10. A Guide to Paths

Monday, January 29, 2018 8:20 PM

You'll soon make a website that displays an image that is stored locally on your computer. In order to display a local image, you need to be able to write a **path**.

tl;dr If there is a file called `index.html` in a directory and there is another directory called `example/` in the same directory, you can access any files in `example/` from `index.html` with the URL (path) `example/filename.html`, e.g. `<a href="example/filename.html">Example Path</a>`.

Paths

A path is a way of describing where a file is stored.

Think of it like this:

Anyone in the world can use the address 1600 Pennsylvania Ave NW, Washington DC, USA 20006 to find the White House. A street address is an absolute path to a location.

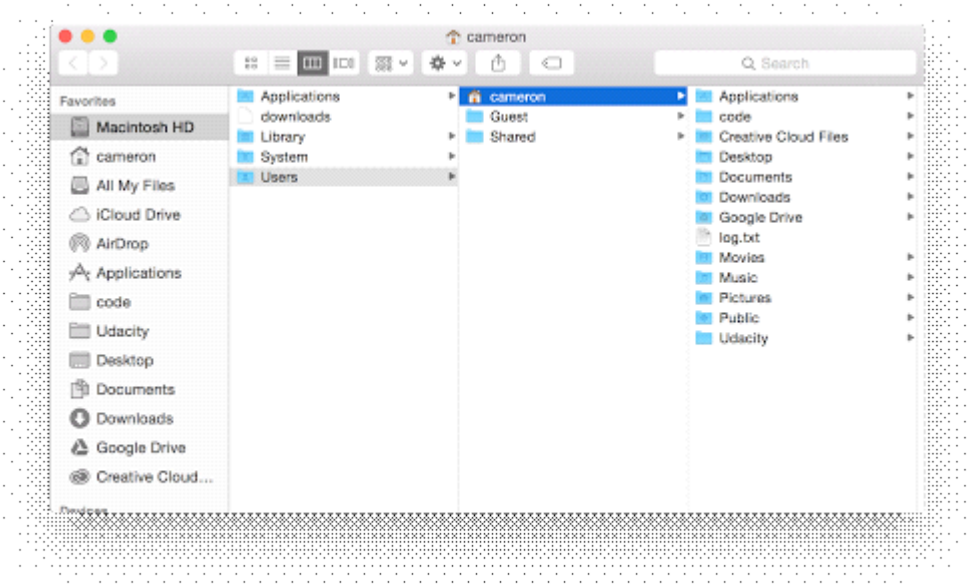
But, if you were at the Eisenhower Executive Office, you could also use the phrase "next door" to find the White House. "Next door" is a relative path because it depends on your current location.

There are essentially two domains for paths that you'll need to consider as a web developer: paths to find files on your computer, **local** files, and paths to find files on other computers, **external** files.

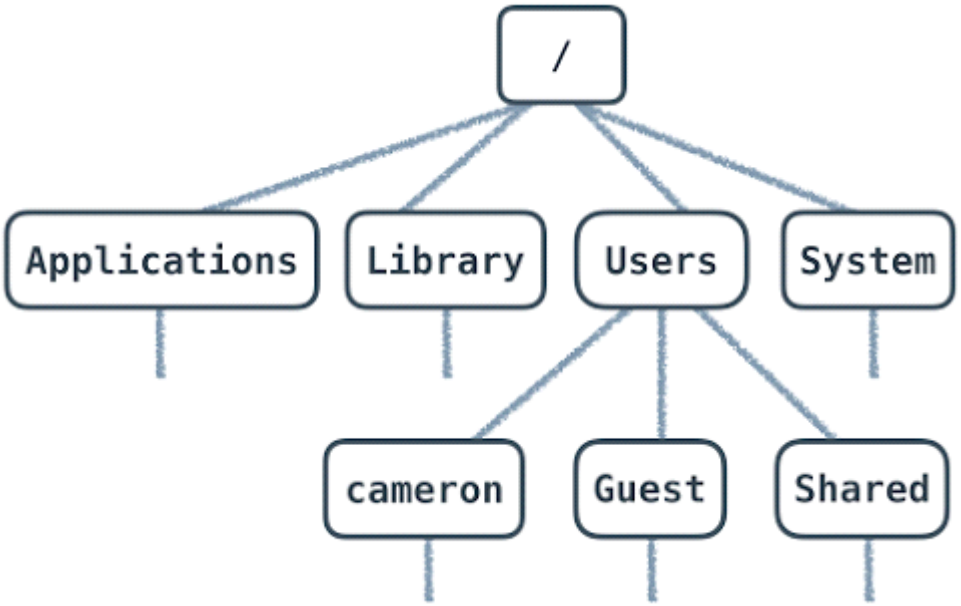
Local Paths

Computers have folders (also called "directories"). Operating systems like Windows, Mac and Linux organize *all* of your files into a tree of directories called a **file system**. There's a top-most directory, often called the **root**, that contains all of the other directories. Within the root, there are files and directories. Within those directories are more files and more directories. And within those directories are even more files and directories, and so on.

