

HTML

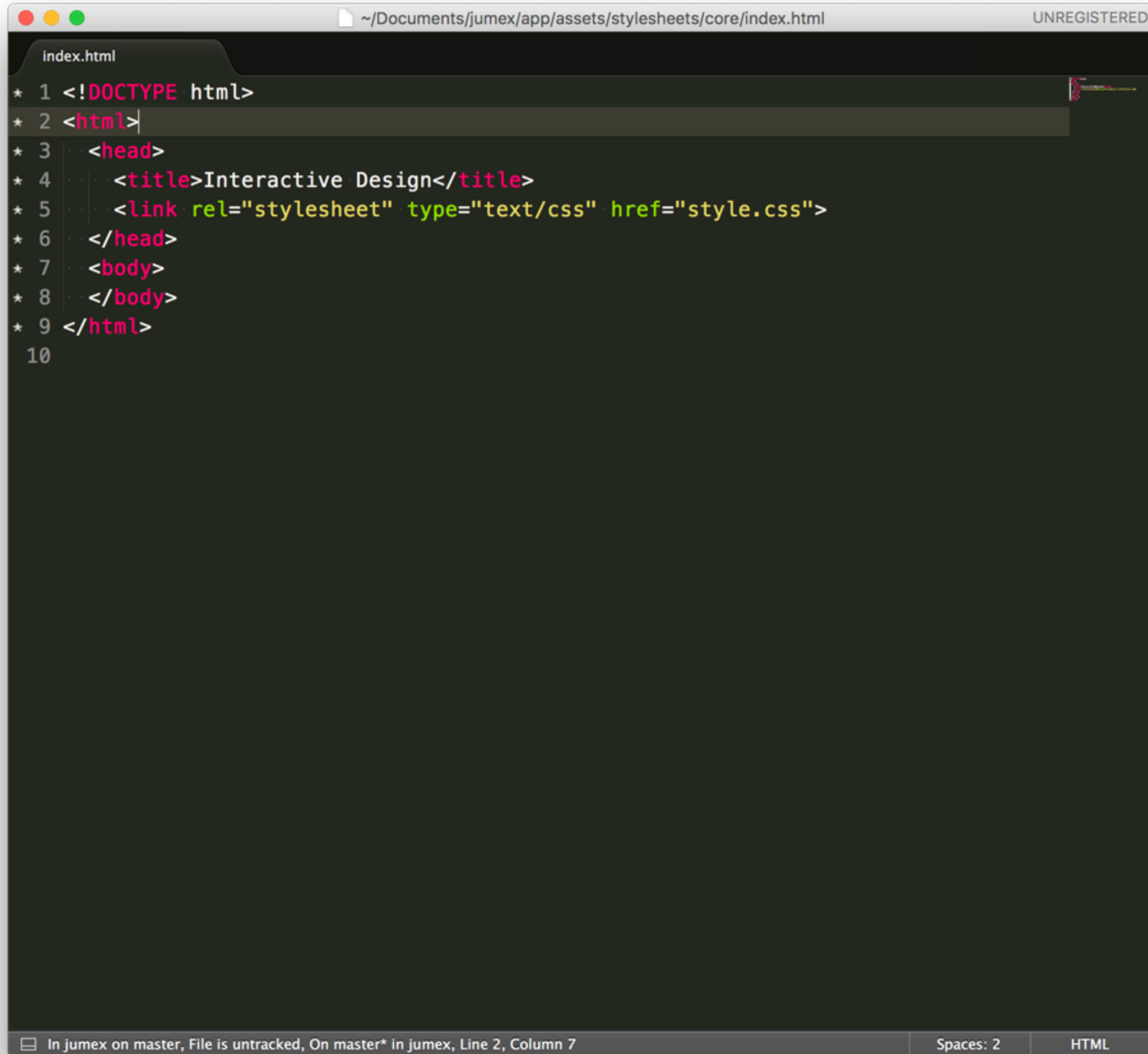
HTML is the language used to create the web pages you visit everyday. It provides a logical way to structure content for web pages.

Let's analyze the acronym "HTML", as it contains a lot of useful information. HTML stands for HyperText Markup Language.

A markup language is a computer language that defines the structure and presentation of raw text. Markup languages work by surrounding raw text with information the computer can interpret, "marking it up" for processing.

HyperText is text displayed on a computer or device that provides access to other text through links, also known as “hyperlinks”.

HTML can be written using any text editor. In this course we will be using Sublime text.

