Intro to HTML & CSS

Learn by doing step by step exercises.

Includes downloadable class files that work on Mac & PC.

EDITION 5.0

noble desktop

Published by:
Noble Desktop
185 Madison Ave, 3rd Floor
New York, NY 10016
nobledesktop.com

Copyright © 2014–2024 Noble Desktop NYC LLC
Publish Date: 7-03-2024

Table of Contents

Table of Contents

Downloading the Class Files

Here's how to get the files you'll need for this training:

1. Navigate to the **Desktop**.
2. Create a **new folder** called **Class Files** (this is where you'll put the files after they have been downloaded).
3. Go to **nobledesktop.com/download**
4. Enter the code **ith-2407-03**
5. If you haven't already, click **Start Download**.
6. After the **.zip** file has finished downloading, be sure to unzip the file if it hasn't been done for you. You should end up with a **Web Dev Class** folder.
7. Drag the downloaded folder into the **Class Files** folder you just made. These are the files you will use while going through the workbook.
8. If you still have the downloaded .zip file, you can delete that. That's it! Enjoy.

A basic website is essentially a collection of files in a folder. People access the files when they are hosted on a web server, accessible via the Internet. A website's address (for example google.com) is known as a URL (uniform resource locator).

Webpages are delivered via HTTP (Hypertext Transfer Protocol) or HTTPS (HTTP Secure), the latter of which uses encryption and provides security for the user. The web browser renders the page content according to its HTML markup instructions.

What Is HTML?

HTML stands for **Hypertext Markup Language**. It's a standardized system for tagging content in webpages. HTML provides a way to structure text semantically into paragraphs, headings, lists, links, etc. It also allows images and objects like form elements to be represented on the page. Webpages can link from one to another.

HTML allows computers of various platforms (Mac, Windows, iOS, Android, etc.) to view information in essentially the same way.

Most users will see the webpage as you intended, but it can appear differently depending on the user's browser or operating system. For example, one person may see the text in Times, while another sees Helvetica (because they uninstalled the Times font). While the webpage may not appear exactly the same to every user, the important thing is that all users have an acceptable experience and can access the content. Blind people can even have webpages read to them!

What Is CSS?

CSS stands for **Cascading Style Sheets**. It's a style language used to define the colors, fonts, spacing, layout, and more of elements in an HTML file. HTML files markup (label) content and CSS formats (styles) that content.

HTML and CSS are continually evolving languages, so webpages viewed in older browsers may not not look/work exactly the same as newer browsers. If only a few people still use the older browser (or it's not a major change) it might not be a problem. Otherwise you could wait until more people upgrade their browser to support the feature/code you want to use.

The Structure of an HTML Tag

Information in HTML is surrounded by tags, which are are wrapped by < >
For example, here's how to mark up (code) a paragraph:

```
<p>This is a paragraph</p>
```

Notice the paragraph is followed by a closing tag `</p>`

Most HTML tags require an ending or closing tag. If you have problems, you may have forgotten to end the tag. Some tags may work even if you don't end them. For example, the `<p>` tag indicates a new paragraph, so it often works without a closing `</p>`. It's a best practice to close tags to ensure they work properly.

Some tags do not have a closing tag, such as `<img>` for an image.

HTML tags are not case sensitive, but some things such as filenames and CSS style names are. It's a best practice to write all tags in lowercase.

File Naming Conventions

- HTML files end with the extension **.html**
- CSS files end with the extension **.css**
- Do not use spaces or special characters. Use hyphens instead of spaces.
- Use only lowercase letters when naming files (some servers are case sensitive).
- The homepage of a website is named **index.html**
- Use descriptive file names that tell what a page is about. For example: **about.html** and **contact.html** rather than **page1.html** and **page2.html**. Descriptive file names are good for SEO (Search Engine Optimization) because they provide keywords that help the page rank better on search engines like Google.

Choosing a Code Editor to Work In

Code editors are specialized text editors which offer code hints and more. Two popular choices are **Visual Studio Code** and **Sublime Text**. You can write code in any editor, but we highly recommend Visual Studio Code.

Installing Visual Studio Code

Visual Studio Code is a great code free editor for Mac and Windows.

1. Visit **code.visualstudio.com** and download **Visual Studio Code**.
2. To install the app:
   - Mac users: Drag the downloaded app into your **Applications** folder.
   - Windows users: Run the downloaded installer.
     During installation check on the **Add "Open with Code" …** options.

Setting Up Visual Studio Code

There are a few things we can do to make Visual Studio Code more efficient.

Setting Preferences

1. Launch **Visual Studio Code**.
2. In Visual Studio Code, at the bottom left of the window, click the Gear and from the menu that appears choose **Settings**.
3. In the search field type: **wrap**
4. Find **Editor: Word Wrap** and change its setting from **off** to **on**

   NOTE: This ensures you'll be able to see all the code without having to scroll to the right (because it will wrap long lines to fit to your screen size).

Setting Up Open in Browser

Visual Studio Code does not include a quick way to preview HTML files in a web browser (which you'll do a lot). We can add that feature by installing an extension:

1. On the left side of the Visual Studio Code window, click on the **Extensions** icon .
2. In the search field type: **open in browser**
3. Under the **open in browser** click **Install**.

Defining a Keystroke for Wrapping Code

Visual Studio Code has a feature to wrap a selection with some code. This is very useful so we'll want to use it frequently. There's no keystroke for it by default, so let's define one:

1. In Visual Studio Code, at the bottom left of the window, click the Gear ⚙ and from the menu that appears choose **Keyboard Shortcuts**.
2. In the search field type: **wrap with**
3. Double–click on **Emmet: Wrap with Abbreviation** and then:
   - Mac users: Hold **Option** and press **W** then hit **Return**.
   - Windows users: Hold **Alt** and press **W** then hit **Enter**.
4. You're done, so close the **Keyboard Shortcuts** tab.

Downloading the Latest Web Browsers

Browsers can handle code slightly differently, so as a web developer it's important to have all the current popular browsers installed for testing.

Chrome (Mac & Windows)

- Get it free at **google.com/chrome**

Firefox (Mac & Windows)

- Get it free at **firefox.com**

Safari (Mac Only)

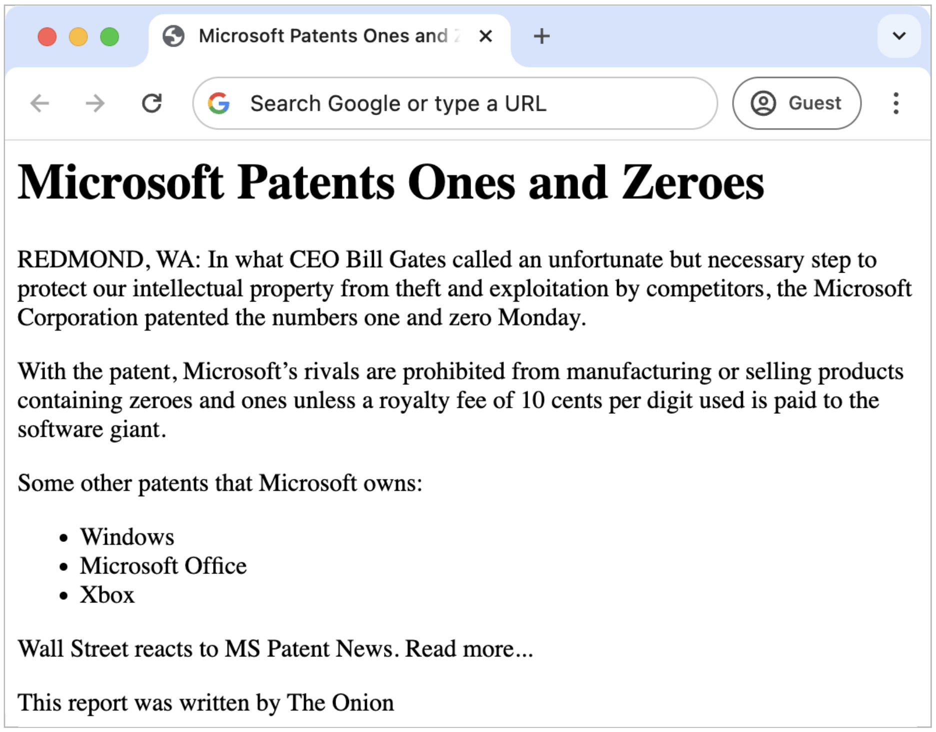
- To get the most recent version of Safari, you may have to update your macOS.

Microsoft Edge (Mac & Windows)

The latest versions of Microsoft Edge are based on Chrome's rendering engine.

- Mac Users: Get it free at **microsoft.com/edge**
- Windows Users: Microsoft Edge comes pre-installed on Windows 10 (it replaced Internet Explorer).

Exercise Preview



Microsoft Patents Ones and Zeroes

REDMOND, WA: In what CEO Bill Gates called an unfortunate but necessary step to protect our intellectual property from theft and exploitation by competitors, the Microsoft Corporation patented the numbers one and zero Monday.

With the patent, Microsoft's rivals are prohibited from manufacturing or selling products containing zeroes and ones unless a royalty fee of 10 cents per digit used is paid to the software giant.

Some other patents that Microsoft owns:

- Windows
- Microsoft Office
- Xbox

Wall Street reacts to MS Patent News. Read more...

This report was written by The Onion

Exercise Overview

In this exercise you'll learn how to write HTML to create a webpage and format headings, paragraphs, and lists.

Getting Started

1. Launch your code editor (**Visual Studio Code**, **Sublime Text**, etc.). If you are in a Noble Desktop class, launch **Visual Studio Code**.

2. All the files for a website are stored in a folder. We'll be working with files in a folder named **News Website**. Open the folder as follows (this should work in most code editors such as Visual Studio Code):

   - Go to **File > Open Folder**.
   - Navigate to **Class Files > Web Dev Class > News Website** and hit **Open** (Mac) or **Select Folder** (Windows).
   - If you get a message about trusting the author, check on **Trust the authors of all files in the parent folder** and click **Yes, I trust the authors**.

3. On the left you should see a panel showing files in the folder you just opened. Click on **microsoft.html** to open it.

4. We've typed out some text, but it doesn't have any HTML tags yet. Add the required tags shown below. The code you need to type is highlighted in bold throughout this book.

