11.2.5

More HTML Elements

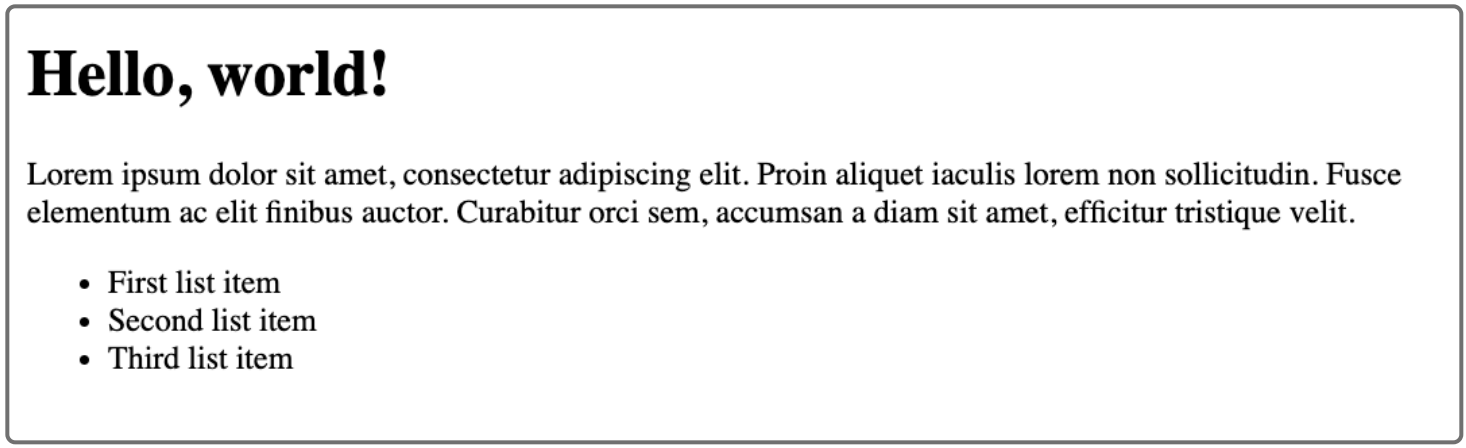
We've discussed the structure of a basic HTML document. Specifically, it has a head section that contains metadata about the document. It also has a body section containing the content that displays on a webpage. Now, we'll explore more HTML elements that we can add to the body section to make a webpage come to life.