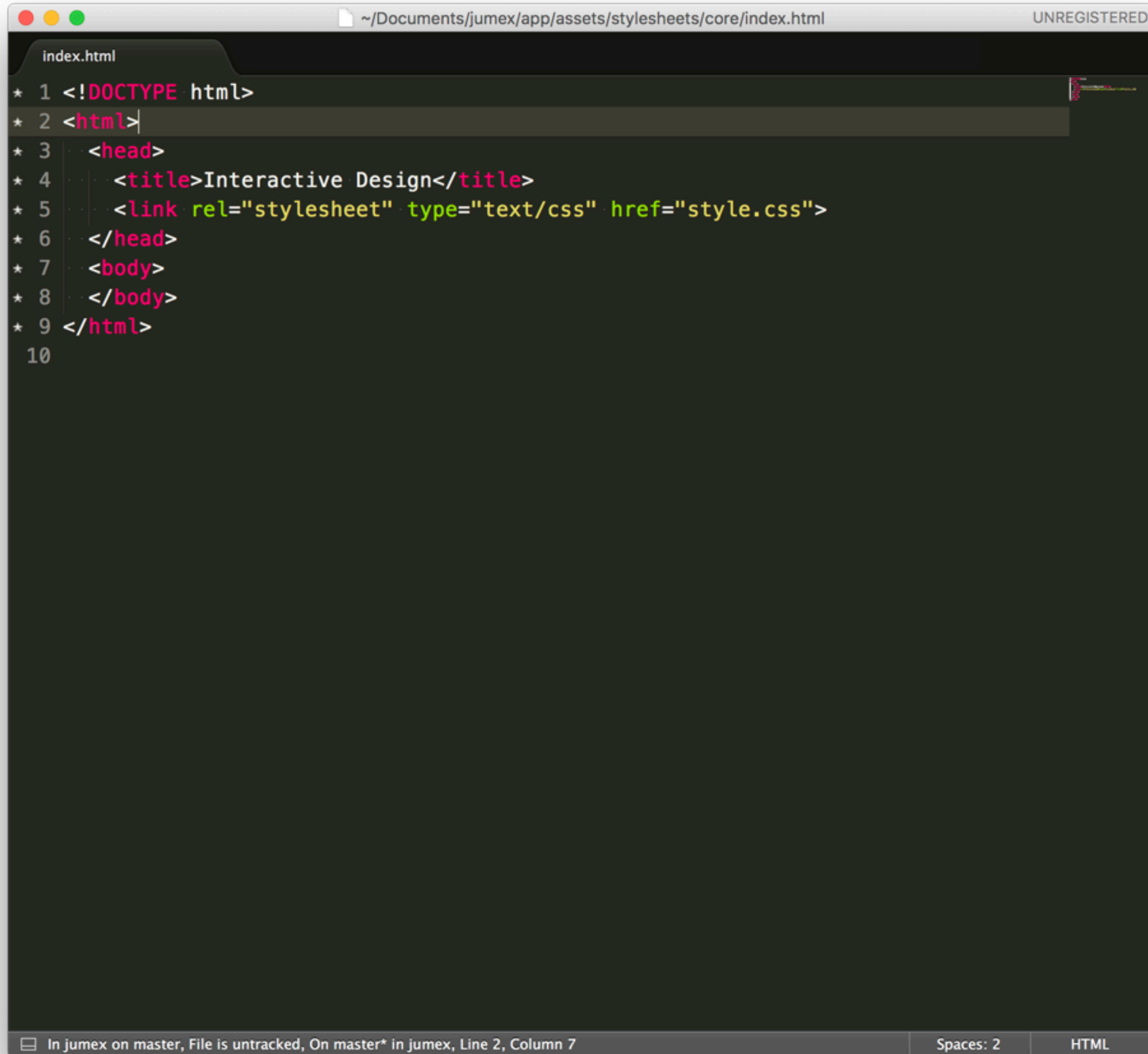


The image shows a code editor window with a dark theme. The title bar at the top indicates the file path is `~/Documents/jumex/app/assets/stylesheets/core/index.html` and the application is "UNREGISTERED". The editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Interactive Design</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7   <body>
8   </body>
9 </html>
```

The status bar at the bottom shows "In jumex on master, File is untracked, On master* in jumex, Line 2, Column 7", "Spaces: 2", and "HTML".

This is the most basic structure an html page can have. Let's look at some of the elements.

A screenshot of a code editor window. The title bar shows the file path: ~/Documents/jumex/app/assets/stylesheets/core/index.html and the word UNREGISTERED. The editor has a tab labeled index.html. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Interactive Design</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7   <body>
8   </body>
9 </html>
```

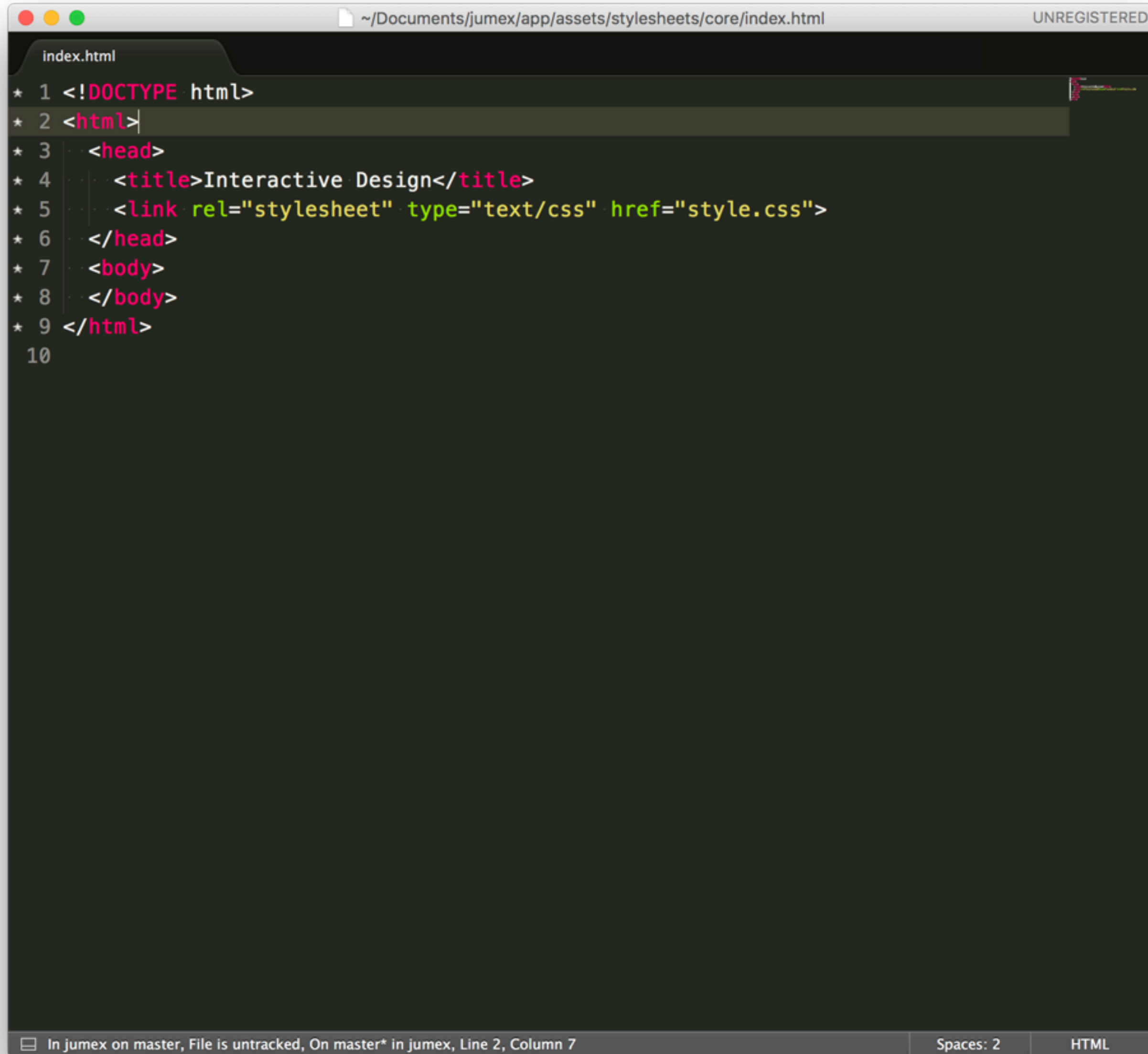
Line numbers 1 through 10 are visible on the left. The status bar at the bottom shows: In jumex on master, File is untracked, On master* in jumex, Line 2, Column 7, Spaces: 2, and HTML.

A *web browser* must know what language a document is written in before they can process the contents of the document.

You can let web browsers know that you are using the HTML language by starting your HTML document with a *document type declaration*.

The declaration is the following
`<!DOCTYPE html>`

This declaration is an instruction. It tells the browser what type of document to expect, along with what version of HTML is being used in the document. `<!DOCTYPE html>` *must* be the first line of code in all of your HTML documents.



```
index.html
~ /Documents/jumex/app/assets/stylesheets/core/index.html
UNREGISTERED

* 1 <!DOCTYPE html>
* 2 <html>
* 3   <head>
* 4     <title>Interactive Design</title>
* 5     <link rel="stylesheet" type="text/css" href="style.css">
* 6   </head>
* 7   <body>
* 8 </body>
* 9 </html>
10

In jumex on master, File is untracked, On master* in jumex, Line 2, Column 7
Spaces: 2
HTML
```

To begin adding HTML structure and content, we *must* first add opening and closing **<html>** tags, like so:

<html>

</html>

Anything between these tags will be considered HTML code.

Before we move forward, let's standardize some vocabulary we'll use as you learn HTML.

Angle brackets – In HTML, the characters `<` and `>` are known as angle brackets.

HTML element (or simply, element) – **HTML** code that lives inside of angle brackets.

Opening tag – the first, or opening, HTML tag used to start an HTML element.

