

Exploring Web Design

Lab 3: Internal, Relative Folders & Resource Links

There are many ways to create a links, and many resources we can link to, such as:

- Links to internal and external sources
- Navigations
- Webpages
- Other websites
- Images
- Videos
- Maps

In this week's lab we will create such links to other web pages. We can use a lab 2 as a template.

Complete the following steps:

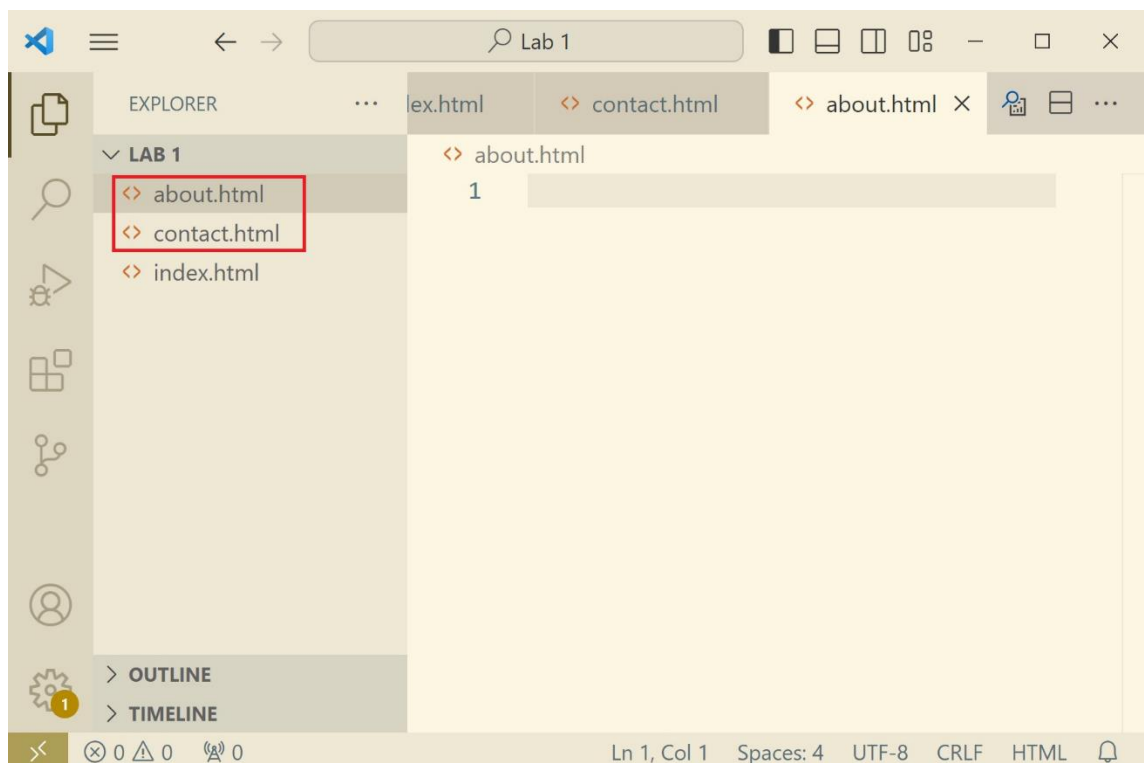
Step 1: Create two new HTML files inside your project folder named:

contact.html and

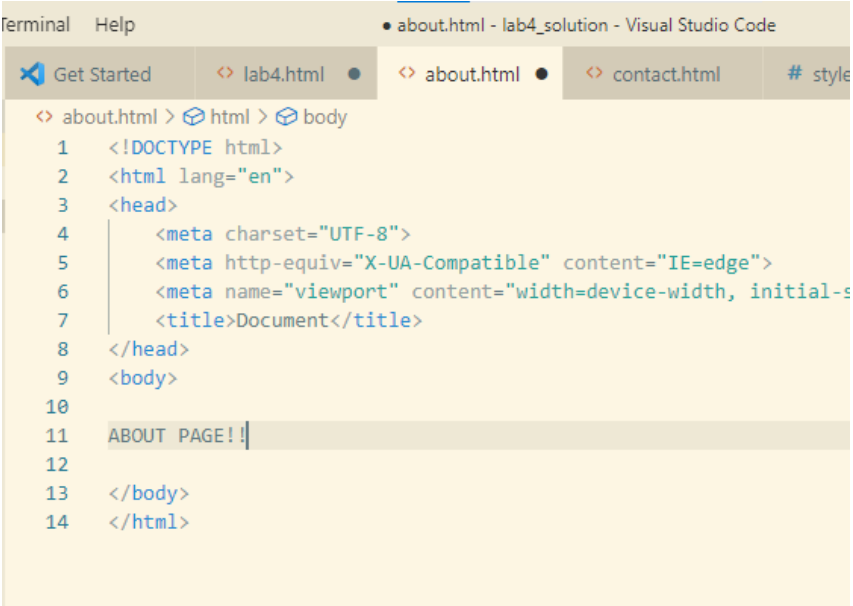
about.html

[See here for how to create files in VS Code.](#)

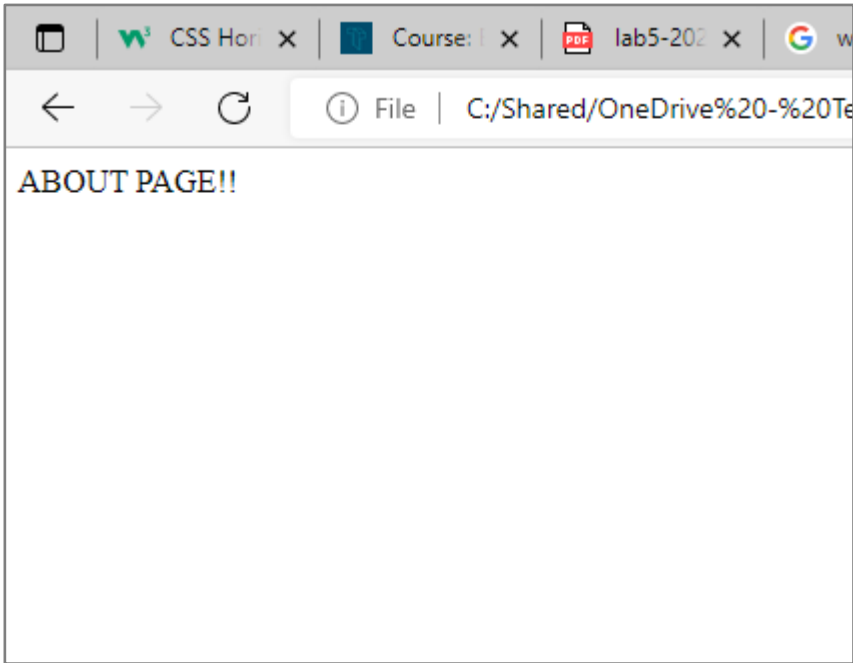
Your project folder in Visual Studio should now contain two new files:



Step 2: Add some text to the *about.html* page and see if the *about* page works in the browser



```
Terminal  Help  • about.html - lab4_solution - Visual Studio Code
Get Started  lab4.html  about.html  contact.html  # style
about.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-s
7      <title>Document</title>
8  </head>
9  <body>
10
11  ABOUT PAGE!!
12
13  </body>
14  </html>
```



Step 3: Repeat step 2 for the *contact.html* page

Step 4: Creating Relative Links

We’ve seen already how to add **Absolute Links**, which are links to exact locations, such as the below web addresses:

```
<a href="https://www.google.com">Click me to go to Google</a>
<a href="http://www.itb.ie">Click me to go to TUD
Blanchardstown</a>
```

This time we will create **Relative Links** to resources inside your project. Relative links are links with an address that is relative to the resource (e.g. the webpage) that contains the link.

We will “nest” the three hyperlink `<a>` tags inside a `<nav>` tag. A `<nav>` tag is where we put navigation links.

