P&A GSO & SPS

Personal Website Workshop

October 24, 2023

CONTENTS

01

Why make a website?

02

What is GitHub?

03

Introduction to GitHub Pages

04

Website Tutorial + HTML Basics

Please follow this link to access the Google Drive folder of resources:

https://drive.google.com/drive/folders/1pe8vpGoRWdowZlNAwaP4-74ZZUl_aIIr?usp=share_link

01

WHY MAKE A WEBSITE?

WHY MAKE A PERSONAL WEBSITE?

Why is it important?

- The simple answer is that it makes it easier for other professionals to find you, learn about your work, and contact you.
- A website is a great place to showcase your published work, conference contributions, and perhaps even write popular dissemination texts about your articles, or perhaps you can have a nerdy blog which would be useful for certain audiences. It's also a great place to make yourself visible to a broader audience.

When/how should I share it?

- Link to it on conference posters / talks
- Link to it on your CV
- Link to it on your social media sites
- If your research group has a website, you can ask your advisor to link to your personal website on that page as well
- Anywhere else you think would be helpful!

Personal Website Content

What should you include on your website?

- About me page - a photo of you, introduction to you and your interests (personal and professional alike) (i.e. research interests, hobbies, sports, travels, etc)
- A link to your CV - link to a pdf copy of your CV
- Research page - a brief description of your research organized by topic
- Contact information - email, links to social media profiles (LinkedIn, Twitter, Bluesky, etc)
- Other ideas: outreach activities, public talks, awards, passion projects, workshops, blog of professional updates, links to articles you've written/interviewed for, press releases etc.

Personal Website Content

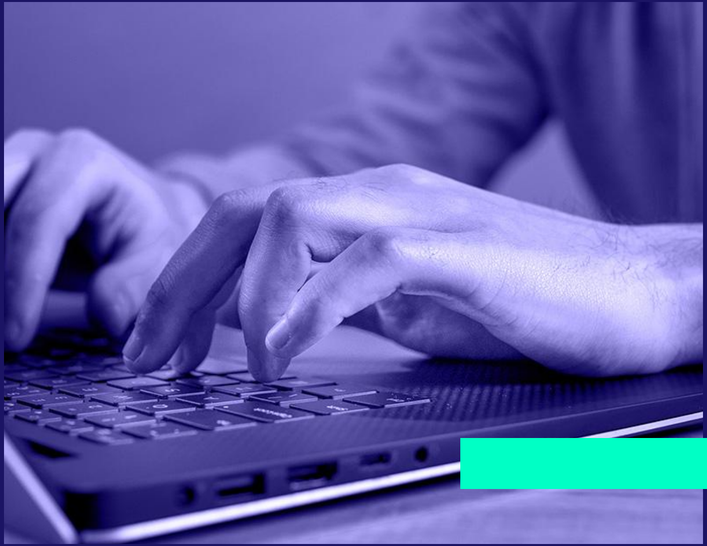
Other notes:

- Include pictures! If you participate in a workshop, present at a conference, have a particular hobby, create a project, etc. pictures will always make it more appealing!
- It is important to keep your website relatively up to date!
- When styling your website, be mindful of using easy to read fonts, font sizes, color combinations, etc. and to not make it too crowded
- Think carefully about the organization of your site! Make it logical and easy to navigate.

EXAMPLE WEBSITES

Looking at others’ website can often give you the best ideas about what to include and how to organize your site!

The document we gave has links to several good examples of websites!



02

QUICK INTRO TO GITHUB

GitHub

GitHub is a web-based app that lets you host files and code in repositories, collaborate on work, and track your changes over time. Version tracking ensures you can always revert to earlier versions of work as you experiment.

And, you are able to host a website through GitHub pages!

If you are unfamiliar with GitHub, please see the quick introduction page in the workshop guide for more helpful information, tutorials, and resources to help get you up to speed with GitHub!

If you don't have a GitHub account, please go to https://github.com/join to make a free account now!

03

Introduction to GitHub Pages

GITHUB PAGES

What is GitHub pages?

- GitHub pages is a simple service to publish a website directly on GitHub from a Git repository. You are able to add files and folders to your repo and GitHub pages will turn it into a website.
- Your domain will be generally be: username.github.io

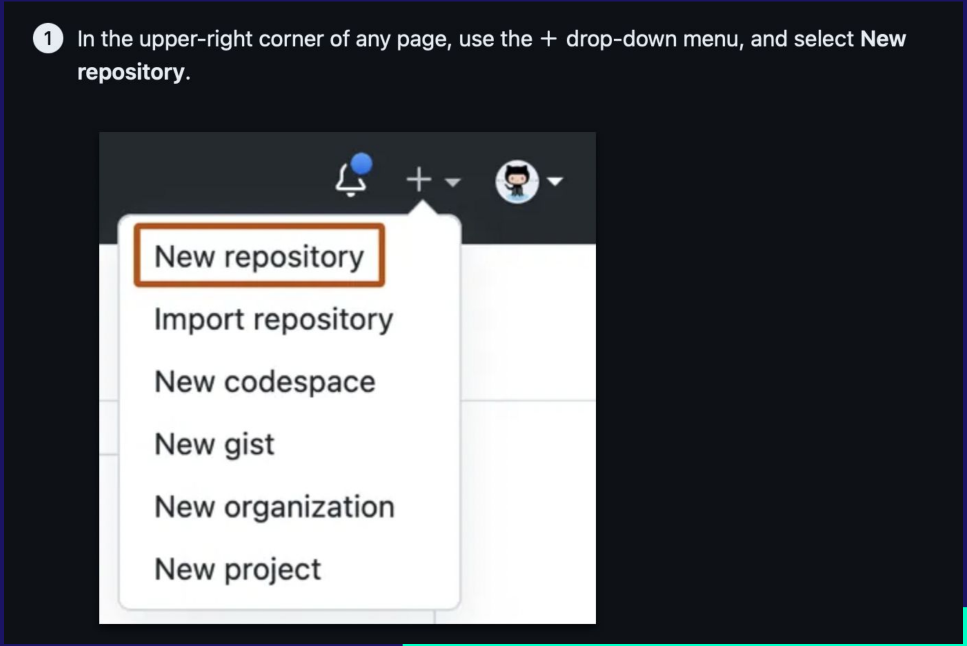
- GitHub Pages sites are have a recommended limit of 1 GB
- GitHub Pages sites have a soft bandwidth limit of 100 GB per month
  - This shouldn't be an issue, but be mindful of what you are uploading to your site's repository

