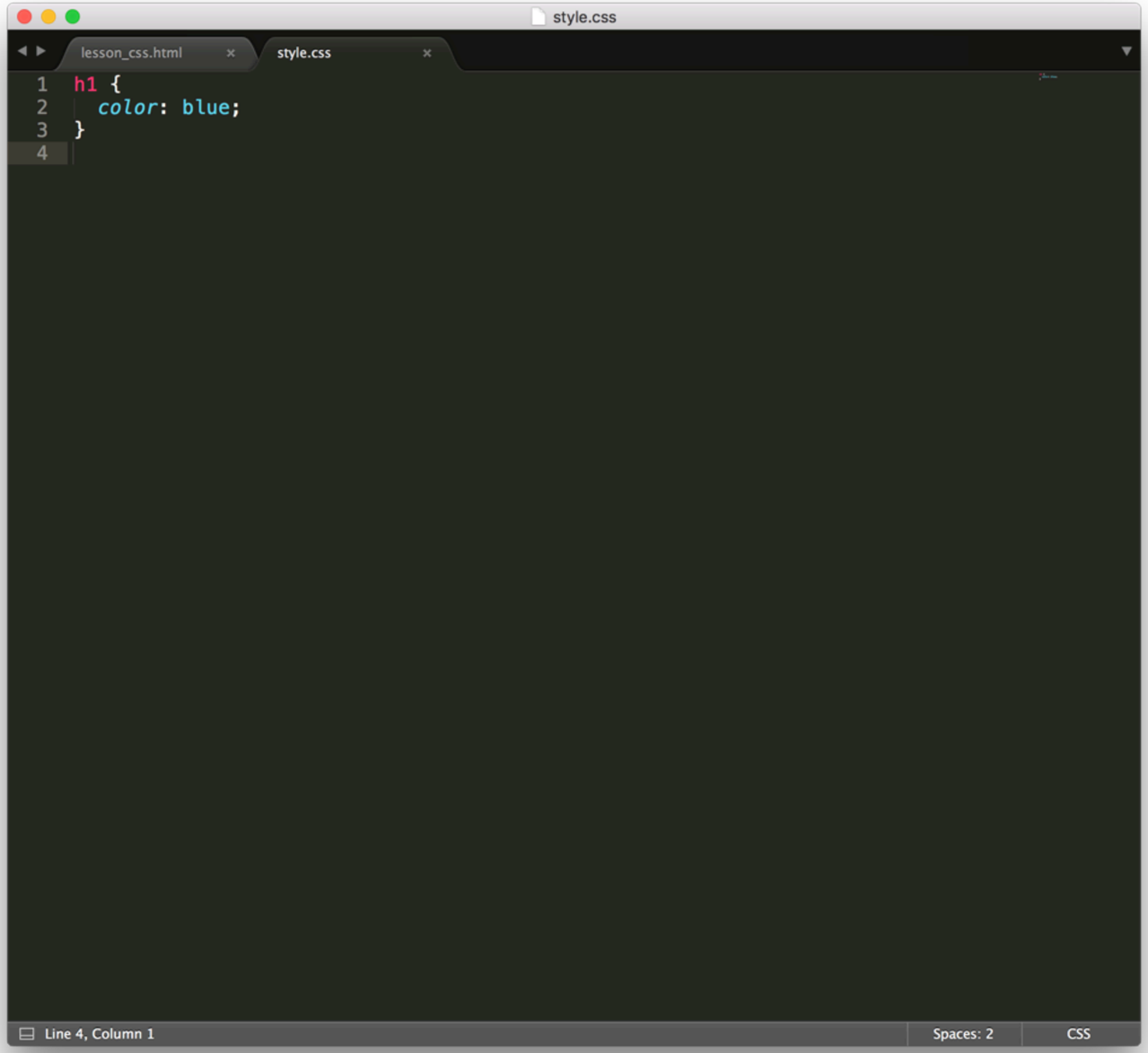
Property - the property you'd like to style of that element (i.e., size, color, etc.).

Value - the value of the property (i.e., 18px for size, Blue for color, etc.).


opening
curly brace
selector
h1 {
value
property
font-size: 42px;
colon
semicolon
}
closing
curly brace



In this example, the <h1> element has been selected. Inside of the selector's body, the heading's color property is set to a value of Blue. The heading will now appear the color blue in the browser.



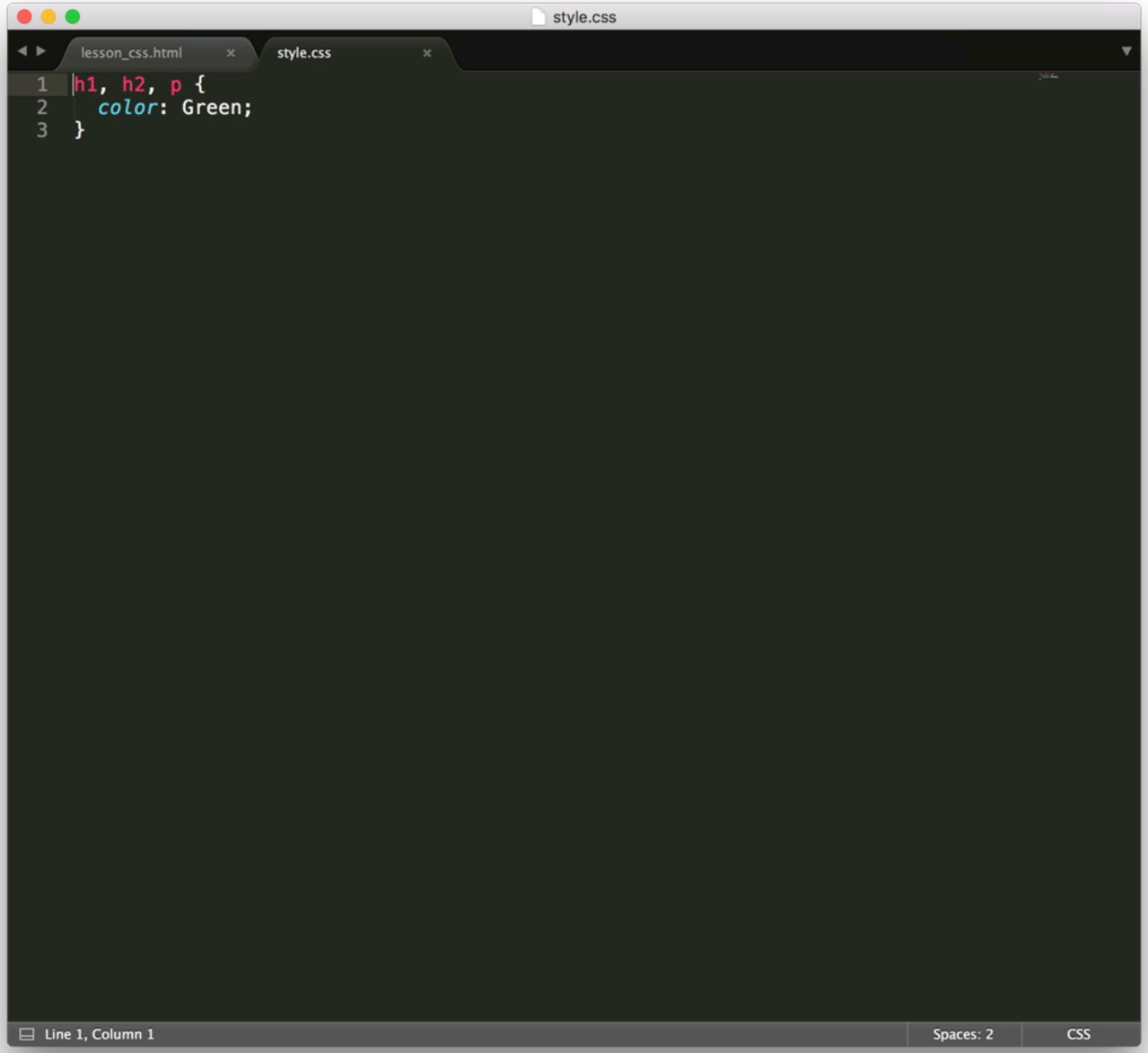
The line color: Blue; is referred to CSS *declaration*. A CSS declaration consists of a property and a value. Note that a semicolon (;) ends all declarations.

Finally, the entire snippet of code in this example is known as a CSS *rule*. A CSS rule consists of the selector and all declarations inside of the selector.

Let’s give this a try. Add a heading tag and paragraph tag to your html document. Change the color of the heading to pink and the font-size of the paragraph to 35px.

Styling with CSS would be very inefficient if you were forced to manually style the same property across many elements.

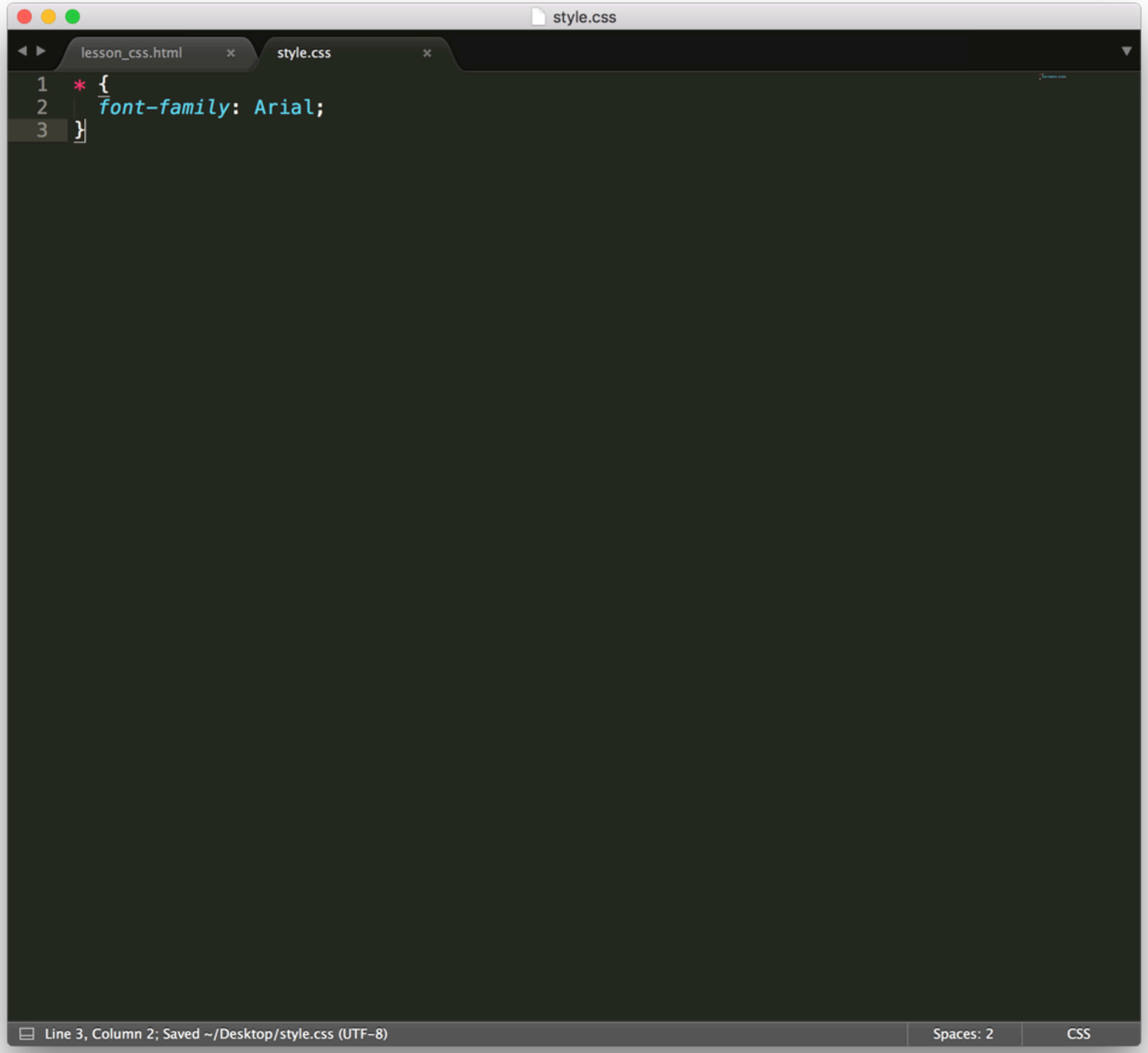
For example, what if you wanted to change the color of 10 different elements to Aquamarine in CSS?.



