File Storage: File Management and HTML File Paths

A website is a collection of multiple files— HTML documents, style sheets, image files, audio or video files, etc. As such, when constructing a website, it is crucial to organize these files into a structured file system on your local computer. This ensures that the files will be able to relate to each other so that your content will render correctly. It is important to review this *before* uploading your files to a server. This assignment outlines some key issues to consider when setting up a file structure for your website.

File Management: Storing Files on Your Local Computer

When you are first building out content for a website, it is natural to store your content locally on your computer. At this stage in the process, it is critical that all files related to your website be stored in this single folder. This folder can live anywhere you choose on your local computer — your desktop, or in another location. What is key is that this folder is placed somewhere where it can be easily located.

Naming Files and Folders

What you choose to name this folder is not important. Feel free to choose anything that makes sense or is meaningful to you. What *is* important is that the name of this folder (and all other folders and files related to your website) be completely in lowercase letters with no spaces. Spaces between words in a file name should be replaced with hyphens (-). For example:

```
file-name.html
```

The reason spaces should be replaced with hyphens is twofold:

1. Web servers and many computers are case-sensitive.

2. Different browsers, programming languages, and web servers handle spaces in the same manner. For example, some systems may treat spaces in a file name as two separate files, or replace spaces with "%20" (the character code for space) which may break links within your website.

Naming Files and Folders

Use hyphens to separate words in file or folder (directory) names: my-file.html vs. my_file.html.

File Structure

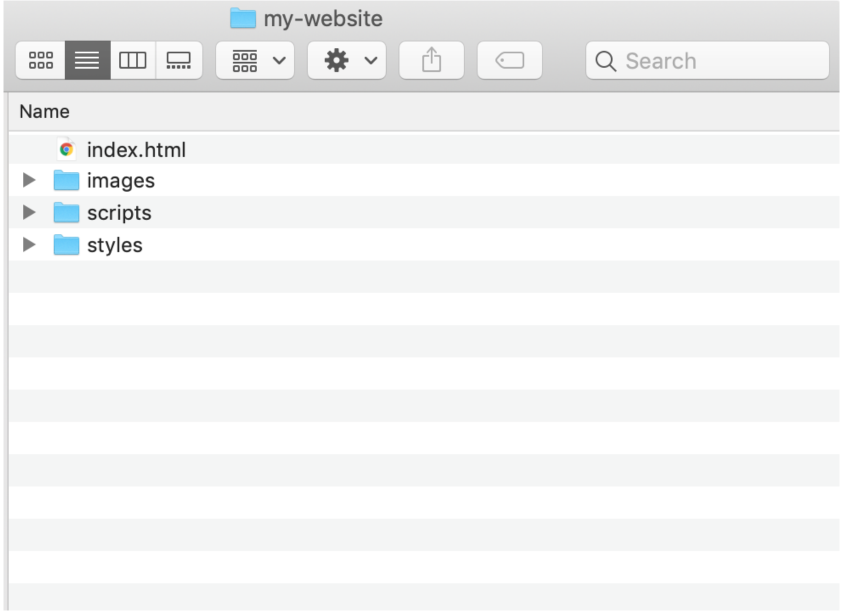
Now that you have identified the main folder where you will store all of your files, the next step is to identify the file structure and subfolders for your website within this folder. For any website project, the most commonly found file and folders within the main folder are listed in the table below:

Common Website Structure	
index.html	The **index.html** file is generally used to contain all of the content (text and images) related to the homepage of a website (the first page visitors see when they visit your site).
images folder	The **images folder** is used to contain all the image files for a website.

styles folder	The **styles folder** is used to contain all the CSS code used to style the content of a website, for example, adding background colors or different styles to text. (CSS is not covered in this course.)
scripts folder	The **scripts folder** contains all the JavaScript code used to add interactive functionality to a website, for example buttons that perform a function when clicked. (JavaScript is not covered in this course.)

