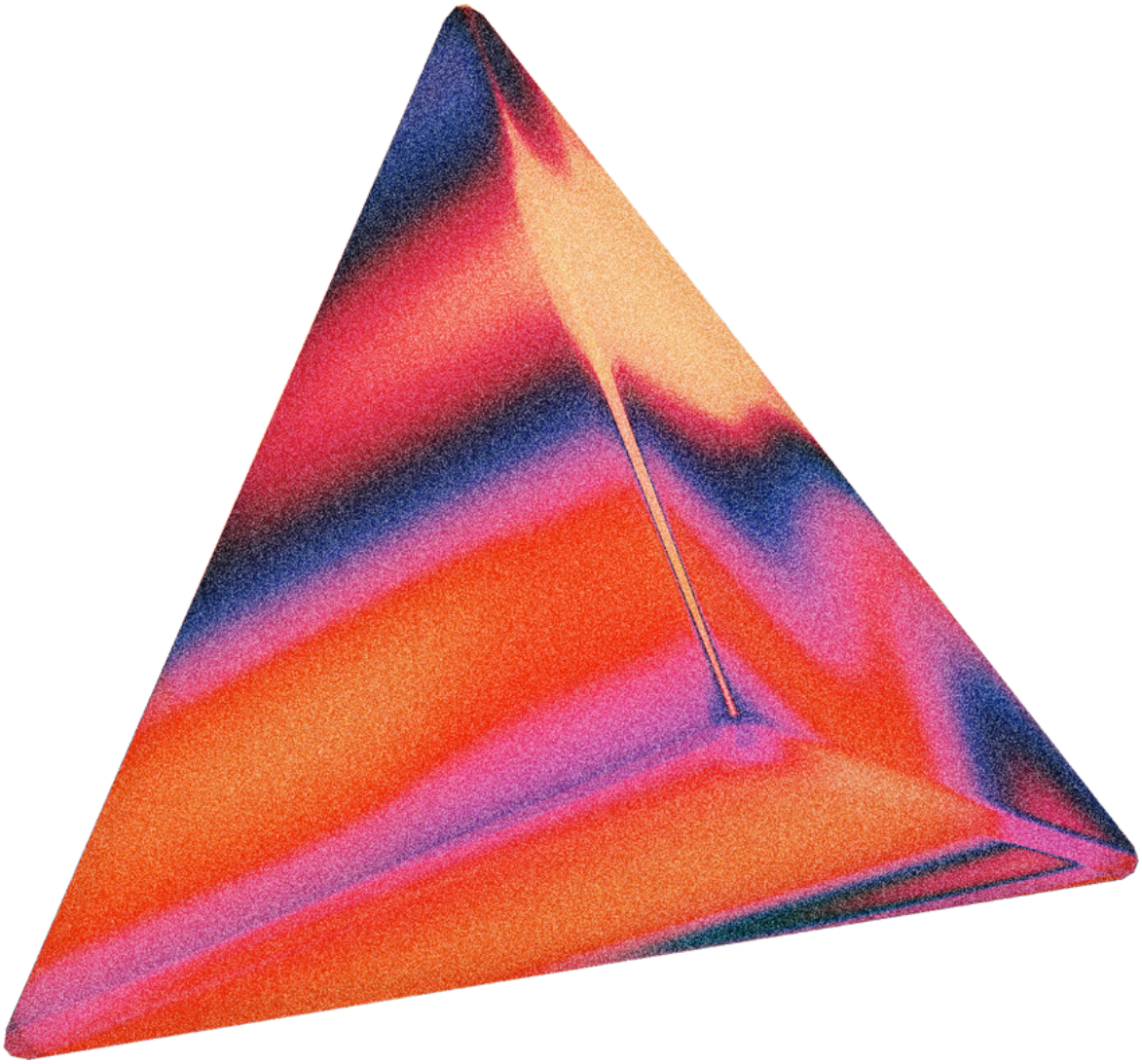
KU PHYSICS & ASTRONOMY GSO AND SPS

Personal Website Workshop Guide

OCTOBER 24, 2023



Contents

This guide accompanies the personal website workshop hosted by the Graduate Student Organization and Society of Physics Students from the Physics and Astronomy Department at the University of Kansas. This guide presents a curated list of public resources and advice for students in making professional personal websites specifically using GitHub Pages.