Fortunately, you can select multiple elements at once so that you can save time styling a shared property.

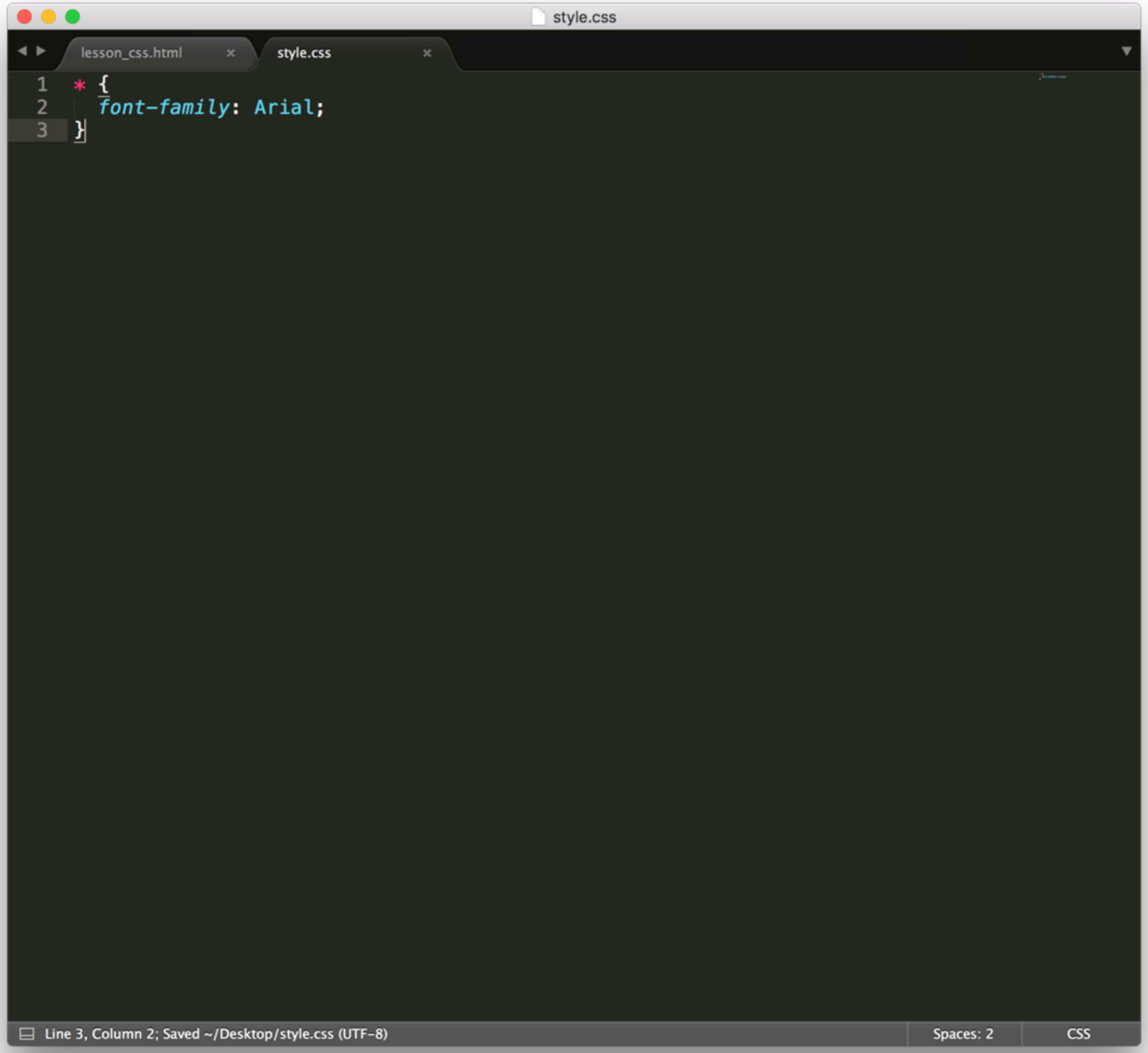
In this example, the <h1> heading, the <h2>heading, and the paragraph have all been styled to appear Green using a *multiple element selector*. A multiple element selector can save you time when you want to style the same property across many elements.

There's a special selector that can instantly select every single element on the web page: the *universal selector*.



The universal selector, *, is used to select every element on the page and set the font to Arial.

What makes the universal selector so special? When all elements on a web page require the same styling, it's often more efficient to set that styling using the universal selector. Afterwards, you can modify (or remove) that styling for specific elements that don't require it.



Just like HTML, CSS follows certain best practices for spacing and indentation.

style.css

lesson_css.html | style.css

```
1  html, body {
2    margin: 0;
3  }
4
5  h1, h2, a {
6    font-family: 'Oswald', sans-serif;
7    text-transform: uppercase;
8  }
9
10 p {
11   font-family: Helvetica, Arial, sans-serif;
12 }
13
14 .container {
15   max-width: 940px;
16   margin: 0 auto;
17 }
18
19 /* Main */
20 .main {
21   text-align: left;
22   background: url("bg.jpg") no-repeat center center;
23   background-size: cover;
24   height: 600px;
25 }
26
27 .main .container {
28   position: relative;
29   top: 100px;
30 }
31
32 .main h1 {
33   color: #fff;
34   margin: 0;
35   font-size: 150px;
36 }
```

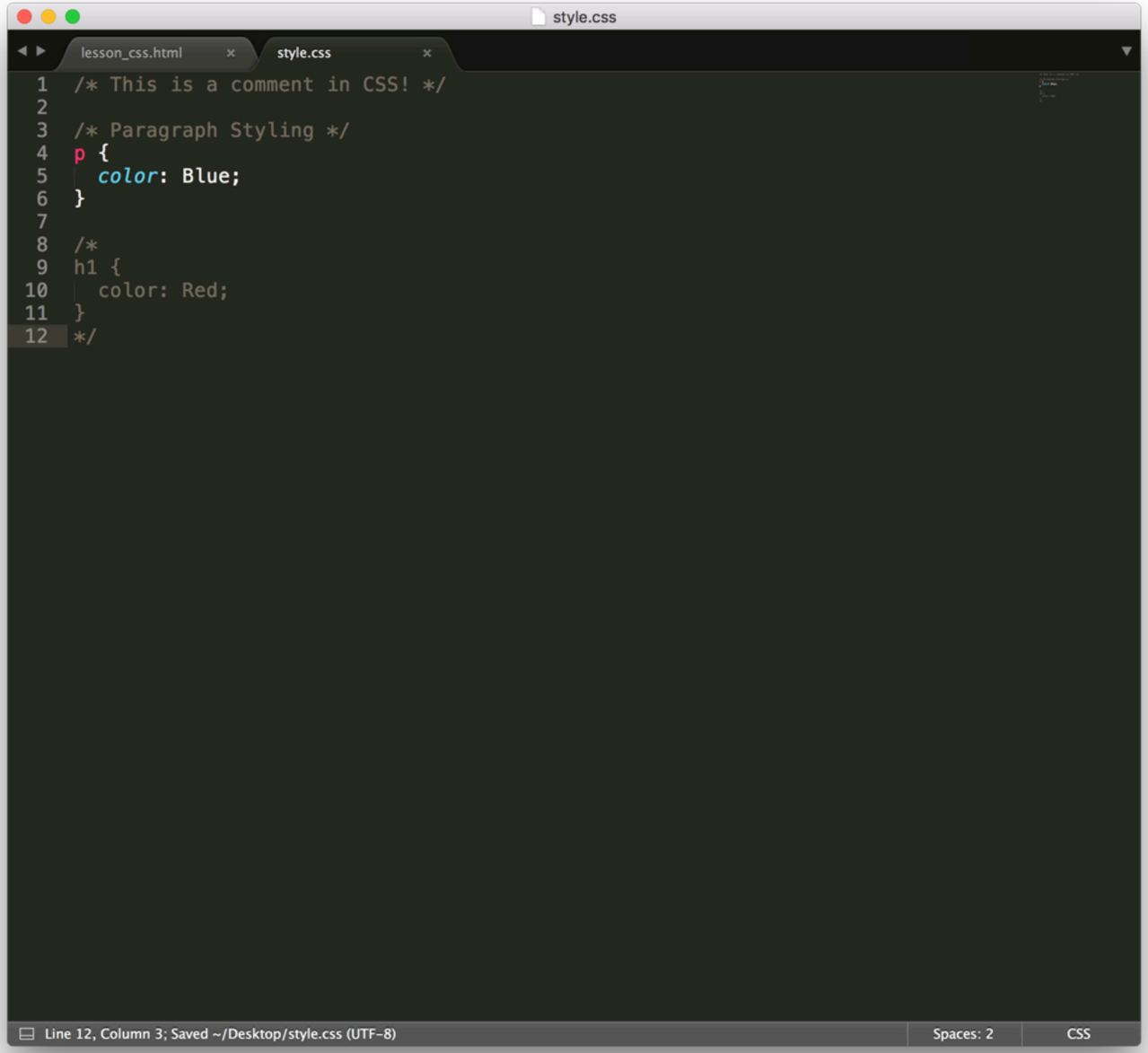
Line 22, Column 20 | Spaces: 2 | CSS

One space should be used between the selector and the opening curly brace ({).

No extra spaces should exist between opening and closing curly braces ({ and }) and CSS declarations.

Two spaces of indentation should be used for CSS declarations.

One line of spacing should exist between CSS rules.



Just like HTML, you can also leave comments in your CSS file. CSS comments begin with `/*` and end with `*/`. Including comments in your code is helpful for many reasons:

_They help you (and others) understand your code if you decide to come back and review it at a much later date.

_They allow you to experiment with new code, without having to delete old code.

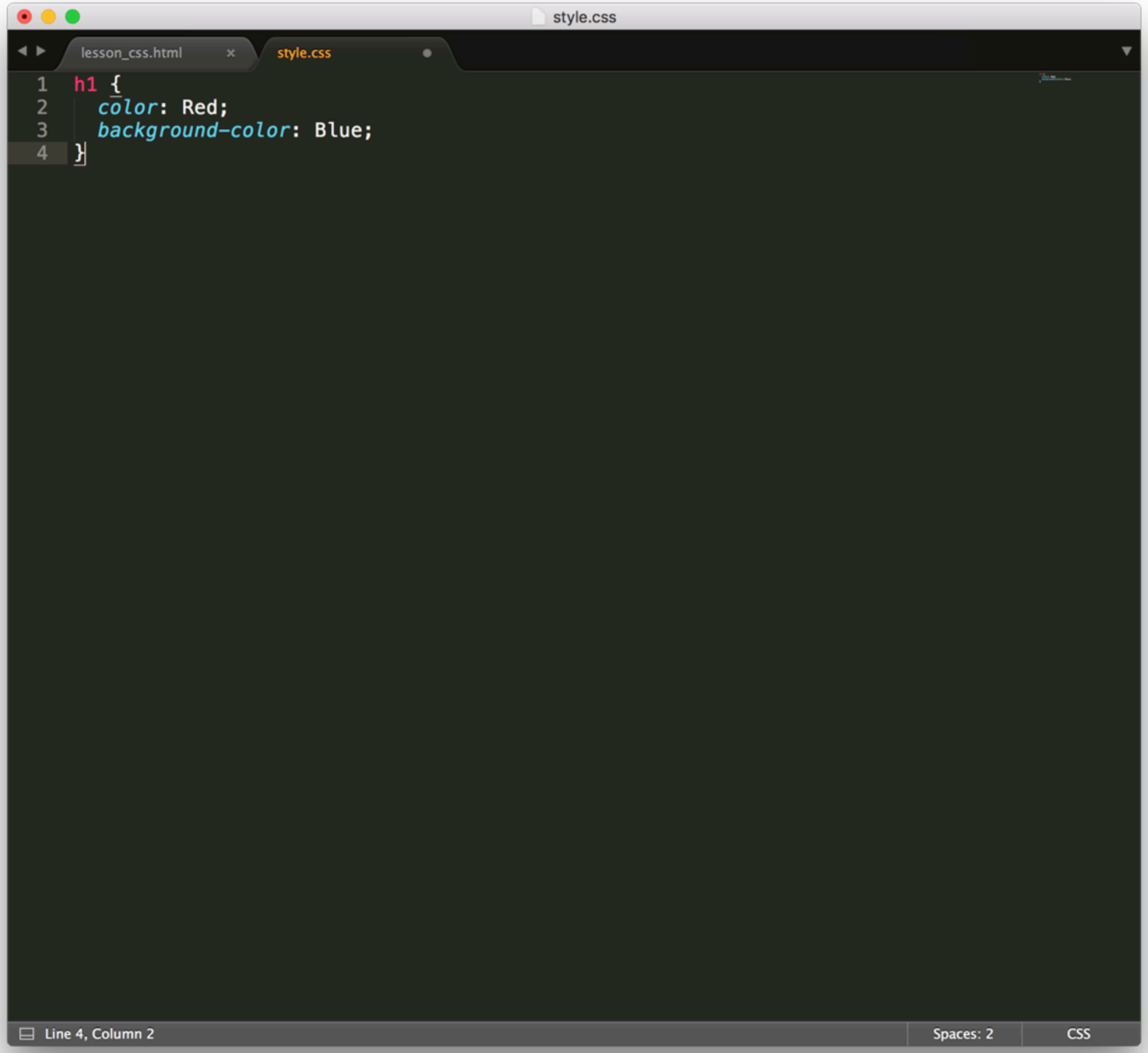
So far we’ve styled different properties of HTML elements, but the details of each property weren't explained. Let’s start by taking a look at color.