04

Website Making Tutorial

Step 1: Build your Domain

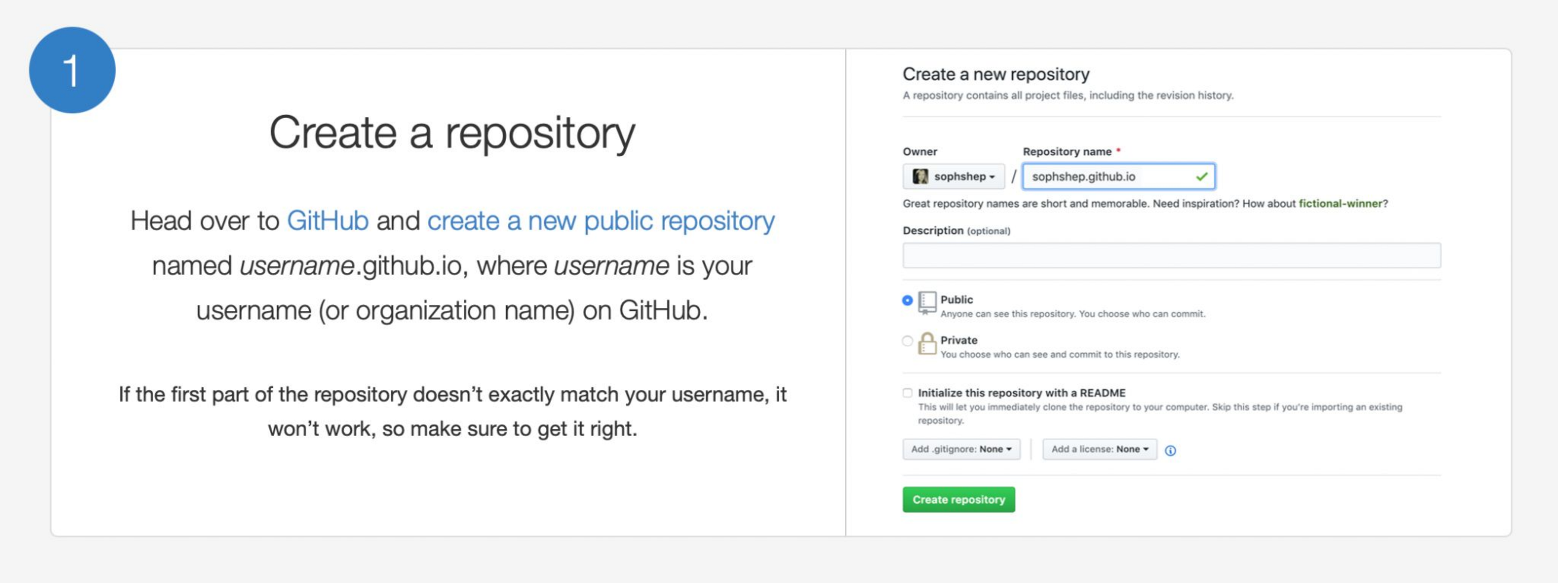
- Navigate to https://pages.github.com or https://docs.github.com/en/pages to follow along at your own pace



1 In the upper-right corner of any page, use the + drop-down menu, and select **New repository**.

Step 1: Build your Domain

- Navigate to https://pages.github.com to follow along at your own pace



1

Create a repository

Head over to GitHub and create a new public repository named *username*.github.io, where *username* is your username (or organization name) on GitHub.

If the first part of the repository doesn't exactly match your username, it won't work, so make sure to get it right.

Create a new repository

A repository contains all project files, including the revision history.

Owner: sophshep / Repository name *: sophshep.github.io ✓

Great repository names are short and memorable. Need inspiration? How about fictional-winner?

Description (optional)

● Public
Anyone can see this repository. You choose who can commit.

