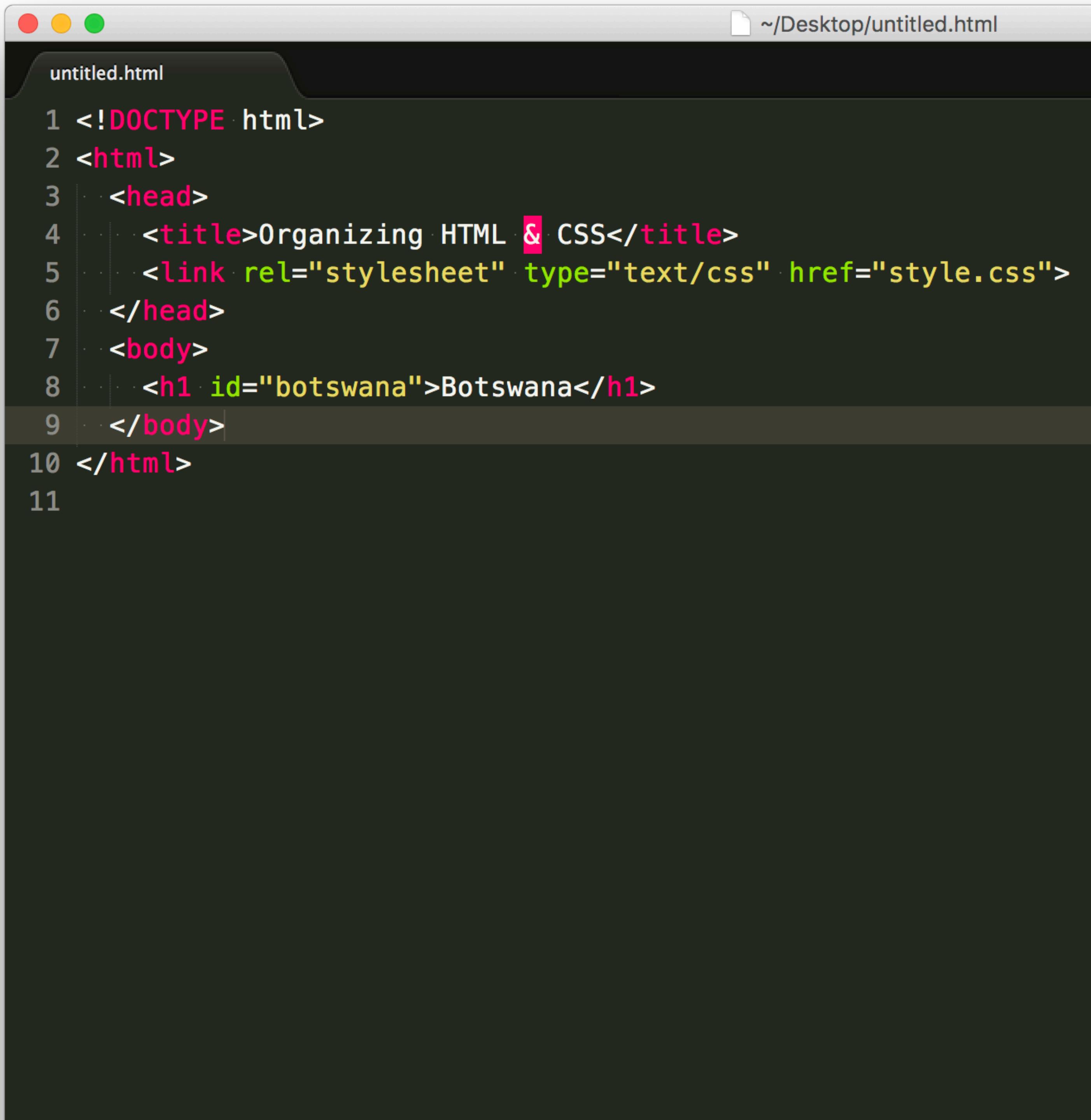


CSS Part II

Up until this point we've been styling every type of a certain element. For example, all the paragraphs or all the h1 headers.

