You may have seen links like /public/logo.png, ./script.js, or ../styles.css before. But what do these special types of links mean? These are called file paths. There are three key syntaxes to know. First is the slash, which can be a backslash (\) or forward slash (/) depending on your operating system. The second is the single dot (.). And finally, we have the double dot (..).

The slash is known as the "path separator". It is used to indicate a break in the text between folder or file names. This is how your computer knows that naomis-files/ points to a directory of that same name, while naomis/files/ points to a files directory in the naomis directory.

A single dot points to the current directory, and two dots point to the parent directory. You will typically see a single dot used to ensure that the path is recognized as a relative path. Remember that you learned in a previous lecture about relative versus absolute paths before.

Double dots, however, are much more common to access files outside of the current working directory.

For example, given this file tree:

my-app/

├─ public/

│ ├─ favicon.ico

│ ├─ index.html

├─ src/

│ ├─ index.css

│ ├─ index.js

If your public/index.html file needed to load the favicon.ico file, you would use a relative path with a single dot to access the current directory: ./favicon.ico. If your public/index.html file needed to load the index.css file, you would use a relative path with double dots to navigate to the parent my-app directory first, then to the src directory, and finally to your file: ../src/index.css.