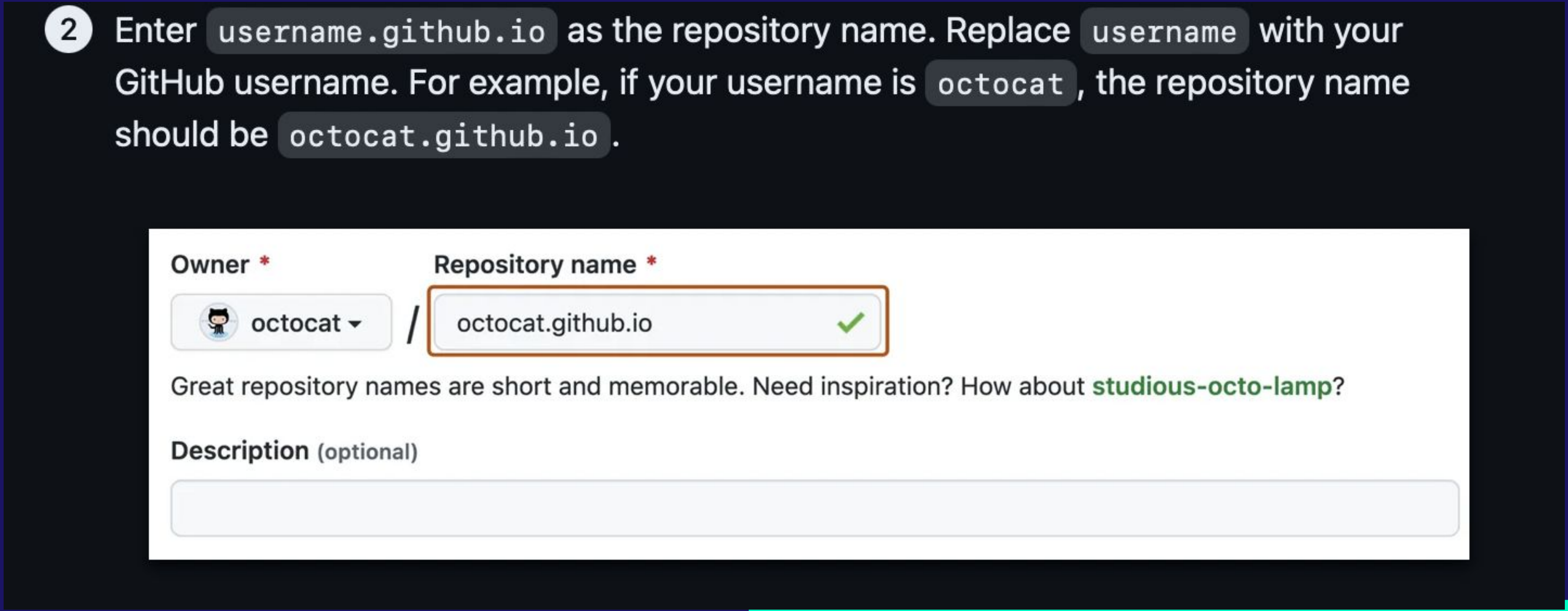
○ Private
You choose who can see and commit to this repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾ | Add a license: None ▾

Step 1: Build your Domain

- Navigate to https://pages.github.com to follow along at your own pace



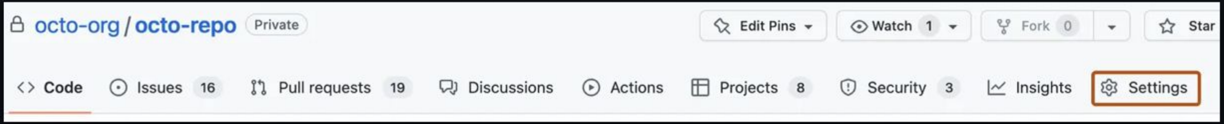
2 Enter `username.github.io` as the repository name. Replace `username` with your GitHub username. For example, if your username is `octocat`, the repository name should be `octocat.github.io`.

Owner * octocat / Repository name * octocat.github.io ✓

Great repository names are short and memorable. Need inspiration? How about studious-octo-lamp?

Description (optional)

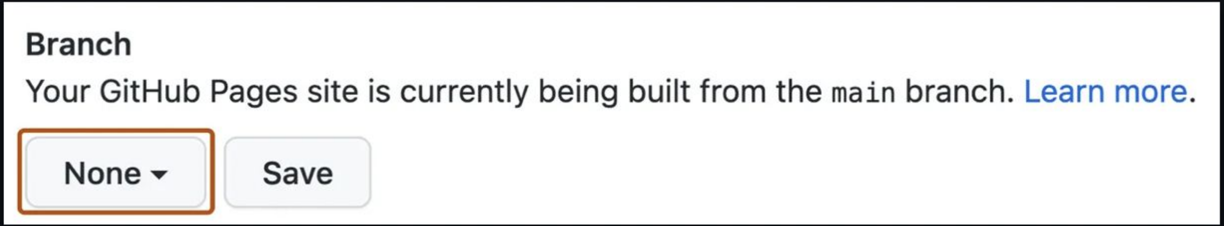
3. Under your repository name, click **Settings**. If you cannot see the "Settings" tab, select the ··· dropdown menu, then click **Settings**.



4. In the "Code and automation" section of the sidebar, click **Pages**.

5. Under "Build and deployment", under "Source", select **Deploy from a branch**.

6. Under "Build and deployment", under "Branch", use the branch dropdown menu and select a publishing source.



Branch

Your GitHub Pages site is currently being built from the `main` branch. Learn more.

Step 2: Import a Template

- **Many templates can be found at https://html5up.net/**

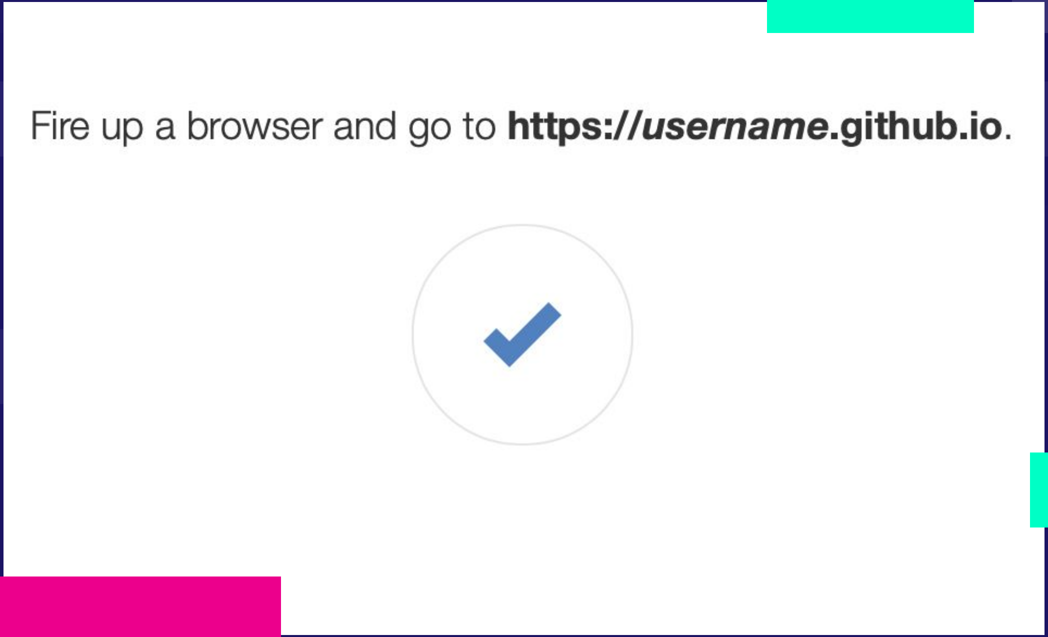
1. Download your favorite template. Unzip folder.
2. Save to your GitHub repository (this can be done from either your terminal or browser – depending on what you are comfortable with)
   - Be sure to upload all of the files that come with the template (configs, etc.) even if you aren’t sure what they are used for.
   - Be sure to upload the contents of the zip folder, not the main folder itself
3. Commit/Push these changes
   - Either from browser using commit button
   - Or if using your computer’s terminal, these commands will help you to push your changes
     - git add –all
     - git commit -m “Initial Commit”
     - git push -u origin main