Before discussing the specifics of color, it's important to make two distinctions about color. Color can affect the following design aspects:

The foreground color
The background color

Foreground color is the color that an element appears in. For example, when a heading is styled to appear green, the *foreground color* of the heading has been styled.

Conversely, when a heading is styled so that its background appears yellow, the *background color* of the heading has been styled



In CSS, these two design aspects can be styled with the following two properties:

color - this property styles an element's foreground color.

background-color - this property styles an element's background color.

Over the past few exercises, you've seen CSS examples that use colors like Red, Blue, or Cyan. In CSS, these colors are technically known as *named colors*. There are a total of 147 named colors.

Although named colors provide 147 different options, this is a small amount when you consider the flexibility of CSS. To take advantage of the full spectrum of colors that CSS supports, you have the option of using *RGB colors*.

style.css

lesson_css.html | style.css

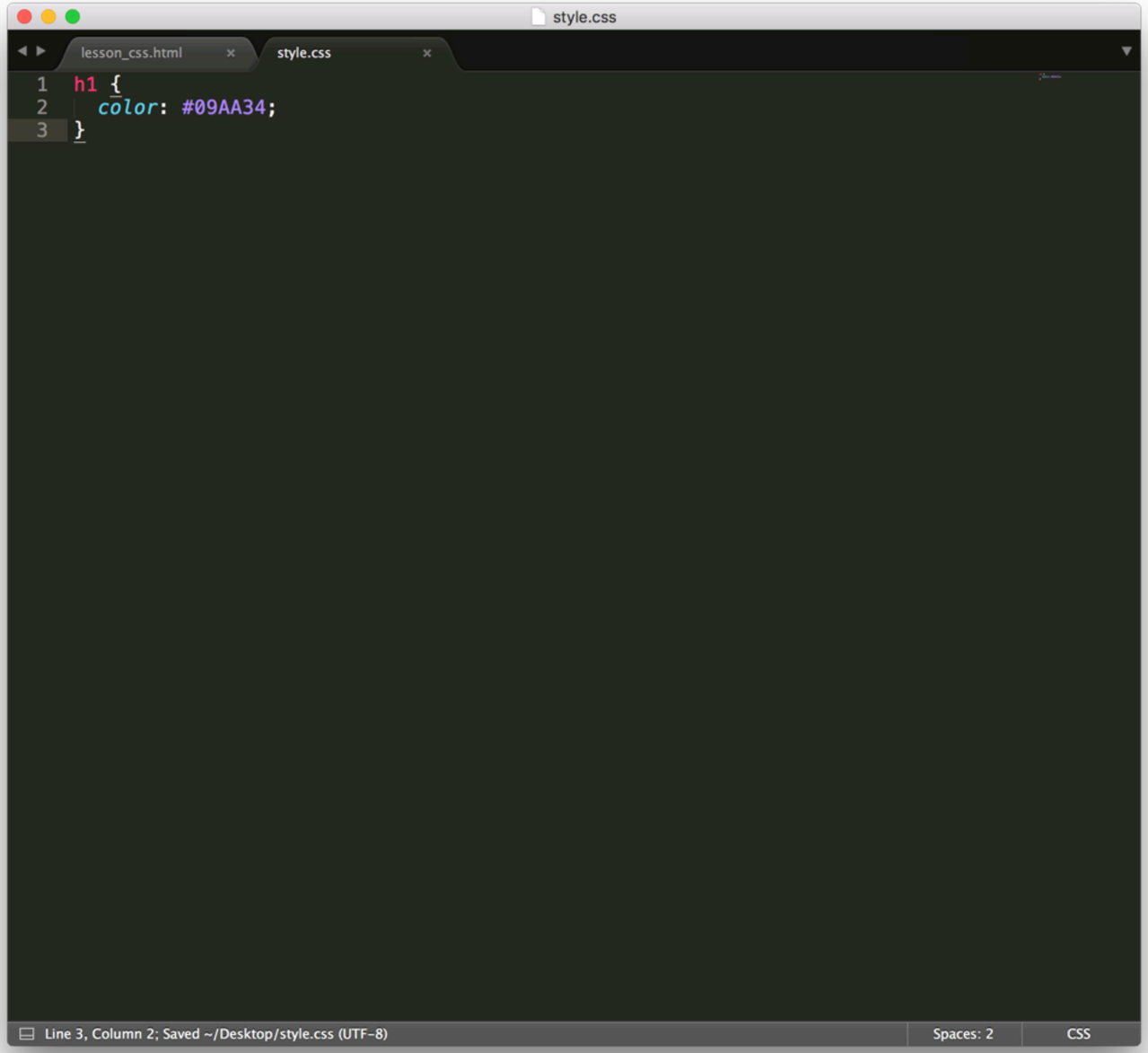
```
1 h1 {
2   color: rgb(123, 20, 233);
3   background-color: rgb(99, 21, 127);
4 }
```

Line 4, Column 2; Saved ~/Desktop/style.css (UTF-8) | Spaces: 2 | CSS

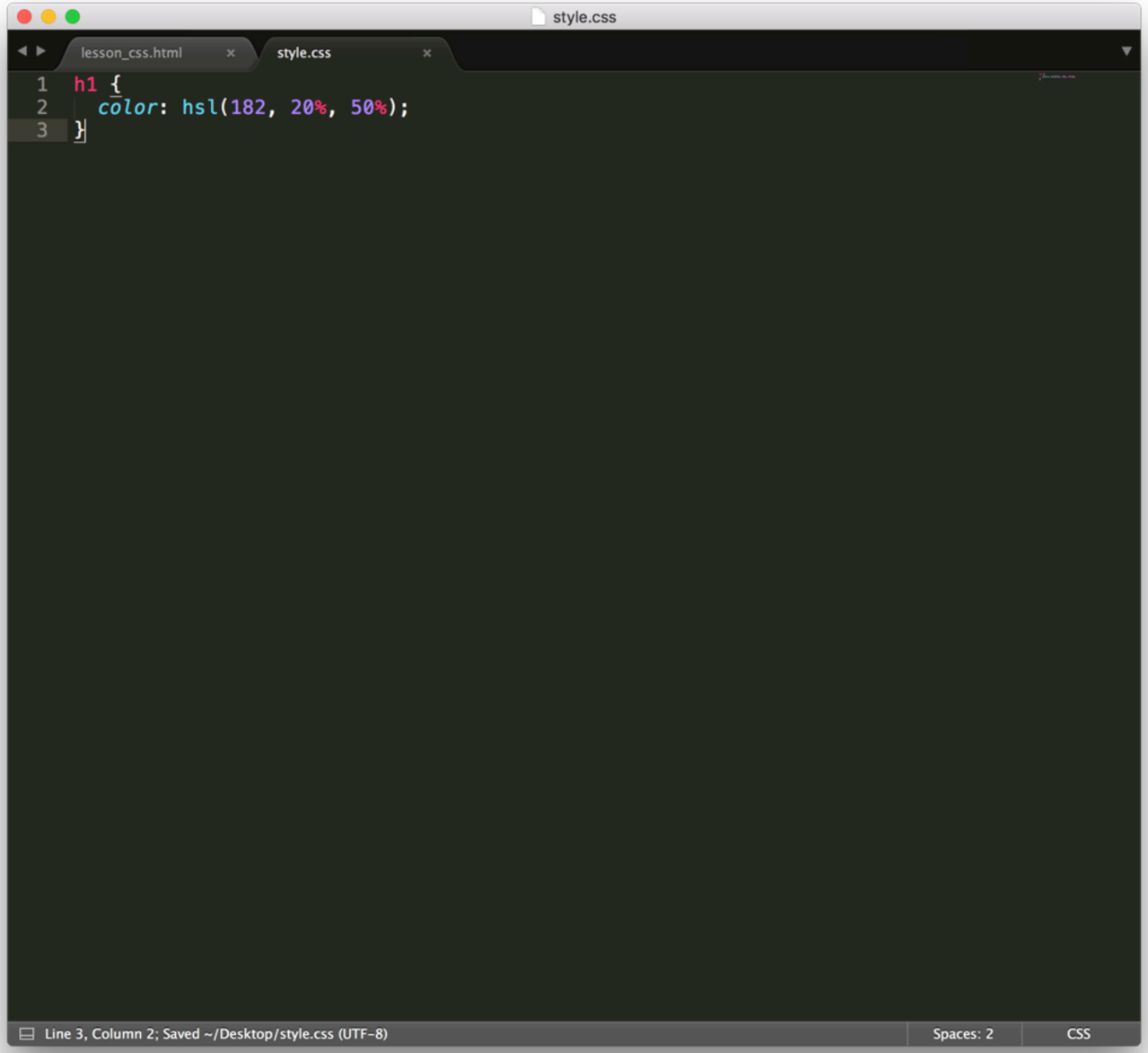
To use RGB colors, you can use the rgb() value when styling a color.

In this example, the value of color is set to rgb(). The three numbers in the parentheses represent the values for R, G, and B, in that order. Note that you can use rgb() for background colors as well.

