- log in to github

- search for project open data

- fork repo

- if you want, you can then rename repo once you created the branch.  to do this, go to repo settings, then change the repo name to be [sdevine188.github.io](http://sdevine188.github.io/).  This base reference format works with the cio structure out of the box, because of the way they referenced their links.  If you don't make this change, the style sheets mess up, and you'll need to go into the html to fix links.

- to update the html link reference, first go to your repo settings, and change the name.  for instance, if you want the final url to be [sdevine188.github.io/test](http://sdevine188.github.io/test), then just write test in the name field.  then go into the \_includes/header.html folder.  a few paragraphs down, find

  <link rel="stylesheet" href="/assets/css/bootstrap.css">

  <link rel="stylesheet" href="/assets/css/site.css">

then insert "test/" before "assets" in these two lines, like:

  <link rel="stylesheet" href="/test/assets/css/bootstrap.css">

  <link rel="stylesheet" href="/test/assets/css/site.css">

then you can use the new url [sdevine188.github.io/test](http://sdevine188.github.io/test)

- next you need to edit the CNAME file.  the url listed in this file can be used to point at thegh-pages.  so if you put [b.site.com](http://b.site.com/) (sub-domain) in this file, the owner of [site.com](http://site.com/) can point [b.site.com](http://b.site.com/) at the gh ip address.

<https://help.github.com/articles/what-ip-addresses-does-github-use-that-i-should-whitelist/>

then when the request gets to gh, it will look through its gh-pages and see which ones have the requesting site ([b.site.com](http://b.site.com/)) listed in the CNAME file, and then route the traffic to this gh page ([sdevine188.github.io/test](http://sdevine188.github.io/test)).

- Also, if your url name is "[username.github.io](http://username.github.io/)", which is a special case since only one site can have this url, then you can commit md changes in the master branch of the repo.  However, if your url name is "[username.github.io/test](http://username.github.io/test)", then github will look for any updates to md files in a "gh-pages" branch.  So any changes committed to the master file would oddly not be reflected for the user.  So, if you want a url with /test, then click on the Branch drop down menu on the repo, and type in the find/create field "gh-pages".  This creates the gh-pages branch, and you can then update you md files there and it will be reflected for the user.  To simplify the updating process so that you don't always need to use the branch drop-down to navigate away from the default master branch to the gh-pages branch, you can go to repo settings and select gh-pages as the default branch pulled up when you click on the repo, thus you can go right into making changes to thegh-pages md files.

to create a link to a new page, click the plus next to the repo name to create a new file.  name the file "[other.md](http://other.md/)", and input whatever markdown text needed.  then go to the original home page [index.md](http://index.md/), and put hyperlinked text in brackets, followed immediately by the new url file path referenced against the original homepage, like [hyperlink to other](other/).  Once this is done, you need to make sure the [other.md](http://other.md/) file has the correctly update permalink and file reference.  To update this (only necessary once), go to the[other.md](http://other.md/) file, and edit the very top section permalink to be "/other/" (actually write quotes), and then the filename can be "[other.md](http://other.md/)" (actually write quotes).  once this updates, the user will be able to click through the hyperlink.

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

to embed file on web site, upload file to gh-pages using gitbash.  importantly, when the last code will be "git push origin gh-pages" not "git push origin master".  Then go to the md file of whatever page you want to embed the doc on, and use the similar syntax for links, [link to doc here](word\_doc.docx),

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

to edit Cascading Style Sheet (CSS), go to Assets folder, then CSS, then go to site.css.  Then you can use Inspect Element on actual web page to see the tags for different types of text, like h1, h2, h3 headers, or p text (main body).  then you can overwrite default rules in CSS by adding new rules like

p {font-family: "Arial";

    font-size: 30px;}