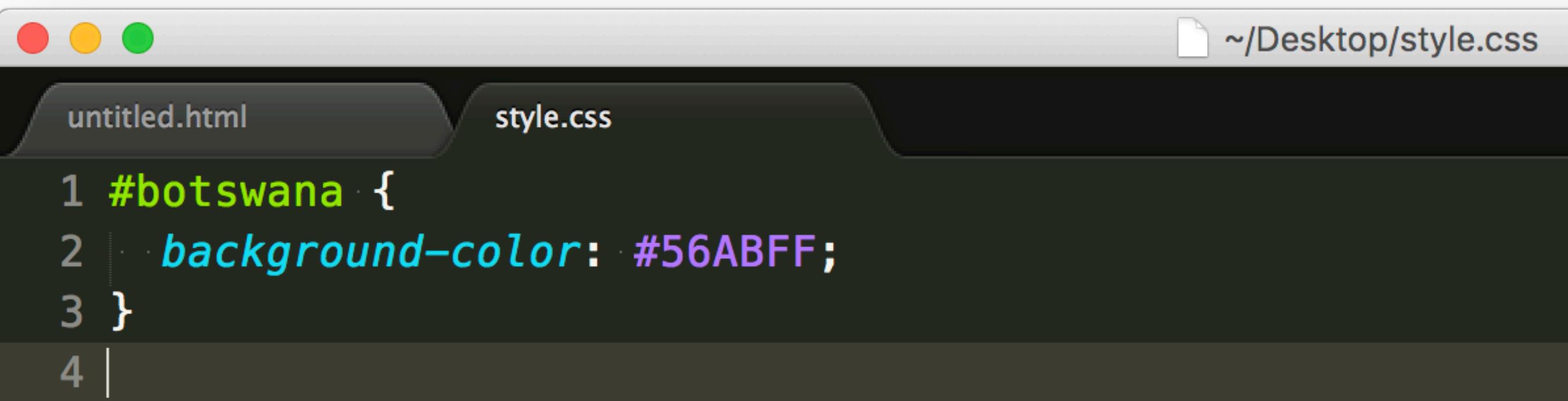
With the proper labels, we can style individual HTML elements. Specifically, we can label HTML elements with a unique identifier, or *ID*. We can then style that specific element in the stylesheet.



A screenshot of a Mac OS X desktop environment. In the top-left corner, there are three colored window control buttons: red, yellow, and green. The title bar of the active window is dark grey with white text. It shows a small icon of a document with a folder symbol, followed by the path `~/Desktop/untitled.html`. Below the title bar, the main content area is a terminal window with a black background and white text. The terminal window has a tab bar at the top with one tab labeled "untitled.html". The code in the terminal is:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Organizing HTML & CSS</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7   <body>
8     <h1 id="botswana">Botswana</h1>
9   </body>
10 </html>
11
```

To label an element with an ID, we can use the **id** attribute on an HTML element.



A screenshot of a Mac OS X desktop environment. At the top, there are three red, yellow, and green window control buttons. Below them is a dark grey dock bar with two icons: "untitled.html" and "style.css". The main window title bar is also dark grey and displays the file path "~/Desktop/style.css". The main content area of the window shows a text editor with the following CSS code:

```
1 #botswana {  
2     background-color: #56ABFF;  
3 }  
4 |
```

To style a specific element labeled with an ID, you can use an *ID selector* in the stylesheet.

The ID selector always starts with a hash symbol.

IDs are great for labeling unique elements, but IDs have a limitation. Because unique IDs should not be used across multiple HTML elements, they are insufficient for quickly styling elements that should all share a specific style.

CSS offers a solution to this limitation. For multiple elements that should share the same styling, *classes* can be used to label them.



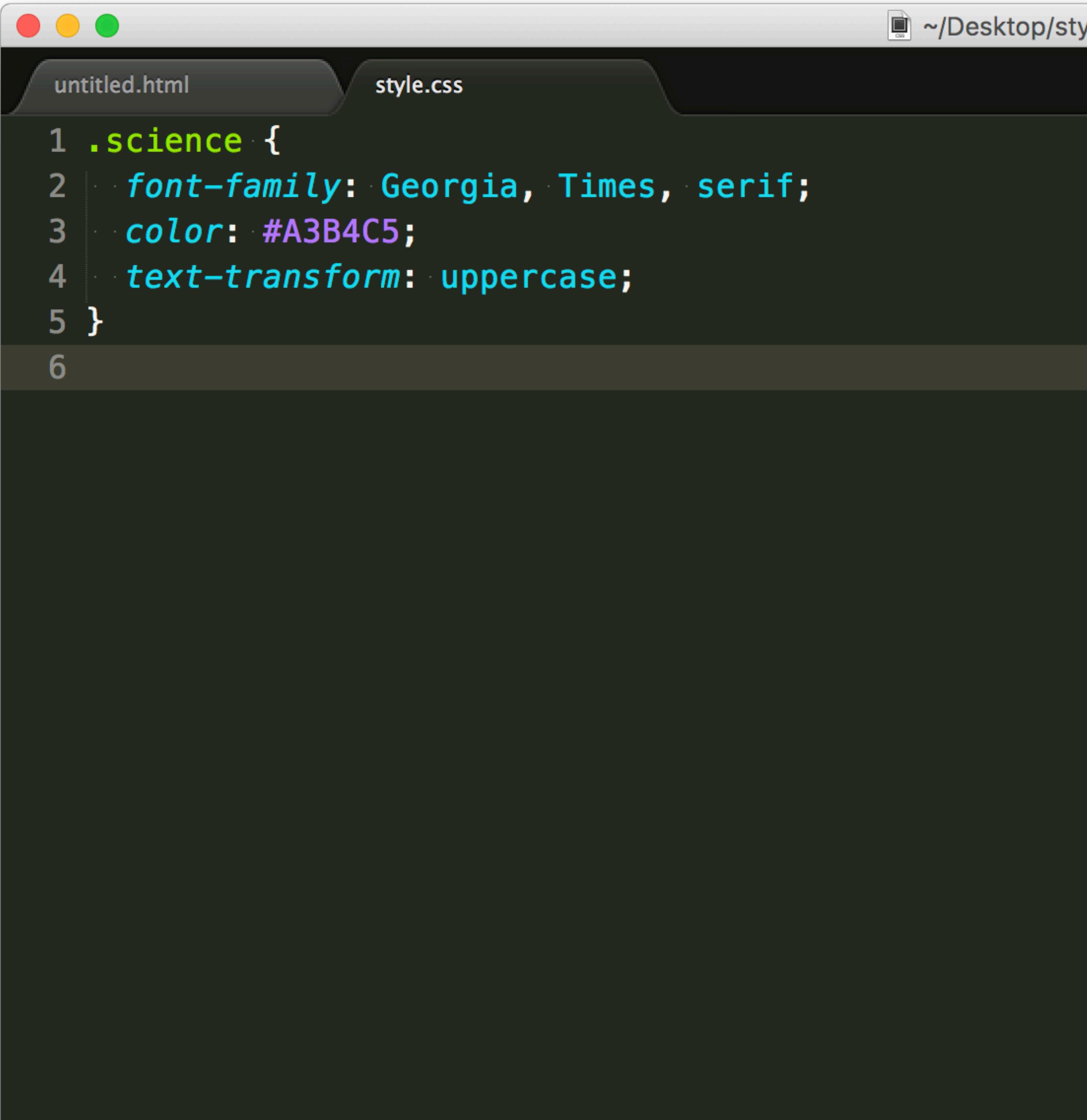
untitled

LICENSE UPGRADE REQUIRED

untitled

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>title</title>
6   </head>
7   <body>
8
9     <h1 class="science"></h1>
10
11   </body>
12 </html>
```





```
untitled.html style.css
```