4.1 Recreate the following HTML in your index.html page under the opening <body> tag.

```
<nav>

  <a href="index.html">Home</a> &nbsp;
  <a href="about.html">About</a> &nbsp;
  <a href="contact.html">Contact</a> &nbsp;

</nav>

<p>
  INDEX PAGE!!
</p>
```

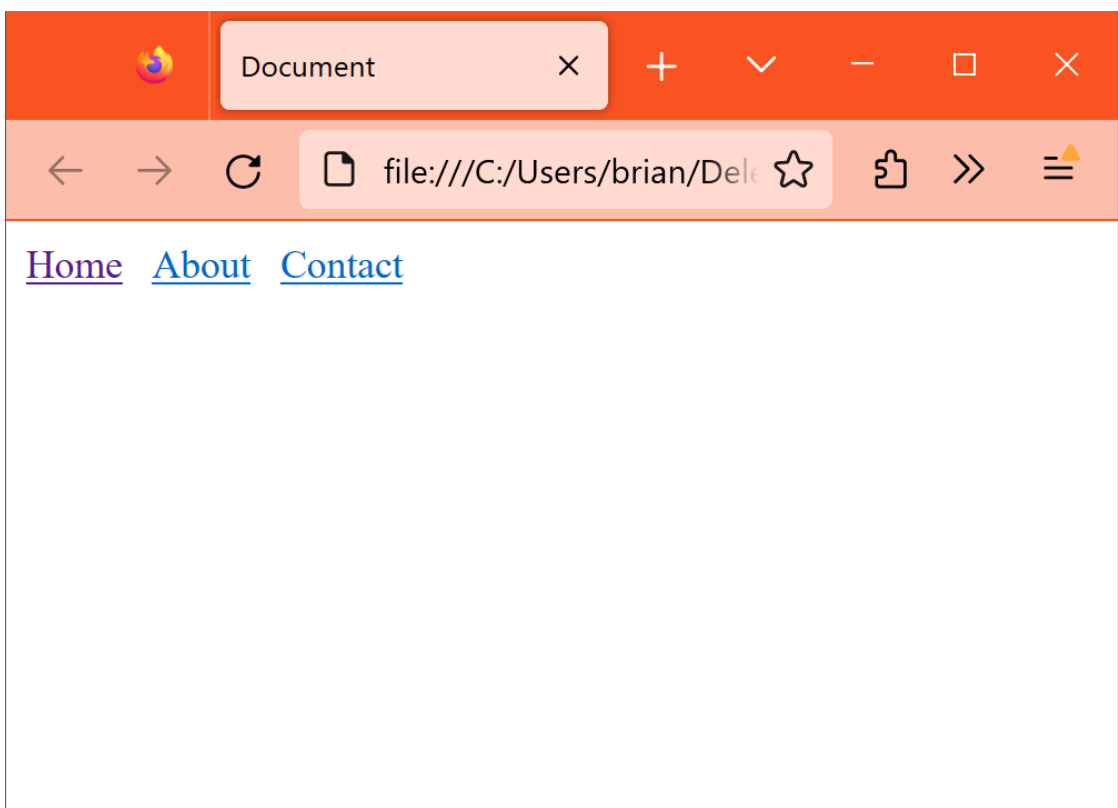
** ** adds an extra space. Why not just add another space using spacebar? HTML only recognises one space generated from spacebar. All other spaces are ignored.

Entire *index.html* page:



Your page should now contain three links.