Headers and Lists

In VS Code, create a new HTML file named `index4.html`, and then paste the following code into the file:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Document</title>
</head>
<body>
    <h1>Hello, world!</h1>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <ul>
      <li>First list item</li>
      <li>Second list item</li>
      <li>Third list item</li>
    </ul>
</body>
</html>
```

Save the file, and then open it. It opens in a web browser and appears like the following image:



In the preceding image, notice that we have three notable elements. The first is a heading that consists of the text "Hello, world!" in a large and bold font. The second is a paragraph of text that begins with “Lorem ipsum.” The third is a bulleted list of three items, as follows:

- First list item
- Second list item
- Third list item

Also notice that in our code, the `<body />` tag contains all these elements.

NOTE

You might be wondering why we just used a new format for the `<body />` tag. Well, we sometimes combine the opening and the closing tags like that. Notice that this combination tag, or **self-closing tag**, consists of a left angle bracket (`<`); the name of the tag (in this case, `body`); and then a space, a slash, and a right angle bracket (`>`).

This is mainly a concise way of referring to a set of opening and closing tags along with any content that appears between them.

Let's now go over the tags for these elements, as they appear in our code:

- In the `<h1 />` tag, the "h" stands for "header." The text in a header tag will display larger and bolder than the other text on the page. Six levels of headers are available, from `h1` to `h6`, with `h1` displaying the largest text.
- The `<p />` tag is a paragraph tag. In our code, it contains nonsensical placeholder text.

- In the `<ul />` tag, the "ul" stands for “unordered list.” An **unordered list** is a list of bulleted items.
- Nested within the unordered list, we have three `<li />` tags. The "li" stands for “list item.”

Now, practice creating headings and lists yourself in the following Skill Drill:

SKILL DRILL

In your `index4.html` file, add headings at the second, third, and fourth levels. To do so, use tags for `h2`, `h3`, and `h4`. Also, add an ordered list of three items. An **ordered list** is a numbered list. Such a list uses numbers that go in order instead of using bullets. To add this list, use an `<ol />` instead of a `<ul />` tag.

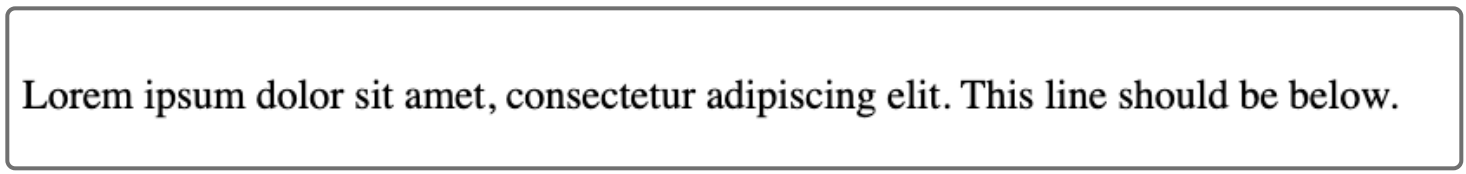
Multiple Paragraphs

Why are paragraph tags necessary? To find out, let's modify the paragraph in `index4.html`. After the line starting with "Lorem ipsum," type another line that contains the text "This line should be below." (But, don’t include the quotation marks in that text.) Your code should be as follows:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Document</title>
</head>

<body>
    <h1>Hello, world!</h1>
    <h2>This is a second-level heading.</h2>
    <h3>This is a third-level heading.</h3>
    <h4>This is a fourth-level heading.</h4>
    <p>
```

```
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        This line should be below.
    </p>
    <ul>
        <li>First list item</li>
        <li>Second list item</li>
        <li>Third list item</li>
    </ul>

    <ol>
        <li>First ordered item</li>
        <li>Second ordered item</li>
        <li>Third ordered item</li>
    </ol>
</body>

</html>
```

When displayed in the browser, the new line of text doesn't appear on a separate line. In fact, it appears in the same line as the "Lorem ipsum" text, as the following image shows:



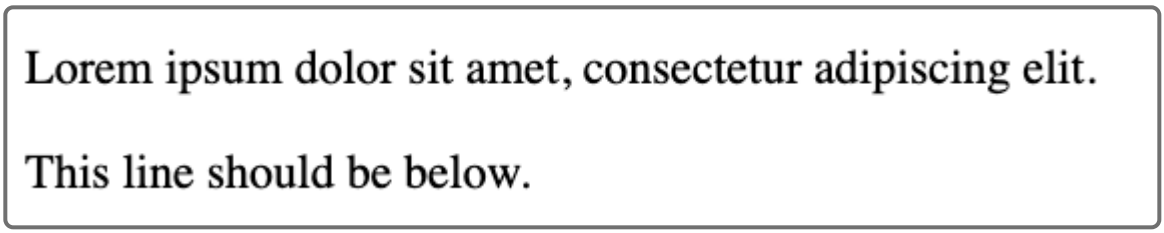
This happens because HTML mostly ignores white space, like return characters and extra spaces.

So, how do we create a new line? We can make the second sentence appear on a separate line by enclosing it in its own paragraph tag, as the following code shows:

```
    <p>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    </p>
```

```
    <p>
        This line should be below.
    </p>
```

The reason this works is that each paragraph starts on a new line. Now, modify your code to add the new paragraph tags, and then open the file by double-clicking it. In the browser, the second sentence now appears on a new line.



Links

What if we want to link to another website or to a file that enhances our content? We can do so by creating a link. An HTML **link** consists of three parts: an anchor tag, an `href` attribute, and the text that the browser will display for the link. (The latter is the text that people will click to go to the link.)