Compare this part of the file system on my computer:



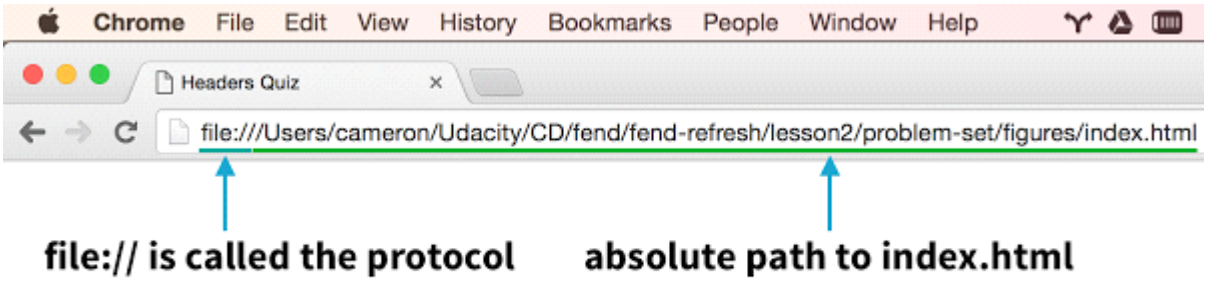
Local path directory structure screenshot
to a tree diagram showing the same directory structure:



Tree diagram of directory structure
Every file has an address, which we call the "path." An absolute path is written in relation to the computer's root directory. For instance, a file in the Documents folder on a Mac has a path that looks like this:

/Users/cameron/Documents/file.txt
file.txt is stored inside Documents/. Users/, cameron/ and Documents/ are all names of directories. Documents/ lives inside cameron/ and cameron/ lives inside Users/. Users/ is inside the root directory, which is represented by the first /. The rest of the / are used to separate directories.

When you open an HTML file in your browser, you're seeing the absolute path to the file on your computer.



Absolute Path to index.html
This URL will *only* work for you on your computer. As no one else has your file system, this URL is unique to your computer. If you want other people to be able to access it, then you need an...

External Paths

The process of loading a website from a URL like https://www.udacity.com mimics opening an HTML file that you've written and saved to your computer. Every website starts with an HTML file. It just so happens that when you want to visit a website, the HTML file that you want to open lives on a different computer. The computer responsible for giving you a website's files is called a **server**.

Pointing your browser to https://www.udacity.com sends a request to Udacity's server for the HTML file (and others) that your computer needs to load the Udacity website. You can think of udacity.com as the root path of Udacity's server (computer) that anyone can access (the reality of the situation is actually much more complicated but the general idea is true). Unlike your personal computer (for now!), Udacity's servers run software that **expose** files to the web, which means that they make them available to anyone who wants them. Servers have an **external path** that anyone can access and is the reason why the web works.

Different websites are just different collections of files. Every website is really just a server (or many servers) with an external address, which we call a URL. Servers store files and send them to computers who request them (the requesting computers are called **clients**).

There are different **protocols** for serving files, the most common of which on the web are HTTP and HTTPS. When you open a file on your own computer, you're using the file protocol. You don't need to know much more about protocols for now, but if you're interested in learning (a lot!) more about HTTP, check out Networking for Web Developers.

Relative Paths

The relative path is similar to the absolute path, but it describes how to find a path to a file from a directory that is not the root directory. Like using the phrase "next door" to tell someone how to find the White House from the Eisenhower Executive Office, a relative path takes advantage of the location of one file to describe where another file can be found.

Relative paths work the same for both local and external paths. Let's break down two examples of absolute paths to see how relative paths work.

External:

```
<a href="http://labs.udacity.com/fend/example/hello-world.html">Hello, world!</a>
```

Local:

```
<a href="/Users/cameron/Udacity/etc/labs/fend/example/hello-world.html"> Hello, world!</a>
```

href is really just a path to a file.

Both examples are links to the same file using absolute paths, **but the external example would work for anyone and only my computer can use the link in the local example.**

Pay attention to the `fend/example/hello-world.html` portion of both paths - they mean the same thing.

Imagine that you are editing `/Users/cameron/Udacity/etc/labs/fend/test.html`. `test.html` can reference `hello-world.html` by describing how to get from it's location in `fend/` to `hello-world.html`. The relative path would look like:

```
example/hello-world.html
```

This relative path takes advantage of the fact that `test.html` and `example/` are in the same directory.

But what if I'm editing a file in `/Users/cameron/Udacity/etc/labs/` and I want to write a path to `hello-world.html`? In that case, the relative path would be:

```
fend/example/hello-world.html
```

Now that I'm in `labs/`, not `fend/`, I have to include `fend/` in a relative path to `hello-world.html`.

To finish this off, let's imagine there are two files:

http://labs.udacity.com/science/sciences.html
and

http://labs.udacity.com/science/physics/relativity.html
In order to write a relative path from `sciences.html` to `relativity.html`, I only need to include the part of the path that describes how to get from `science/` to `relativity.html`:

```
<a href="physics/relativity.html">Einstein's Special Relativity</a>
```

And that's it! Now it's time to apply your new skills.

From <https://classroom.udacity.com/courses/ud304-gwg/lessons/7222405183/concepts/73276037150923>