   NOTE: As you begin typing a tag, your code editor may display a list of suggested tags. For now, we recommend ignoring those code hints. Also, when you type the **</** part of an ending tag, your code editor may automatically finish the closing tag for you. This is a nice time-saver, but be sure to double-check your code.

   ```
   <html>
   <head>
      <title>Microsoft Patents Ones and Zeroes - The Onion</title>
   </head>
   <body>
      Microsoft Patents Ones and Zeroes
   ```

5. Scroll to the bottom and add the following closing tags highlighted in bold:

   ```
      Wall Street reacts to MS Patent News. Read more...
      This report was written by The Onion
   </body>
   </html>
   ```

   Let's break this code down:

   - HTML tag names are enclosed in less-than (<) and greater-than (>) signs.
   - Each tag wraps the content it describes, so there is a start and an end tag.
   - The end tag has a slash (/) before the tag name.

   Notice that the `<html>` tag wraps around the entire document. Inside that, there are two sections: the `<head>` and the `<body>`. The **head** contains information about the document, such as its title and other data that the visitor to the site will not interact with, whereas the **body** contains the document's actual content that a visitor will see and interact with.

   The `<title>` should be a concise description of this page's content. That is important for Search Engine Optimization (SEO), as the title plays an important role in most search engines' ranking. Additionally, it's often the text search engines display in their search results. Titles appear in the current browser tab, as well as a browser's bookmarks (if a user bookmarks the page). Because the **title** is inside the **head** tag, we indented it to make it easier to see the structure.

6. Let's see what this page looks like in a browser. Note that when you preview this file, the text is going to be one big blob of text because we haven't formatted it yet. Before we preview, we need to save the file. Hit **Cmd–S** (Mac) or **Ctrl–S** (Windows).

7. Keep this file open in your code editor, but switch to the Desktop (the **Finder** on Mac or **Windows Explorer** on Windows).

8. Go into the **Class Files** folder, then **Web Dev Class** folder, then **News Website**.
9. **Ctrl–click** (Mac) or **Right–click** (Windows) on **microsoft.html**, go to **Open with** and select your favorite browser.
10. In the browser's tab (at the top of the window), you should see the text you put the **`<title>`** tag. (In Safari you won't see the title unless the tab bar is visible.)
11. Next, take a look at the page. The actual content of the page—which is what is wrapped inside the **`<body>`** tag—looks like one long paragraph of text. That's because line breaks in the code are ignored by web browsers.

    NOTE: We recommend leaving **microsoft.html** open in the browser as you work, so you can reload the page to see the changes as you make them in the code.
12. Return to **microsoft.html** in your code editor.

Headings

Headings help organize content, so visitors can quickly skim a page. They also make the page more accessible (visually impaired visitors using a screen reader can hear the headings, and jump between them using keystrokes). Headings also help search engines (like Google) to better understand what the page's content is about. There are six levels of headings from **h1** (the main topic) to **h6** (the most nested subtopic).

1. Our page has a single heading, with no subtopics. In your code editor, at the top of the **body** section, wrap the heading text in an **h1** tag as shown below in bold:

```
<body>
   <h1>Microsoft Patents Ones and Zeroes</h1>
```

2. Save the file.
3. Return to the browser and click the **Reload** button to refresh the browser window with the new code.
4. Notice that the heading is bold. More important headings would also be larger than less important ones. These defaults, along with the space around the heading, can be modified using Cascading Style Sheets (CSS). For now, we are just focusing on the markup, so we won't change their appearance.

Paragraphs

Even though there appear to be paragraphs in the provided text, the browser doesn't know where the paragraphs start and end. We have to code that.

1. Return to **microsoft.html** in your code editor.

2. For the first paragraph, add the following opening and closing tags shown in bold:

```
<h1>Microsoft Patents Ones and Zeroes</h1>

<p>REDMOND, WA: In what CEO Bill Gates called an unfortunate but necessary
step to protect our intellectual property from theft and exploitation by
competitors, the Microsoft Corporation patented the numbers one and zero
Monday.</p>
```

3. Wrap the remaining paragraphs, as shown below:

```
<p>With the patent, Microsoft's rivals are prohibited from manufacturing or
selling products containing zeroes and ones unless a royalty fee of 10 cents
per digit used is paid to the software giant.</p>

<p>Some other patents that Microsoft owns:</p>

   Windows
   Microsoft Office
   Xbox

<p>Wall Street reacts to MS Patent News. Read more...</p>
<p>This report was written by The Onion</p>
```

4. Save the file.

5. Return to the browser and click the **Reload** button. You should now see multiple paragraphs that are separated by some vertical space.

Lists

To create a bulleted list, we must use two tags: **`<ul>`** and **`<li>`**:

- The **`<ul>`** tag creates an **unordered list**, which is a bulleted list. There is also an **`<ol>`** tag for an ordered list, which uses numbers or letters (1, 2, 3 or A, B, C).
- The **`<li>`** tag is wrapped around the contents of each list item.

1. Return to **microsoft.html** in your code editor. Mark up a bulleted list of Microsoft's patents by adding the following code highlighted in bold:

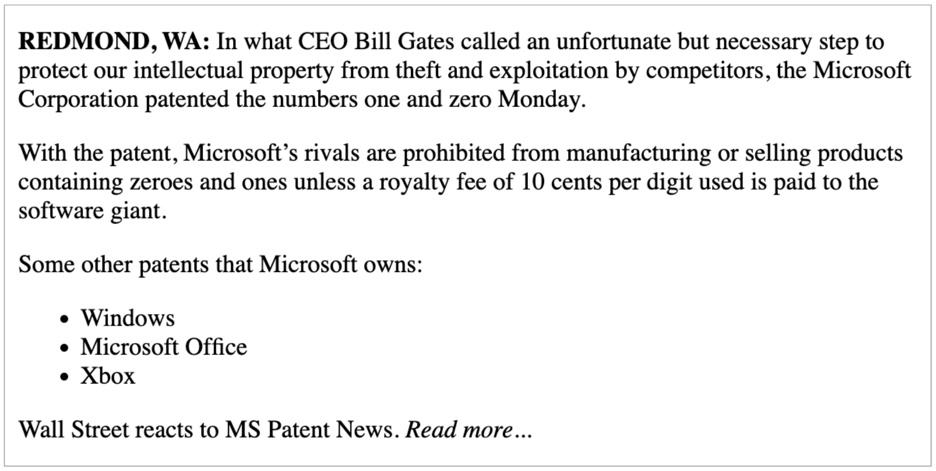
```
<p>Some other patents that Microsoft owns:</p>
<ul>
   <li>Windows</li>
   <li>Microsoft Office</li>
   <li>Xbox</li>
</ul>

<p>Wall Street reacts to MS Patent News. Read more...</p>
```

2. Save the file.
3. Return to the browser and click the **Reload** button to refresh the browser to see a bulleted list with three items.

   Congratulations, you've made a webpage! Leave the file open so you can use it in the next exercise.

Exercise Preview



REDMOND, WA: In what CEO Bill Gates called an unfortunate but necessary step to protect our intellectual property from theft and exploitation by competitors, the Microsoft Corporation patented the numbers one and zero Monday.

With the patent, Microsoft's rivals are prohibited from manufacturing or selling products containing zeroes and ones unless a royalty fee of 10 cents per digit used is paid to the software giant.

Some other patents that Microsoft owns:

- Windows
- Microsoft Office
- Xbox

Wall Street reacts to MS Patent News. *Read more…*

Exercise Overview

In this exercise you'll learn how to bold and italicize text, as well as some important tags and attributes that every webpage needs.

1. If you completed the previous exercise, **microsoft.html** should still be open in your code editor, and you can skip the following sidebar. If you closed **microsoft.html**, re-open it now (from the **News Website** folder). We recommend you finish the previous exercise (1A) before starting this one. If you haven't finished it, do the following sidebar.

If You Did Not Do the Previous Exercise (1A)

1. Close any files you may have open.
2. In your code editor, open **microsoft-ready-for-doctype.html** from the **News Website** folder.
3. Do a **File > Save As** and save the file as **microsoft.html**, replacing the older version in your folder.

Strong & Em (Bold & Italic)

The **`<strong>`** tag is rendered as **bold** and **`<em>`** (short for emphasis) is rendered as *italic* in a browser. These tags may change the speaking style of a screen reader, which is an app used by people who are blind or visually impaired.

1. Return to **microsoft.html** in your code editor and add the following tags to make some text bold:

   ```
   <p><strong>REDMOND, WA:</strong> In what CEO Bill Gates called an unfortunate
   but necessary step to protect our intellectual property from theft and
   exploitation by competitors, the Microsoft Corporation patented the numbers
   one and zero Monday.</p>
   ```

2. Near the bottom, add the following tags to italicize some text:

   ```
   <p>Wall Street reacts to MS Patent News. <em>Read more...</em></p>
   <p>This report was written by The Onion</p>
   ```

3. Save the file.

4. Return to the browser and click the **Reload** button to see the bold and italic text.

Adding Attributes

1. Return to **microsoft.html** in your code editor.

2. Add the following attribute (highlighted in bold) to the **html** tag:

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

   Take a moment to review the code and check for typos. Notice that **`lang="en"`** is written after the tag name but before the greater-than (>) sign.

   The **lang** attribute declares the language of a page (or part of a page) and **en** is the international standard code for English. This is helpful for accessibility and Search Engine Optimization (SEO).