Special thanks to Joseph Hand, Alex Polanski, Keaton Donaghue, Yoni Brande, Ashley Lieber and the KU Department of Physics & Astronomy for making this workshop possible! We hope it's helpful!

Why Make a Website?

Why should you create a personal website?

As a scientist, why is it important for you to create a website to promote yourself and your work? 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.

How to make a website?

- GitHub Pages (what this workshop will introduce)
- Google Sites
- WordPress.com

What should I put on my website?

- About me page
  - This could include a photo of you, an introduction to your interests both research-wise and even personal (i.e. rock climbing, surfing, photography etc).
- A link to your CV
- Research page
  - This is where you can detail your research projects/internship work (REUs) and put recent results.
  - If you have published, this is a great place to list your publications and link to them on places like: Arxiv, SAO/NASA ADS (Astrophysics Data System), ORCID, Research Gate, Google Scholar, etc.
  - If your research group has a page, you could link to that site too.
  - You may also want to link to your GitHub as well
- Contact information
  - It is good to have a place for people to contact you. This can be a form on your website, listing your professional (usually .edu) email.
  - You can also link to professional social media sites such as LinkedIn, Twitter, Bluesky, etc.
- Other ideas:
  - Outreach activities -- pictures are always nice if possible
  - Blog of professional updates
  - Any other professional activities: conferences, workshops, passion projects, etc.

Advice adapted from this paper by Emily Moravec (2020) about crating professional websites:
https://arxiv.org/abs/2012.08553

Example Websites

Looking at other's websites can give you a lot of great insight into what you could include on yours. Please find several examples below bother from our department and beyond.

Astrophysics Examples

- Dr. Ian Crossfield Research Website: https://crossfield.ku.edu
- Dr. Elisabeth Mills Research Website: https://mills.ku.edu
- Dr. Allison Kirkpatrick Research Website: https://kirkpatrick.ku.edu
- Dr. Greg Rudnick Research Website: https://rudnick.ku.edu

Astroparticle Examples

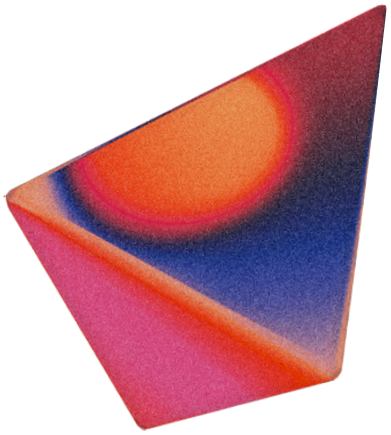
- Dr. Steven Prohira Radar Echo Telescope Project Website: https://www.radarechotelescope.org/#home

Condensed Matter Examples

- Dr. Dmitry Ovchinnikov Research Website: https://www.ovchinnikovlab.com
- Dr. Judy Wu Research Site: https://thinfilm.ku.edu
- Dr. Hui Zhao Research Site: https://ultrafast.ku.edu
- Dr. Qunfei Zhou Research Group: https://sites.google.com/view/zhougroup

Student Website Examples:

- Dr. Mindy Townsend: https://www.ad-astra-per-esquire.com
- Alex Polanski: https://aspolanski.github.io
- Yoni Brande: https://jbrande.github.io



Quick Introduction to GitHub

If you've never used GitHub, or have very limited experience, please follow the advice and information on this page to learn about it and get set up before embarking on makring your own website!

GitHub is a web-based app that lets you host files in repositories, collaborate on work, and track changes to files over time. Version tracking on GitHub is powered by the open source software Git. Whenever you update a repository on GitHub, Git tracks the changes you make.