Each of these folders (also called directories) may also contain other files or sub-directories, nested inside of them. The organization structure may contain as many levels as desired. Where this becomes important is discussed in the next section on file paths.

The image below illustrates the file structure for a web project named my-website.



File names and Windows Computers

If you are using a Windows computer, you may have trouble seeing file names. This is because Windows has a default feature named 'Hide extensions for known file types.'

The feature can be turned off by going to Windows Explorer, selecting Folder options, unchecking the Hide extensions for known file type checkbox, and then clicking OK.

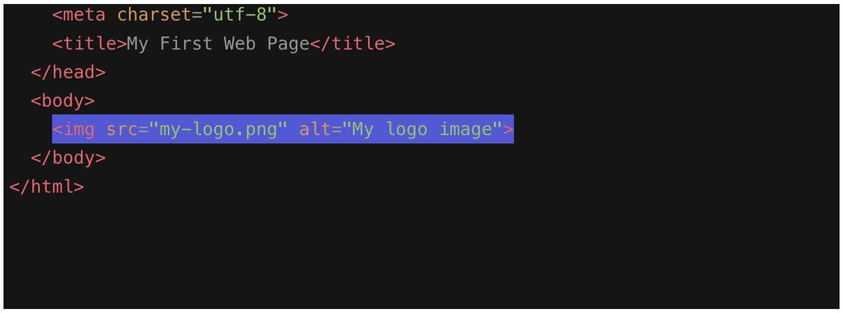
HTML File Paths

Once you have identified and created the file structure for your website, the next step is to create a path between files so that they can talk to one another. In order for a website to render properly, it is important for a file to know where another file is located. Let's take a look at an example to illustrate what this looks like.

In the example below, let's examine the file path for an image that we want to appear on a webpage. To do this, we need to identify where the image is located within the HTML code, so the browser will know where to find it.

For example, let's say we wanted to place an image file named "my-logo.png" on a webpage. In the HTML code below, the line <img src="my-logo.png" alt="My logo image"> is the HTML code that will insert this image into the page. The file path for the image in this example is for an image file that is located in the same folder (or directory) as the HTML document. (Don't focus on the other various structures in the HTML code below — each will be discussed in more detail in later assignments.)

```
<!DOCTYPE html>
<html>
  <head>
```



```
    <meta charset="utf-8">
    <title>My First Web Page</title>
  </head>
  <body>
    <img src="my-logo.png" alt="My logo image">
  </body>
</html>
```

File path links can also point to files that are located in folders or directories other than where the invoking HTML file resides. The table below lists general rules for creating file paths:

File Paths: General Rules		
File Location	**Path**	**Example**
Same Folder/Directory	Use just the filename	my-logo.png
Folder/Directory One Level Above	For a file that is in a file one level above, place two dots and a forward slash in front of the file name.	../my-logo.png
Folder/Directory One Level Below	For a file that is in a file one level below, place the sub-folder/directory name in front of the file name, plus a forward slash.	subfolder/my-logo.png

Naming Files and Folders

Windows tends to use backslashes (\) for file systems, i.e., C: \windows. This distinction doesn't matter when you are using HTML on a Windows machine. When creating your website on a Windows machine, you should still use forward slashes (/) in your code.

Review Checkpoint

To test your understanding of the content presented in this assignment, please click on the Questions icon below. If you have trouble answering any of the questions presented here, you are always free to return to this or any assignment to re-read the material.

1. Which of the following is the proper file path syntax for a file located in a directory one level above?

   a. ..image.png

   Incorrect. Try again.

b. ../image.png

Correct. To create a file path for a file that is in a file one level above, place two dots and a forward slash in front of the file name.

c. /image.png

Incorrect. Try again.

d. None of the above

Incorrect. Try again.

2. Spaces between words in a file name should be separated using which of the following?

a. backslash (\)

Incorrect. Try again.

b. forward slash (/)

Incorrect. Try again.

c. hyphen (-)

Correct. Spaces between words in a file name should be replaced with hyphens (-)

d. underscore (_)

Incorrect. Try again.