Attributes

Many HTML tags can be enhanced with **attributes**, modifiers of HTML elements. Most attributes have a name and a value. Attributes are only added to a start tag and are written like this `<tag attribute="value">`

Meta & Doctype Tags

1. Add the following new line of code highlighted in bold:

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Microsoft Patents Ones and Zeroes - The Onion</title>
</head>
```

Meta tags can do various things, but this one says that special characters are encoded as Unicode (UTF-8). This means that special characters (like ©, ™, etc.) can be typed into the code and they'll display properly across platforms and devices.

This particular meta tag must occur within the first 512 bytes of the file, so it is considered good practice for this to be the first child of the `<head>` element.

2. Place the cursor at the very beginning of the code—directly before the **html** tag—and hit **Return** (Mac) or **Enter** (Windows) to get a new line on top.

3. Now add the following code highlighted in bold:

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

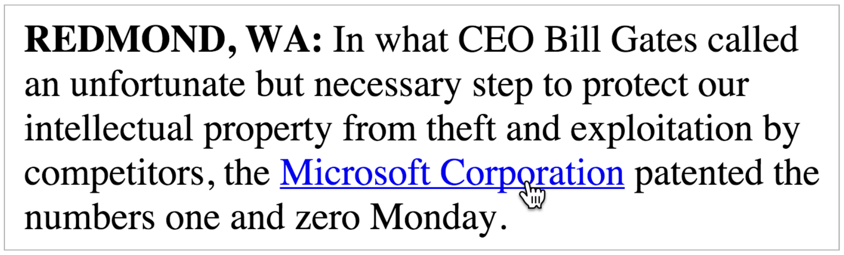
This **doctype** is for HTML5 and beyond. It tells the browser to render the page according to the latest HTML and CSS specifications (which is called "standards mode"). Using older doctypes (or omitting a doctype) can render the page in "quirks mode", which emulates the nonstandard behavior of old browsers.

4. Notice the doctype and meta tags do not have an ending tag (unlike most of the tags we've used so far). That's because they do not wrap around content.

5. Save the file.

Congratulations, you've finished your first webpage! Leave the file open so you can use it in the next exercise.

Exercise Preview


REDMOND, WA: In what CEO Bill Gates called an unfortunate but necessary step to protect our intellectual property from theft and exploitation by competitors, the Microsoft Corporation patented the numbers one and zero Monday.

Exercise Overview

What would a webpage be without links? In this exercise, you'll code a few links to learn how it's done for both external websites and other pages on your site.

1. If you completed the previous exercise, **microsoft.html** should still be open in your code editor, and you can skip the following sidebar. If you closed **microsoft.html**, re-open it now (from the **News Website** folder). We recommend you finish the previous exercise (1B) before starting this one. If you haven't finished it, do the following sidebar.

If You Did Not Do the Previous Exercise (1B)

1. Close any files you may have open.
2. In your code editor, open **microsoft-ready-for-links.html** from the **News Website** folder.
3. Do a **File > Save As** and save the file as **microsoft.html**, replacing the older version in your folder.

The Anchor Tag & HREF Attribute

1. In the first paragraph, wrap an **<a>** tag (which stands for anchor) around **Microsoft Corporation**, as shown below in bold:

```
<p><strong>REDMOND, WA:</strong> In what CEO Bill Gates called an unfortunate
but necessary step to protect our intellectual property from theft and
exploitation by competitors, the <a href="https://www.microsoft.com">Microsoft
Corporation</a> patented the numbers one and zero Monday.</p>
```

NOTE: The anchor tag wouldn't do anything without the **href** attribute. Href stands for hyperlink reference. Its value is equal to the URL (uniform resource locator) of the page you're linking to.

2. Take a moment to review the code and check for typos. Remember that attributes are only added to a start tag and are written like this `<tag attribute="value">`

3. Save the file.

4. Preview **microsoft.html** in a browser.

Browser Preview Shortcuts

If you're using Visual Studio Code with the **open in browser** extension installed, hit **Option–B** (Mac) or **Alt–B** (Windows) to open the saved HTML document in your default browser. If asked what browser you want to open it with, ideally you should choose Google Chrome (which is the most widely used browser).

5. Click the link to make sure it works. Links that include the entire address (including the **`http://www`**) are called **absolute** links.

6. Return to **microsoft.html** in your code editor. Create another link, this time to The Onion's website (in the last paragraph, just before the end **body** tag):

```
   <p>Wall Street reacts to MS Patent News. <em>Read more...</em></p>
   <p>This report was written by <a href="https://www.theonion.com">
The Onion</a></p>
</body>
```

7. Save the file.

8. Preview **microsoft.html** in a browser and test the new link.

   We just made two links to outside websites, but what about a link to a page in this site? We made another page (called **wall-street.html**) which is in the same folder as the page you're currently editing.

9. Return to **microsoft.html** in your code editor and below the bulleted list, create that link by adding the code shown in bold:

```
   <p>Wall Street reacts to MS Patent News. <a href="wall-
street.html"><em>Read more...</em></a></p>
   <p>This report was written by <a href="https://www.theonion.com">The
Onion</a></p>
</body>
```

   Links like this, which don't include the full address, are called **relative** links. The link is relative to the current .html file. In this case, the link will look for a page called **wall-street.html** in the same folder as this file (**microsoft.html**).

10. Save the file.

11. Preview **microsoft.html** in a browser and test out the links. If the browser is still open to the page, you can hit the **Reload** button.

Opening a Link in a New Browser Tab/Window

We can make a link open in a new browser tab or window (depending on the user's preference) using the **target** attribute.

1. Return to **microsoft.html** in your code editor.
2. In the first paragraph, add the **target** attribute to the **microsoft.com** link as shown below. NOTE: Attributes must be separated by a space character.

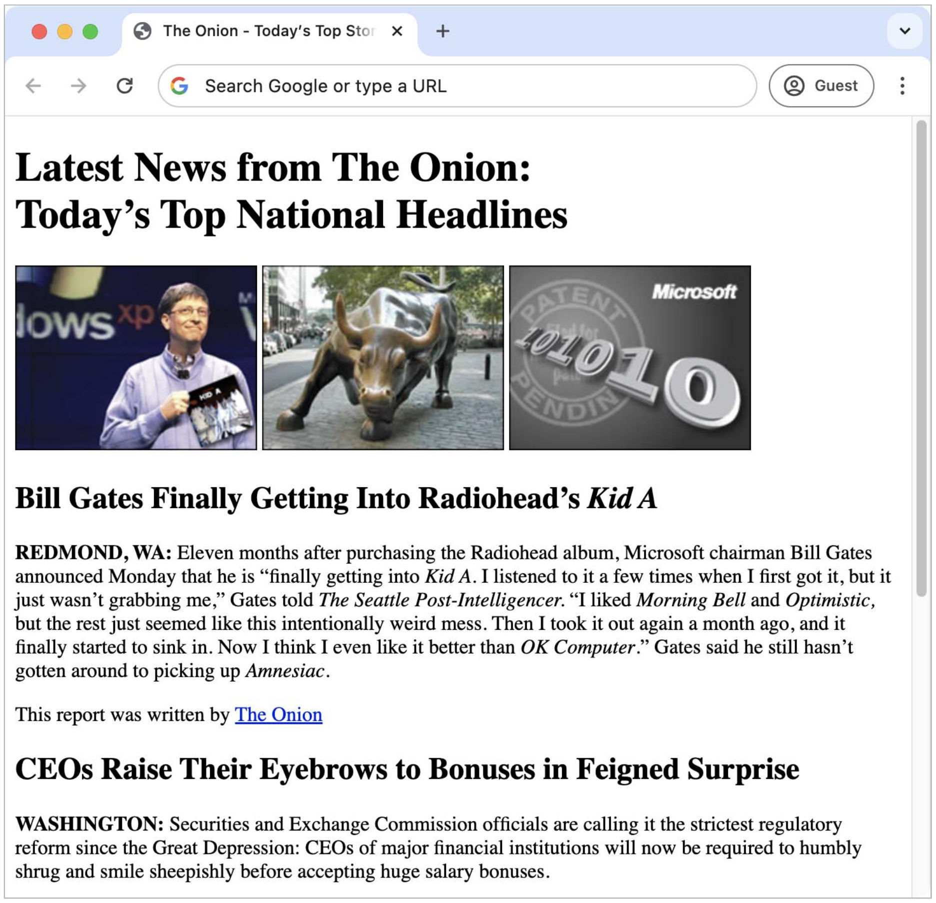
```
<p><strong>REDMOND, WA:</strong> In what CEO Bill Gates called an unfortunate
but necessary step to protect our intellectual property from theft and
exploitation by competitors, the <a href="https://www.microsoft.com"
target="_blank">Microsoft Corporation</a> patented the numbers one and zero
Monday.</p>
```

3. In the last paragraph, add the **target** attribute to the **theonion.com** link as shown below:

```
<p>This report was written by <a href="https://www.theonion.com"
target="_blank">The Onion</a></p>
```

4. Save the file.
5. Preview **microsoft.html** in a browser. Click the links for **Microsoft Corporation** and **The Onion** and those pages should open in a new browser tab or window!
6. Return to your code editor and close any files you have open. We'll be moving on to a new file in the next exercise.

Exercise Preview



Latest News from The Onion:
Today's Top National Headlines

Bill Gates Finally Getting Into Radiohead's *Kid A*

REDMOND, WA: Eleven months after purchasing the Radiohead album, Microsoft chairman Bill Gates announced Monday that he is "finally getting into *Kid A*. I listened to it a few times when I first got it, but it just wasn't grabbing me," Gates told *The Seattle Post-Intelligencer*. "I liked *Morning Bell* and *Optimistic*, but the rest just seemed like this intentionally weird mess. Then I took it out again a month ago, and it finally started to sink in. Now I think I even like it better than *OK Computer*." Gates said he still hasn't gotten around to picking up *Amnesiac*.

This report was written by The Onion

CEOs Raise Their Eyebrows to Bonuses in Feigned Surprise

WASHINGTON: Securities and Exchange Commission officials are calling it the strictest regulatory reform since the Great Depression: CEOs of major financial institutions will now be required to humbly shrug and smile sheepishly before accepting huge salary bonuses.

Exercise Overview

In this exercise, you'll learn how to add images to a page, and include alternative text for accessibility. You will also learn about two other tags: one for creating line breaks and another for creating the appearance of content division. These three tags have something in common… none of them wrap around content.

1. In your code editor, go to **File > Open**.
2. Navigate to **Desktop > Class Files > Web Dev Class > News Website**.
3. Double–click on **index.html** to open it.

   NOTE: **index.html** is a special filename reserved for the first page (the homepage) of a website. When you go to a .com URL in a browser, the **index.html** is the first page that will be displayed.

4. Preview in a browser to get a feeling for the page. TIP: If you're using Visual Studio Code with the **open in browser** extension installed, hit **Option–B** (Mac) or **Alt–B** (Windows) and your saved HTML document will open in your default browser.

Adding a Line Break

The main heading at the top (which is an h1) is a bit long. A **break** tag `<br>` would help to make it more legible.

1. Return to **index.html** in your code editor. Add a break tag to push Today's Top National Headlines to the next line:

```
<body>
   <h1>
      Latest News from The Onion:<br>
      Today's Top National Headlines
   </h1>
```

Like the **doctype** and **meta** tags, the **`<br>`** tag does not have a closing tag because it has no content inside of it. It simply performs its own function.

2. Save the file.
3. Return to the browser and reload the page to see the line break in the top heading.

Adding Image Files

Images are inserted into the HTML document with a single tag (they have no end tag). There are two main graphic file formats we'll use in this book: JPEG and PNG. Please see the **Graphic File Formats** reference in the front of the workbook for more information on file formats.

1. Return to **index.html** in your code editor.
2. To add an image below the main heading, add the following line of code highlighted in bold. It may not look like a single line in the book, but be sure to enter it on one line in your code editor.