4.2 Open the index.html file in your browser and it should look like the below.

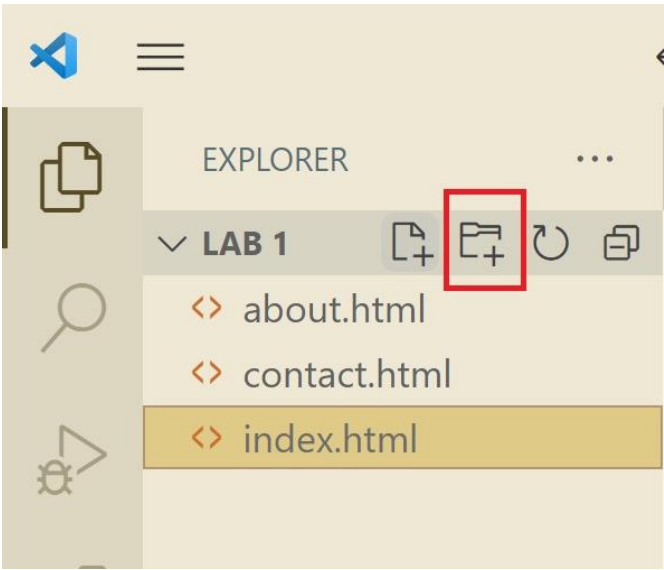


4.3 Click each link to make sure the corresponding page loads.

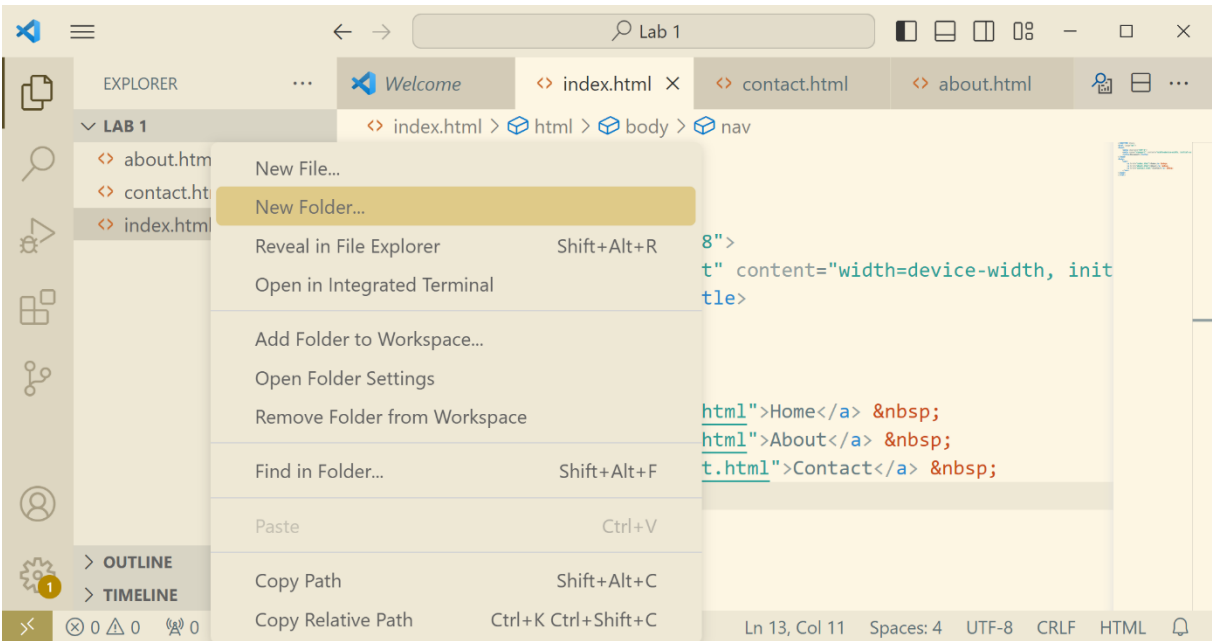
Step 5 – Putting the webpages in their own folder

Sometimes for large projects, we may wish to separate resources to make it easier to navigate our project files.