- First things first, sign up for a free account on GitHub.com: https://github.com/join
- Then you can follow the "Hello World" GitHub exercise to get familiar with the most common GitHub commands: https://docs.github.com/en/get-started/quickstart/hello-world
- Fun intro video: https://www.youtube.com/watch?v=pBy1zgt0XPc
- Confused on a term? Look at the glossary here: https://docs.github.com/en/get-started/quickstart/github-glossary
- Forget a quick command: Look at the Git Cheatsheet here: https://docs.github.com/en/get-started/quickstart/git-cheatsheet

Introduction to GitHub Pages

GitHub Pages is a simple service to publish a website directly on GitHub from a Git repository. You add some files and folders to a repository and GitHub Pages turns it into a website. You can use HTML directly if you like, but they also provide Jekyll, which renders Markdown into HTML and makes it really easy to setup a blog or a template-based website. or you can upload an HTML template and edit that. You can edit and work with your GitHub files either online in your browser, or from your terminal. (link)

HOW TO SET UP A PERSONAL WEBSITE (THE EASY WAY)

Step 1: Build you Domain

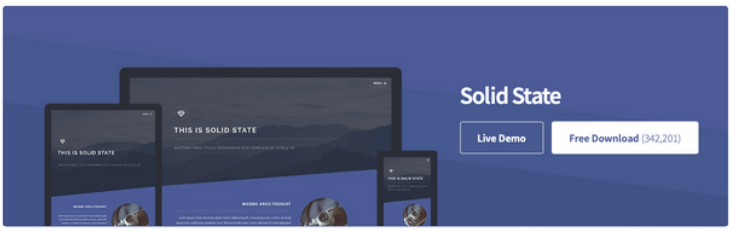
- Follow along on https://pages.github.com
- GitHub Pages Documentation: https://docs.github.com/en/pages
- GitHub pages is very user friendly and since it is attached to you, and not KU, you can take it with you as you progress throughout your career.
- You can choose to make changes to your GitHub repository either from your terminal or in the browser.

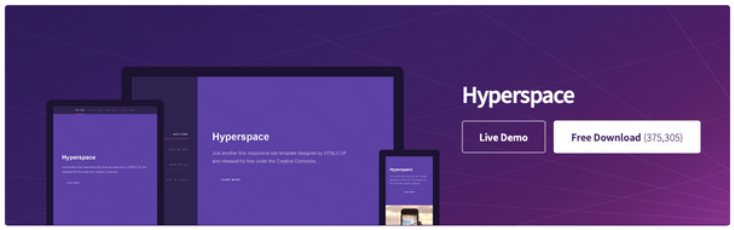
Step 2: Import a Template

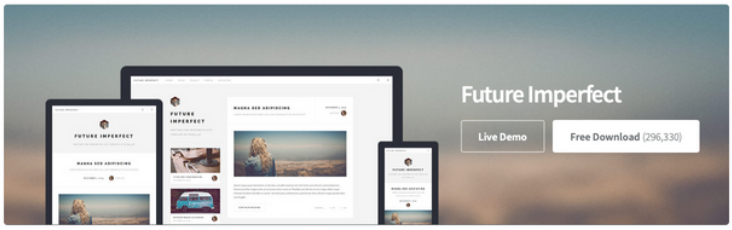
- There are a bunch here https://html5up.net/
- Download your fave template. Unzip folder
- Save to your github repo (either from terminal or in browser - browser upload and editing is generally easier in my opinion)
  - You should upload all of the files that come with the template (configs, etc.) even if you're not sure what they're for. Upload the contents of the zip folder and not the main folder itself.
- Push changes
  - If using the terminal, these commands will push your changes
    - git add --all
    - git commit -m "Initial commit"
    - git push -u origin main