Step 2: Import a Template

- **Many templates can be found at https://html5up.net/**

1. Download your favorite template. Unzip folder.
2. Save to your GitHub repository (this can be done from either your terminal or browser – depending on what you are comfortable with)
    - Be sure to upload all of the files that come with the template (configs, etc.) even if you aren't sure what they are used for.
    - Be sure to upload the contents of the zip folder, not the main folder itself
3. Commit/Push these changes
    - Either from browser using commit button
    - Or if using your computer's terminal, these commands will help you to push your changes
        - git add –all
        - git commit -m "Initial Commit"
        - git push -u origin main

LET'S TAKE A LOOK!

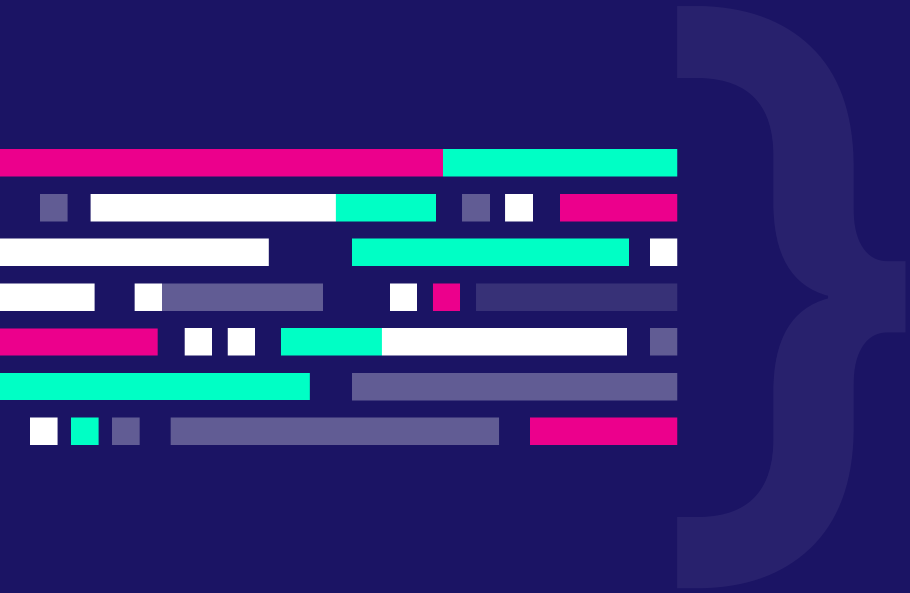


Fire up a browser and go to **https://*username*.github.io**.

Note: sometimes it may take up to 10 minutes
for pushed changes to appear on your site

05

CUSTOMIZING YOUR WEBSITE



HTML

(HYPER TEXT MARKUP LANGUAGE)

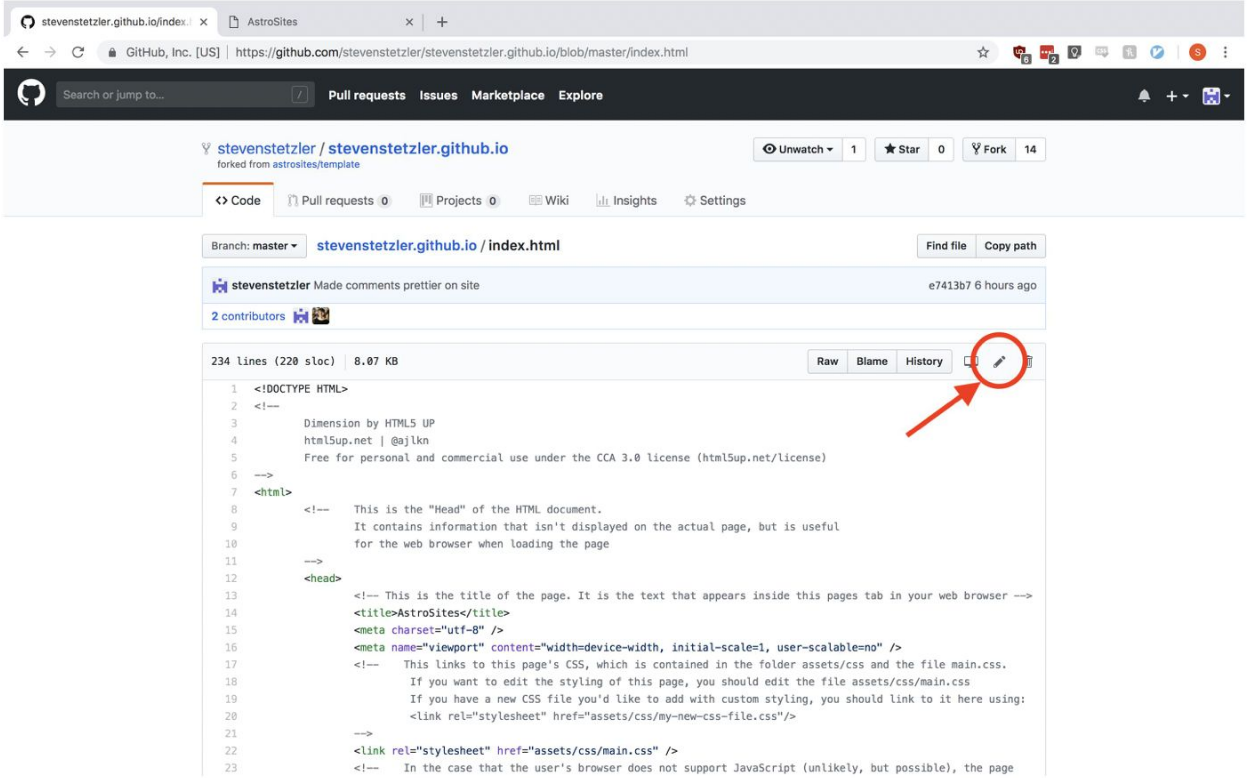
- HTML is the standard markup language for creating web pages.