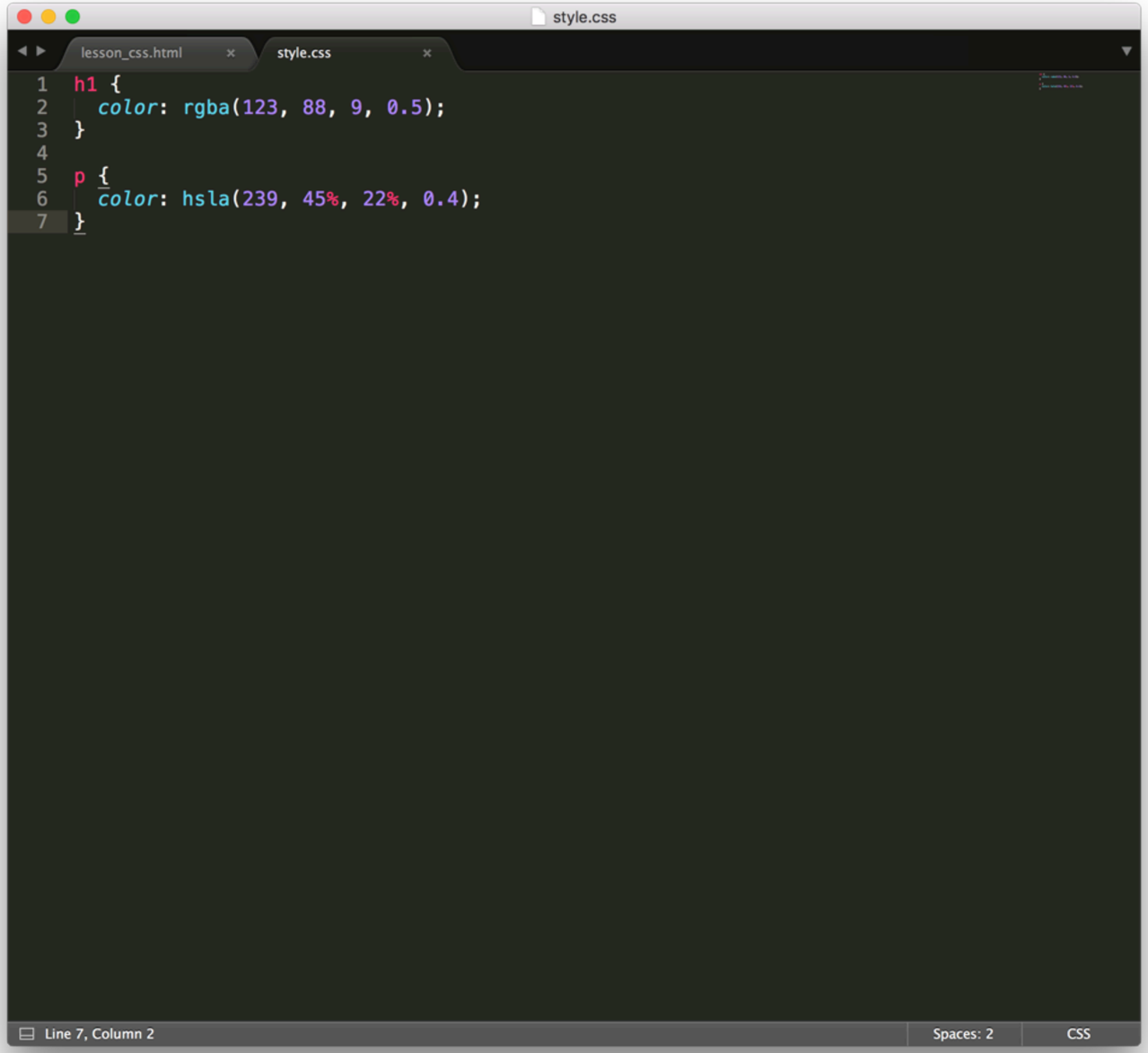
There's an additional way to specify colors in CSS: *hexadecimal color codes*, often referred to as "hex color codes" for short.



When read from left to right, each group of two characters responds to a value for red, green and blue, respectively. In the example above, 09 refers to the value for red, AA refers to the value for green, and 34 refers to the value for blue. All hex color codes begin with a # character.



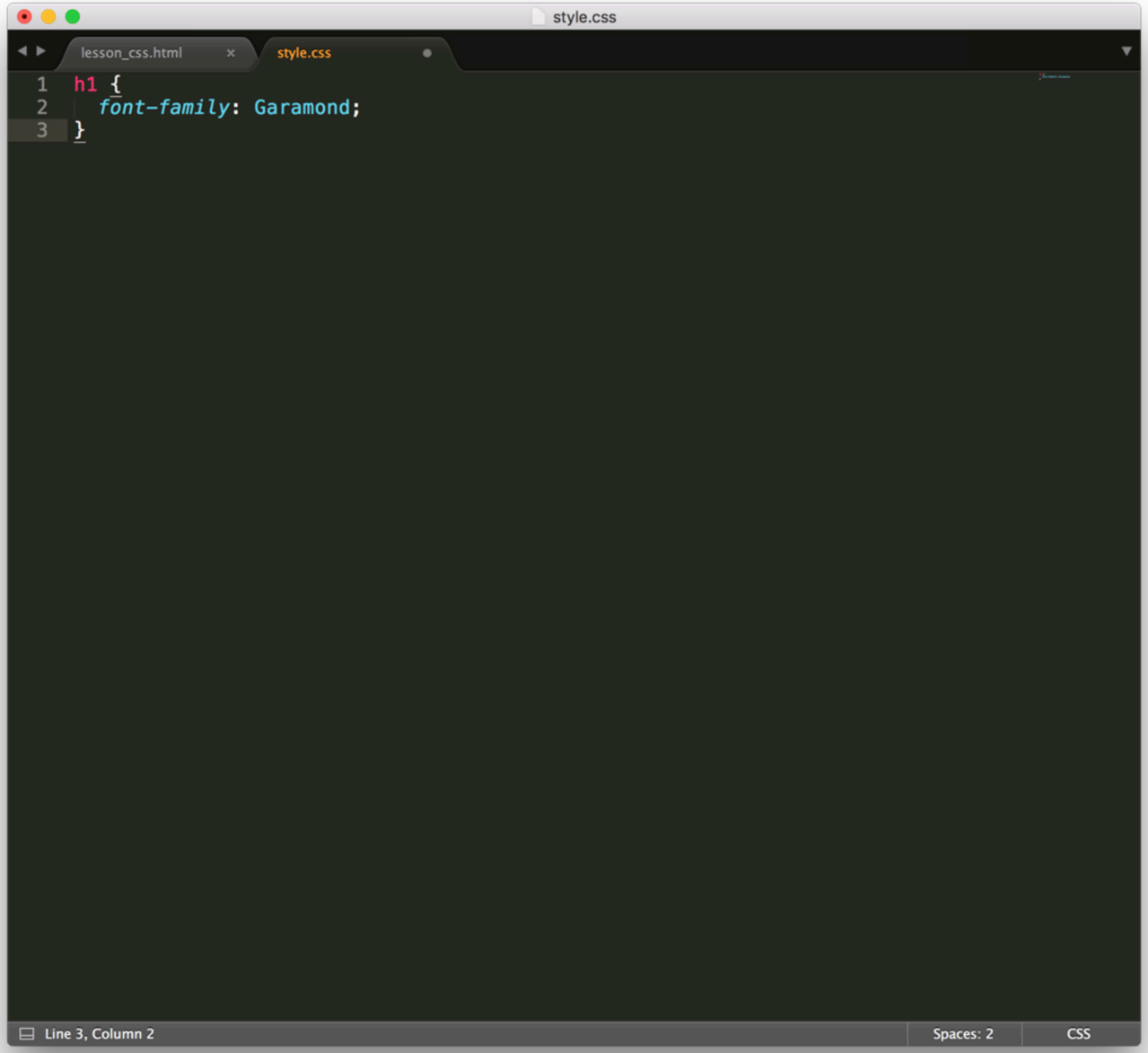
Here’s an example of hsl color.
Very similar to rgb syntax.



Opacity is a measure of how transparent a color is. To modify opacity in RGB colors, CSS offers the rgba() value. Note the slight difference in rgb() and rgba().

The extra a character in the rgba() value is known as the *alpha value*. It represents the opacity of a color. The alpha value can be a number between 0 or 1, inclusive.

Now let’s take a look at fonts and typography



To change the typeface of text on your web page, you can use the font-family property.

When setting typefaces on a web page, keep the following points in mind:

The font specified in a stylesheet must be installed on a user's computer in order for that font to display when a user visit the web page. We'll learn how to work around this issue soon.

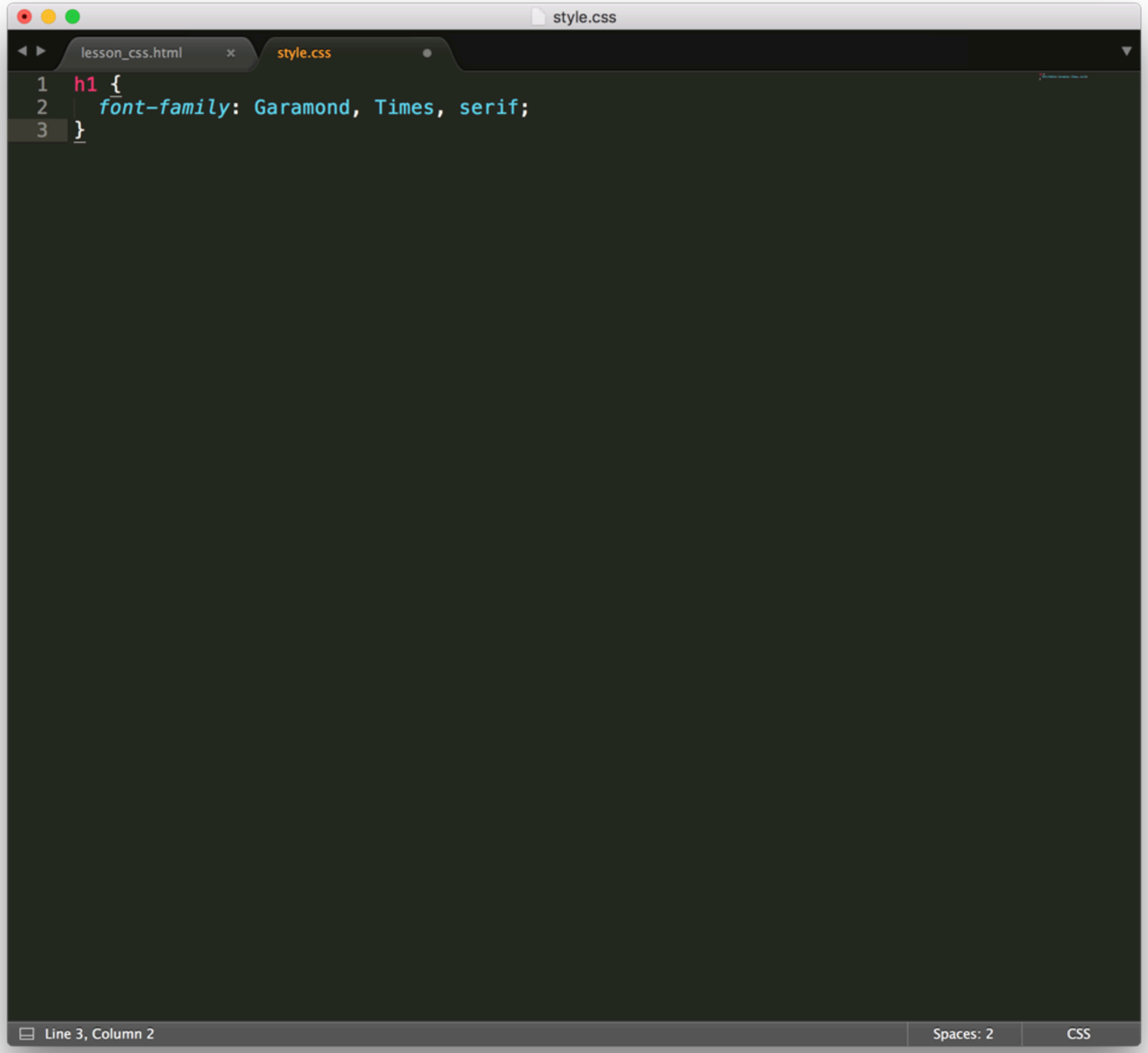
How exactly does the browser know what typeface to use when displaying the web page? The default typeface for all HTML elements is Times New Roman.

When the name of a typeface consists of more than one word, it must be enclosed in double quotes (otherwise it will not be recognized), like so:
"Courier New”

What happens when a font is not installed on a user's computer?

Most computers have a small set of typefaces pre-installed. This small set includes serif fonts and sans-serif fonts, like Times New Roman and Arial, respectively.

When the stylesheet specifies a font not installed on a user's computer, the pre-installed fonts can be used as *fallback fonts* for users.



The CSS rule above says: "Use the Garamond font for all <h1> elements on the web page. If that font is not available, use the Times font. If both of those fonts are not available, use any serif font pre-installed on the user's computer." The fonts specified after Garamond are the fallback fonts.

It would be unrealistic to expect users to have all the fonts you might want to design with installed on their computers.

Fortunately, you don't have to predict which fonts are installed on a user's computer.

Google offers Google Fonts, a directory of thousands of open-source fonts that are free to use by anyone.