```
<body>
   <h1>
      Latest News from The Onion:<br>
      Today's Top National Headlines
   </h1>
   <img src="images/newsthumb-bill-gates.jpg" height="145" width="190"
alt="Bill Gates, Radiohead Fan">

   <h2>Bill Gates Finally Getting Into Radiohead’s <em>Kid A</em></h2>
```

3. Take a moment to review the code you just typed. The **`<img>`** tag requires a **src** attribute (**src** is an abbreviation for **source**) to call the appropriate image file. Images are typically stored in a subfolder of the website, which we've named **images**. There's nothing special about that name (it could be named **img** if you want a shorter name), but we do have to know the folder name to link to images inside it.

   Notice that the code includes a **width** and a **height** attribute. Specifying the dimensions of an image can slightly speed up the rendering of the page.

   Alt Text

   The img tag's **alt** attribute (commonly referred to as **alt text**) is important for accessibility and Search Engine Optimization (SEO). It is a brief text description of a graphic that is used by screen readers, search engines, and is displayed if the graphic does not load.

   It's best practice to add alt text to all graphics. The only time no alt text is needed is if the graphic is purely decorative. In those rare cases, add an empty alt attribute (**`alt=""`**) or use CSS techniques so the graphics can be ignored by screen readers.

4. Save the file.

5. Return to the browser and reload the page to see the image.

   NOTE: Image files are not actually embedded into an HTML page. The image tag is a placeholder for the linked source file, so images must be uploaded (along with the HTML pages) to your remote web server in order for visitors to see them.

6. Return to your code editor, and add another image below the previous one by typing the following code shown in bold:

```
<body>
   <h1>
      Latest News from The Onion:<br>
      Today's Top National Headlines
   </h1>
   <img src="images/newsthumb-bill-gates.jpg" height="145" width="190"
alt="Bill Gates, Radiohead Fan">
   <img src="images/newsthumb-wall-street-bull.jpg" height="145" width="190"
alt="Charging Bull">

   <h2>Bill Gates Finally Getting Into Radiohead’s <em>Kid A</em></h2>
```

7. Save the file.

8. Return to the browser and reload the page to see that the images sit in a line, rather than stacking on top of each other.

Img Is an Inline Element

Images, text, and anchor tags are considered **inline** elements because each one goes next to the other to form a line of content.

If you want to force inline elements (such as images) to stack on top of each other, you can put them in a container that is too narrow for them to fit next to each other. You could also change their CSS display property.

9. Return to the code and add a third image below the second one you just added:

```
<img src="images/newsthumb-bill-gates.jpg" height="145" width="190" alt="Bill
Gates, Radiohead Fan">
<img src="images/newsthumb-wall-street-bull.jpg" height="145" width="190"
alt="Charging Bull">
<img src="images/newsthumb-patent.jpg" height="145" width="190" alt="Microsoft
Patent Illustration">

<h2>Bill Gates Finally Getting Into Radiohead's <em>Kid A</em></h2>
```

10. Save the file, return to the browser, and reload the page to preview the new image.

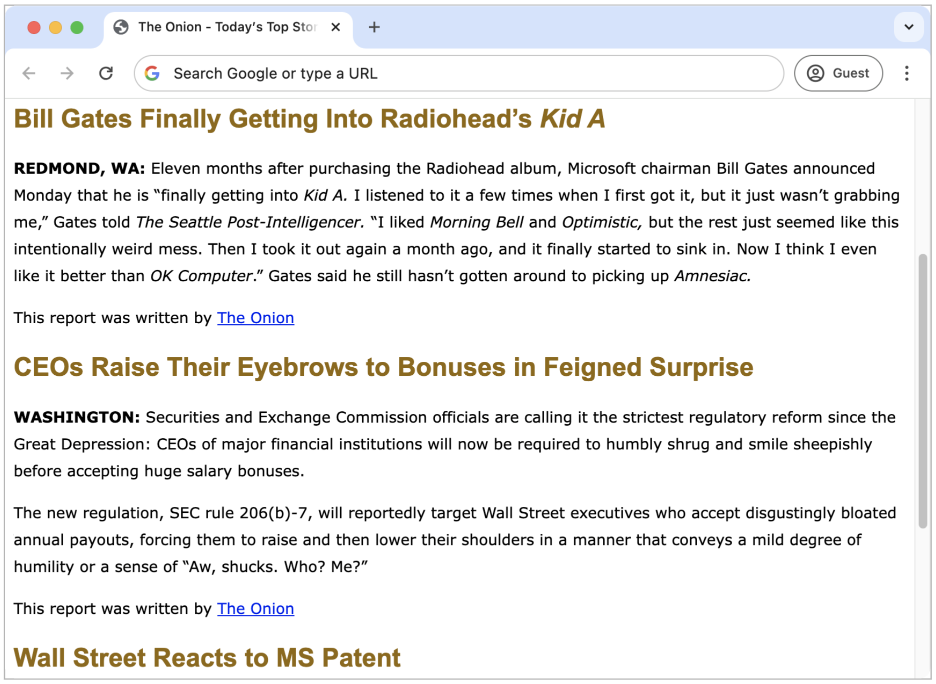
11. You can leave the file open in the browser and the code editor so you can use it in the next exercise.

Graphic File Formats: JPEG, PNG, etc.

To learn more about which file formats to use for graphics, read the **Graphic File Formats for the Web** reference at the back of this book.

Intro to Cascading Style Sheets (CSS)

Exercise Preview



Bill Gates Finally Getting Into Radiohead's *Kid A*

REDMOND, WA: Eleven months after purchasing the Radiohead album, Microsoft chairman Bill Gates announced Monday that he is "finally getting into *Kid A*. I listened to it a few times when I first got it, but it just wasn't grabbing me," Gates told *The Seattle Post-Intelligencer.* "I liked *Morning Bell* and *Optimistic,* but the rest just seemed like this intentionally weird mess. Then I took it out again a month ago, and it finally started to sink in. Now I think I even like it better than *OK Computer*." Gates said he still hasn't gotten around to picking up *Amnesiac.*

This report was written by The Onion

CEOs Raise Their Eyebrows to Bonuses in Feigned Surprise

WASHINGTON: Securities and Exchange Commission officials are calling it the strictest regulatory reform since the Great Depression: CEOs of major financial institutions will now be required to humbly shrug and smile sheepishly before accepting huge salary bonuses.

The new regulation, SEC rule 206(b)-7, will reportedly target Wall Street executives who accept disgustingly bloated annual payouts, forcing them to raise and then lower their shoulders in a manner that conveys a mild degree of humility or a sense of "Aw, shucks. Who? Me?"

This report was written by The Onion

Wall Street Reacts to MS Patent

Exercise Overview

In this exercise, you'll style text (font, color, size, etc) using Cascading Style Sheets (CSS). HTML defines the type of content: heading, paragraph, list, etc. CSS tells the browser how to style that content.

1. If you completed the previous exercise, **index.html** should still be open in your code editor, and you can skip the following sidebar. If you closed **index.html**, re-open it now (from the **News Website** folder). We recommend you finish the previous exercise (1D) before starting this one. If you haven't finished it, do the following sidebar.

If You Did Not Do the Previous Exercise (1D)

1. Close any files you may have open.
2. In your code editor, open **index-ready-for-styles.html** from the **News Website** folder.
3. Do a **File > Save As** and save the file as **index.html**, replacing the older version in your folder.

2. Preview the file in a browser so you can see how it looks. TIP: If you're using Visual Studio Code with the **open in browser** extension installed, hit **Option–B** (Mac) or **Alt–B** (Windows) and your saved HTML document will open in your default browser.
3. Notice the main heading (h1), three images, and then three stories. Each story has a subheading (h2) and some paragraphs of text.

Tag Selectors: A Way to Set "Default" Appearance

1. Return to **index.html** in your code editor.
2. CSS is a list of stylistic rules for the browser to follow. Because CSS works behind the scenes, it is written in the **head** section rather than in the **body**. Add the following bold code in the **head**, just below the **title** tag:

```
   <title>The Onion - Today’s Top Stories</title>
   <style>

   </style>
</head>
```

3. A CSS selector tells the browser where to apply a style. The first type of CSS selector we'll use is called a **tag selector**. It tells the browser to find all instances of a particular HTML tag and style them. Add the following bold **h1** tag selector inside the `<style>` tag:

```
<style>
   h1 {

   }
</style>
```

   This rule will target all the h1's on the page.

4. Inside the h1 tag selector add the following bold code:

```
h1 {
   font-family: Arial, Helvetica, sans-serif;
   font-size: 35px;
   color: #006b3a;
}
```

   A CSS **rule** is the selector (in this case **h1**) through the closing curly brace. Inside the curly braces {} we add **property declarations**. For each predefined CSS property (font-size, color, etc.) we set a value (35px, #006b3a, etc.).

   We recommend putting each property on its own indented line, which your code editor may do automatically. It's not required, but makes the code easier to read.

5. Save the file.

6. Return to the browser and reload the page to see that the heading at the top has changed. Awesome!

The Font-Family Property

The **font-family** property is a wish list. The first font will be applied if it's available on the user's computer. The second, third, or any additional fonts will only be used if the preceding font is not available. This list is called the font stack.

Hexadecimal Color Codes

Web colors are commonly coded as a 6-digit hexadecimal value that represents RGB color values: the first 2 digits are red, the next 2 green, and the last 2 are blue. Hexadecimal values must start with a # sign: **`#00ff33`**

The letters in hex values are not case sensitive—they can be written in either upper or lowercase.

7. Return to **index.html** in your code editor.
8. Inside the **style** tags, below the **h1** style, add a rule for the subheadings (h2 tags). Type only the new code that is highlighted in bold:

```
   h1 {
      font-family: Arial, Helvetica, sans-serif;
      font-size: 35px;
      color: #006b3a;
   }
   h2 {
      font-family: Arial, Helvetica, sans-serif;
      font-size: 24px;
      color: #8f6800;
   }
</style>
```

9. Save the file.
10. Return to the browser and reload the page again to see your new heading styles. You have styled all elements in the page that have been tagged as h1 or h2.
11. Let's add another rule. Return to your code editor.

12. Below the **h2** rule, add a rule for all paragraphs (type only the new rule that is highlighted in bold):

```
   h2 {
      font-family: Arial, Helvetica, sans-serif;
      font-size: 24px;
      color: #8f6800;
   }
   p {
      font-family: Verdana, sans-serif;
      font-size: 14px;
      line-height: 25px;
   }
</style>
```

    NOTE: We use line-height to adjust the space between lines of text (it's called leading in print design). Line-height defines how tall a line of text is. Characters are vertically centered within this space, so increasing line-height adds space above and below the characters. Readability can be dramatically improved by increasing line-height, particularly if the lines of text are long.

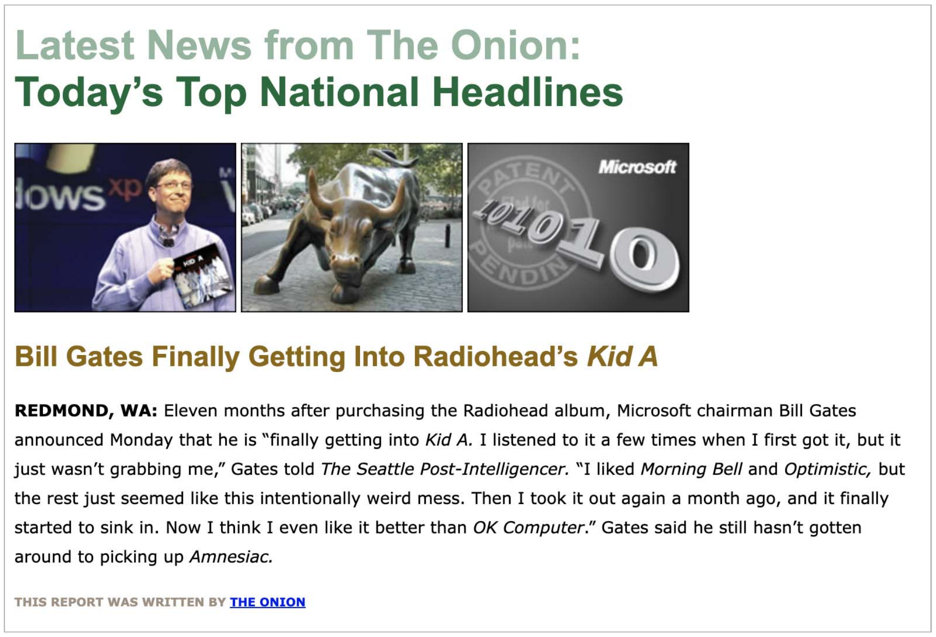
13. Save the file.

14. Return to the browser and reload the page. What a difference!

Units of Measure for Type in CSS

We measure screens in **pixels**, so pixels are commonly used to size type in webpages. There are other types of measurement, such as **ems** and **rems**. They are proportional, and require a bit of math to figure out the type size. We're starting with pixels because they are easier.

Exercise Preview


Latest News from The Onion:
Today’s Top National Headlines

Bill Gates Finally Getting Into Radiohead’s *Kid A*

REDMOND, WA: Eleven months after purchasing the Radiohead album, Microsoft chairman Bill Gates announced Monday that he is “finally getting into *Kid A.* I listened to it a few times when I first got it, but it just wasn’t grabbing me,” Gates told *The Seattle Post-Intelligencer.* “I liked *Morning Bell* and *Optimistic,* but the rest just seemed like this intentionally weird mess. Then I took it out again a month ago, and it finally started to sink in. Now I think I even like it better than *OK Computer*.” Gates said he still hasn’t gotten around to picking up *Amnesiac.*

THIS REPORT WAS WRITTEN BY THE ONION

Exercise Overview

In the previous exercise you learned that CSS tag selectors are a quick way to style all instances of a tag. But what if you want a one paragraph to look different from the rest? In this exercise you’ll learn how to use the **class selector** to override a tag selector anywhere you like.

1. If you completed the previous exercise, **index.html** should still be open in your code editor, and you can skip the following sidebar. If you closed **index.html**, re-open it now (from the **News Website** folder). We recommend you finish the previous exercises (1D–2A) before starting this one. If you haven’t finished it, do the following sidebar.

If You Did Not Do the Previous Exercises (1D–2A)

1. Close any files you may have open.
2. In your code editor, open **index-ready-for-classes.html** from the **News Website** folder.
3. Do a **File > Save As** and save the file as **index.html**, replacing the older version in your folder.

2. Preview the file in a browser so you can see how it looks.

   TIP: If you're using Visual Studio Code with the **open in browser** extension installed, hit **Option–B** (Mac) or **Alt–B** (Windows) and your HTML file will open in your default browser.

Class Selectors: Making Exceptions to the Defaults

We want to style some paragraphs differently than the rest. To do this we must give them a name (called a **class**) that we'll use to target them for styling. We can add a **class** attribute to any HTML tag.

1. There are three paragraphs that say **This report was written by**… Find the first one (around line 36).

2. On the opening **p** tag, add **`class="author"`** as shown below in bold:

   ```
   <p class="author">This report was written by <a href="https://
   www.theonion.com" target="_blank">The Onion</a></p>
   ```

   NOTE: There are no predefined class names. You choose the name, ideally a name that describes what the element is (such as **author**), rather than how it will look (such as **uppercase**). We don't want to have to change the class name if we later decide to make it look different.

3. Now that we have a way to refer to this specific paragraph, we can define a CSS rule for how it should look. In the **style** tag, under the **p** style, add the following bold code. Be sure to type the period (.) before **author**, which is what tells the browser it's a **class** selector!

   ```
      p {
         font-family: Verdana, sans-serif;
         font-size: 14px;
         line-height: 25px;
      }
      .author {
         font-size: 10px;
         text-transform: uppercase;
         font-weight: bold;
         color: #a18f81;
      }
   </style>
   ```

4. Save the file.

5. Return to the browser and reload the page. Notice that only the first instance of **THIS REPORT WAS WRITTEN BY** should be uppercase and sand brown.

6. Return to **index.html** in your code editor.

7. Classes can be reused as many times as needed. Find the second **This report was written**… paragraph (around line 48) and add **class="author"** as shown below in bold:

```
<p>The new regulation, SEC rule 206(b)-7, will reportedly target Wall Street
executives who accept disgustingly bloated annual payouts, forcing them to
raise and then lower their shoulders in a manner that conveys a mild degree of
humility or a sense of "Aw, shucks. Who? Me?"</p>
<p class="author">This report was written by <a href="https://
www.theonion.com" target="_blank">The Onion</a></p>
```

8. One more to go. Near the bottom, find the last **This report was written**… paragraph and add **class="author"** as shown below in bold:

```
   <p class="author">This report was inspired by, but not written by <a
href="https://www.theonion.com" target="_blank">The Onion</a></p>
</body>
</html>
```

9. Save the file.

10. Return to the browser and reload the page to see that the author style applies to all three author paragraphs.

The Span Tag

1. Return to **index.html** in your code editor.

2. We want to make the first part of the h1 tag a bit lighter. We need an HTML tag to add the **class** attribute, so we'll need to wrap a **<span>** tag around the words we want to group together.

   Find the **h1** near the start of the **body** section, and add the following bold code:

```
<h1>
   <span class="muted">Latest News from The Onion:</span><br>
   Today's Top National Headlines
</h1>
```

   NOTE: Class names are case sensitive. They cannot contain spaces, so use hyphens or underscores instead. They cannot start with a number, and you should avoid using special characters.

3. Now we can make the CSS rule to style our new class. Under the **.author** style, add the following bold code:

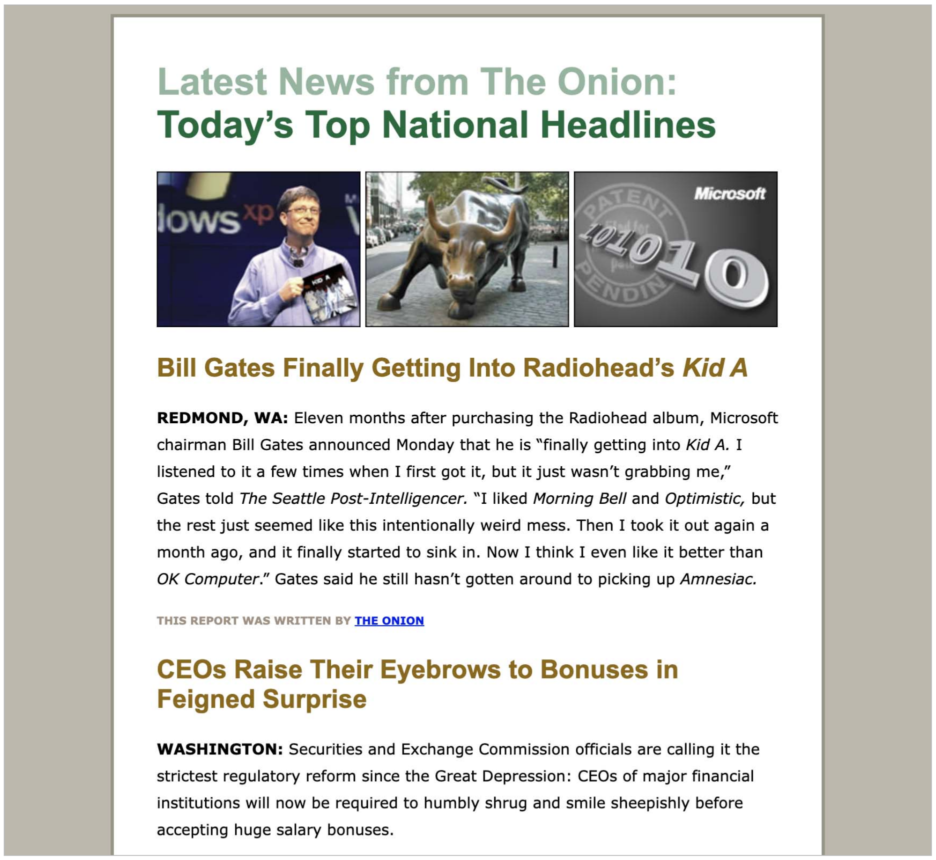
   ```
     .author {
        font-size: 10px;
        text-transform: uppercase;
        font-weight: bold;
        color:#a18f81;
     }
     .muted {
        opacity: .5;
     }
   </style>
   ```

   NOTE: Opacity takes a value between 0 and 1 (1 is 100% visible, and 0 is invisible). A decimal such as .5 or 0.5 (the preceding 0 is optional) would be 50% transparent.

4. Save the file.

5. Return to the browser and reload the page. You should see that **Latest News from The Onion:** is lighter (because it's partially transparent and showing through to the white background behind the text).

Exercise Preview


Latest News from The Onion:
Today's Top National Headlines

Bill Gates Finally Getting Into Radiohead's *Kid A*

REDMOND, WA: Eleven months after purchasing the Radiohead album, Microsoft chairman Bill Gates announced Monday that he is "finally getting into *Kid A.* I listened to it a few times when I first got it, but it just wasn't grabbing me," Gates told *The Seattle Post-Intelligencer.* "I liked *Morning Bell* and *Optimistic,* but the rest just seemed like this intentionally weird mess. Then I took it out again a month ago, and it finally started to sink in. Now I think I even like it better than *OK Computer.*" Gates said he still hasn't gotten around to picking up *Amnesiac.*

THIS REPORT WAS WRITTEN BY THE ONION

CEOs Raise Their Eyebrows to Bonuses in Feigned Surprise

WASHINGTON: Securities and Exchange Commission officials are calling it the strictest regulatory reform since the Great Depression: CEOs of major financial institutions will now be required to humbly shrug and smile sheepishly before accepting huge salary bonuses.

Exercise Overview

In this exercise you'll take more control over the layout of the page, using a **div** tag (div is short for division). Wrapping content in div tags lets us create sections of content that are grouped together and can be styled with CSS.

Div versus Span

In the previous exercise we used a **span** tag to do something similar, but here's the main difference:

- **Span** tags are **inline** elements, meaning multiple span tags go next to each other in a line.
- **Div** tags are **block** elements, meaning multiple div tags stack on top of each other.

1. If you completed the previous exercise, **index.html** should still be open in your code editor, and you can skip the following sidebar. If you closed **index.html**, re-open it now (from the **News Website** folder). We recommend you finish the previous exercises (1D–2B) before starting this one. If you haven't finished them, do the following sidebar.

If You Did Not Do the Previous Exercises (1D–2B)

1. Close any files you may have open.
2. In your code editor, open **index-ready-for-divs.html** from the **News Website** folder.
3. Do a **File > Save As** and save the file as **index.html**, replacing the older version in your folder.

Wrapping Content in a Div & Setting a Page Width

1. Preview the file in a browser so you can see how it looks. Take a moment to click and drag the edge of the browser window in and out to resize the window. Notice how all the content, including the images, wrap to the edge of the browser window. This isn't all that problematic for the more narrow window size, but if you make the window rather wide, the content stretches out to a point where it is incredibly hard to read.

   We can use the **`<div>`** tag to get more control over the layout.

2. Return to **index.html** in your code editor.
3. Let's start to wrap a **div** tag around all the content in the page. Type the following opening tag (highlighted in bold) just below the opening **body** tag:

```
</head>
<body>
   <div>
   <h1>
```

4. As shown below in bold, close the **div** tag just above the closing **body** tag (at the bottom of the file):

```
   <p class="author">This report was inspired by, but not written by <a
href="https://www.theonion.com" target="_blank">The Onion</a></p>
   </div>
</body>
</html>
```

5. We recommend indenting the lines in between the opening and closing **div** tags. It's not required for the code to work, but it makes your code vastly more legible.

Code Indentation Shortcuts

Highlight the line(s) of code and try one of these keystrokes:

- **Tab** to indent. **Shift–Tab** to unindent.
- **Cmd–]** (Mac) or **Ctrl–]** (Windows) to indent.
  Cmd–[(Mac) or **Ctrl–[** (Windows) to unindent.

6. Although we currently have only one **div**, pages typically have multiple divs. Assuming more will be added later, to style this div we should give it a name (either a **class** or **ID**). Add the following ID to the opening **div** tag, as follows:

```
<body>
   <div id="wrapper">
      <h1>
```

We'll use this ID to apply CSS to this specific div. Do not use spaces or special characters in an ID. How is an ID different than a class? You can apply a class name to multiple elements, but an ID can only be applied to one element in a page.

7. In the **style** tag, between the **p** and **.author** rules, add the following new rule (highlighted in bold):

```
p {
   CODE OMITTED TO SAVE SPACE
}
#wrapper {
   width: 580px;
}
.author {
   CODE OMITTED TO SAVE SPACE
}
```

ID selectors use a **hash mark** (#) prior to the ID name, which tells the browser to find/style an element with that ID. Only one element can use an ID within a page, making ID selectors the most specific type of selector.

8. Save the file.

9. Return to the browser and reload the page to see the content reflows to fit within the 580 pixel-wide container.

10. Resize the browser window narrower than the width of the content. Notice that you have to scroll horizontally to see whatever does not fit inside the window. This is not ideal, but we wanted you to see how width works. There's an easy way to change this to a flexible, fluid layout that work a bit better in both large and small windows.

11. Return to **index.html** in your code editor.

12. In the **#wrapper** rule, change **width** to **max-width** as shown below in bold:

```
#wrapper {
   max-width: 580px;
}
```

13. Save the file.

14. Return to the browser and reload the page.

15. Resize the browser window narrower than the width of the content and notice that now the content reflows to stay inside the window and you are no longer forced to scroll horizontally. This will be much better for small screens.

Adding a Background-Color

1. Return to **index.html** in your code editor.

2. In the **style** tag, above all the other rules add a new rule for **body** (as shown below in bold):

```
<style>
   body {
      background-color: #bdb8ad;
   }
   h1 {
      font-family: Arial, Helvetica, sans-serif;
      font-size: 35px;
      color: #006b3a;
   }
```

3. Save the file.

4. Return to the browser and reload the page to see that the color applies to the entire page background (behind all the content).

5. Return to **index.html** in your code editor.

6. Let's add a white background behind only the text area to make it pop off the page's background. In the **style** tag, find the rule for **#wrapper** and add the following property declaration (in bold):

```
#wrapper {
   max-width: 580px;
   background-color: #ffffff;
}
```

7. Save the file.
8. Return to the browser and reload the page to see that the white background is applied only to the content inside the div tag. The body has a different background color that is behind the div.

Adding Padding Inside the Div

1. The content is very close to the edge of the browser and white background. Let's add some **padding** (space between the content and the edge of the containing element) to make it more legible. Return to **index.html** in your code editor and add the following property declaration (in bold) to the rule for **#wrapper**:

```
#wrapper {
   max-width: 580px;
   background-color: #ffffff;
   padding: 40px;
}
```

   This sets 40 pixels of padding on all four sides: top, right, bottom, and left.

2. Save the file.
3. Return to the browser and reload to see more white space around the content.

Centering Content in the Window

1. Return to **index.html** in your code editor and add the following two new property declarations (in bold) to the rule for **#wrapper**:

```
#wrapper {
   max-width: 580px;
   background-color: #ffffff;
   padding: 40px;
   margin-left: auto;
   margin-right: auto;
}
```

   NOTE: Margin is space outside of an element. Padding is space inside an element.

2. Save the file.
3. Return to the browser and reload the page to see how the browser centers the div in the browser window.

   Why does this work? Setting left and right margins (space outside an element) to **auto**, will automatically make them equal to one another. When the width for an element (such as the **div**) is smaller than its container (such as the **body**), the effect horizontally centers the div with its container.

Adding a Border

1. Return to **index.html** in your code editor and add the following three new property declarations (in bold) to the rule for **#wrapper**:

```
#wrapper {
   max-width: 580px;
   background-color: #ffffff;
   padding: 40px;
   margin-left: auto;
   margin-right: auto;
   border-width: 3px;
   border-style: solid;
   border-color: #959485;
}
```

   NOTE: The **border-style** property tells the browser to render the border as a solid line. Other possible values include dashed and dotted.

2. Save the file.
3. Return to the browser and reload the page to see the 3-pixel solid border around the white background div.

CSS Clean-up: Shorthand & "DRY"

Instead of specifying the width, style, and color for the border as three separate property declarations, CSS has a method of "shorthand" we can use to write a border style more elegantly.

1. Return to **index.html** in your code editor.

2. Edit your rule for **#wrapper** as shown below. Note that, instead of three lines of code to describe the border properties, one **border** declaration can be written like so:

```
#wrapper {
   max-width: 580px;
   background-color: #ffffff;
   padding: 40px;
   margin-left: auto;
   margin-right: auto;
   border: 3px solid #959485;
}
```

3. While we're cleaning up our CSS, take note of how often we declare **font-family: Arial, Helvetica, sans-serif;**

   A core principle of coding is **DRY**, which stands for "don't repeat yourself". We try to avoid redundancies and write the most elegant code possible.

   We can declare the default font-family once (in the body rule) and have all elements inherit this property, because they sit inside the body tag.

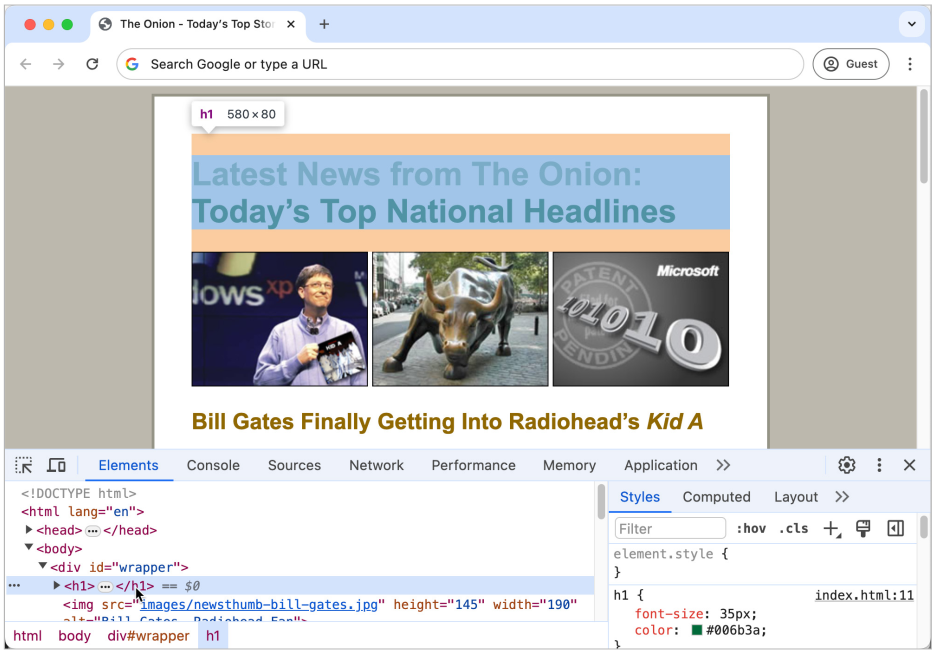
4. Edit your CSS rules as shown below. Delete the **font-family** declaration from **h1** and **h2**, and instead, place it inside the **body** rule:

```
body {
   font-family: Arial, Helvetica, sans-serif;
   background-color: #bdb8ad;
}
h1 {
   font-size: 35px;
   color: #006b3a;
}
h2 {
   font-size: 24px;
   color: #8f6800;
}
```

5. Save the file.

6. Return to the browser and reload the page. It should look the same as it did before the change, but the code is leaner and more elegant.

Exercise Preview


The Onion - Today's Top Sto
Search Google or type a URL
Guest
h1 580 × 80
Latest News from The Onion: Today's Top National Headlines
Bill Gates Finally Getting Into Radiohead's *Kid A*
Elements
Console
Sources
Network
Performance
Memory
Application
Styles
Computed
Layout
Filter
:hov
.cls
html
body
div#wrapper
h1

```
<!DOCTYPE html>
<html lang="en">
 ▶<head> ⋯ </head>
 ▼<body>
   ▼<div id="wrapper">
···  ▶<h1> ⋯ </h1> == $0
      <img src="images/newsthumb-bill-gates.jpg" height="145" width="190"
```

```
element.style {
}
h1 {                                    index.html:11
   font-size: 35px;
   color: ■ #006b3a;
```

Exercise Overview

All major browsers have built-in tools that let you see and alter HTML and CSS on-the-fly. In this exercise, you'll see how useful they can be when experimenting with code changes.

We'll be using Chrome's DevTools, but the tools in other browsers work similarly.

1. If you completed the previous exercise, **index.html** should still be open, and you can skip the following sidebar. If you closed **index.html**, re-open it now. We recommend you finish the previous exercises (1D–2C) before starting this one. If you haven't finished them, do the following sidebar.

If You Did Not Do the Previous Exercises (1D–2C)

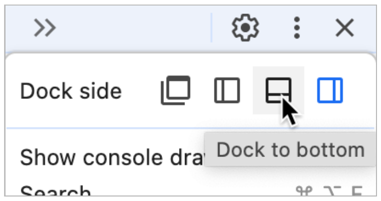
1. Close any files you may have open.
2. In your code editor, open **index-ready-for-dev-tools.html** from the **News Website** folder.
3. Do a **File > Save As** and save the file as **index.html**, replacing the older version in your folder.

Using Chrome's DevTools

1. Preview index.html in **Chrome**.
2. To open the DevTools, **Ctrl–click** (Mac) or **Right–click** (Windows) on any of the images near the top of the page and choose **Inspect**.

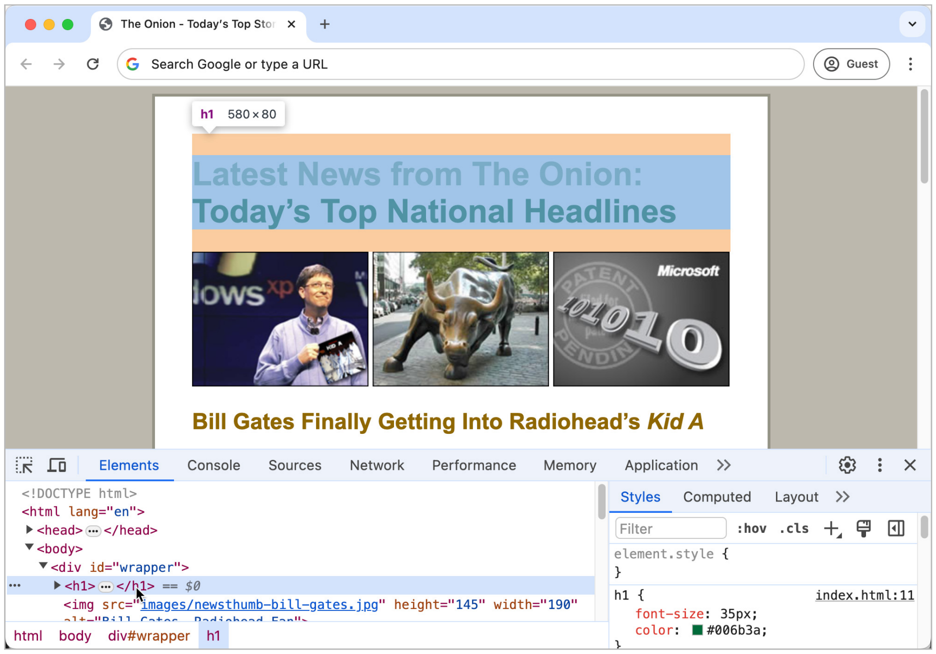
   NOTE: The element you **Ctrl–click** (Mac) or **Right–click** (Windows) on will be initially selected in the DevTools.

3. We want the DevTools docked to the bottom. If the DevTools are docked to the right side of the browser window, at the top right of the DevTools panel click the ⋮ button and then choose **Dock to bottom** as shown below:



4. The DevTools are organized into tabbed panels. The **Elements** panel should currently be open on the left side. Here, you can view and edit all elements in the **DOM tree** (DOM is short for Document Object Model), which is a fancy way of saying you can see the structure of the document and the way each HTML element is nested inside another.

   Hover over some of the tags, noticing that as you do so, the element will be highlighted in the browser window above (along with a tooltip that shows the HTML element's width and height).

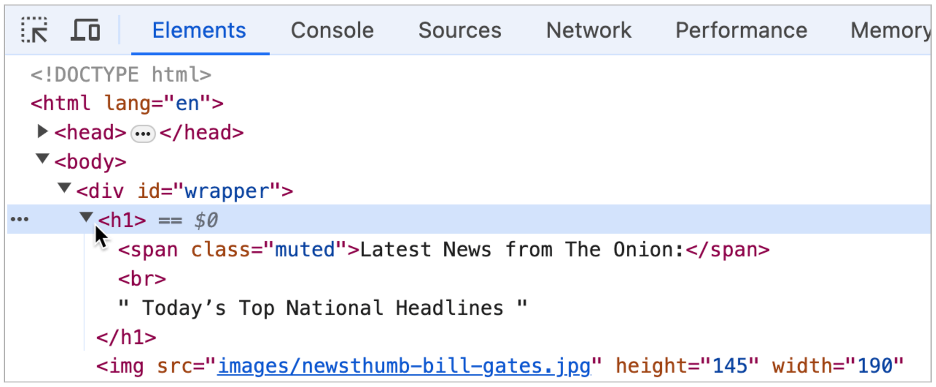

The Onion - Today's Top Sto
Search Google or type a URL
Guest
h1 580 × 80
Latest News from The Onion: Today's Top National Headlines
Bill Gates Finally Getting Into Radiohead's *Kid A*
Elements
Console
Sources
Network
Performance
Memory
Application
Styles
Computed
Layout
Filter
:hov
.cls
html
body
div#wrapper
h1

```
<!DOCTYPE html>
<html lang="en">
  ▶<head> … </head>
  ▼<body>
    ▼<div id="wrapper">
      ▶<h1> … </h1> == $0
        <img src="images/newsthumb-bill-gates.jpg" height="145" width="190"
```

```
element.style {
}
h1 {                                    index.html:11
   font-size: 35px;
   color: 🟩 #006b3a;
```

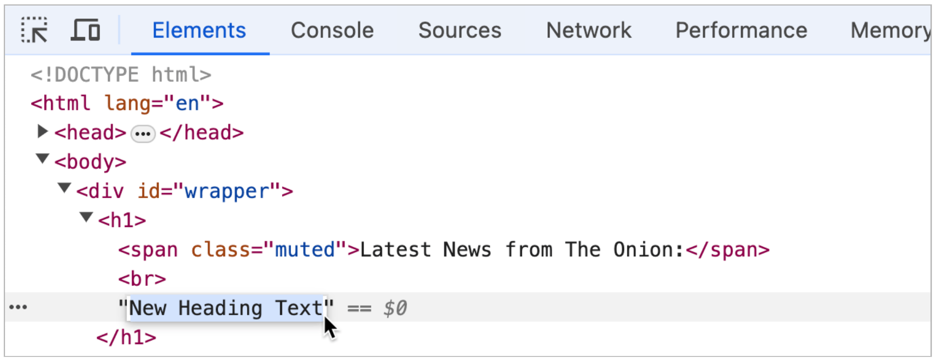
5. While you're here, also take note of the arrows that can be opened and closed to show the contents of the HTML elements. Experiment a little by opening up the **`<h1>`** tag (directly above the three **img** tags) to see the content.

6. Notice the nested **`<span>`** and **`<br>`** tags. It is helpful to know that the `<h1>` is considered the parent element and the `<span>` and `<br>` are child elements within the `<h1>`. Following this logic, the `<span>` and `<br>` elements are siblings according to the structure of the DOM tree.