Closing tag – the second, or closing, HTML tag used to end an HTML element. Closing tags have a forward slash (`/`) inside of them.

element

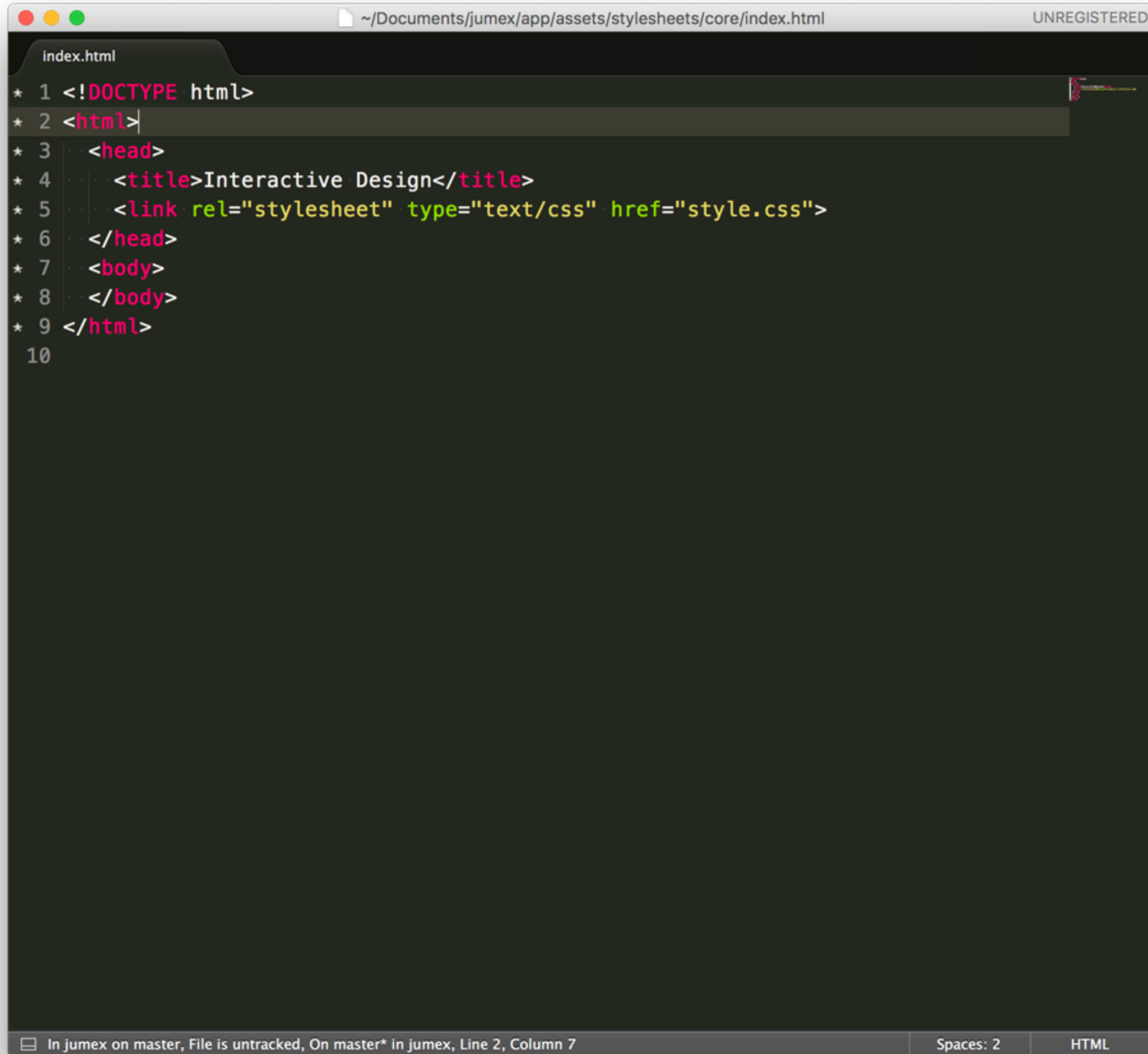
angle bracket

<p>Hello World!</p>

opening tag

closing tag

With the exception of a few HTML elements, most elements consist of an opening and closing tag.

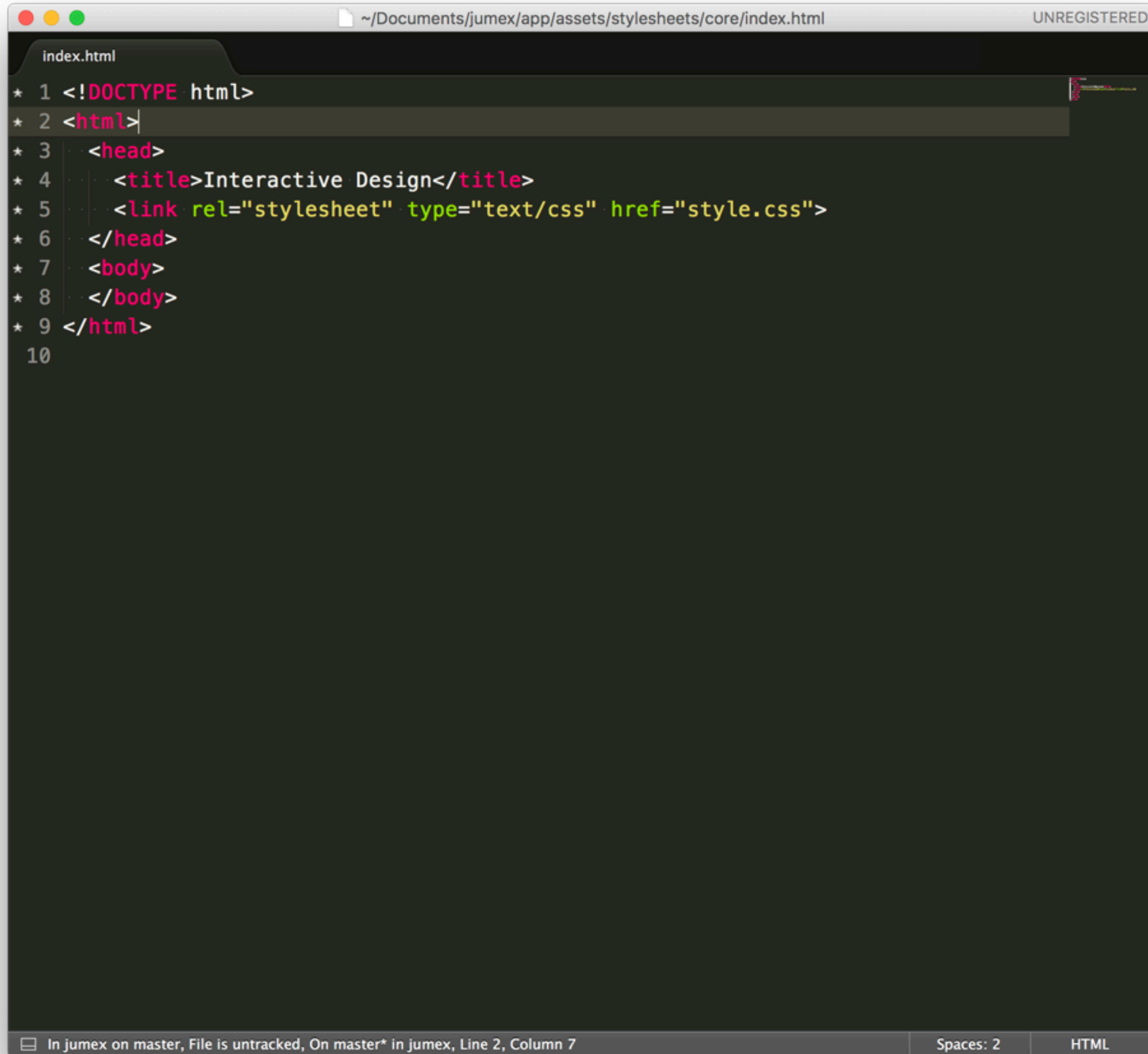


```
index.html
★ 1 <!DOCTYPE html>
★ 2 <html>
★ 3   <head>
★ 4     <title>Interactive Design</title>
★ 5     <link rel="stylesheet" type="text/css" href="style.css">
★ 6   </head>
★ 7   <body>
★ 8 </body>
★ 9 </html>
10
```

In jumex on master, File is untracked, On master* in jumex, Line 2, Column 7 Spaces: 2 HTML

Let's also give the browser some information about the page. We can do this by adding a **<head>** element.