To use these fonts, you can link to a specific Google Font in your HTML code and use it immediately in your stylesheet. Because the HTML file links directly to the Google Font, a user's browser can display the typeface you specify. This avoids having to determine whether or not that font is installed on a user's computer.

lesson_css.html

lesson_css.html | style.css

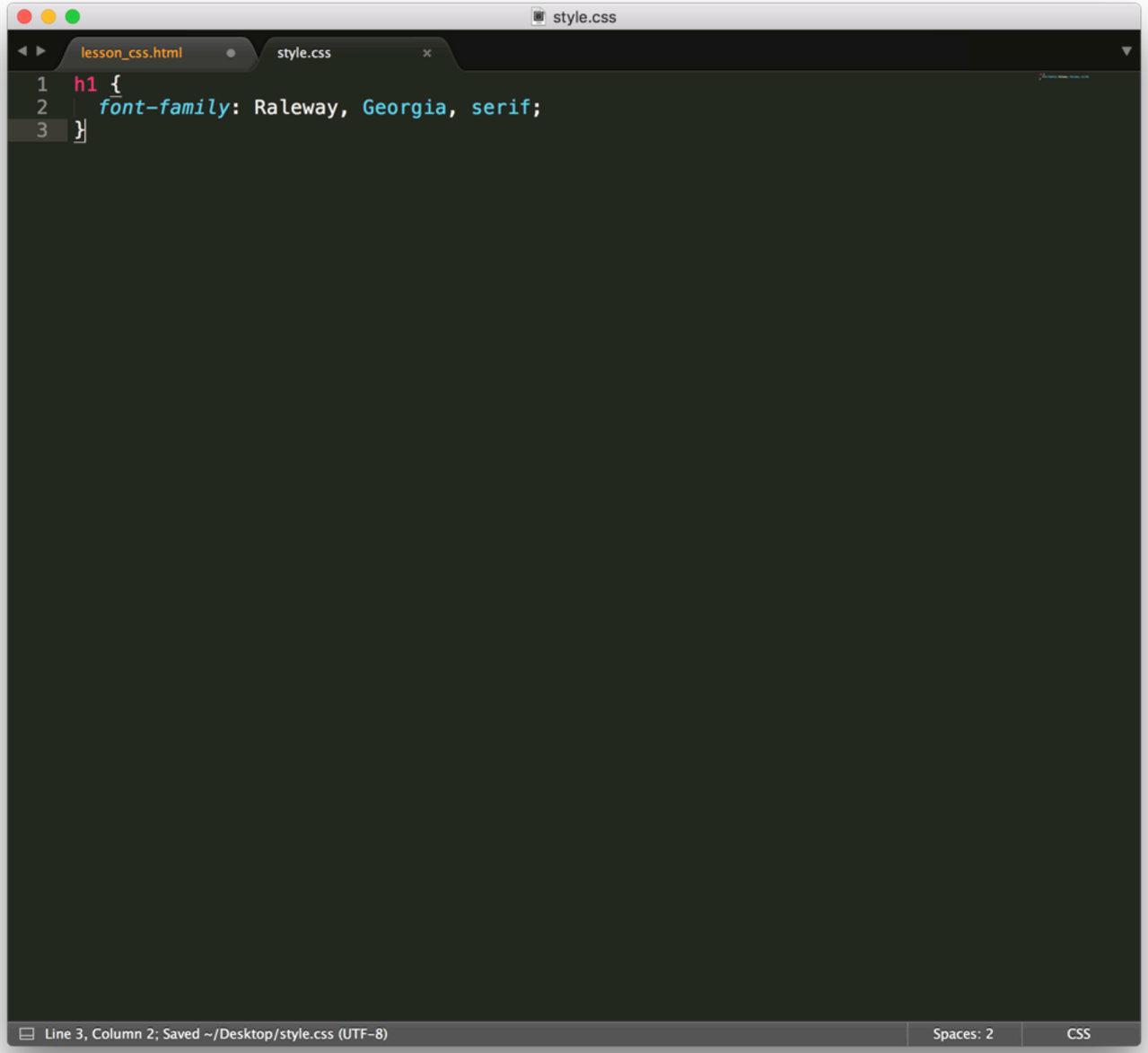
```
1 <head>
2   <link href="https://fonts.googleapis.com/css?family=Raleway" type="text/css"
    rel="stylesheet" >
3 </head>
```

Line 3, Column 8 | Spaces: 2 | HTML

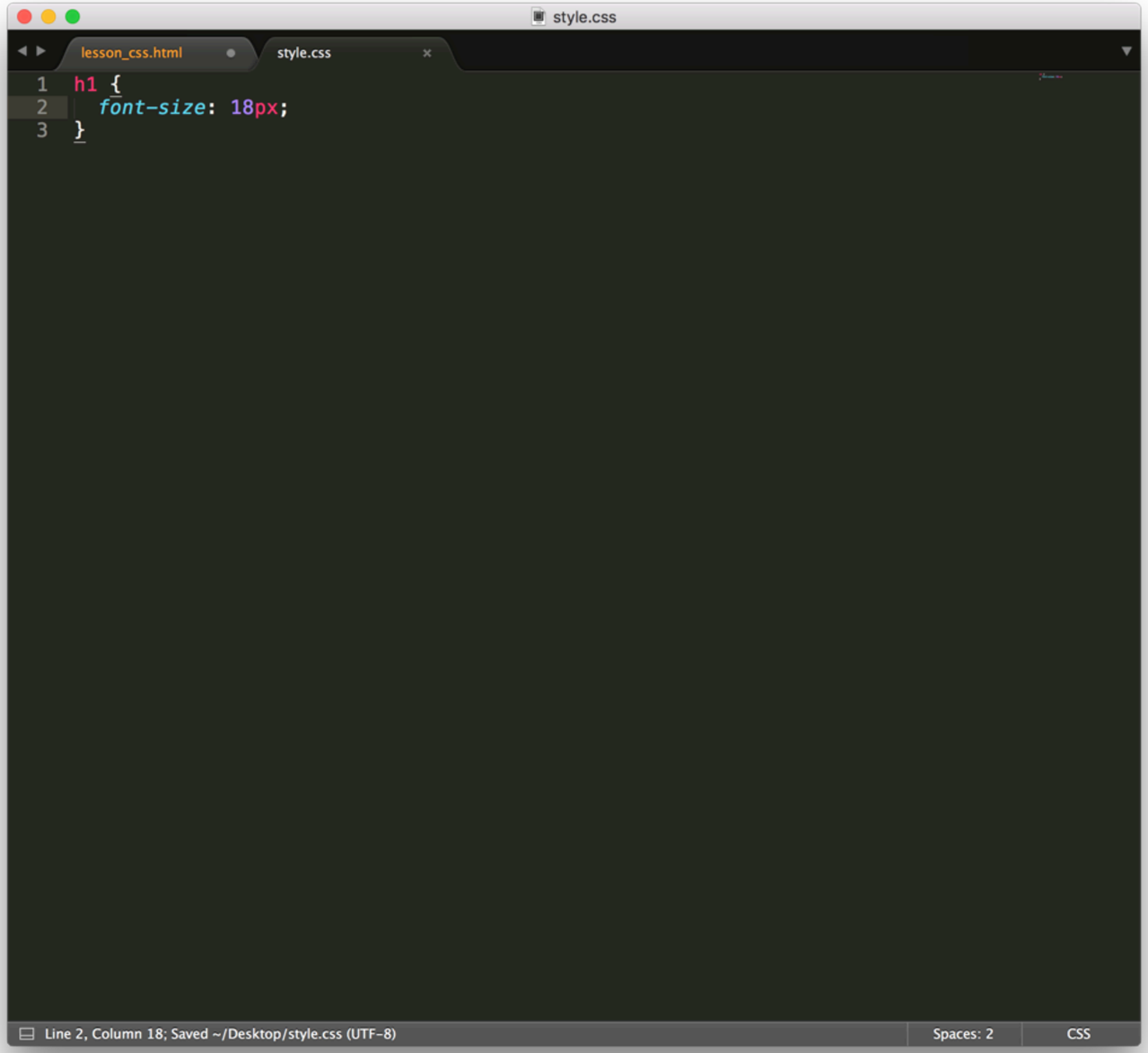
The **href** attribute is set to the following URL, which was retrieved from Google Fonts:

https://fonts.googleapis.com/css?family=Raleway

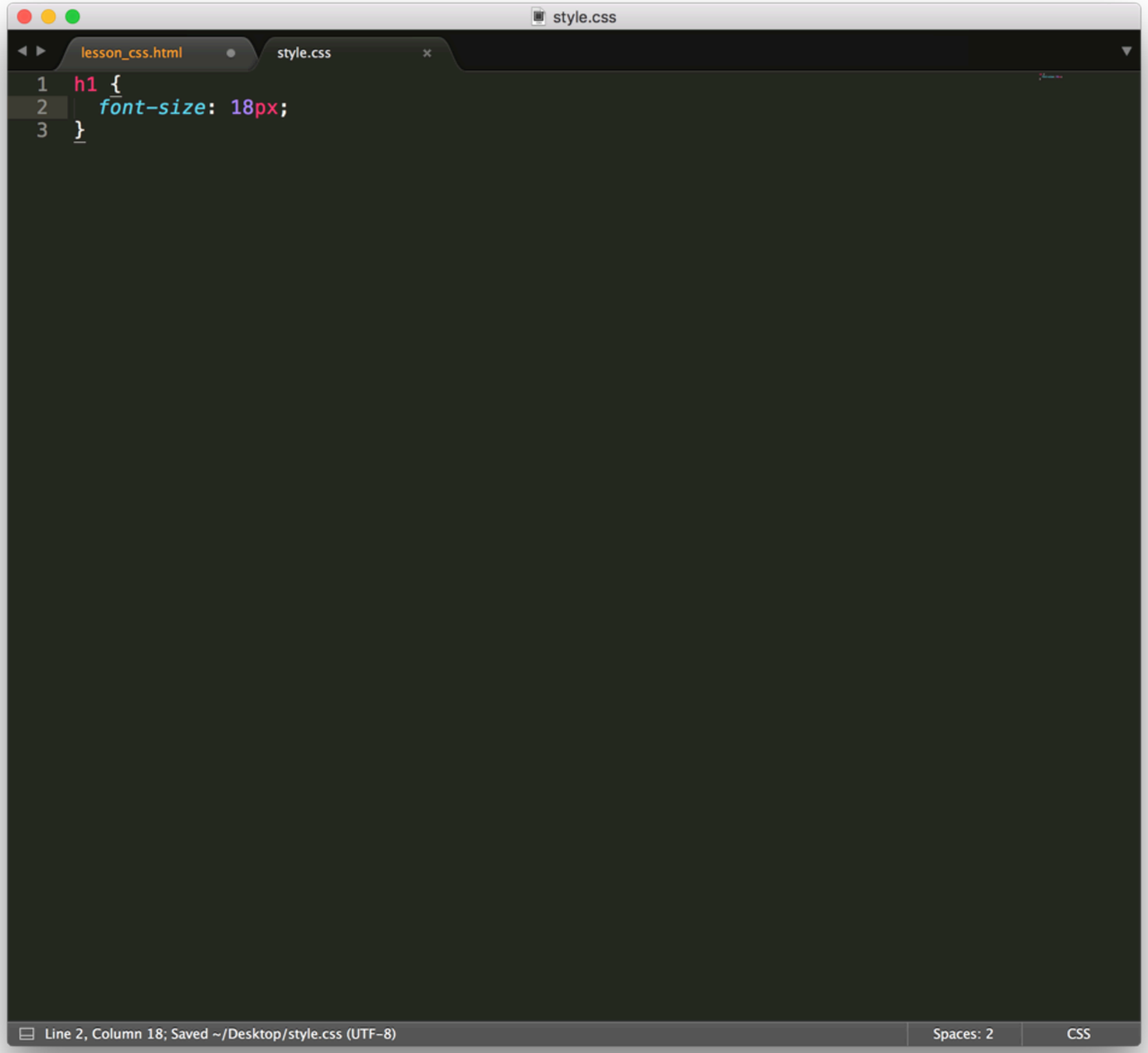
The URL in the example above specifies the **Raleway** typeface from Google Fonts.



You can now use this just as you would any other font.