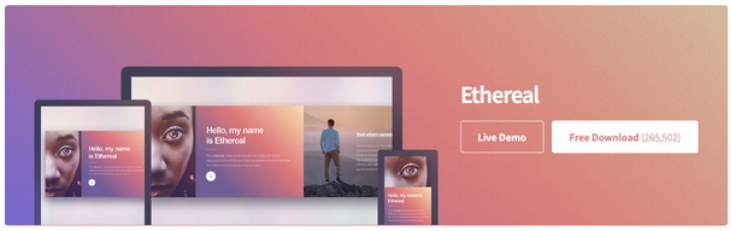
Website Templates

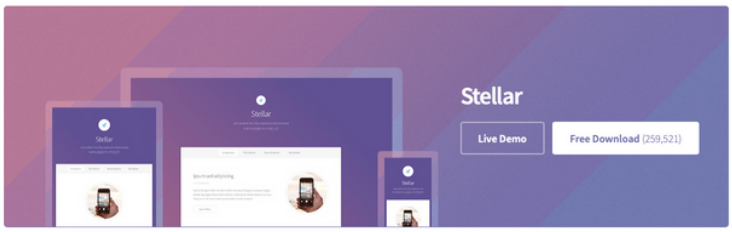
No need to reinvent the wheel, unless you want to. There are tons of good drag and drop templates for Github Pages based websites is HTML5up. Some examples of templates you can download for free are shown below.
Find a good resource of templates here: **LINK**











Additional GitHub Pages Resources

GitHub Pages is a simple service to publish a website directly on GitHub from a Git repository. You add some files and folders to a repository and GitHub Pages turns it into a website. You can use HTML directly if you like, but they also provide Jekyll, which renders Markdown into HTML and makes it really easy to setup a blog or a template-based website. or you can upload an HTML template and edit that. You can edit and work with your GitHub files either online in your browser, or from your terminal. (link)

- CodeAcademy Startup Guide: **LINK**

GITHUB PAGES SKILL TUTORIAL

This GitHub skill page will take you through some more tutorial on working with GitHub pages. https://github.com/skills/github-pages

TUTORIAL VIDEO

Tutorial Link

LIMITS

There are some general usage limits on GitHub Pages sites. In general, GitHub Pages sites have a recommended limit of 1 GB and a soft bandwidth of 100 GB per month. This shouldn't be too much of an issue, but be careful of what all you are uploading to your website's repository to make sure you aren't exceeding this limit!

ADDITIONAL PAGES TUTORIAL

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

CUSTOMIZING YOUR WEBSITE

Introduction to 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: **https://www.w3schools.com/html/default.asp**

QUICK HTML HELP (ADAPTED FROM HAWAII REU WORKSHOP: LINK)

- `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 to help fill in and organize your information:

- 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

Continued on the next page...

QUICK HTML CONTINUED (ADAPTED FROM HAWAII REU WORKSHOP: LINK)