- HTML elements tell your browser how to display the content of your web page.

- For a comprehensive tutorial on HTML please follow this link: **HTML Tutorial**
- Website Building Blocks: **Website Building Blocks**

Editing the Template

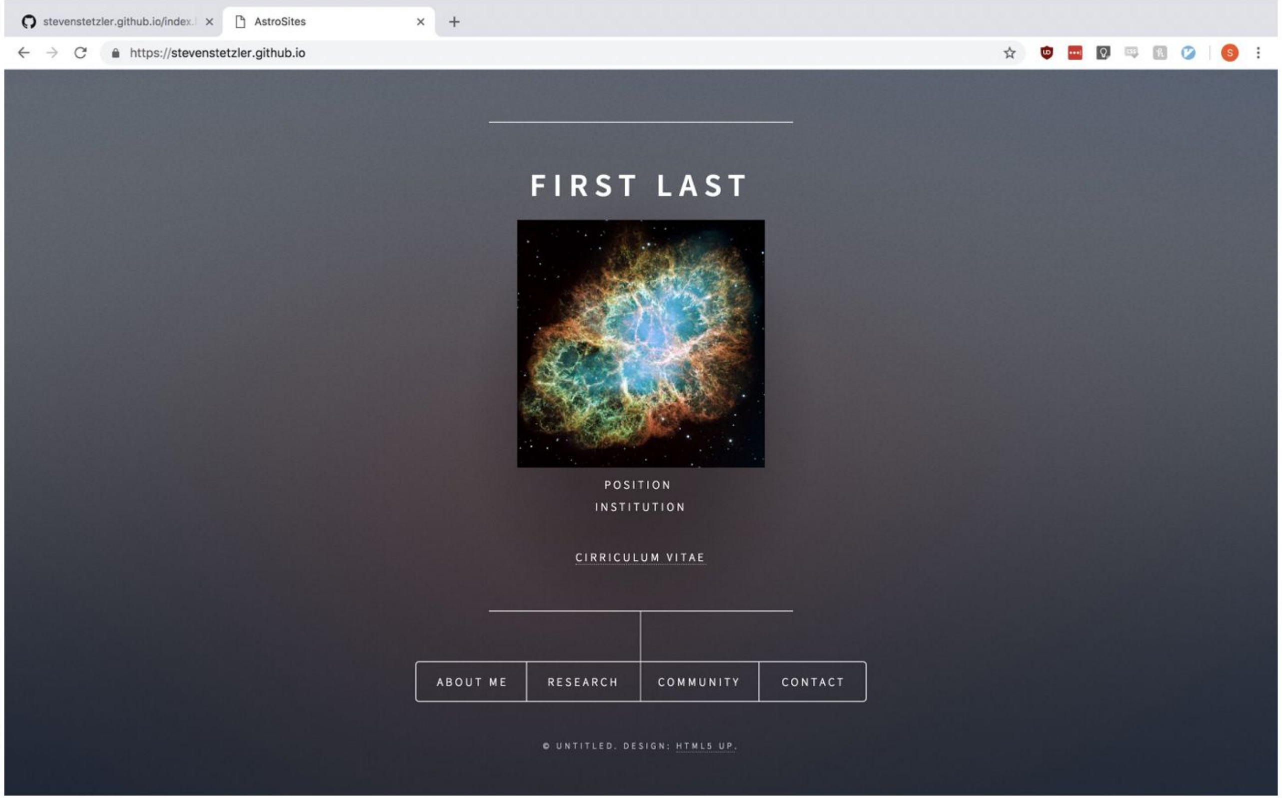
Most of the edits we will perform will be on `index.html`. Click on that file in your repository to open it up. Once the file is loaded, click on the **pencil** icon to start making edits.


stevenstetzler.github.io/index.
AstroSites
GitHub, Inc. [US] | https://github.com/stevenstetzler/stevenstetzler.github.io/blob/master/index.html
Search or jump to...
Pull requests
Issues
Marketplace
Explore
stevenstetzler / stevenstetzler.github.io
forked from astrosites/template
Unwatch 1
Star 0
Fork 14
Code
Pull requests 0
Projects 0
Wiki
Insights
Settings
Branch: master
stevenstetzler.github.io / index.html
Find file
Copy path
stevenstetzler Made comments prettier on site
e7413b7 6 hours ago
2 contributors
234 lines (220 sloc) | 8.07 KB
Raw
Blame
History

```
<!DOCTYPE HTML>
<!--
        Dimension by HTML5 UP
        html5up.net | @ajlkn
        Free for personal and commercial use under the CCA 3.0 license (html5up.net/license)
-->
<html>
        <!--    This is the "Head" of the HTML document.
                It contains information that isn't displayed on the actual page, but is useful
                for the web browser when loading the page
        -->
        <head>
                <!-- This is the title of the page. It is the text that appears inside this pages tab in your web browser -->
                <title>AstroSites</title>
                <meta charset="utf-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
                <!--    This links to this page's CSS, which is contained in the folder assets/css and the file main.css.
                         If you want to edit the styling of this page, you should edit the file assets/css/main.css
                         If you have a new CSS file you'd like to add with custom styling, you should link to it here using:
                         <link rel="stylesheet" href="assets/css/my-new-css-file.css"/>
                -->
                <link rel="stylesheet" href="assets/css/main.css" />
                <!--    In the case that the user's browser does not support JavaScript (unlikely, but possible), the page
```