The **<head>** element will contain information about the page that isn't displayed directly on the actual web page.



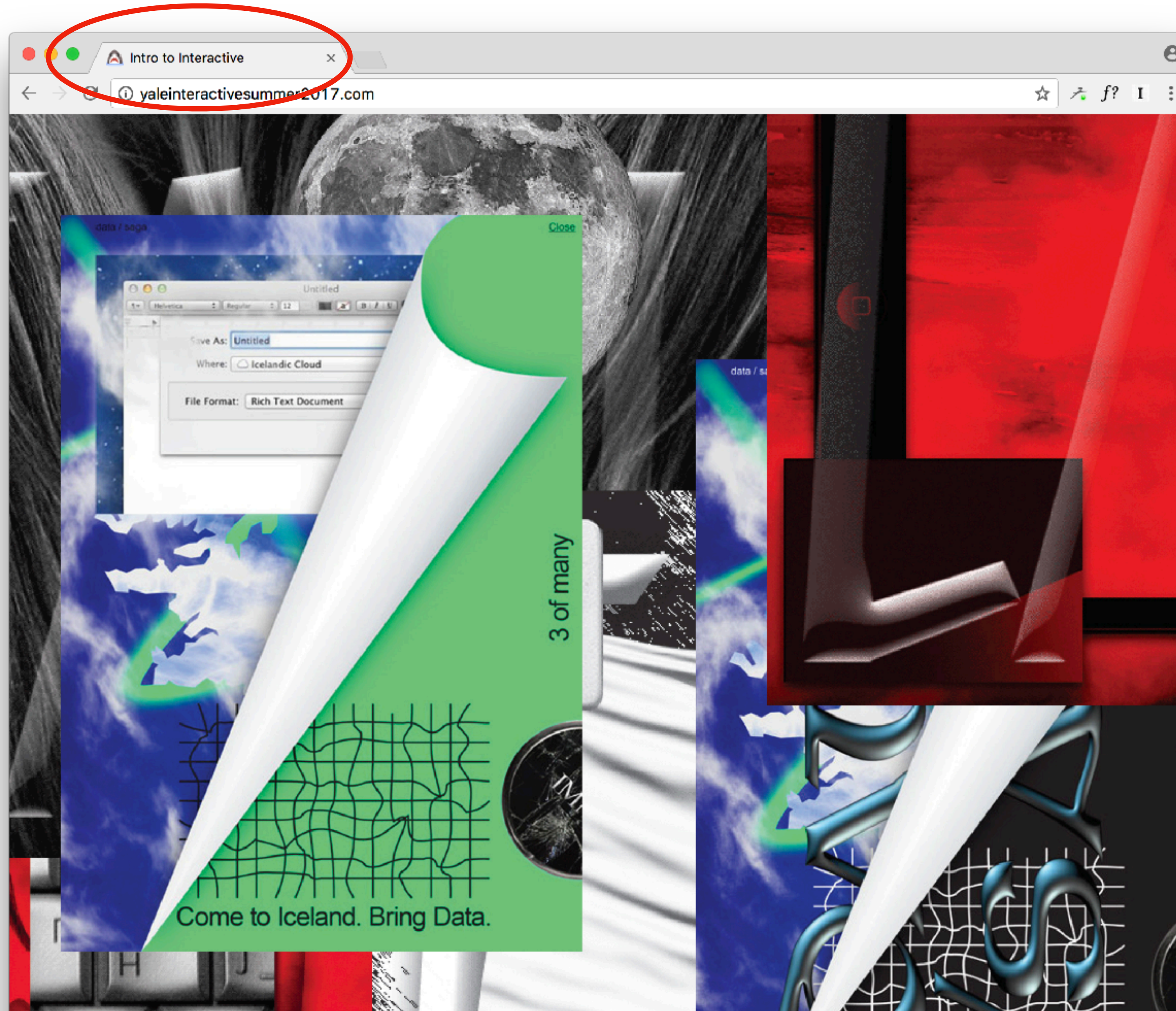
```
index.html
~ /Documents/jumex/app/assets/stylesheets/core/index.html
UNREGISTERED

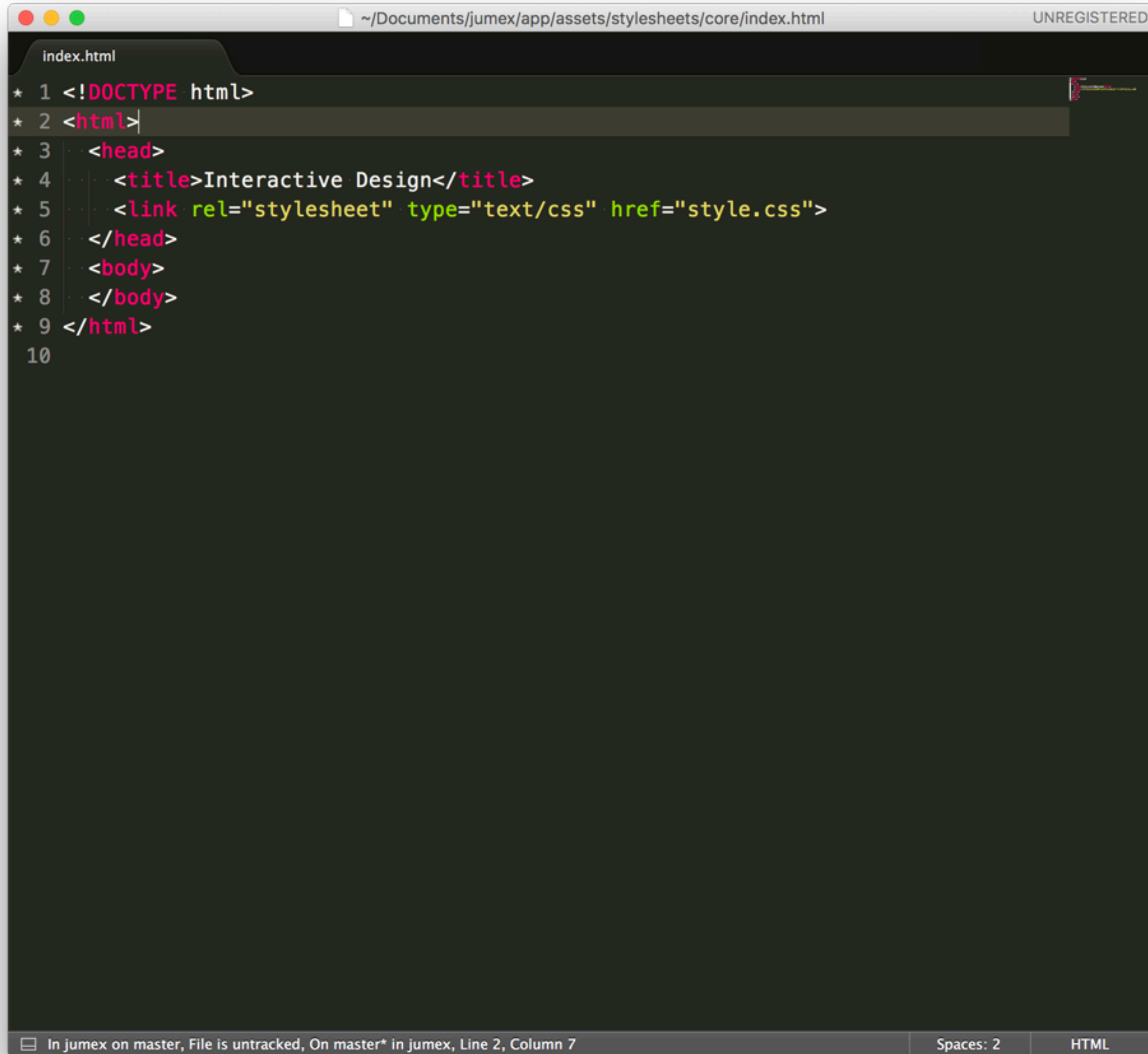
* 1 <!DOCTYPE html>
* 2 <html>
* 3   <head>
* 4     <title>Interactive Design</title>
* 5     <link rel="stylesheet" type="text/css" href="style.css">
* 6   </head>
* 7   <body>
* 8   </body>
* 9 </html>
10

In jumex on master, File is untracked, On master* in jumex, Line 2, Column 7
Spaces: 2
HTML
```