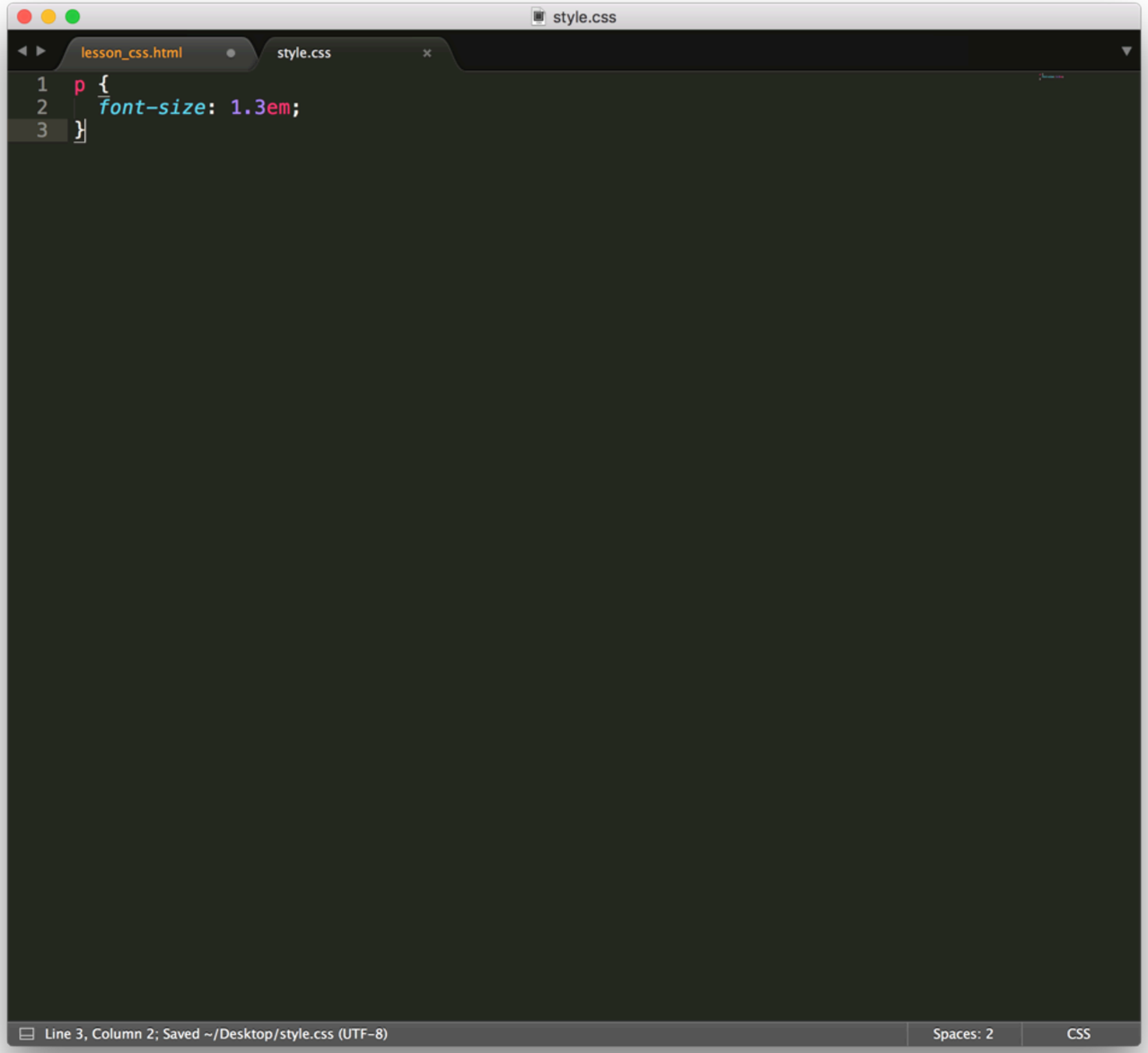


To change the size of text on your web page, you can use the font-size

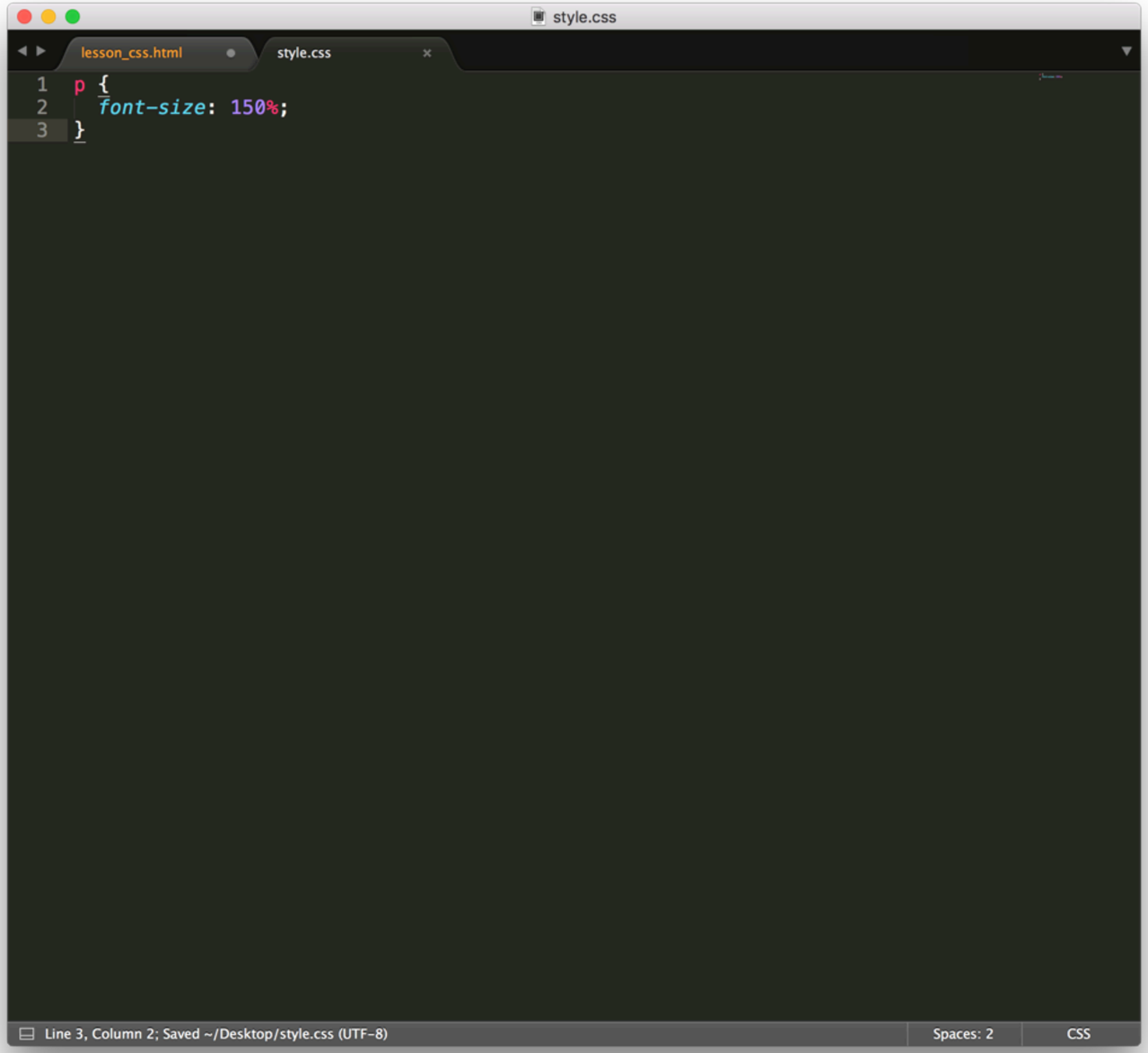


There are three units of measurement for font size:

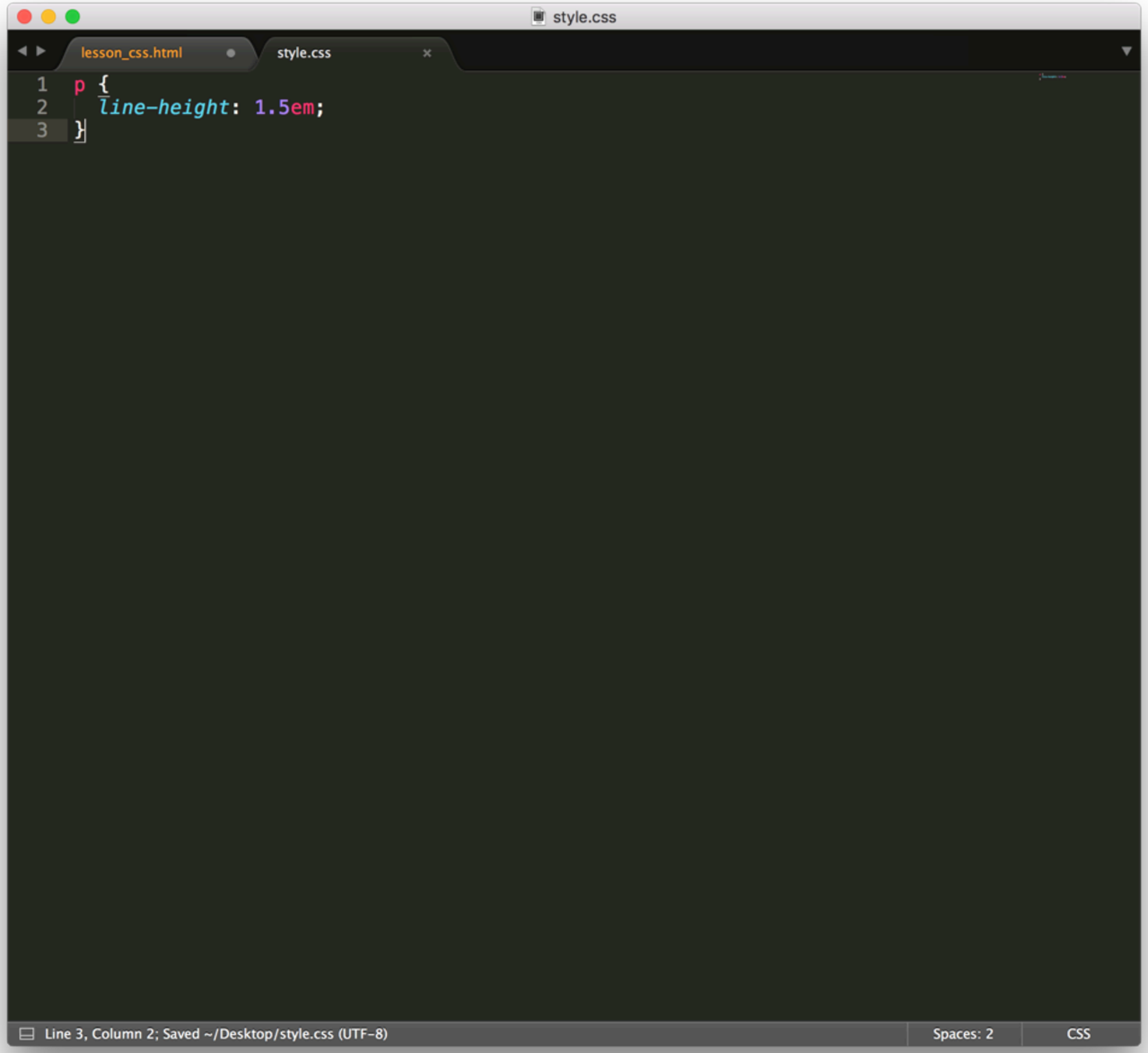
px – Represents the unit of *pixels*. The display of a computer monitor can be measured in pixels. A pixel is a small point on the display. Most computer monitors have a resolution of 72 pixels per inch, so a pixel represents about 1/72nd of an inch. Pixels are sometimes also referred to as *points*. Pixels are used to set the exact size of an element.



em – Pronounced just as it looks, "em." An em is equal to the width of the letter "m". Ems are a relative unit of measurement. They change the size of text relative to the parent element's size of text.



% – Percentages are also a relative unit of measurement. The default size of text in web browsers is 16 pixels, or 16px. When percentages are used, they set the size of text relative to this default size. For example, setting the font size to 200% would be equivalent to setting it to 32px.