Let's create a link now. To do so, create a new HTML file named `index5.html`, paste the following code into that file, save the changes, and then open the file in a browser:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
    <title>Document</title>
</head>

<body>
```

```
    <a href="https://pandas.pydata.org/">Pandas Website</a>
</body>

</html>
```

The preceding code creates an HTML link to the Pandas library website. How does the syntax work? The code is `<a href="https://pandas.pydata.org/">Pandas Website</a>`. The "a" in the `<a />` tag stands for "anchor." That tag lets the browser know that a link is coming. (The link can go either to somewhere on the same HTML webpage or to an external webpage or file.) We then set the `href` attribute to the URL of the Pandas website. That's the link that the anchor tag let the browser know was coming. Finally, we place the text that we want the browser to display (in this case, "Pandas Website") between the opening and closing tags.

Now, practice creating an HTML link yourself in the following Skill Drill:

SKILL DRILL

Create a new HTML file named `index6.html`, and then add the standard code for a basic HTML page. Then create a link to the News - NASA Mars Exploration website (https://mars.nasa.gov/news/). The URL is `https://mars.nasa.gov/news/`. Use "Mars News" as the display text.

Terrific job!

Divisions

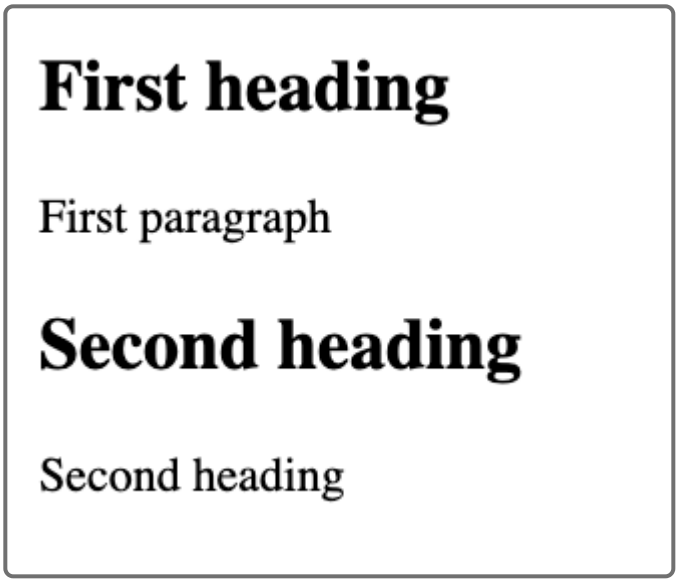
Let's explore one more tag: `<div>`, which stands for "division." This tag differs in an important way from the HTML tags that we've learned about so far. Specifically, it doesn't add an element to the webpage but helps us meaningfully organize our HTML elements. It does this by establishing divisions, or sections, within the body.

Now, create a new HTML file named `index7.html`, and then paste the following code into that file:

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
    <title>Document</title>
</head>

<body>
    <div>
        <h2>First heading</h2>
        <p>First paragraph</p>
    </div>

    <div>
        <h2>Second heading</h2>
        <p>Second heading</p>
    </div>
</body>

</html>
```

Next, open the file in a browser. Notice that the webpage appears with our two sections, each of which has a heading and a paragraph.



Now, practice creating the same file without divisions in the following Skill Drill:

SKILL DRILL

Create a new HTML file named `index8.html`, and then add the same code that `index7.html` has—with one difference. Specifically, remove the `<div>` tags. Then open the file in the browser. Does it appear differently?

You might have noticed that the `<div>` tag doesn't change the appearance of your HTML code in the browser. As mentioned earlier, its only purpose is to meaningfully organize our HTML elements for ourselves. In `index7.html`, the `<div>` tags divided the body into two logical sections, where each section contained a heading and a paragraph. The reason for the appropriate use of `<div>` tags is that they make the code easier to read. And, code readability becomes more important as the code gets longer.

Congratulations! You've learned and practiced the skills that you need to identify and create HTML elements on a webpage. Next, you'll apply what you've learned to a web scraping project.