5.1 Create a new folder by using the new folder icon:

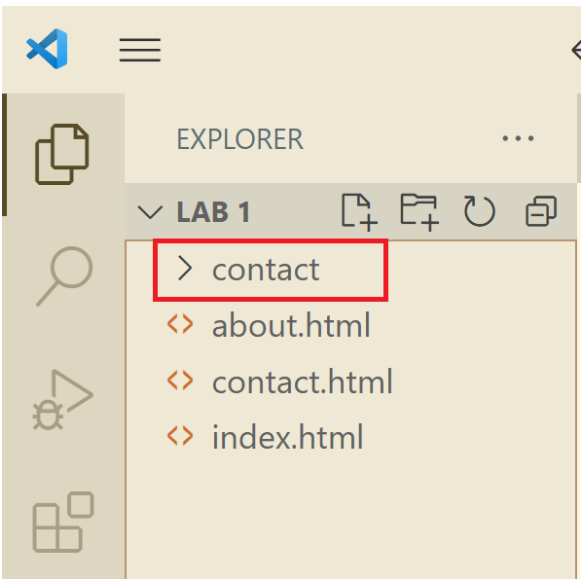


Or by right clicking the folder navigation area and selecting, “New Folder”:

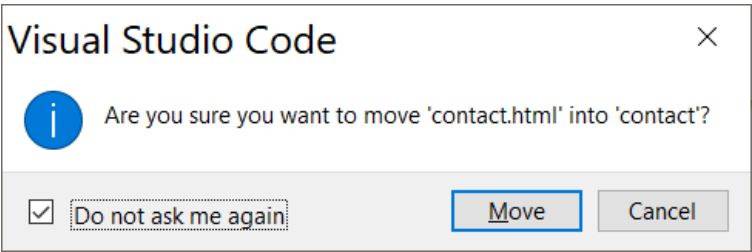


5.2 Name the first folder *contact*

You can see from the image that the contact folder has been created:

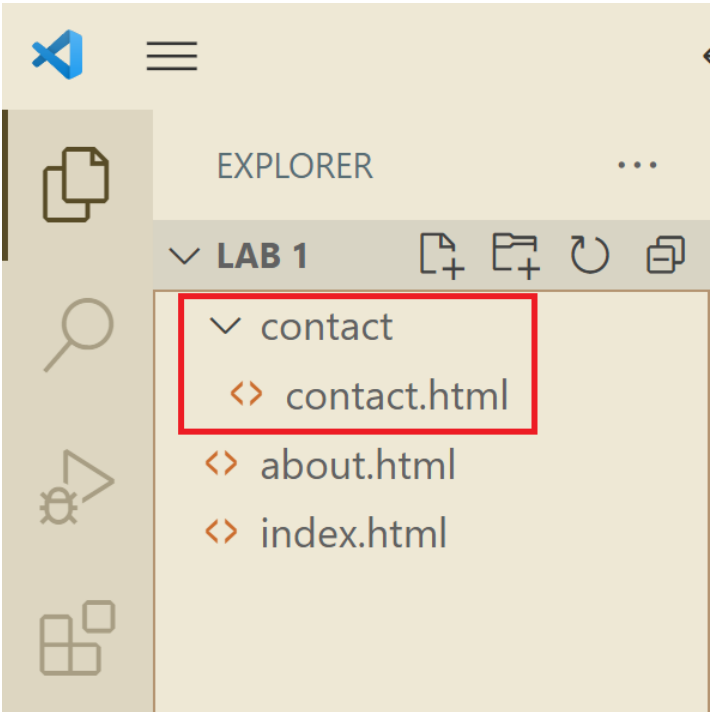


5.3: Drag *contact.html* into the newly created *contact* folder and then select Move on the below pop-up window.



The warning appears because moving resources like web pages or images means any links to them will no longer work, i.e., the links will become ‘broken’. That’s OK as our next step will be to fix the broken links.

Your *contact.html* page should now appear in the contact folder, as shown below.



Step 6: Complete step 5 again for the about page, creating a new *about* folder

Step 7: In *index.html*, change the HTML within the nav tag to reflect the location changes made to *about.html* and *contact.html*

```
<nav>
  <a href="index.html">Home</a> &nbsp;
  <a href="about/about.html">About</a> &nbsp;
  <a href="contact/contact.html">Contact</a> &nbsp;
</nav>
```

Notice the *about* page and *contact* page’s HTML location have changed to indicate their new folder locations with the syntax *folder_name/file_name.extension* - for example, *about/about.html*

7.1: Test links in the browser

Test to see if the contact page link still works in the browser. Remember to add some text in between the <body> tags of the *contact* and *about* html pages so that you can tell they are working correctly (otherwise you would be looking at blank pages in your browser.)