What kind of information about the web page can the **<head>** element contain?

One element is **<title>** which appears in the tab above a page.





```
index.html
~ /Documents/jumex/app/assets/stylesheets/core/index.html
UNREGISTERED

* 1 <!DOCTYPE html>
* 2 <html>
* 3   <head>
* 4     <title>Interactive Design</title>
* 5     <link rel="stylesheet" type="text/css" href="style.css">
* 6   </head>
* 7   <body>
* 8 </body>
* 9 </html>
10

In jumex on master, File is untracked, On master* in jumex, Line 2, Column 7
Spaces: 2
HTML
```

We've added some HTML, but still haven't seen any results in the web browser to the right. Why is that?

Before we can add content that a browser will display, we have to add a *body* to the HTML file. Once the file has a body, many different types of content can be added within the body, like text, images, buttons, and much more.

All of the code above demonstrates what is sometimes referred to as "boilerplate code".

The term "boilerplate code" is used to describe the basic HTML code required to begin creating a web page. Without *all* of the elements in the boilerplate code, you'll risk starting without the minimum requirements considered to be best practice.

HTML TAGS

Now that we have the basic structure. Let's start adding content to the page.

<h1> - <h6>

Headings in HTML can be likened to headings in other types of media. For example, in newspapers, large headings are typically used to capture a reader's attention. Other times, headings are used to describe content, like the title of a movie or an educational article.

HTML follows a similar pattern. In HTML, there are six different *headings*, or *heading elements*. Headings can be used for a variety of purposes, like titling sections, articles, or other forms of content.

```
index.html
* 1 <!DOCTYPE html>
* 2 <html>
* 3   <head>
* 4     <title>Interactive Design</title>
* 5     <link rel="stylesheet" type="text/css" href="style.css">
* 6   </head>
* 7   <body>
* 8     <h1>Most Important</h1>
* 9     <h2>Less Important</h2>
*10     <h3>Less Important</h3>
*11     <h4>Less Important</h4>
*12     <h5>Less Important</h5>
*13     <h6>Less Important</h6>
*14   </body>
*15 </html>
16
```

In jumex on master, File is untracked, On master* in jumex, Line 1, Column 1 Spaces: 2 HTML

The following is the list of heading elements available in HTML. They are ordered from largest to smallest in size

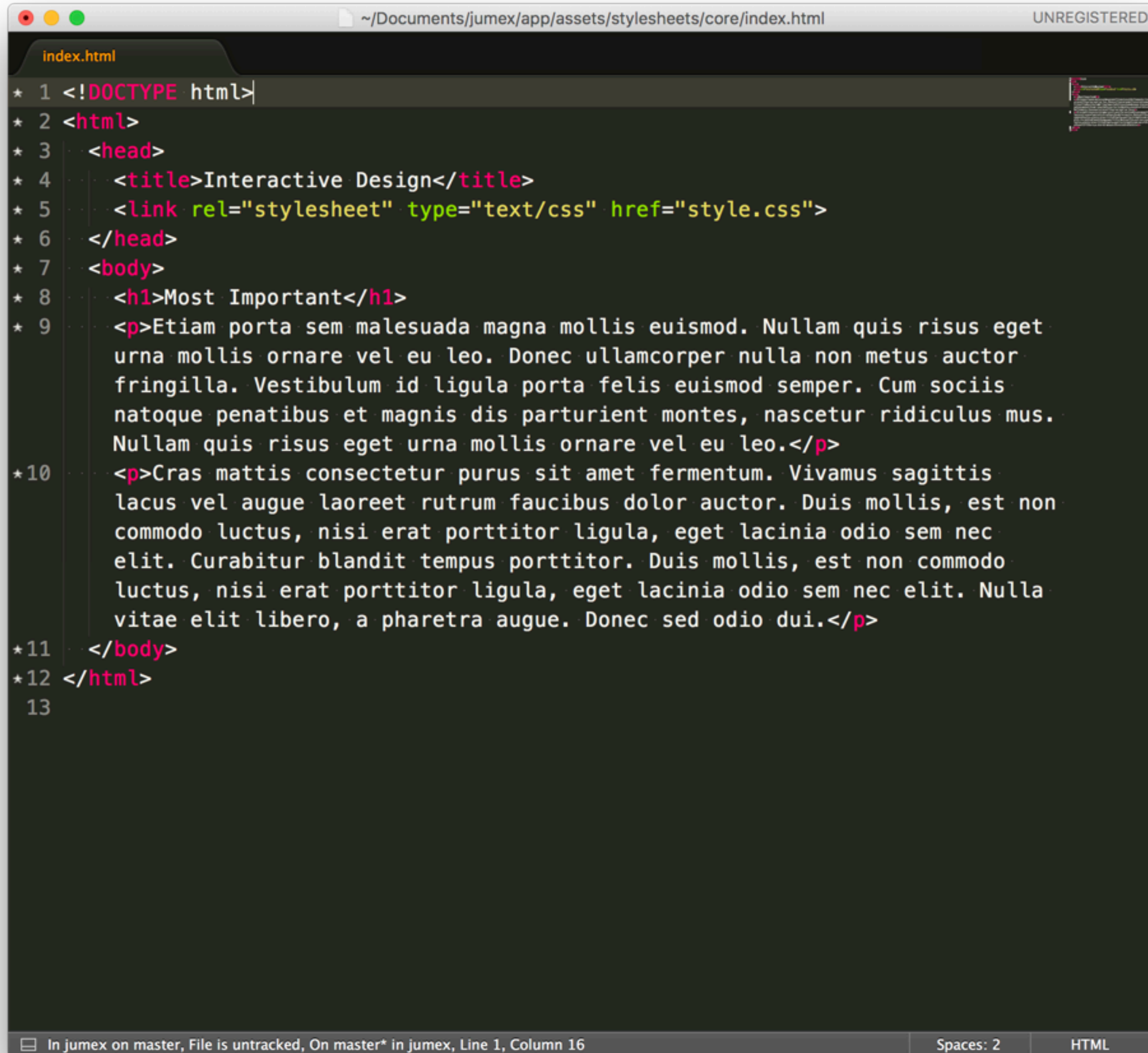
<h1> – used for main headings, all other smaller headings are used for subheadings.