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

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

- 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>
  <li>item</li>
</ul>
```

  - `Ordered: <ol> <li>item</li> <li>item</li> </ol>`

Better, more thorough guide here: https://www.w3schools.com/html/default.asp Also, please see the HTML cheatsheet on the following pages for a quick reference on HTML formatting.

HTML Cheatsheet

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

Creates bold text (should use instead)

Creates italicized text (should use 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

Emphasizes a word (usually processed in italics)

Emphasizes a word (usually processed in bold)

Sets size of font - 1 to 7 (should use CSS instead)

Sets font color (should use CSS instead)

Defines the font used (should use CSS instead)

Links

clickable text
Creates a hyperlink to a Uniform Resource Locator

clickable text
Creates a hyperlink to an email address

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

clickable text
Creates a link to that target location

Formatting

Creates a new paragraph

AInserts a line break (carriage return)

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

Used to format block content with CSS

Used to format inline content with CSS

Lists

Creates an unordered list

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

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

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

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

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

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)

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

Additional Resources

HTML Cheatsheets:

- Stony Brook HTML Cheatsheet: https://www3.cs.stonybrook.edu/~pramod.ganapathi/doc/CSE102/CSE102-CheatSheetHTML.pdf
- CSS Cheatsheet: https://htmlcheatsheet.com/css/
- Geeks for Geeks CSS Basic Guide: https://www.geeksforgeeks.org/css-cheat-sheet-a-basic-guide-to-css/
- CSS Cascading Style Sheets Cheatsheets (included in this guide): https://web.stanford.edu/group/csp/cs21/csscheatsheet.pdf

GitHub:

- GitHub Command Cheatsheet: (included in this guide) https://education.github.com/git-cheat-sheet-education.pdf
- w3 Schools GitHub Tutorials: https://www.w3schools.com/git/git_remote_getstarted.asp?remote=github

GitHub Pages:

- Medium Article on GitHub Pages: https://abhiappmobiledeveloper.medium.com/github-pages-a-complete-tutorial-151cb6cade9c
- Alternative options for building out your website (other than HTML5up Templates): https://astrosites.github.io/building-options/
- A wonderful tutorial on how to edit your website step by step using exactly the same templates we are using can be found here: https://astrosites.github.io/tutorial/
- Website Building Blocks: https://astrosites.github.io/site-building-blocks/

GitHub
GIT CHEAT SHEET

Git is the free and open source distributed version control system that's responsible for everything GitHub related that happens locally on your computer. This cheat sheet features the most important and commonly used Git commands for easy reference.

INSTALLATION & GUIS

With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

GitHub for Windows
https://windows.github.com

GitHub for Mac
https://mac.github.com

For Linux and Solaris platforms, the latest release is available on the official Git web site.

Git for All Platforms
http://git-scm.com

SETUP

Configuring user information used across all local repositories

`git config --global user.name "[firstname lastname]"`
set a name that is identifiable for credit when review version history
`git config --global user.email "[valid-email]"`
set an email address that will be associated with each history marker
`git config --global color.ui auto`
set automatic command line coloring for Git for easy reviewing

SETUP & INIT

Configuring user information, initializing and cloning repositories

`git init`
initialize an existing directory as a Git repository
`git clone [url]`
retrieve an entire repository from a hosted location via URL

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

`git status`
show modified files in working directory, staged for your next commit
`git add [file]`
add a file as it looks now to your next commit (stage)
`git reset [file]`
unstage a file while retaining the changes in working directory
`git diff`
diff of what is changed but not staged
`git diff --staged`
diff of what is staged but not yet committed
`git commit -m "[descriptive message]"`
commit your staged content as a new commit snapshot

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

`git branch`
list your branches. a* will appear next to the currently active branch
`git branch [branch-name]`
create a new branch at the current commit
`git checkout`
switch to another branch and check it out into your working directory
`git merge [branch]`
merge the specified branch's history into the current one
`git log`
show all commits in the current branch's history

INSPECT & COMPARE

