

# COMP 1017

Day 09  
Site Navigation

So far, we've been creating  
single page websites.

But single-page sites are  
not really the norm.

So, how can we create multiple-page websites with functional navigation between pages?

# Absolute vs. Relative Pathing

The battle of the century!

So far, we've been using `<a>` to link to `external` websites.

```
<a href="#">site name</a>
```

The `href` has always been a URL to somewhere else - that is, on some other site, hosted on some other server.

These URLs are an example of what we'd call **absolute** paths.



Think of it like a street address.  
**Absolute** paths do not change.

Example:

The HP Centre at NAIT is located at 10504 Princess Elizabeth Ave NW.

However, we've also been using  
**relative** paths for other things.

Whenever we use an `<img>`, our `src` value is *relative* to where the `index.html` is located.

Our images are inside of our image folders (img), so we have to tell our browser to go there in order to get the image file it needs.

This is more like giving directions.

If we were in the HP Building on campus and you wanted to get to class, giving you an absolute address wouldn't be very helpful.

( We would already be on campus, after all. )

Instead, I'd tell you how to get there starting from where we currently are.

This is exactly how linking within the  
same framework folder  
(i.e. the same website) works.



# Linking in the Same Directory

It's all relative, really.

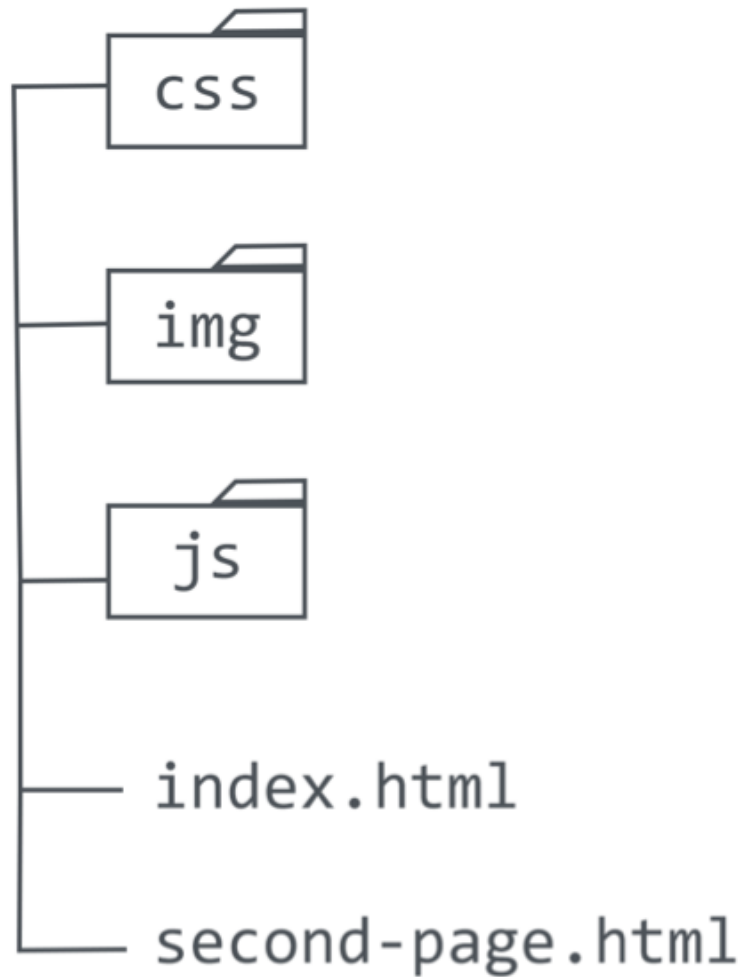
Let's say we've got a bunch of pages for our website, all in the **root** of our framework folder.

Note: When I say root, I mean **immediately inside** of the folder.

Instead of giving an **absolute** path or address, we will give a **relative** path.

So, let's say our framework folder looks like the following ...

framework



How can we give 'directions' from  
`index.html` to `second-page.html`?

```
<a href="second-page.html">Link to Second Page</a>
```

... that's literally it. If your other pages are in the root folder, you just have to point to them.



In this case, you do not have to tell your browser to go somewhere else in order to find the files as the page is in the root folder.

# Link to Pages in Subdirectories

AKA tossing a spanner into the works.

So, what happens if we keep all of our pages inside of a sub-folder?

In that case, if we want to link to the page, we have to tell our browser to go into the **sub-folder** and find that HTML file.

framework



Additional pages  
(name.html)

```
<a href="pages/second-page.html">Link to Second Page</a>
```

... but what if we're linking from a page  
**inside** of the sub-folder and want to link  
back to the **index.html**?

We have to tell our browser to  
go back (up one level, to the root)  
and find the index.html



```
<a href="../index.html">Return to Index</a>
```

Go up one folder from where I am and then go from there.

(../ Up a directory) (../.. Up 2 directories)