```
<!DOCTYPE html>
<html lang="en">
 ▶<head>…</head>
 ▼<body>
   ▼<div id="wrapper">
…    ▼<h1> == $0
        <span class="muted">Latest News from The Onion:</span>
        <br>
        " Today’s Top National Headlines "
      </h1>
      <img src="images/newsthumb-bill-gates.jpg" height="145" width="190"
```

7. Double–click on the text inside the **`<h1>`** element and type out some new text (whatever you want), as shown below:



```
<!DOCTYPE html>
<html lang="en">
 ▶<head>…</head>
 ▼<body>
   ▼<div id="wrapper">
     ▼<h1>
        <span class="muted">Latest News from The Onion:</span>
        <br>
…       "New Heading Text" == $0
      </h1>
```

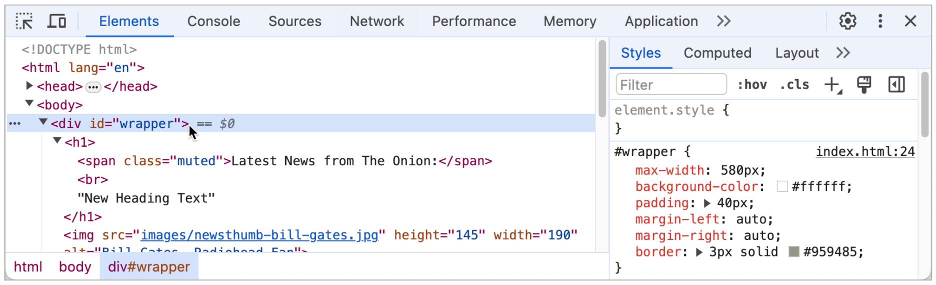
8. Hit **Return** (Mac) or **Enter** (Windows) to see the change in the browser window above the DevTools. Wow, that's cool!

   Play around for a bit and have fun. Don't worry, the changes you make in the DevTools are temporary (they are not saved into the HTML file). You need to modify the content in your actual HTML file to save the changes.

9. Reload the browser to see the original content once again.

10. Click on the **`<div id="wrapper">`** tag in the DevTools, which is just below the **`<body>`** tag.

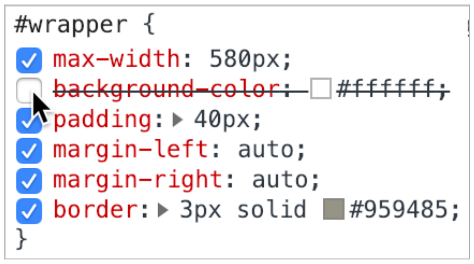
11. Look at the **Styles** panel on the right to see the CSS for **#wrapper**



```
<!DOCTYPE html>
<html lang="en">
  ▶<head> ··· </head>
  ▼<body>
···  ▼<div id="wrapper"> == $0
      ▼<h1>
         <span class="muted">Latest News from The Onion:</span>
         <br>
         "New Heading Text"
       </h1>
       <img src="images/newsthumb-bill-gates.jpg" height="145" width="190"
```

```
element.style {
}
#wrapper {                                  index.html:24
   max-width: 580px;
   background-color: □#ffffff;
   padding: ▶ 40px;
   margin-left: auto;
   margin-right: auto;
   border: ▶ 3px solid ■#959485;
}
```

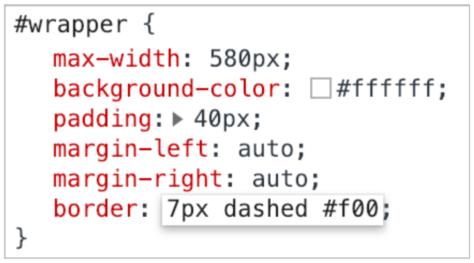
12. Hover over the properties for the **#wrapper** rule and notice that **checkboxes** appear next to each property. Try unchecking and checking the properties to see how you can temporarily disable and re-enable the property in the browser. When a property is disabled, it will appear with a strike-through:



```
#wrapper {
✓ max-width: 580px;
☐ ~~background-color: □#ffffff;~~
✓ padding: ▶ 40px;
✓ margin-left: auto;
✓ margin-right: auto;
✓ border: ▶ 3px solid ■#959485;
}
```

13. Find the **border** property and click on the value that's there (3px solid #959485).

14. Type in **7px dashed #f00** to see how the browser displays the change:



```
#wrapper {
   max-width: 580px;
   background-color: □#ffffff;
   padding: ▶ 40px;
   margin-left: auto;
   margin-right: auto;
   border: 7px dashed #f00;
}
```

NOTE: Other possible values for border style are **dotted**, **double**, **groove**, **ridge**, **inset**, and **outset**.

Hexadecimal Shorthand

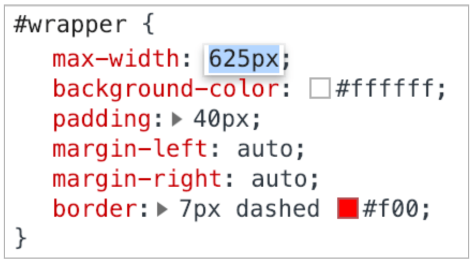
When a CSS color hex value contains **3 identical pairs**, it can be written as a 3-digit shorthand **`#fff`** (instead of the 6-digit **`#ffffff`**). Shorthand can only be used for colors with **3 identical pairs** of characters. Instead of **`#33cc66`** you could write **`#3c6`**. Colors such as **`#0075a4`** cannot be shortened.

15. Play around for a bit and have fun. Remember, the changes you make to the code in the DevTools are temporary. To save the changes you would need to remember the new values and then code them into your CSS file in your code editor.

16. Find the **width** property and click on the value to highlight it.

17. As you do the following, watch how the wrapper div's width is updated live in the browser window:

    - Hit the **Up Arrow** key on your keyboard to increase the value **1** pixel at a time.
    - Hit the **Down Arrow** key to decrease the value.
    - Hold **Shift** while hitting the Arrow keys to change the value **10** pixels at a time.



```
#wrapper {
   max-width: 625px;
   background-color: #ffffff;
   padding: ▸ 40px;
   margin-left: auto;
   margin-right: auto;
   border: ▸ 7px dashed #f00;
}
```

18. Reload the browser to return to the originally assigned values.

    We don't want to actually modify this element's width or border, but this gives you a sense of how a browser's DevTools can help you understand and figure out your CSS by providing immediate visual feedback on your changes.

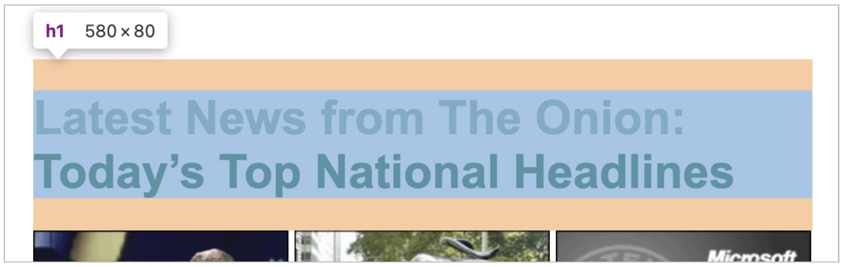
    Let's put what we've learned to some use in fine-tuning this particular page.

Fine-Tuning Margins for the Main Heading

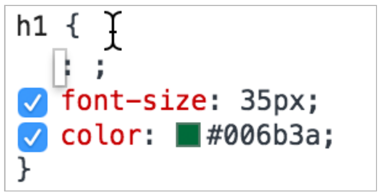
1. Look at the main heading of the page (Latest News from the Onion: Today's Top National Headlines) and notice that the space around the heading is uneven. There's way more space above the heading than on the left. Let's balance out this space.

2. On the left side of the DevTools, find the **`<h1>`** tag.

3. Hover over the **`<h1>`** tag and, as you do so, look at the way it is highlighted in the browser window above. You should see an orange-colored highlight above and below the header. The orange color indicates this element has margin:

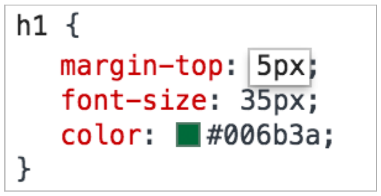


4. Click on the **<h1>** tag in the DevTools to select it. You will now see the **Styles** associated with the **h1** on the right side of the DevTools.
5. In the **Styles** panel of the DevTools, notice that we did not add any margin to **h1** to the CSS in the style tag embedded into **index.html** but, just below that rule, you'll see rules added by the **user agent stylesheet**. This is styling the browser does by default. All browsers add margin to the top and bottom of headings, paragraphs, and lists. Let's override this.
6. In the **Styles** section of DevTools, to edit our rule for **h1**, click once inside the opening curly brace:



```
h1 {
    : ;
✓ font-size: 35px;
✓ color: 🟩#006b3a;
}
```

7. Type **margin-top** and hit **Tab**. Then type **5px**



```
h1 {
    margin-top: 5px;
    font-size: 35px;
    color: 🟩#006b3a;
}
```

8. Use the **Up Arrow** and **Down Arrow** keys on your keyboard to modify the margin. Notice that the space looks more even when **margin-top** is set to **0px**. Let's add this CSS to our actual code to lock it in.
9. You should still have **index.html** open in your code editor. If not, open it now.
10. Find the **h1** rule (near the top of the **style** tag) and add the following code shown below in bold:

```
h1 {
   margin-top: 0;
   font-size: 35px;
   color: #006b3a;
}
```

NOTE: In CSS, you almost always specify a unit of measurement (such as px) for a numeric value. Because zero is always zero, regardless of the unit of measurement, you do not need to specify px when coding 0 (zero).

11. Save the file.
12. Return to the browser and reload **index.html** to see the final change to the code.

Checking for Errors: Validating Your Code

Validating your code is a way to test for coding mistakes such as unclosed HTML elements, typos, etc. If you have a problem with some code and can't find the mistake, running the code through a validator may help you to find it. Even if your page seems to be working properly, validating it may catch unseen errors. We'll use an online validation tool.

1. Open a browser and go to: **validator.w3.org**
2. Click the **Validate by File Upload** tab.
3. Click the **Choose File** or **Browse** button.
   - Navigate to **Class Files > Web Dev Class > News Website**.
   - Double–click on **index.html** to select it.
4. Click the **Check** button.

   It should say **Document checking completed. No errors or warnings to show.**

Noble Desktop Training

Learn live online or in person in NYC

Front-End Web Development	Motion Graphics & Video Editing
Full-Stack Web Development	Adobe Premiere Pro
JavaScript	Adobe After Effects
Python	InDesign, Illustrator, & Photoshop
Software Engineering	Web, UX, & UI Design
Data Science & Data Analytics	Figma
SQL	Digital & Social Media Marketing
WordPress	and much more...

noble desktop.com

HTML vs. XHTML Syntax

HTML can be written using HTML or XHTML syntax. There are only minor stylistic differences between the two. We prefer HTML syntax, which is what we use in this book. It's even a bit less typing.

If you prefer to code using XHTML syntax, here are some things to keep in mind:

- You must specify a document type at the top of the file.
- All tags must be lowercase.
- All attributes must have quotes.
- All tags must have a close tag `<p>this is a paragraph</p>`
- If tags do not have a close tag, such as a `<br>` or `<img>` tag, they must be written as self-ending: `<br />` or `<img src="image.gif" />`