Examining logs, diffs and object information

git log
show the commit history for the currently active branch
git log branchB..branchA
show the commits on branchA that are not on branchB
git log --follow [file]
show the commits that changed file, even across renames
git diff branchB...branchA
show the diff of what is in branchA that is not in branchB
git show [SHA]
show any object in Git in human-readable format

SHARE & UPDATE

Retrieving updates from another repository and updating local repos

git remote add [alias] [url]
add a git URL as an alias
git fetch [alias]
fetch down all the branches from that Git remote
git merge [alias]/[branch]
merge a remote branch into your current branch to bring it up to date
git push [alias] [branch]
Transmit local branch commits to the remote repository branch
git pull
fetch and merge any commits from the tracking remote branch

TRACKING PATH CHANGES

Versioning file removes and path changes

git rm [file]
delete the file from project and stage the removal for commit
git mv [existing-path] [new-path]
change an existing file path and stage the move
git log --stat -M
show all commit logs with indication of any paths that moved

REWRITE HISTORY

Rewriting branches, updating commits and clearing history

git rebase [branch]
apply any commits of current branch ahead of specified one
git reset --hard [commit]
clear staging area, rewrite working tree from specified commit

IGNORING PATTERNS

Preventing unintentional staging or commiting of files

logs/ ***.notes** **pattern*/**
Save a file with desired patterns as .gitignore with either direct string matches or wildcard globs.
git config --global core.excludesfile [file]
system wide ignore pattern for all local repositories

TEMPORARY COMMITS

Temporarily store modified, tracked files in order to change branches

git stash
Save modified and staged changes
git stash list
list stack-order of stashed file changes
git stash pop
write working from top of stash stack
git stash drop
discard the changes from top of stash stack

GitHub Education

Teach and learn better, together. GitHub is free for students and teachers. Discounts available for other educational uses.

✉ **education@github.com**
education.github.com

Cascading Style Sheets Cheatsheet (CSS 1&2)

Font Properties

Font-Family
Changes the font family of certain words, sentences, paragraphs, etc.
P { font-family: "New Century Schoolbook", Times, serif; }

Font-Style
Changes text: normal, oblique, and italics.
H1 { font-style: oblique; }
P { font-style: normal; }

Font-Variant
Used to display font in normal or small-caps.
SPAN { font-variant: small-caps; }

Font-Weight
Used to specify the weight of the font.
H1 { font-weight: 800; } or P { font-weight: normal; }

Font-Size
Used to modify the size of the displayed font.
H1 { font-size: large; } or P { font-size: 12pt; }
LI { font-size: 90%; }
STRONG { font-size: larger; }

Font
Used to combine all properties of fonts.
P { font: italic bold 12pt/14pt Times, serif; }

Color and Background Properties

Color
Changes the color of text.
H1 { color: blue; } or H2 { color: #000080; }

Background-Color
Sets the background color of an element.
BODY { background-color: white; }
H1 { background-color: #000080; }

Background-Image
Sets the background image of an element.
BODY { background-image: url(/images/foo.gif); }
P { background-image: url(http://www.htmlhelp.com/bg.png); }

Background-Repeat
Determines how a specified background image is repeated. The repeat-x value will repeat the image horizontally while the repeat-y value will repeat the image vertically.
BODY { background: white url(candybar.gif);
background-repeat: repeat-x; }

Background-Attachment
Determines if a specified background image will scroll with the content or be fixed with regard to the canvas.
BODY { background: white url(candybar.gif);
background-attachment: fixed; }

Background
Used to combine all properties of background.
BODY { background: white url(http://www.htmlhelp.com/foo.gif); }
BLOCKQUOTE { background: #7fffd4; }
P { background: url(../backgrounds/pawn.png) #f0f8ff fixed; }
TABLE { background: red url(leaves.jpg) no-repeat bottom right; }

CREDIT: MARK BRANOM
STANFORD UNIVERSITY

Text Properties

Word-Spacing
Defines an additional amount of space between words.
P EM { word-spacing: 0.4em; }
P.note { word-spacing: -0.2em; }

Letter-Spacing
Defines an additional amount of space between characters.
H1 { letter-spacing: 0.1em; }
P.note { letter-spacing: -0.1em; }

Text-Decoration
Allows text to be decorated through one of five properties: underline, overline, line-through, blink, none.
A:link, A:visited, A:active { text-decoration: none; }

Vertical-Align
Used to alter the vertical positioning of an inline element, relative to its parent element or to the element's line.
IMG.middle { vertical-align: middle; }
IMG { vertical-align: 50%; }
.exponent { vertical-align: super; }

Text-Transform
Allows for capitalizing the first letter of each word (capitalize), capitalizing all letters of a word (uppercase), using all small letters in each word(lowercase), and the inital value(none).