<h2>
<h3>
<h4>
<h5>
<h6>

<p> </p>

Often times, headings are meant to emphasize or enlarge only a few words.

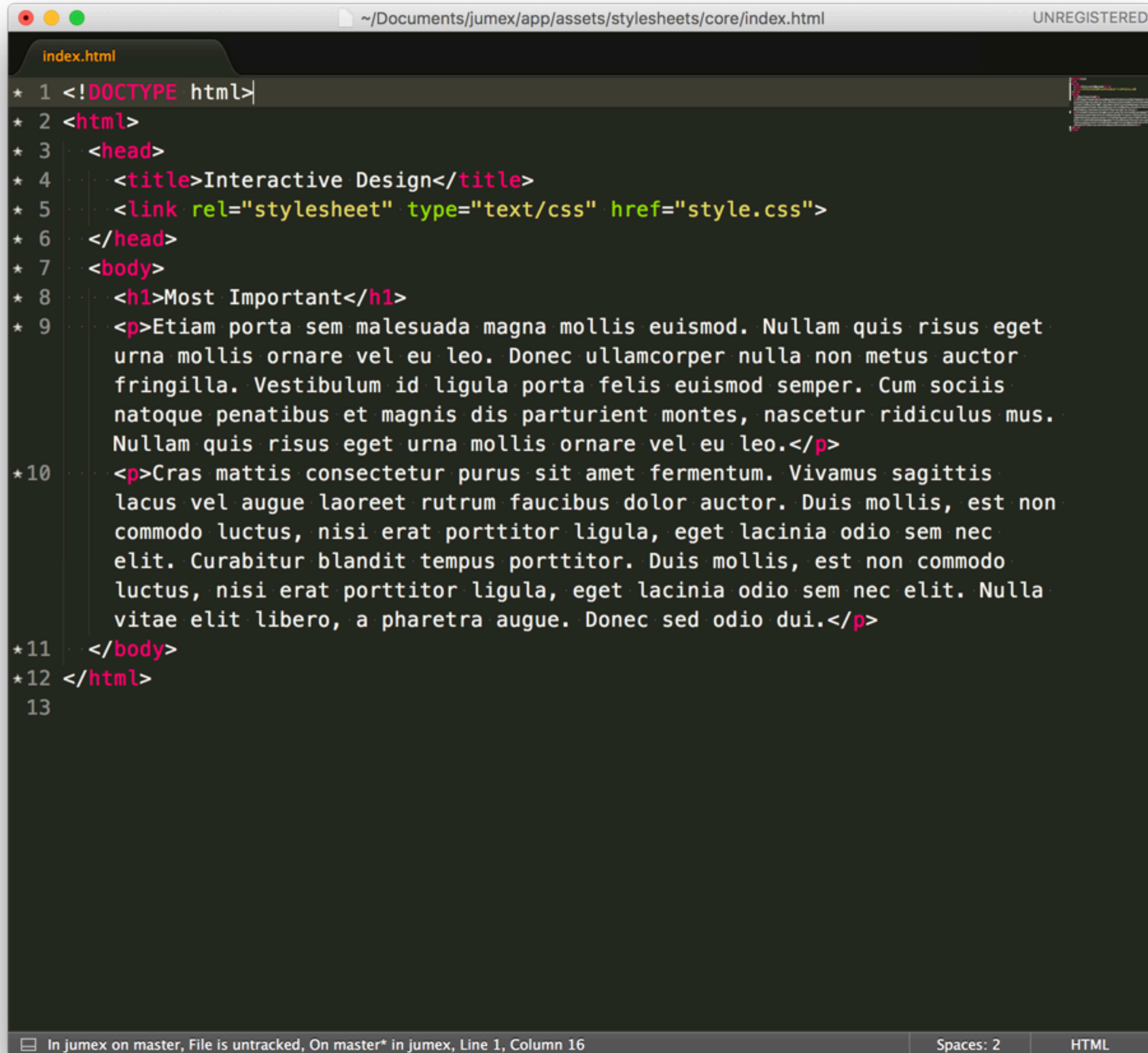
If you want to add content in paragraph format, you can add a *paragraph* using the paragraph element <p>