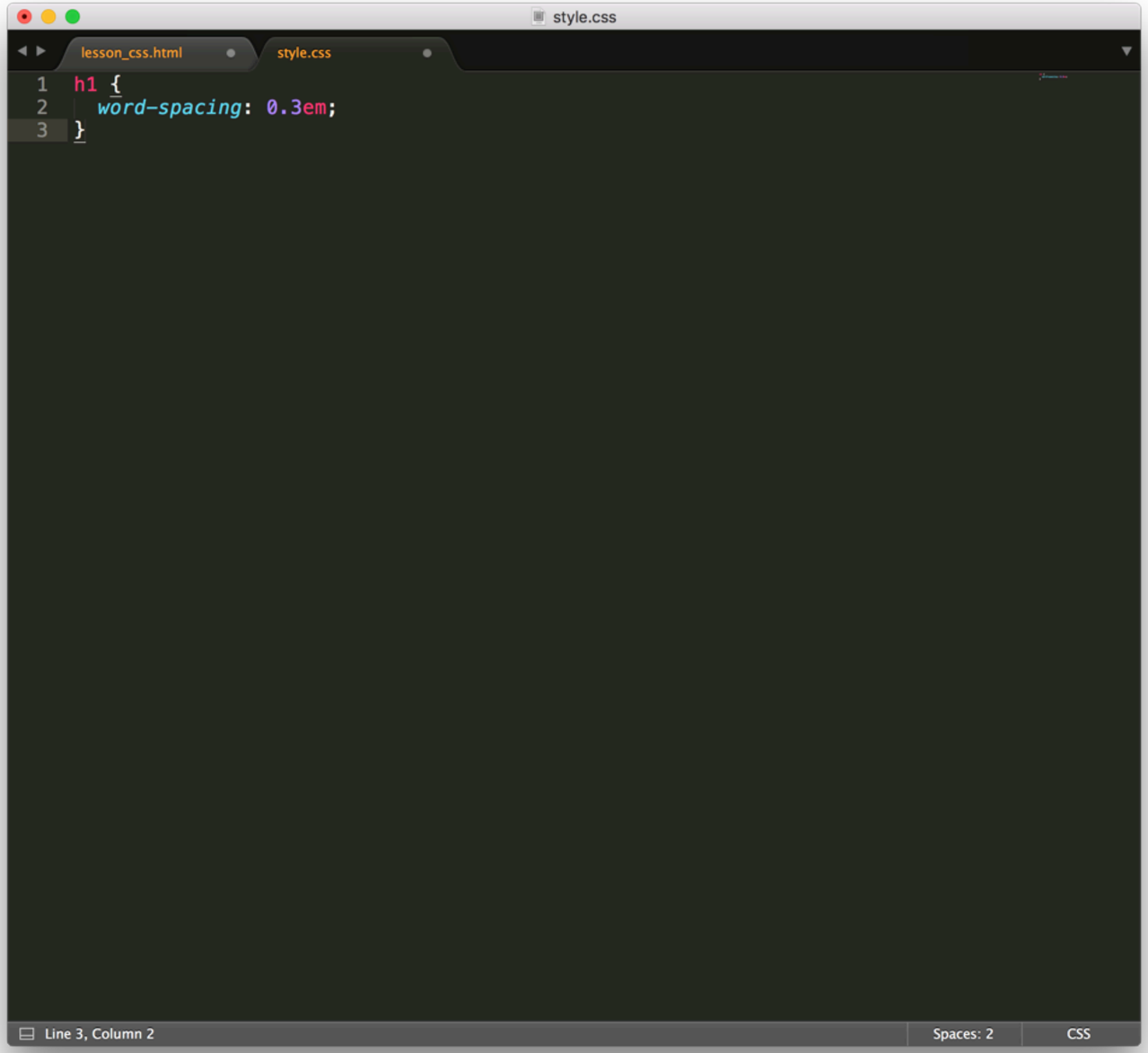


You can modify the spacing between lines of text with the line-height property.

When the line height of an element is modified, you are manipulating the *leading* (pronounced "ledding") of the font. When the line height is increased, the spacing between lines of text increases.

The line height can be modified using pixels or ems, but the unit of ems is preferred, since ems offer a spacing relative to the size of the text on the page.

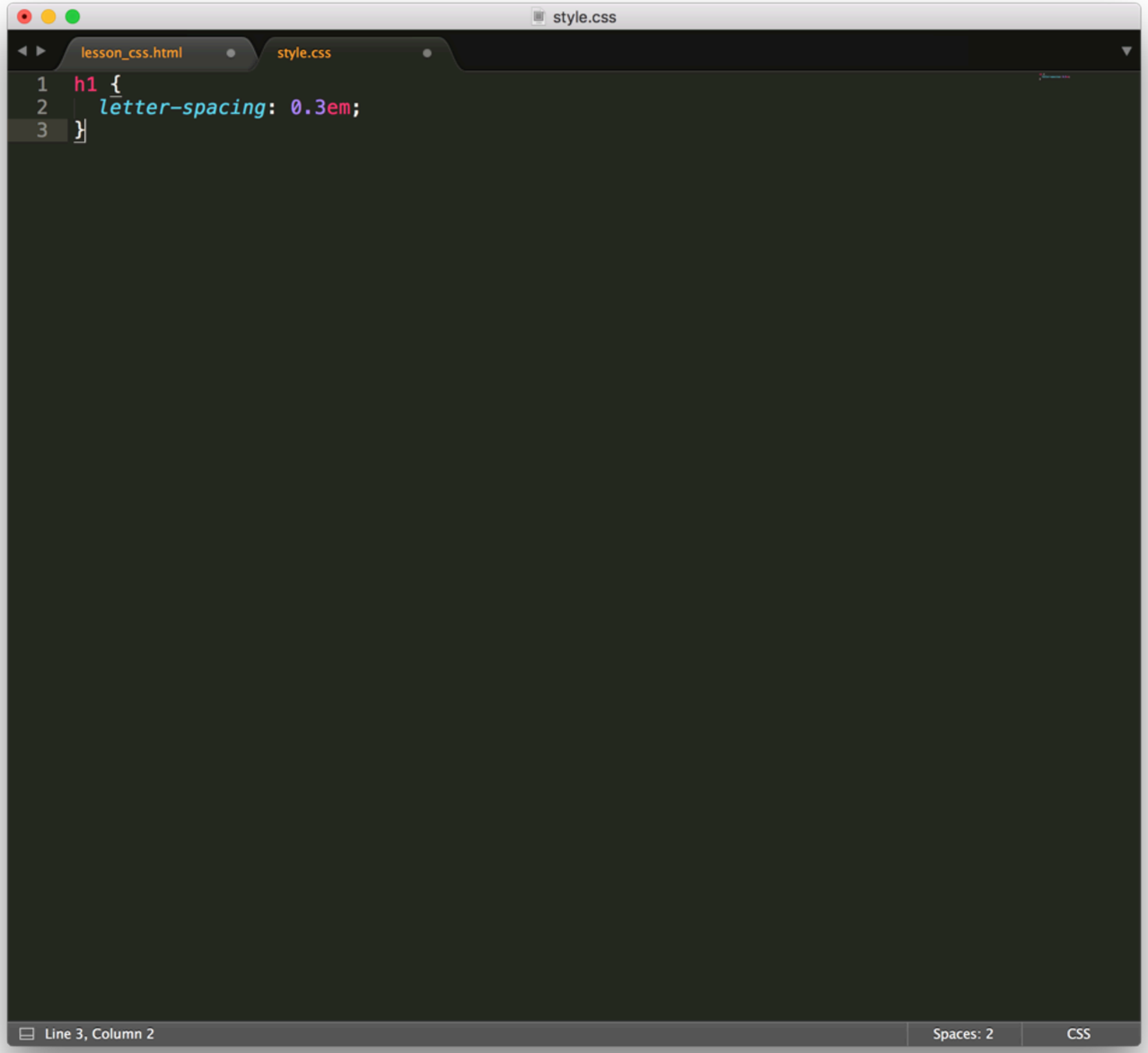

The fastest cat
can race at 75
leading
font size
line height



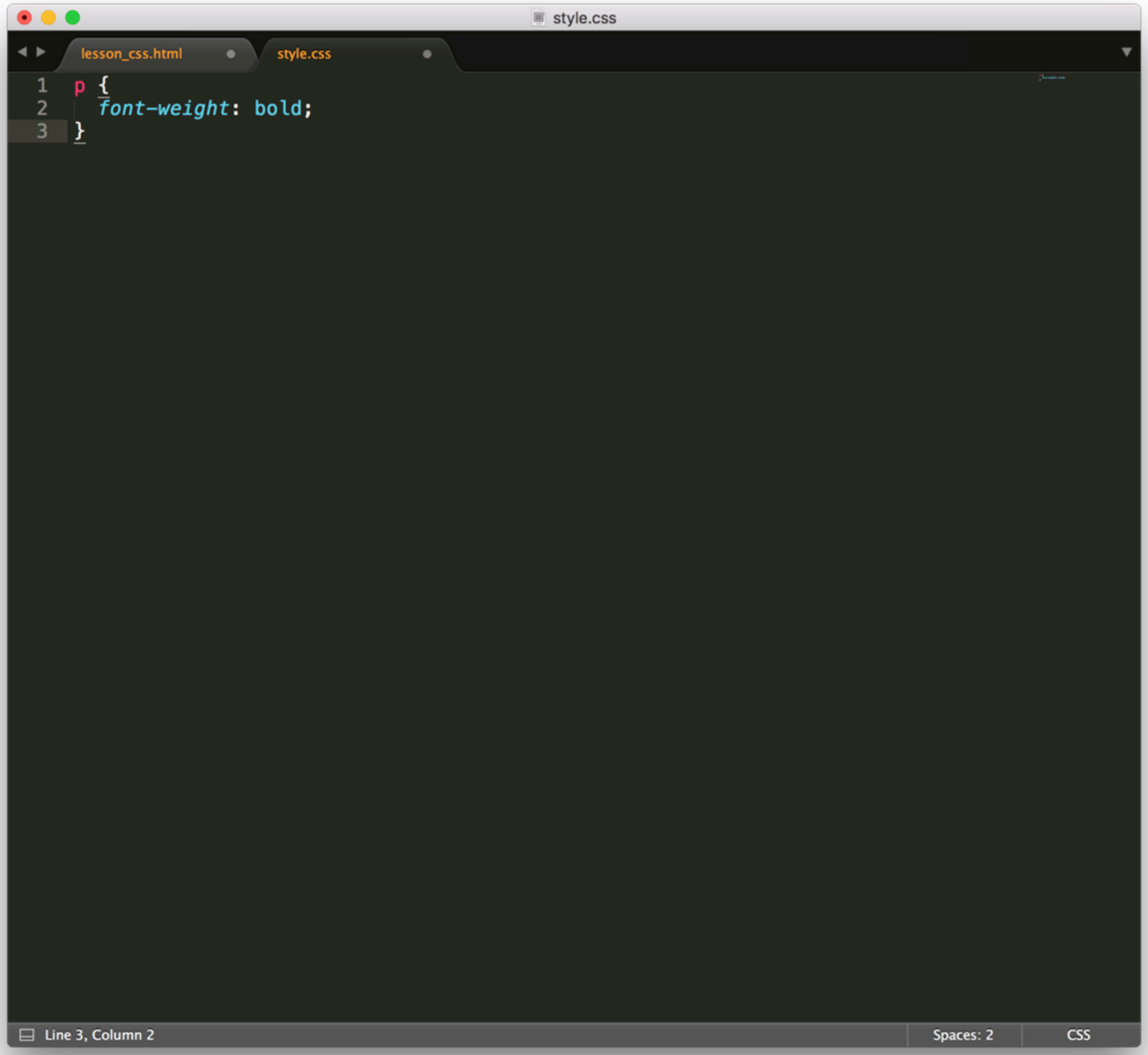
You can also increase the spacing between words in a body of text, technically known as *word spacing*.

To do so, you can use the word-spacing property.

The default amount of space between words is usually 0.25em. In this example above, the word spacing is set to 0.3em, which represents an increase of only .05em in word spacing.



