INFO/CS 1300: Lab 2 (9/1)

By Grant Storey

Due at the end of your lab section. Show the TA your work to get your participation credit.

Overview

In this lab, we'll cover linking multiple pages, absolute and relative paths, and URLS. We'll also look at organizing information with LATCH and card sorting.

What You Need

Bring your **laptop, paper, pen/pencil, and scissors** to your lab section. You may also want to **print** out this lab.

Part 1: Relative Paths vs. Absolute URLs

Create a New Webpage

Create a folder called **lab2-<NetID>** (example: lab2-kjh235).

Create a new **index.html** with the following contents:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Relative vs. Absolute Paths</title>
</head>

<body>
</body>

</html>
```

Insert a Hotlinked Image

At the moment, this page is a little bland! So let's review how to add an image to a webpage by creating a page that showcases my favorite animal. I found a great picture here: https://cdn.pixabay.com/photo/2017/07/07/15/54/giraffe-2481840_640.jpg

To add this image to our webpage, add the following HTML snippet to the **body** of **index.html**:

```
<!-- Image Source: https://pixabay.com/en/giraffe-zoo-mammal-animal-head-2481840/ -->
<!-- Image License: CC0 Creative Commons -->
<img alt="giraffe" src="https://cdn.pixabay.com/photo/2017/07/07/15/54/giraffe-2481840_640.jpg"/>
```

Recall that HTML is all about structure. We indent our code to communicate this structure to others. Make sure you properly indent this code snippet after you insert it. **Tip**: Bring up the command palette and search for **Auto Indent**.

View your webpage with the preview in Atom and/or in Firefox or Chrome.

It works! Even though you didn't download the image and put it in the same folder as your index.html, you can still see it. This makes it really easy to include images from all over the web in your personal webpages!

You should **never** do this. This practice is called **hotlinking**. There are several problems with hotlinking: 1) someone else has this image on their web server, which means they have to pay every time someone downloads this file, even when it's on your webpage! That's not fair! 2) If the hotlinked image was ever deleted from the original server, your webpage would also loose access to the image and then you would have a broken image on your webpage!

TL;DR Do not hotlink images!

Insert an Image using a Relative Path

Open the giraffe image in your web browser and **download** it. Typically you just need to right click on the image and choose **Save Image As...**. If you are given a choice, download the image to your lab2 folder. If not, move the downloaded image to your lab2 folder.

We could just replace the hotlink image by changing the code to this:

```
<!-- Image Source: https://pixabay.com/en/giraffe-zoo-mammal-animal-head-2481840/ -->
<!-- Image License: CC0 Creative Commons -->
<img alt="giraffe" src="giraffe-2481840_640.jpg"/>
```

However, **giraffe-2481840_640.jpg** just isn't a great filename. It's a little clunky. Let's make it something easier for use to deal with. **Rename** the image to **giraffe-face.jpg**.

Now remove the hotlink URL and replace it with this **relative path**:

```
<!-- Image Source: https://pixabay.com/en/giraffe-zoo-mammal-animal-head-2481840/ -->
<!-- Image License: CC0 Creative Commons -->
<img alt="giraffe" src="giraffe-face.jpg"/>
```

This is a relative path because we are telling the web browser where to giraffe-face.jpg **relative to** index.html. In this case, they sit in the same folder, so we can just give the filename of the JPG image.

Note about Naming Conventions

You might be wondering why we chose giraffe-face.jpg with a dash rather than "giraffe face.jpg" with a space. If you include a space in the filename, it is easy for computer to think that you are referring to two different files; using our example, these two files would be "giraffe" and "face.jpg". This will lead to all sorts of sneaky errors and issues, so **don't put spaces in your filenames or folder names!**

There are all sorts of alternatives to using spaces, like snake case, with underscores (_) instead of spaces (`snake_case_example.jpg`), camel case, with capital letters denoting new words (`camelCaseExample.jpg`), or kebab case, with dashes instead of spaces (`kebab-case-example.jpg`).

Organizing your Website's Files

So you have a picture of a giraffe and you know how to name files. But you will notice that the image is sitting in the same area as your HTML file in the lab2 folder. This is fine when there is one image, but what if you had hundreds of images of giraffes? It would get very cluttered.

In order to address this problem, we want to organize your website's files into folders. Create a new folder inside of your lab2 folder called **images**. Use your file manager to move giraffe-face.jpg to the images folder.

Moving the image has now created a broken link in your index.html. Just try it… view your page in the browser! Your image no longer works!

In order to fix this, you will need to properly specify the location of the image **relative** to the location of index.html (this is called the "relative path"). In this case, that means adding the folder where the image sits to the image's filename. You can add folders to your filename by prefixing the folder name to the filename. When adding the folder prefix, separate the two with a forward slash (/). For example: **folder-1/folder-2/folder-3/image.jpg**.

Ask for help if you need it. We love helping you!

Part 2: Multi-Page Websites

Creating a Second Webpage

Create another HTML file in the lab2 folder called **about.html**.

Take a few minutes to create the HTML for about.html. You should quickly describe something about giraffes. We won't provide a code snippet here; it's time for you to venture out on your own in terms of creating this. **Hint:** you can always start using what is in your index.html file as a base.

Linking Your Page Together

You now have two pages: **index.html** and **about.html**. However, your pages are totally disconnected; there is no way to navigate from one to the other. To fix this, we are going to link up the two pages using a navigation menu. Navigation is usually done with the **unordered list** (ul) and **list item** (li) tags. Later, when we learn CSS, we can style these tags to work like menus!

In order to create a navigation menu, copy the following code snippet into both index.html and about.html:

```
<nav>
  <ul>
    <li><a href="">Home</a></li>
    <li><a href="">About</a></li>
  </ul>
</nav>
```

The links (**a**) currently go nowhere, that is, the **href** (hypertext reference) attribute is set to empty. Use your knowledge of relative paths to fill in the proper locations of your main page and your about page.

You should now be able to navigate between your two pages using the navigation menu. You now also have the beginnings of a multi-page website!

When you are finished, show the TA your work!

Part 3: Organizing your Website's Pages

For Project 1, Milestone 2 you will have a multi-page website. For a multi-page site you will need to think more about how these pages will be organized. Your users will need to be able to quickly find what they are looking for when navigating it. To help us start thinking about how to organize the content on your site, check out this video on LATCH: https://www.youtube.com/watch?v=Tgi1JQGHENI.

Introduction to Card Sorting

A classic technique for organizing information is called **Card Sorting**. This is a great way to figure out how your target audience might order things in a way that makes sense to them. The basic idea is to create cards for all the content you have for your website and then ask some of your potential users to sort them into groupings that make sense to them. After creating the groupings, give each group a name. You can use this grouping information to set up the navigation structure for your page.

Let's try card sorting! Find a group of 3-4 people. Take the last page of this write-up and print it. Then cut apart the cards. Now spend some time discussing the different ways you could organize the cards. Feel free to move them around. Once you have your groupings, give them each a name and show it to a TA.

Card Sorting Your Project Website

Having practiced with card sorting, it's time to apply it to your project. Create some cards for your Project 1 website. The cards should have the content you plan to have on the site. You aren't committing to these topics, but the more work you put into this exercise the more valuable it will be.

Find a partner and exchange cards, so you sort their website's cards and they sort your cards. Debrief with each other on the categories you came up with. Repeat this 3 times total so that you get 3 different opinions of potential categories for your websites. For each card sort, take a picture so that you remember how your partner organized your cards. This will be really useful for you when writing your rationale for Project 1, Milestone 2.

Credit for Lab 2

When you show a TA your lab2 webpage and also your produce card sort, you'll get credit for participating in lab 2. You do not need to submit anything to CMS.

Banana	**Grape**	**Zucchini**
Peach	**Carrot**	**Broccoli**
Apple	**Bell Pepper**	**Strawberry**
Squash	**Raspberry**	**Avocado**
Green Bean	**Tomato**	**Potato**
Beet	**Cherry**	**Cauliflower**
Onion	**Blueberry**	**Eggplant**