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Interactive Design</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7   <body>
8     <h1>Most Important</h1>
9     <p>Etiam porta sem malesuada magna mollis euismod. Nullam quis risus eget
10    urna mollis ornare vel eu leo. Donec ullamcorper nulla non metus auctor
11    fringilla. Vestibulum id ligula porta felis euismod semper. Cum sociis
12    natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.
13    Nullam quis risus eget urna mollis ornare vel eu leo.</p>
14    <p>Cras mattis consectetur purus sit amet fermentum. Vivamus sagittis
15    lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non
16    commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec
17    elit. Curabitur blandit tempus porttitor. Duis mollis, est non commodo
18    luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Nulla
19    vitae elit libero, a pharetra augue. Donec sed odio dui.</p>
20  </body>
21 </html>
```

Paragraphs are great for expanding the amount of content (text) on your web page. As you begin to add more text to your web page, however, keep in mind that large amounts of text in paragraph format can overwhelm web page visitors.

For example, if multiple paragraphs on your web page each contain large amounts of text, your web page could become difficult to consume.



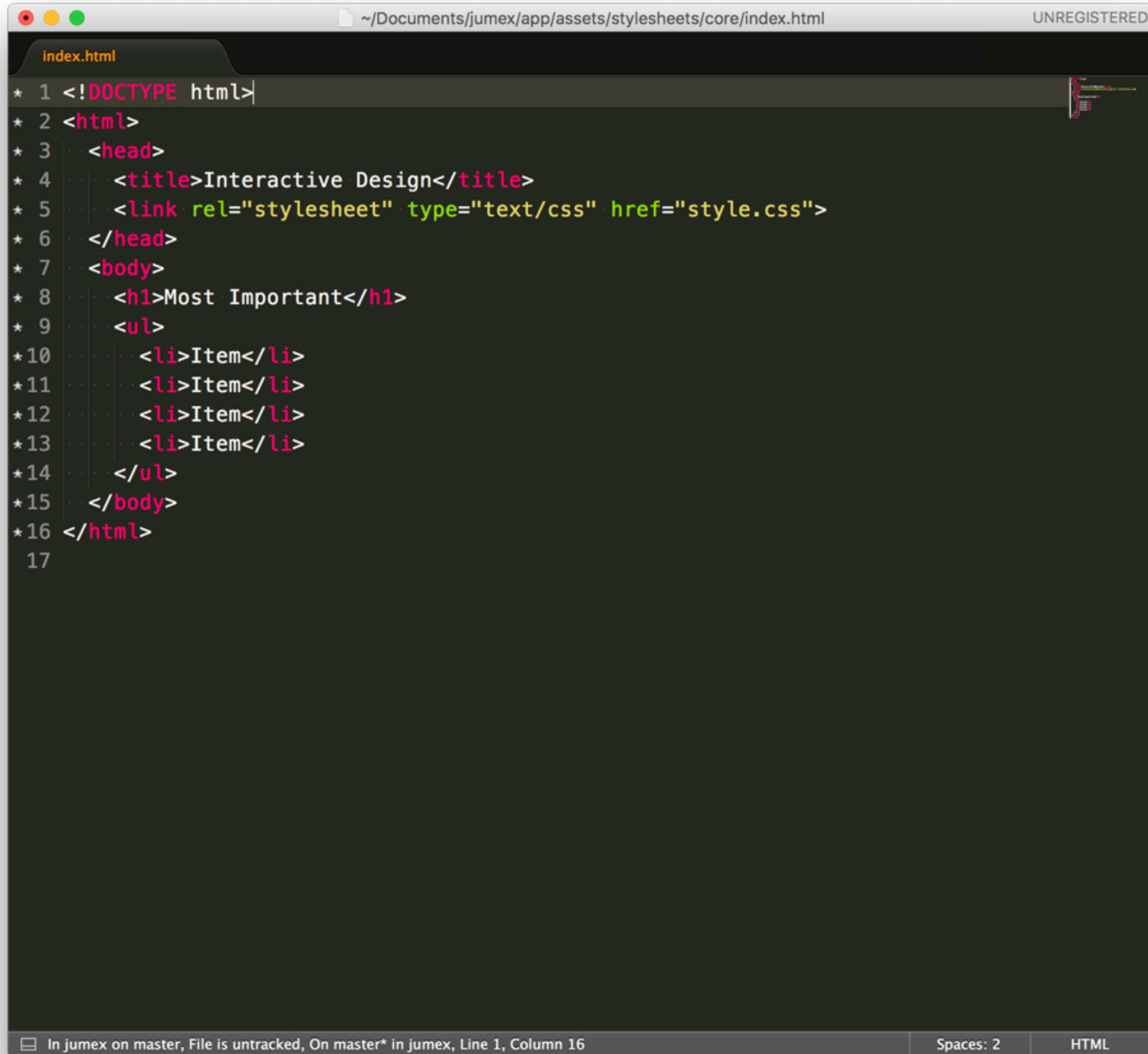
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Interactive Design</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7   <body>
8     <h1>Most Important</h1>
9     <p>Etiam porta sem malesuada magna mollis euismod. Nullam quis risus eget
10    urna mollis ornare vel eu leo. Donec ullamcorper nulla non metus auctor
11    fringilla. Vestibulum id ligula porta felis euismod semper. Cum sociis
12    natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.
13    Nullam quis risus eget urna mollis ornare vel eu leo.</p>
14    <p>Cras mattis consectetur purus sit amet fermentum. Vivamus sagittis
15    lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non
16    commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec
17    elit. Curabitur blandit tempus porttitor. Duis mollis, est non commodo
18    luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Nulla
19    vitae elit libero, a pharetra augue. Donec sed odio dui.</p>
20  </body>
21 </html>
```

Paragraphs are great for expanding the amount of content (text) on your web page. As you begin to add more text to your web page, however, keep in mind that large amounts of text in paragraph format can overwhelm web page visitors.

For example, if multiple paragraphs on your web page each contain large amounts of text, your web page could become difficult to consume.

` `

In HTML, you can use the *unordered list* for text you decide to format in bullet points. An unordered list outlines individual *list items* with a bullet point. You've probably used an unordered list when writing down a grocery list or school supplies list.

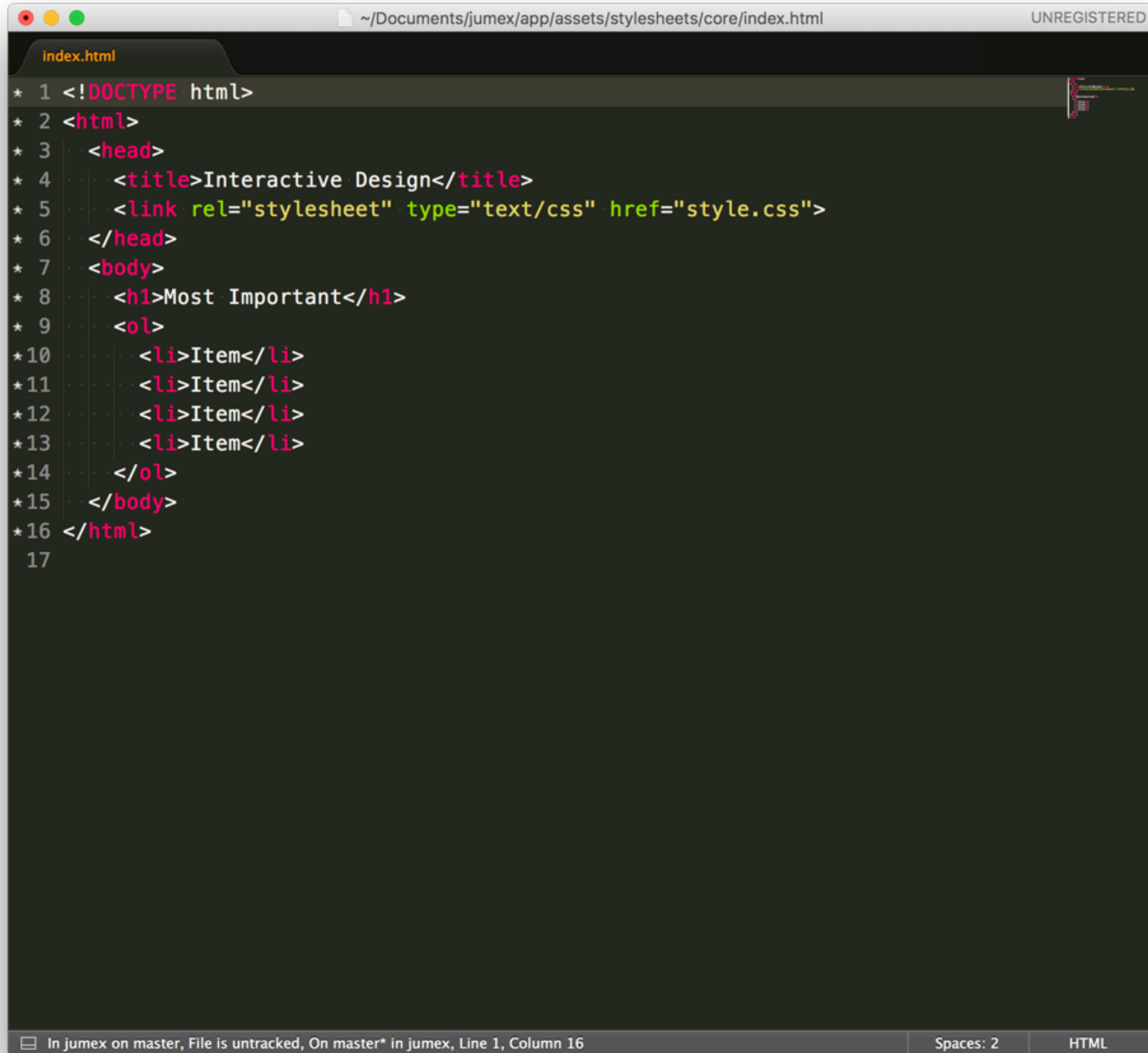


```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Interactive Design</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7   <body>
8     <h1>Most Important</h1>
9     <ul>
10      <li>Item</li>
11      <li>Item</li>
12      <li>Item</li>
13      <li>Item</li>
14    </ul>
15  </body>
16 </html>
17
```

To create a unordered list using HTML, you can use the `` element. The `` element, however, cannot hold raw text and cannot automatically format raw text with bullet points. Individual list items must be added to the unordered list using the `` element.