Step 8: Adding navigation to the about and contact pages

Notice when we leave the *index* page, to visit the *about* or *contact* page, we must click the browser's back arrow to return to the *index* page.

Let's create a nav bar for the about and contact page so we don't have to rely on the browser's back and forward buttons.

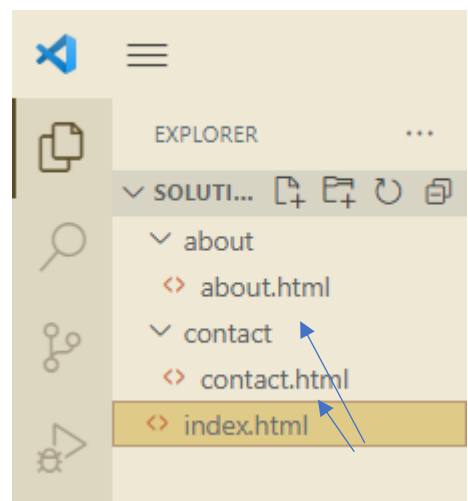
8.1 Copy the below from *index* page to the *about* page

```
<nav>
  <a href="index.html">Home</a> &nbsp;
  <a href="about/about.html">About</a> &nbsp;
  <a href="contact/contact.html">Contact</a> &nbsp;
</nav>
```

8.2 Open the about page in the browser and see if the links to the index and contact pages work.

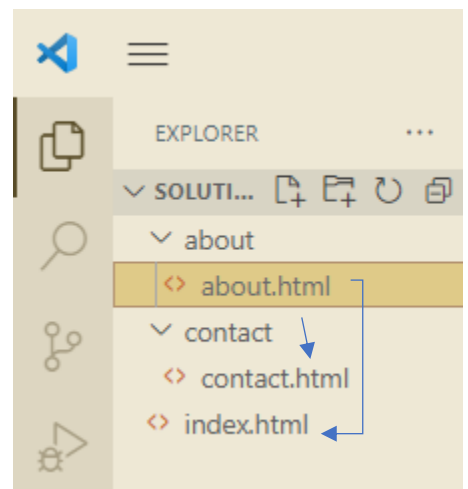
None of the links will work. Why is this? Remember links are relative to current path or resource. In the above code snippet, we are coding on the index page. This means we were creating links from the index page to other resources (i.e., the *about* and *contact* pages). So, we were only concerned with how we get from the index page (current path / resource) to the about and contact pages.

The links to the about and contact pages were **relative** to the index page.



However, now we are in the *about* page, so the direction is from the *about* page to the *index* and *contact* pages. So, we need to make the links to the index and contact pages from the about page.

When we navigated from the *index.html* to the *about* and *contact* pages we were going into folders from the main 'root' folder / directory. Now we are going out of the about folder to get to the *index* and *contact* pages. This is just like in Windows File Explorer – to go from one folder to another folder outside the current folder, we need to first go back.



To go back in HTML, we write two dots before the link ..

8.3 In your about.html page, change the HTML within the nav tags to:

```
<nav>
<
  <a href="../index.html">Home</a> &nbsp;
  <a href="about.html">About</a> &nbsp;
  <a href="../contact/contact.html">Contact</a> &nbsp;
</nav>
About Page!!!
```

Step 9: Using the same method we just used for the about page, create working navigation links for the contact page.

Step 10: Creating an image folder

The images folder will be accessed just like the about and contact page. Using the methods described in this lab:

10.1 Create a new folder called *images* in the main root folder

10.2. Save an image from the web, and instead of saving it directly into to your lab 3 folder, save the image into the folder called *images* within your lab 3 folder.

10.3. Just like in lab 1, create an image tag on index.html, and using the same syntax as shown in steps 5, 6 and 7, make the src attribute equal to *images/yourImageName.fileExtention*

For example: *images/dog.jpg*

10.4. Test to see that your image loads in the browser