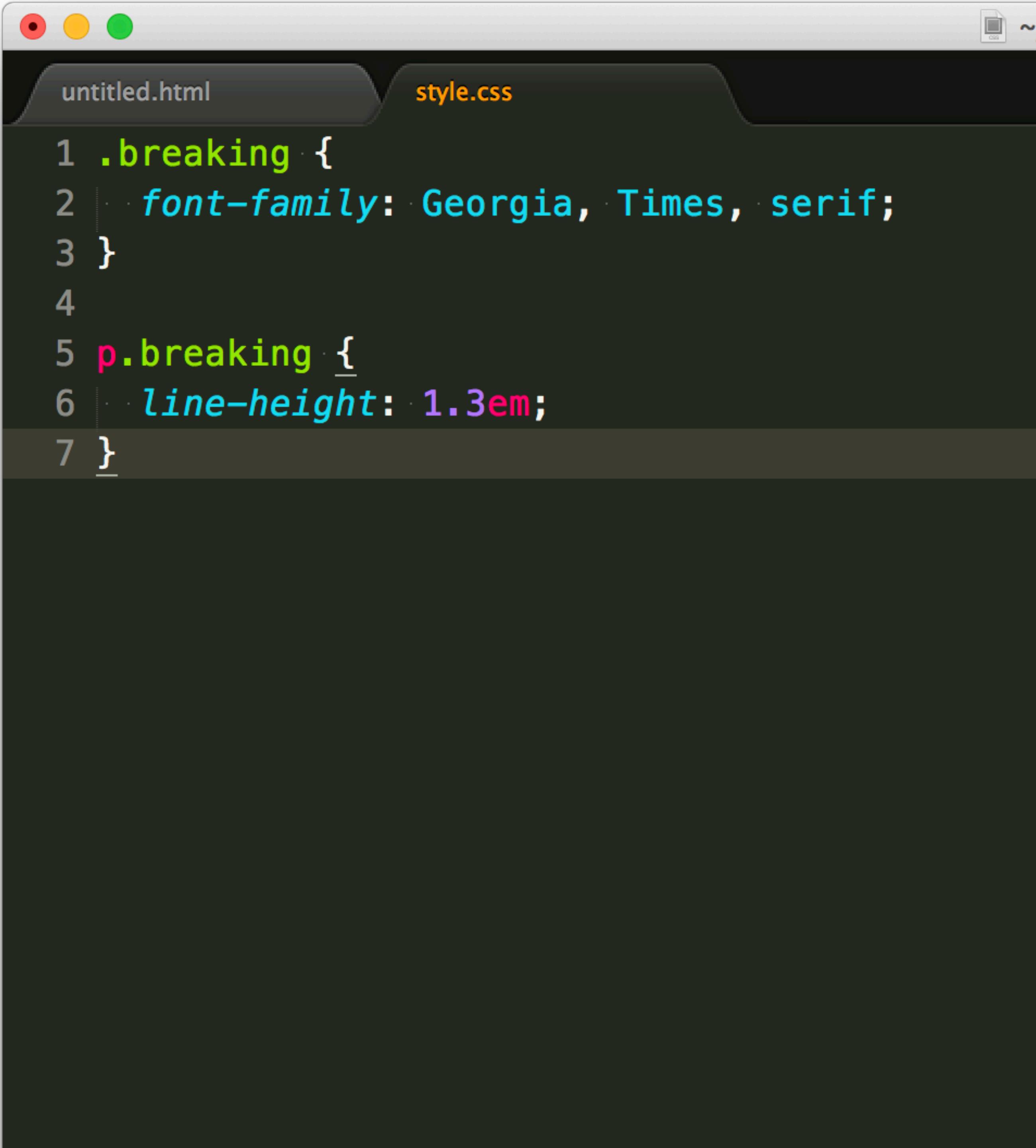
```
1 .science {  
2   font-family: Georgia, Times, serif;  
3   color: #A3B4C5;  
4   text-transform: uppercase;  
5 }  
6
```

Class selectors begin with a period (.) and are immediately followed by the name of the class.