` `

Ordered lists are like unordered lists, except that each list item is numbered. You can create the ordered list with the `` element and then add individual list items to the list using `` elements.



The image shows a code editor window with a dark theme. The title bar at the top indicates the file path is `~/Documents/jumex/app/assets/stylesheets/core/index.html` and the application is `UNREGISTERED`. The editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Interactive Design</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7   <body>
8     <h1>Most Important</h1>
9     <ol>
10      <li>Item</li>
11      <li>Item</li>
12      <li>Item</li>
13      <li>Item</li>
14    </ol>
15  </body>
16 </html>
```

The code is syntactically highlighted, with tags in red, attributes in green, and text in white. A small icon of a document with a red squiggle is visible in the top right corner of the editor area. The status bar at the bottom shows the file is untracked, the current position is Line 1, Column 16, and the file type is HTML.

Ordered list full syntax.

```
<a href="https://www.wikipedia.org/"> </a>
```

You can add links to a web page by adding an anchor element `` and including the text of the link in between the opening and closing tags.

The anchor element in the example above is incomplete without the *href attribute*.

Attributes provide even more information about an element's content. They live directly inside of the opening tag of an element. Attributes are made up of the following two parts:

The *name* of the attribute.

The *value* of the attribute.

For anchor elements, the name of the attribute is **href** and its value must be set to the URL of the page you'd like the user to visit


```
index.html
* 1 <!DOCTYPE html>
* 2 <html>
* 3   <head>
* 4     <title>Interactive Design</title>
* 5     <link rel="stylesheet" type="text/css" href="style.css">
* 6   </head>
* 7   <body>
* 8     <h1>Most Important</h1>
* 9     <ol>
*10       <li>Item</li>
*11       <li>Item</li>
*12       <li>Item</li>
*13       <li>Item</li>
*14     </ol>
*15     <a href="https://www.wikipedia.org/">This Is A Link To Wikipedia</a>
*16   </body>
*17 </html>
18
```

In jumex on master, File is untracked, On master* in jumex, Line 15, Column 73 Spaces: 2 HTML UNREGISTERED

The anchor syntax.

```
<a href="https://www.wikipedia.org/" target="_blank"> </a>
```

The **target** attribute specifies that a link should open in a new window. Why is it beneficial to open links in a new window?

It's possible that one or more links on your web page link to an entirely different website. In that case, you may want users to read the linked website, but hope that they return to your web page. This is exactly when the **target** attribute is useful!

```

```

The `` element lets you add images to a web page. This element is special because it does not have a closing tag, it only has an opening tag. This is because the `` element is a *self-closing* element.

```
~/Documents/jumex/app/assets/stylesheets/core/index.html UNREGISTERED
index.html
* 1 <!DOCTYPE html>
* 2 <html>
* 3   <head>
* 4     <title>Interactive Design</title>
* 5     <link rel="stylesheet" type="text/css" href="style.css">
* 6   </head>
* 7   <body>
* 8     <h1>Most Important</h1>
* 9     <ol>
*10       <li>Item</li>
*11       <li>Item</li>
*12       <li>Item</li>
*13       <li>Item</li>
*14     </ol>
*15     
*16   </body>
*17 </html>
18
```

In jumex on master, File is untracked, On master* in jumex, Line 15, Column 54 Spaces: 2 HTML

Note that the **** element has a required attribute called **src**, which is similar to the **href** attribute in links. In this case, the value of **src** must be the URL of the image. Also note that the end of the **** element has a forward slash **/**. This is required for a self-closing element.

You've probably visited websites where not all links were made up of text. Maybe the links you clicked on were images, or some other form of content.

HTML allows you to turn nearly any element into a link by wrapping that element with an anchor element. With this technique, it's possible to turn images into links by simply wrapping the `` element with an `<a>` element.

```
index.html
* 1 <!DOCTYPE html>
* 2 <html>
* 3   <head>
* 4     <title>Interactive Design</title>
* 5     <link rel="stylesheet" type="text/css" href="style.css">
* 6   </head>
* 7   <body>
* 8     <h1>Most Important</h1>
* 9     <ol>
*10       <li>Item</li>
*11       <li>Item</li>
*12       <li>Item</li>
*13       <li>Item</li>
*14     </ol>
*15     <a href="https://en.wikipedia.org/wiki/Opuntia" target="_blank">
*16       
*17   </a>
*18 </body>
*19 </html>
20
```

In jumex on master, File is untracked, On master* in jumex, Line 17, Column 9 Spaces: 2 HTML

The full syntax for wrapping images as links.

White space

It's important to understand that the formatting of the code in `index.html` will not affect the positioning of the elements within the browser.

For example, if you wanted to increase the space between a paragraph and an image on your web page, you would *not* be able to accomplish this by simply adding more spacing between the paragraph element and image element within `index.html`. This is because the browser ignores *whitespace* present in HTML files like `index.html`.

On the other hand, using whitespace in HTML files is important so that your code is easy to read and follow.

`
`

If you *are* interested in modifying the spacing in the browser, you can use HTML's *line break* element.

The line break element is one self-closing tag. You can use it anywhere within your HTML code and a line break will be shown in the browser.

```
index.html
* 1 <!DOCTYPE html>
* 2 <html>
* 3   <head>
* 4     <title>Interactive Design</title>
* 5     <link rel="stylesheet" type="text/css" href="style.css">
* 6   </head>
* 7   <body>
* 8     <h1>Most Important</h1>
* 9     <ol>
*10       <li>Item</li>
*11       <li>Item</li>
*12       <li>Item</li>
*13       <li>Item</li>
*14     </ol>
*15     <a href="https://en.wikipedia.org/wiki/Opuntia" target="_blank">
*16       
*17     </a>
*18     Item <br />
*19     Item <br />
*20     Item <br />
*21     Item <br />
*22     Item <br />
*23   </body>
*24 </html>
25
```

In jumex on master, File is untracked, On master* in jumex, Line 18, Column 16 Spaces: 2 HTML

The full syntax for adding line breaks.

INDENTATION!

Whitespace makes code easier to read by increasing (or decreasing) the spacing between lines of code. To make the structure of code easier to read, web developers often use *indentation*.

The World Wide Web Consortium is responsible for maintaining the style standards of HTML. At the time of writing, the W3C recommends 2 spaces of indentation when writing HTML code. Indentation is intended for elements nested within other elements.

`<!-- This won't appear -->`

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.

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

You can add headings of different sizes using the different headings elements: `<h1>` through `<h6>`.

Paragraphs are added with the `<p>` element.

Unordered lists are created with the `` element and list items are added using the `` element.

file. You can add a comment with `<!-- This is a comment -->`.

Ordered lists are created with the `` element and list items are added using the `` element.

You can add links to your web page using the `<a>` element – don't forget the `href` attribute!

Images can be added with the `` element – don't forget the `src` attribute!

Images help support visually impaired users when `` elements include the `alt` attribute.

You can turn anything into a link by wrapping it with an `<a>` element.

White space in the HTML file does *not* affect the positioning of elements in the browser.

The W3C recommends 2 spaces of indentation for nested HTML elements.

Comments are used to take notes inside of an HTML