Credit: https://astrosites.github.io/tutorial/

In a second tab or window, pull up your site by navigating to https://your-github-name.github.io. This will allow you to see your edits happen as you make them in the HTML file.



stevenstetzler.github.io/index.
AstroSites
https://stevenstetzler.github.io
FIRST LAST
POSITION
INSTITUTION
CIRRICULUM VITAE
ABOUT ME
RESEARCH
COMMUNITY
CONTACT
© UNTITLED. DESIGN: HTML5 UP.

Credit: https://astrosites.github.io/tutorial/

Customizing your Website

For an in depth look at how to change elements piece by piece, please reference this article! It slowly steps through how to change elements one by one on an HTML5UP template:

https://astrosites.github.io/tutorial/

HTML Basics

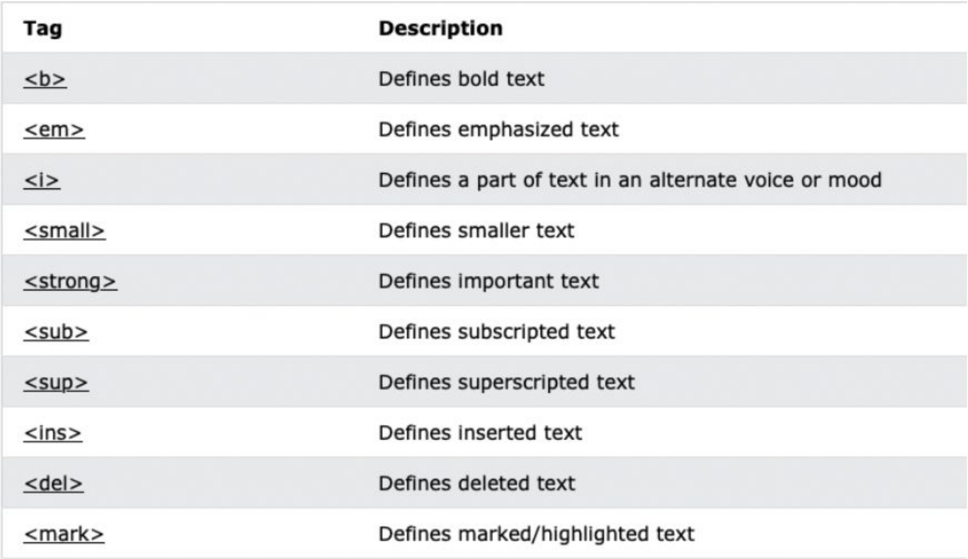
- Tags: open <> and close </>
- Attributes: additional pieces of information within tags, i.e. att="attribute"
  - Based on the template you choose, much of this should be rather straightforward, but you are able to customize it, as you wish.
  - Be sure to keep track of the order of your text, attributes, and close tags especially when you have tags within tags.

HTML BASICS

- Creating a new paragraph:
  - <p> paragraph </p>
  - New line: <br> this is an empty tag, no end </> needed)
- Adding a link:
  - Absolute: <a href= “https://url.com“> link </a>
  - Relative: <a href= “other_page.asp“>link within site</a>
  - Add attribute target=“_blank“ to open link in new tab (default is _self)
- Buttons
  - There should be some javascript in your template already
  - <button onclick=“document.location=‘default.asp‘“>text</button>
  - Adding an image:
    - Save image to your github repo
    - Add an image tag with appropriate attributes
    - <img src=“image.jpg“ width=“pix“ height=“pix“>
    - Replace image.jpg with your path/filename from your repo

TEXT FORMATTING

- Formatting will all be like this: <b>bold text</b>

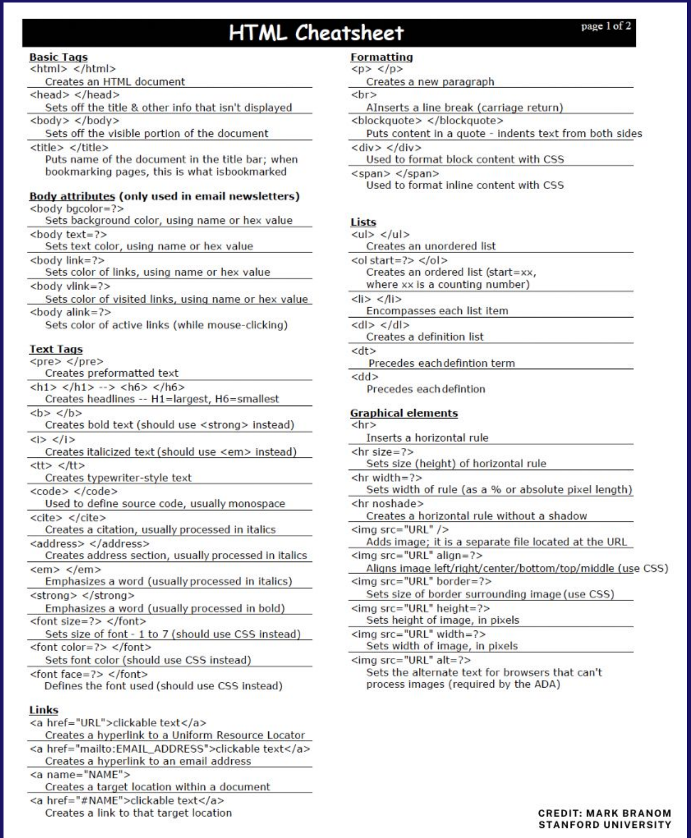


Tag	Description
<b>	Defines bold text
<em>	Defines emphasized text
<i>	Defines a part of text in an alternate voice or mood
<small>	Defines smaller text
<strong>	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