Classes are by far the most commonly used for styling groups of elements.

The class selector targets *all* elements of a particular class. It's possible, however, for multiple elements on a web page to share a specific styling, but for one of those elements to differ slightly.

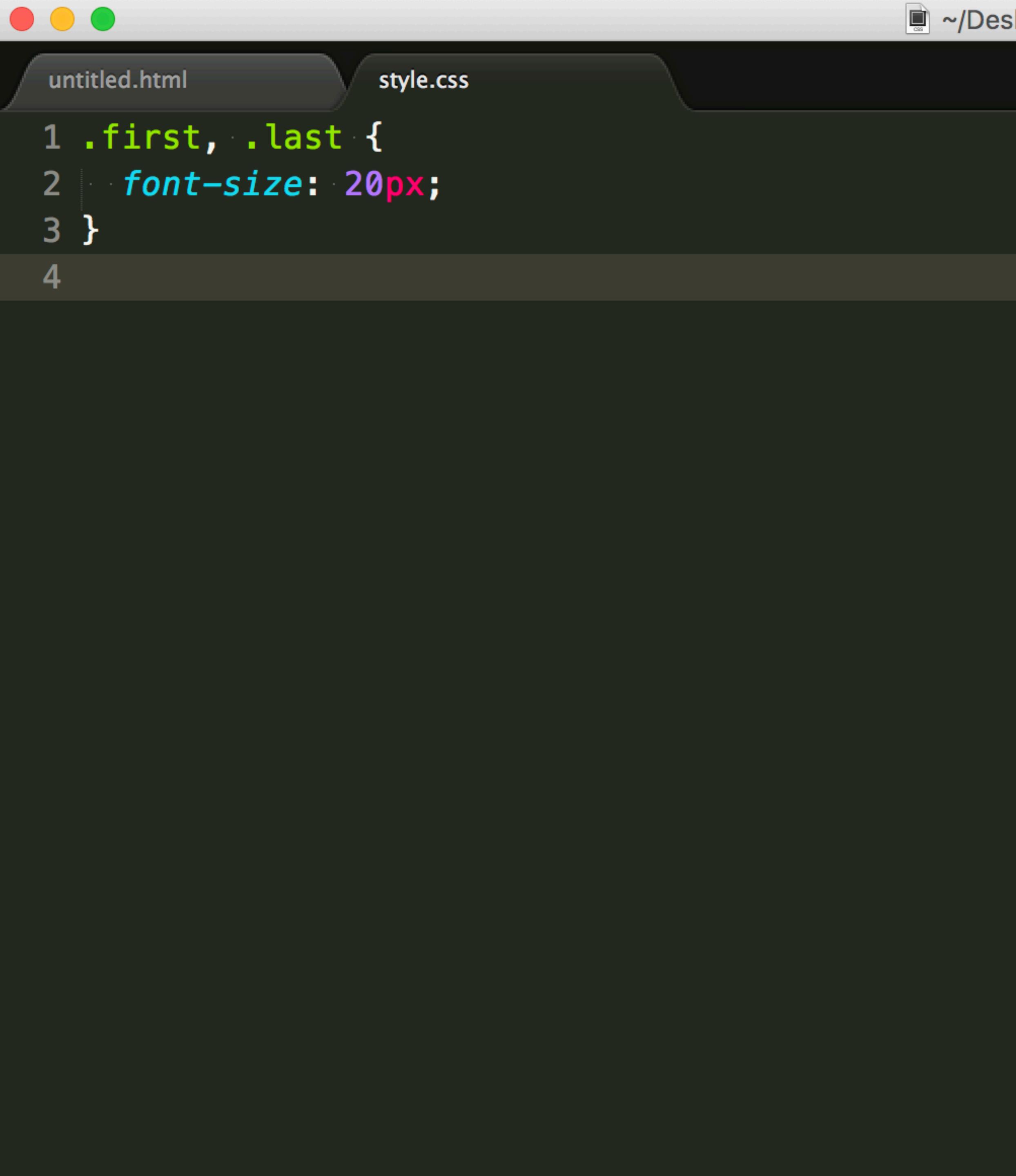
For example, a heading and a paragraph (both with a class of **breaking**) may need to share the same typeface, but the paragraph may require a styling better suited for paragraphs, as in the following example.