H1 { text-transform: uppercase; }
H2 { text-transform: capitalize; }

Text-Align
Used to justify text left, center, right, and justify.
H1 { text-align: center; }
P.newspaper { text-align: justify; }

Text-Indent
Used to specify the amount of indentation prior to the first line of text.
P { text-indent: 5em; }

Line-Height
Used to control the spacing between baselines of text.
P { line-height: 200%; }

Classification Properties

List-Style-Type
Specifies the type of list-item marker, and is used if list-style-image is none or if image loading is turned off.
LI.square { list-style-type: square; }
UL.plain { list-style-type: none; }
OL { list-style-type: upper-alpha; } /* A B C D E etc. */
OL OL { list-style-type: decimal; } /* 1 2 3 4 5 etc. */
OL OL OL { list-style-type: lower-roman; } /* i ii iii iv v etc. */

List-Style-Image
Specifies the image that will be used as list-item marker when image loading is turned on, replacing the marker specified in the list-style-type property.
UL.check { list-style-image: url(/LI-markers/checkmark.gif); }
UL LI.x { list-style-image: url(x.png); }

List-Style-Position
Determines where the marker is placed in regard to the list item. If the value *inside* is used, the lines will wrap under the marker instead of being indented. *outside* is default.
UL { list-style-position: inside; }

Cascading Style Sheets Cheatsheet (CSS 1&2)

Box Properties

Margin-Top
Sets the top margin of an element by specifying a length or a percentage.
```
BODY { margin-top: 5pt; }
```

Margin-Right
Sets the right margin of an element by specifying a length or a percentage.
```
P.narrow { margin-right: 50%; }
```

Margin-Bottom
sets the bottom margin of an element by specifying a length or a percentage.
```
DT { margin-bottom: 3em; }
```

Margin-Left
sets the left margin of an element by specifying a length or a percentage.
```
ADDRESS { margin-left: 50%; }
```

Margin
Sets the margins of an element by specifying top, bottom, left and right margins -- all either specifying length or percentage.
```
BODY { margin: 5em; } /* all margins 5em */
P { margin: 2em 4em; } /* top & bottom 2em, left & right 4em */
DIV { margin: 1em 2em 3em 4em; }
    /* top margin 1em, right 2em, bottom 3em, left 4em */
```

Padding-Top
Describes the amount of space between the top border and the content of the selector.
```
P { padding-top: 20%; }
```

Padding-Right
Describes the amount of space between the right border and the content of the selector.
```
P { padding-right: 20 px; }
```

Padding-Bottom
Describes the amount of space between the bottom border and the content of the selector.
```
P { padding-bottom: 5 em; }
```

Padding-Left
Describes the amount of space between the left border and the content of the selector.
```
P { padding-left: 15 pt; }
```

Padding
Shorthand for the padding-top, padding-right, padding-bottom, and padding-left properties.
```
BLOCKQUOTE { padding: 2em 4em 5em 4em; }
```

Border-Top-Width
Used to specify the width of an element's top border.
```
P { border-top: 20%; }
```

Border-Right-Width
Used to specify the width of an element's right border.
```
P { border-right: 20%; }
```

Border-Bottom-Width
Used to specify the width of an element's bottom border.
```
P { border-bottom: 20%; }
```

Border-Left-Width
Used to specify the width of an element's left border.
```
P { border-left: 20%; }
```

Border-Width
Used to set the width of an element's border (either all borders, or specifying top border, right border, bottom border, left border).
```
P { border-width: 20%; }
P { border-width: 10px 5px 10px 5px; }
```

Border-Color
Used to set the color of an element's border.
```
P { border-color: #00000; }
```

Border-Style
Sets style of a border - none, dotted, dashed, solid, double.
```
P { border-style: dotted; }
```

Border-Top
Sets the width, style, and color of an element's top border.
```
P { border-top: 10px, red, double; }
```

Border-Right
Sets the width, style, and color of an element's right border.
```
P { border-right: 10px, red, double; }
```

Border-Bottom
Sets the width, style, and color of an element's bottom border.
```
P { border-bottom: 10px, red, double; }
```

Border-Left
Sets the width, style, and color of an element's left border.
```
P { border-left: 10px, red, double; }
```

Border
Sets the width, style, and color of an element's border.
```
P { border: 10px, red, double; }
```

Width
Each block-level or replaced element can be given a width, specified as a length, a percentage, or as auto.
```
P { width: 15px; }
H1 { width: 35%; }
.foo { width: auto; }
```

Height
Each block-level or replaced element can be given a height, specified as a length or as auto.
```
P { height: 15px; }
H1 { height: 35%; }
.foo { height: auto; }
```

Float
Allows text to wrap around an element (left, right, none).
```
P { float: left; }
H1 { float: right; }
.foo { float: none; }
```

Clear
Specifies whether an element allows floating elements to its sides (left, right, none).
```
P { clear: left; }
H1 { clear: right; }
.foo { clear: none; }
```

CREDIT: MARK BRANOM
STANFORD UNIVERSITY