HTML BASICS

- Style attribute (within tags such as <p>)
    - Colors
        - **List of named colors**
        - Text color style=”color:Black;”
        - Background color style=”background-color:Black;”
        - RGB: style="color:rgb(0, 0, 0);"
        - Hex: style="color:#000000;"
    - Fonts
        - **List of named fonts**
        - style=”font-family:courier;”
        - style=”font-size:150%;”
    - Alignment
        - style=”text-align:center;”
- Lists (tags within tags! Indenting doesn’t really matter but it looks neater)
    - Unordered: <ul> <li>item</li> <li>item</li> </ul>
    - Ordered: <ol> <li>item</li> <li>item</li> </ol>

HTML CHEATSHEET



HTML Cheatsheet

page 1 of 2

Basic Tags

<html> </html>
Creates an HTML document

<head> </head>
Sets off the title & other info that isn't displayed

<body> </body>
Sets off the visible portion of the document

<title> </title>
Puts name of the document in the title bar; when bookmarking pages, this is what isbookmarked

Body attributes (only used in email newsletters)

<body bgcolor=?>
Sets background color, using name or hex value

<body text=?>
Sets text color, using name or hex value

<body link=?>
Sets color of links, using name or hex value

<body vlink=?>
Sets color of visited links, using name or hex value

<body alink=?>
Sets color of active links (while mouse-clicking)

Text Tags

<pre> </pre>
Creates preformatted text

<h1> </h1> --> <h6> </h6>
Creates headlines -- H1=largest, H6=smallest

<b> </b>
Creates bold text (should use <strong> instead)

<i> </i>
Creates italicized text (should use <em> instead)

<tt> </tt>
Creates typewriter-style text

<code> </code>
Used to define source code, usually monospace

<cite> </cite>
Creates a citation, usually processed in italics

<address> </address>
Creates address section, usually processed in italics

<em> </em>
Emphasizes a word (usually processed in italics)

<strong> </strong>
Emphasizes a word (usually processed in bold)

<font size=?> </font>
Sets size of font - 1 to 7 (should use CSS instead)

<font color=?> </font>
Sets font color (should use CSS instead)

<font face=?> </font>
Defines the font used (should use CSS instead)

Links

<a href="URL">clickable text</a>
Creates a hyperlink to a Uniform Resource Locator

<a href="mailto:EMAIL_ADDRESS">clickable text</a>
Creates a hyperlink to an email address

<a name="NAME">
Creates a target location within a document

<a href="#NAME">clickable text</a>
Creates a link to that target location

Formatting

<p> </p>
Creates a new paragraph

AInserts a line break (carriage return)

<blockquote> </blockquote>
Puts content in a quote - indents text from both sides

<div> </div>
Used to format block content with CSS

<span> </span>
Used to format inline content with CSS

Lists

<ul> </ul>
Creates an unordered list

<ol start=?> </ol>
Creates an ordered list (start=xx, where xx is a counting number)

<li> </li>
Encompasses each list item

<dl> </dl>
Creates a definition list

<dt>
Precedes each defintion term

<dd>
Precedes each defintion

Graphical elements

<hr>
Inserts a horizontal rule

<hr size=?>
Sets size (height) of horizontal rule

<hr width=?>
Sets width of rule (as a % or absolute pixel length)

<hr noshade>
Creates a horizontal rule without a shadow

<img src="URL" />
Adds image; it is a separate file located at the URL

<img src="URL" align=?>
Aligns image left/right/center/bottom/top/middle (use CSS)

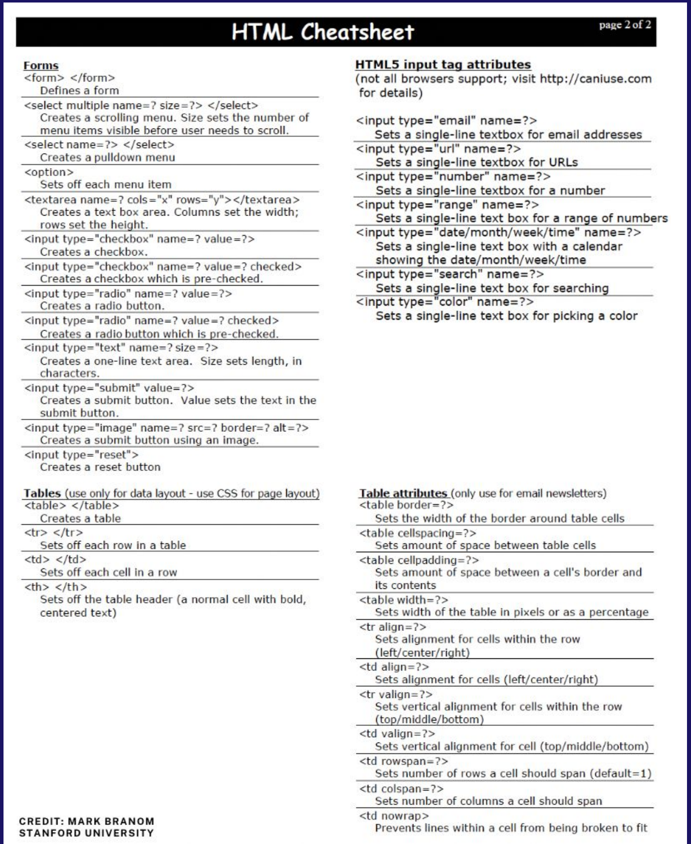
<img src="URL" border=?>
Sets size of border surrounding image (use CSS)

<img src="URL" height=?>
Sets height of image, in pixels

<img src="URL" width=?>
Sets width of image, in pixels

<img src="URL" alt=?>
Sets the alternate text for browsers that can't process images (required by the ADA)

CREDIT: MARK BRANOM
STANFORD UNIVERSITY



HTML Cheatsheet

page 2 of 2

Forms

<form> </form>
Defines a form

<select multiple name=? size=?> </select>
Creates a scrolling menu. Size sets the number of menu items visible before user needs to scroll.