A screenshot of a Mac OS X desktop environment. At the top, there's a menu bar with standard Apple menu items. Below the menu bar, a Dock contains icons for various applications like Finder, Safari, and Mail. A terminal window is open, showing the command `git status` with some output. To the right of the terminal is a file browser window titled "untitled.html" which is currently selected. Another tab in the file browser is labeled "style.css". The content of "style.css" is displayed in the terminal window, showing CSS code for styling elements with a class of "breaking".

```
1 .breaking {  
2   font-family: Georgia, Times, serif;  
3 }  
4  
5 p.breaking {  
6   line-height: 1.3em;  
7 }
```

The **.breaking** selector targets *all* elements with a class of **breaking**. The **p.breaking** selector targets *only* **<p>** elements with a class of **breaking**. This type of selector allows you to be even more specific about a particular element's styling, even when that element must share some styling with other elements.



A screenshot of a dark-themed code editor window. The title bar shows three colored window control buttons (red, yellow, green) and the path ~/Desktop. The editor has two tabs: "untitled.html" and "style.css". The "style.css" tab is active, displaying the following CSS code:

```
1 .first, .last {  
2   font-size: 20px;  
3 }  
4
```

**Using a multiple class selector
is an efficient way of styling
multiple HTML elements.**

It's also possible to label HTML elements with more than one class.

The screenshot shows a web browser window with two tabs open. The left tab, titled "untitled.html", contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Organizing HTML & CSS</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7   <body>
8     <h1 class="book domestic">The Way of the Deep</h1>
9     <h1 class="book foreign">A Night in the Sky</h1>
10  </body>
11 </html>
```

The right tab, titled "style.css", contains the following CSS code:

```
1 .book {
2   font-family: Georgia, serif;
3 }
4
5 .domestic {
6   font-color: #0902CC;
7 }
8
9 .foreign {
10  font-color: #B097DD;
11 }
```

HTML offers an element that is the backbone of code organization: the *div*, represented by <div> in HTML.

You can think of the `div` as a box, or container, that groups elements that belong together.

For example, a `<div>` can be used to group together all of the elements that make up the navigation for a web page, or any other section of a page.

The image shows a Mac OS X desktop with two windows open. The window on the left is titled 'untitled.html' and contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Organizing HTML & CSS</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7   <body>
8     <div class="container">
9       <h1 class="title">Alice In Wonderland</h1>
10      <p> ... </p>
11    </div>
12  </body>
13 </html>
14
```

The window on the right is titled 'style.css' and contains the following CSS code:

```
1 div.container {
2   background-color: rgb(252, 255, 205);
3   font-family: Roboto, Helvetica, sans-serif;
4 }
5
6 h1.title {
7   color: #0D1A2F;
8 }
```

When a `div` is styled, *all elements inside* of the `div` will *inherit* the styling applied to the `div`. This example illustrates how easy it is to style sections of a web page using `div`.

The div is one of the most commonly used elements in all of HTML. Modern web pages make extensive use of the div, and learning how to use divs for organization and styling is a critical skill.

Take a look at the next page and think about what elements might be grouped into divs

Intro to Interactive

yaleinteractivesummer2017.com

data / sage

Untitled

Save As: Untitled

Where: Icelandic Cloud

File Format: Rich Text Document

Close

any

Intro to Interactive

Schedule

Assignments

Resources

Week 1

Studio

Mon 7/3

Authorship & Design

Lab

Mon 7/3

Git, Command Line, HTML

Studio

Wed 7/5

Starting Points