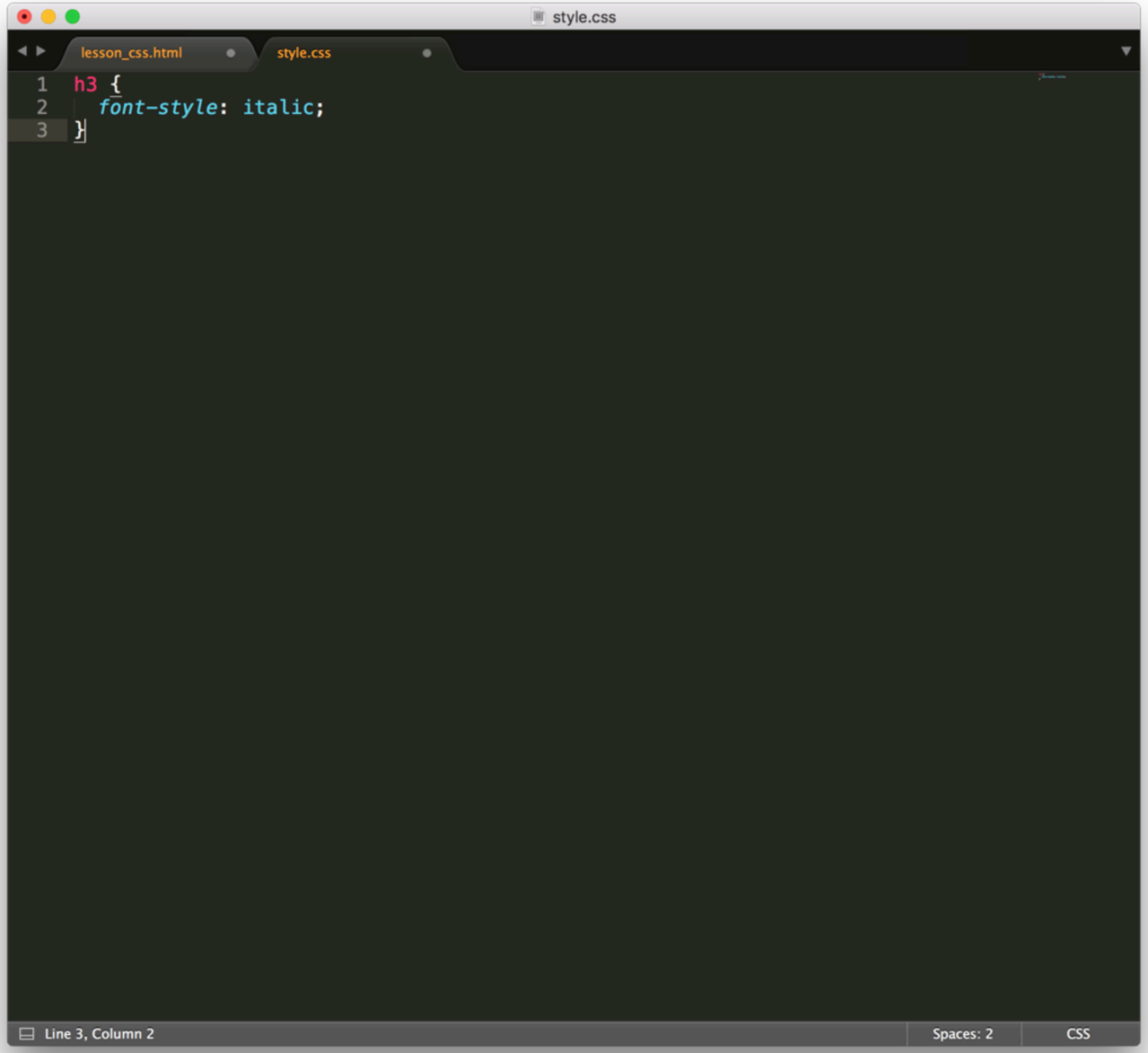
The technical term for adjusting the spacing between letters is called "kerning". Kerning can be adjusted with the letter-spacing property in CSS



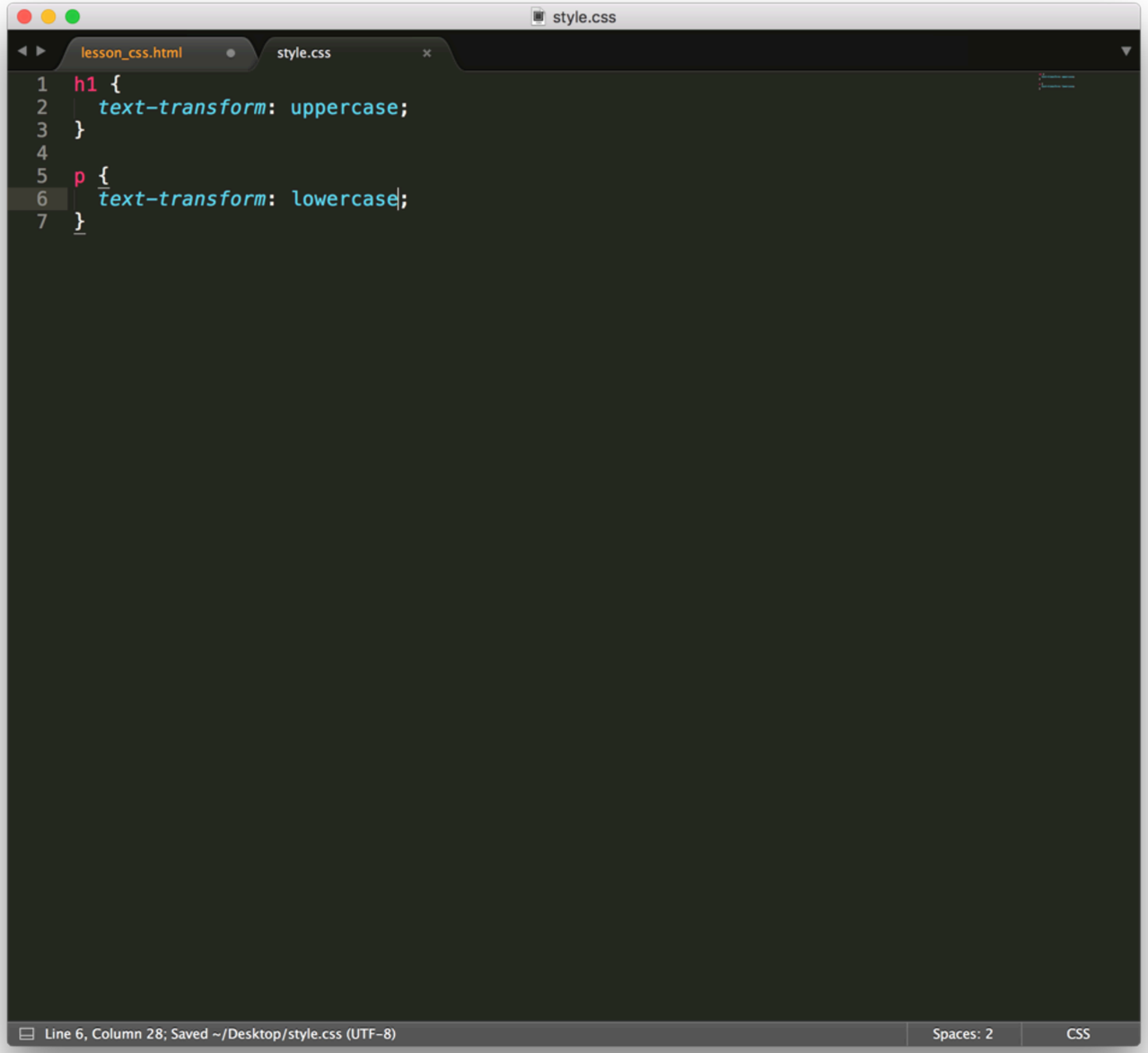
font-weight property turns bold on or off.

The font-weight property has a second value: normal. Why does it exist?

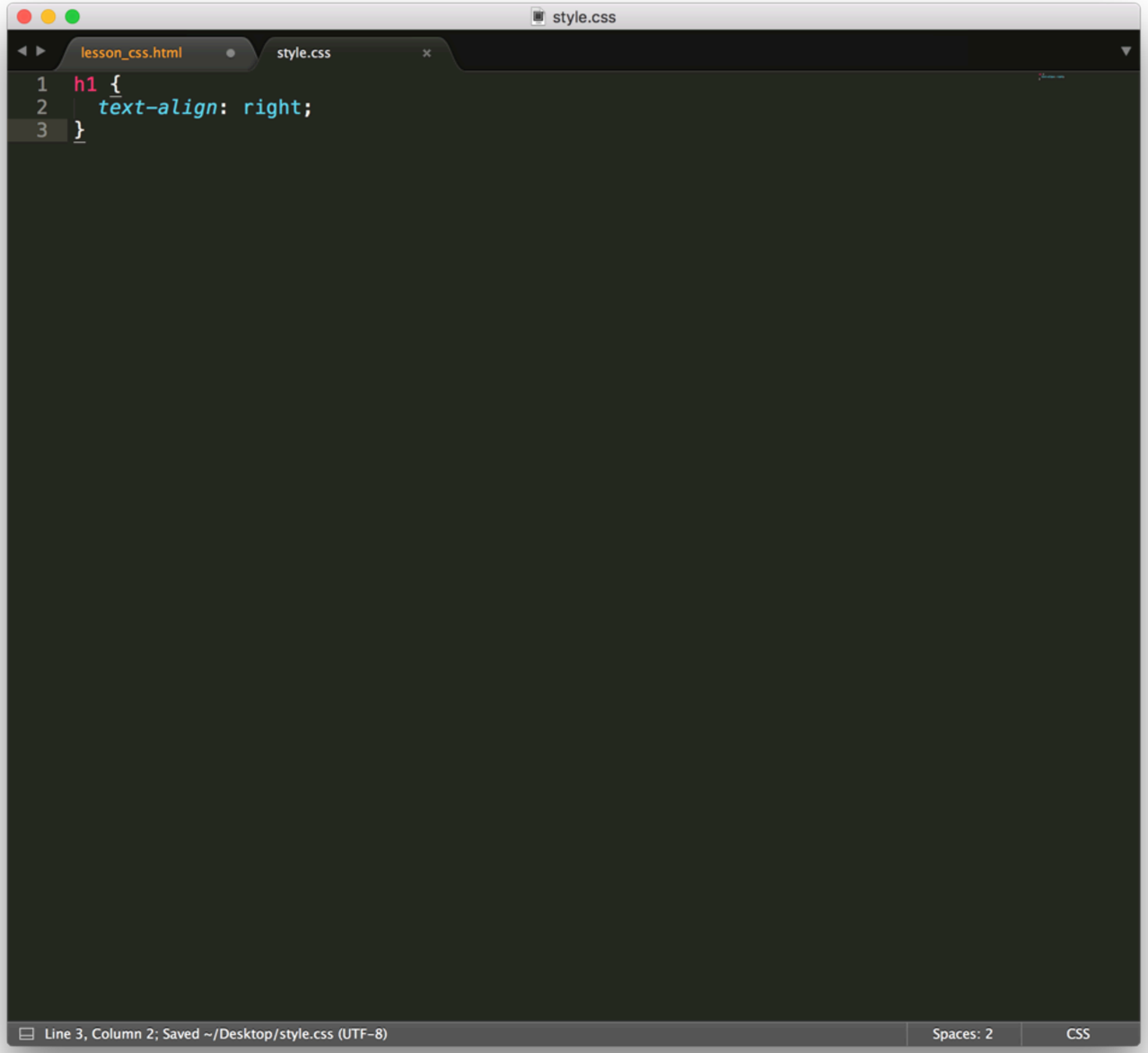
If we wanted *all* text on a web page to appear bolded, we could select all text elements and change their font weight to bold. If a certain section of text was required to appear normal, however, we could set the font weight of that particular element to normal, essentially "shutting off" bold for that element.



You can also *italicize* words with the font-style property



Text can also be styled to appear in either all uppercase or lowercase with the text-transform property.



To move, or align, text, we can use the text-align property.

The text-align property can be set to one of the following three values:

left - aligns text to the left hand side of the browser.

center - centers text.

right - aligns text to the right hand side of the browser.