<select name=?> </select>
Creates a pulldown menu

<option>
Sets off each menu item

<textarea name=? cols="x" rows="y"></textarea>
Creates a text box area. Columns set the width; rows set the height.

<input type="checkbox" name=? value=?>
Creates a checkbox.

<input type="checkbox" name=? value=? checked>
Creates a checkbox which is pre-checked.

<input type="radio" name=? value=?>
Creates a radio button.

<input type="radio" name=? value=? checked>
Creates a radio button which is pre-checked.

<input type="text" name=? size=?>
Creates a one-line text area. Size sets length, in characters.

<input type="submit" value=?>
Creates a submit button. Value sets the text in the submit button.

<input type="image" name=? src=? border=? alt=?>
Creates a submit button using an image.

<input type="reset">
Creates a reset button

Tables (use only for data layout - use CSS for page layout)

<table> </table>
Creates a table

<tr> </tr>
Sets off each row in a table

<td> </td>
Sets off each cell in a row

<th> </th>
Sets off the table header (a normal cell with bold, centered text)

HTML5 input tag attributes
(not all browsers support; visit http://caniuse.com for details)

<input type="email" name=?>
Sets a single-line textbox for email addresses

<input type="url" name=?>
Sets a single-line textbox for URLs

<input type="number" name=?>
Sets a single-line textbox for a number

<input type="range" name=?>
Sets a single-line text box for a range of numbers

<input type="date/month/week/time" name=?>
Sets a single-line text box with a calendar showing the date/month/week/time

<input type="search" name=?>
Sets a single-line text box for searching

<input type="color" name=?>
Sets a single-line text box for picking a color

Table attributes (only use for email newsletters)

<table border=?>
Sets the width of the border around table cells

<table cellspacing=?>
Sets amount of space between table cells

<table cellpadding=?>
Sets amount of space between a cell's border and its contents

<table width=?>
Sets width of the table in pixels or as a percentage

<tr align=?>
Sets alignment for cells within the row (left/center/right)

<td align=?>
Sets alignment for cells (left/center/right)

<tr valign=?>
Sets vertical alignment for cells within the row (top/middle/bottom)

<td valign=?>
Sets vertical alignment for cell (top/middle/bottom)

<td rowspan=?>
Sets number of rows a cell should span (default=1)

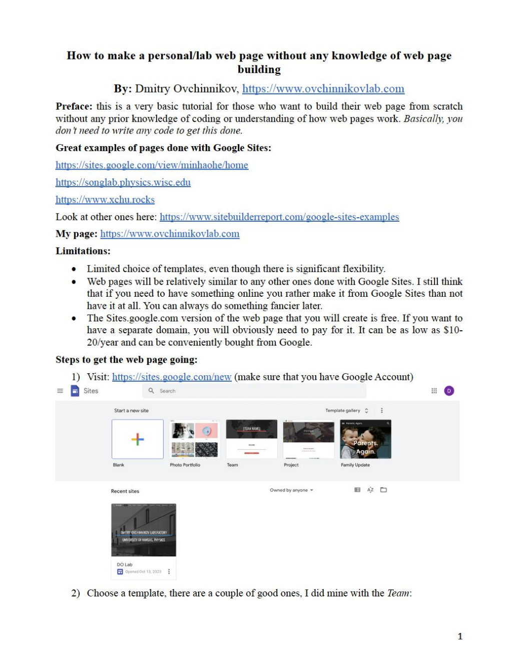
<td colspan=?>
Sets number of columns a cell should span

<td nowrap>
Prevents lines within a cell from being broken to fit

CREDIT: MARK BRANOM
STANFORD UNIVERSITY

GitHub Pages Alternative: Google Sites

Dr. Dmitry Ovchinnikov utilized Google sites to make his website which is a much more user friendly way to make a first website – no HTML editing involved! He created a quick guide on how to make one of these web pages. We've included that in the Google Drive folder for those that might be interested in checking that out!



How to make a personal/lab web page without any knowledge of web page building

By: Dmitry Ovchinnikov, https://www.ovchinnikovlab.com

Preface: this is a very basic tutorial for those who want to build their web page from scratch without any prior knowledge of coding or understanding of how web pages work. *Basically, you don't need to write any code to get this done.*

Great examples of pages done with Google Sites:

https://sites.google.com/view/minhaohe/home

https://songlab.physics.wisc.edu

https://www.xchu.rocks

Look at other ones here: https://www.sitebuilderreport.com/google-sites-examples

My page: https://www.ovchinnikovlab.com

Limitations:

- Limited choice of templates, even though there is significant flexibility.
- Web pages will be relatively similar to any other ones done with Google Sites. I still think that if you need to have something online you rather make it from Google Sites than not have it at all. You can always do something fancier later.
- The Sites.google.com version of the web page that you will create is free. If you want to have a separate domain, you will obviously need to pay for it. It can be as low as $10-20/year and can be conveniently bought from Google.

Steps to get the web page going:

1) Visit: https://sites.google.com/new (make sure that you have Google Account)

2) Choose a template, there are a couple of good ones, I did mine with the *Team*:

THANKS! Happy Website Building!

CREDITS: This presentation template was created by Slidesgo, incluiding icons by Flaticon, and infographics & images by Freepik.