Name: Kevin Tran PID: A16809193 (page 1)

https://www.google.com/search?q=cse15l

HTTP(S) URL: Where a single web resource is located, potentially including extra "arguments" (the query after ?)

Domain: The part of the url after the https:// and before the first slash, usually ending with .com, .edu, .net, .org, .gov, and so on, or country codes, like .uk, .jp, etc. This usually corresponds to some organization or person, like www.google.com above

Path (part): The part of the URL after the domain and before any ?. /search in the example above

Query: The part of the URL after the first ? and before the anchor. ?q=cse151 in the example above.

Anchor (or Fragment): The part of the URL after the #. Not in the example above; usually page-specific data (like where to scroll)

URLs are a *lot* like paths! Github Pages gives you your own domain (it's a subdomain of github.io) at <username>.github.io What are the domains, paths, queries, and fragments of these URLs?

https://edstem.org/us/courses/41619/discussion/3541501

https://ucsd-cse151-f23.github.io/week/week1/#week1-lab-report

https://map.concept3d.com/?id=1005#!m/576556

When you write a link in Markdown, just like in the filesystem, it can be **relative** or an **absolute** URL (kind of like an **absolute path**). Say you have a Github Pages site with the following structure (and say the username for the account is cs151-student):

cs151-lab-reports

- l- index.md
- |- lab-report-1.md
- |- profile-pic.jpg
- |- more-images/
 - screenshot.png

What are **two ways** to write an image link in lab-report-1.md that refer to profile-pic.jpg? What about to screenshot.png? What are some tradeoffs between them?

![sshot](more-images/screenshot.png)

![profile](profile-pic.jpg)

![sshot](https://cs15l-student.github.io/cs15l-lab-reports/

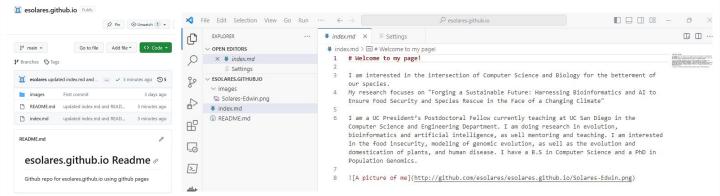
![profile](https://cs15l-student.github.io/cs15l-lab-reports/profile-pic.jpg) more-images/screenshot.png)

What are **two ways** to write a link from index.md that refers to the lab report page?

[Lab Report 1](lab-report-1.html) --> .html, not .md

[Lab Report 1](/lab-report-1.html) --> start from the domain and replace whole path

A developer is working on a Github Pages site. They are adding an image to their readme.md file. The image looks great in their Code preview, but on their Github Pages site it appears as a broken image link. Here's their code preview and their Github repository. You can see this repository at https://github.com/esolares/esolares.github.io



What's going on? What mistake or mistakes might they have made?

It could be the link provided in the "()" where it should be: https://esolares.github.io/images/Solares-Edwin.png

```
Assume the \mbox{WhereAmI.java} file we've been using as an example is stored in the root folder of a GitHub repository at \mbox{https://github.com/ucsd-cse151-f23/WhereAmI}
```

```
local $ ssh cs15lfa23zz@ieng6.ucsd.edu
remote $ git clone https://github.com/ucsd-cse15l-f23/WhereAmI
remote $
```

What commands should we use to compile and run the WhereAmI. java file at this point?

javac WhereAml.java, java WhereAml

```
import java.io.IOException;
                                                                                               Counter.java
import java.net.URI;
// This interface is defined in another file (we provide it in lab 2)
// interface URLHandler { String handleRequest(URI url); }
// URI is a built-in Java class with methods like getPath() and getQuery()
class Counter implements URLHandler {
    int num = 0;
    public String handleRequest(URI url) {
        System.out.println(url);
        if (url.getPath().equals("/")) { return String.format("Number: %d", num); } If URL ends with "/" then return
                                                                                      num
        // FILL in the block below to match the behavior in the browser below!
        else if (url.getPath().equals("/count")) {
           num++:
                                                       else if URL ends with "/count" --> num += 1, return num
           return String.format("Number: %d", num);
        } else {
            return "Don't know what to do with that path!";
    }
}
class CounterMain {
    public static void main(String[] args) throws IOException {
        int port = Integer.parseInt(args[0]);
        // We wrote Server; it is a very short class (you can see it in Server.java in lab 2)
        Server.start(port, new Counter()); --> starts server
    }
}
local $ javac Server.java Counter.java
local $ java CounterMain
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 0 out of bounds for length 0
        at CounterMain.main(Counter.java:24)
local $ java CounterMain 4000
Server Started! Visit http://localhost:4000 to visit.
```

