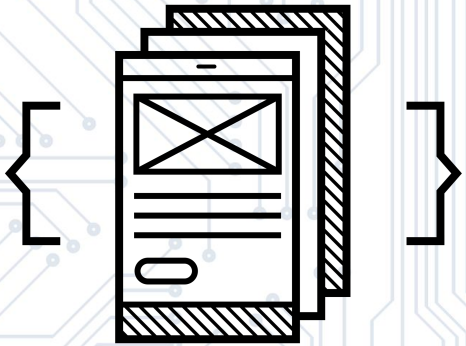
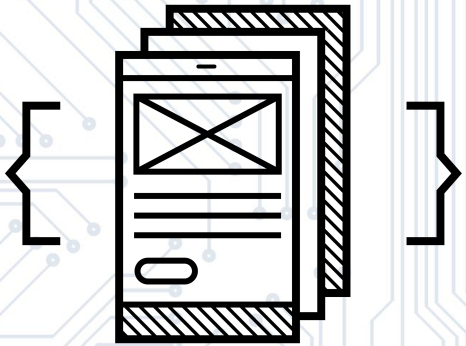


Positioning Elements on the Web

UX/UI Design Boot Camp

Lesson 17.3

45 minutes



Positioning Elements on the Web

UX/UI Design Boot Camp

Lesson 17.3

Today's Objectives

By the end of class today, you will:

Position HTML elements using CSS floats and automatic margins.

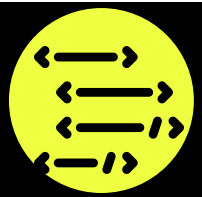
Position inline, block, and inline-block HTML elements to build webpages.

Create HTML divs that overlap by positioning elements absolutely with CSS.

Code your first webpage and host it via GitHub Pages to become your first website.

Front-End Development Units

Today we'll continue with the basics of front-end development.

 UI Design Process	 Dev GRIDS	 **HTML**	 **HTML Structure**	 **CSS3**	 Bootstrap 4	 JavaScript/ jQuery
 **Git/GitHub/ GitHub Pages**	 Responsive Web Design	 Atomic Design Mobile First	 Debugging	 Templates, Layout, and Pages	 Web Prototyping	 UI Testing and A/B Testing

CSS Float and Display Properties

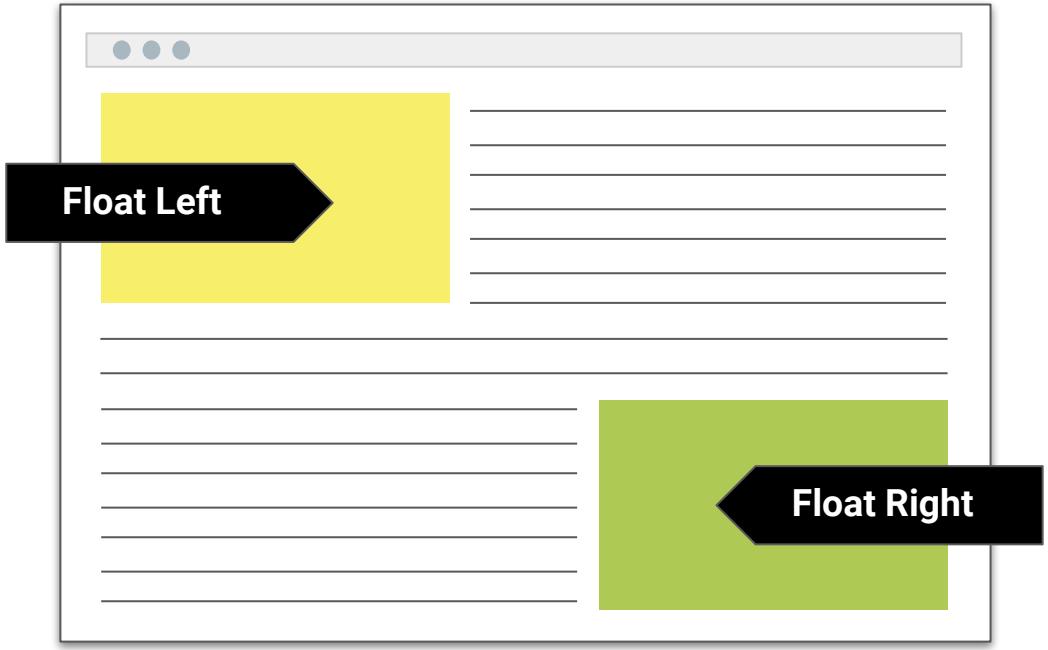
CSS Float Property

Float CSS

Float CSS

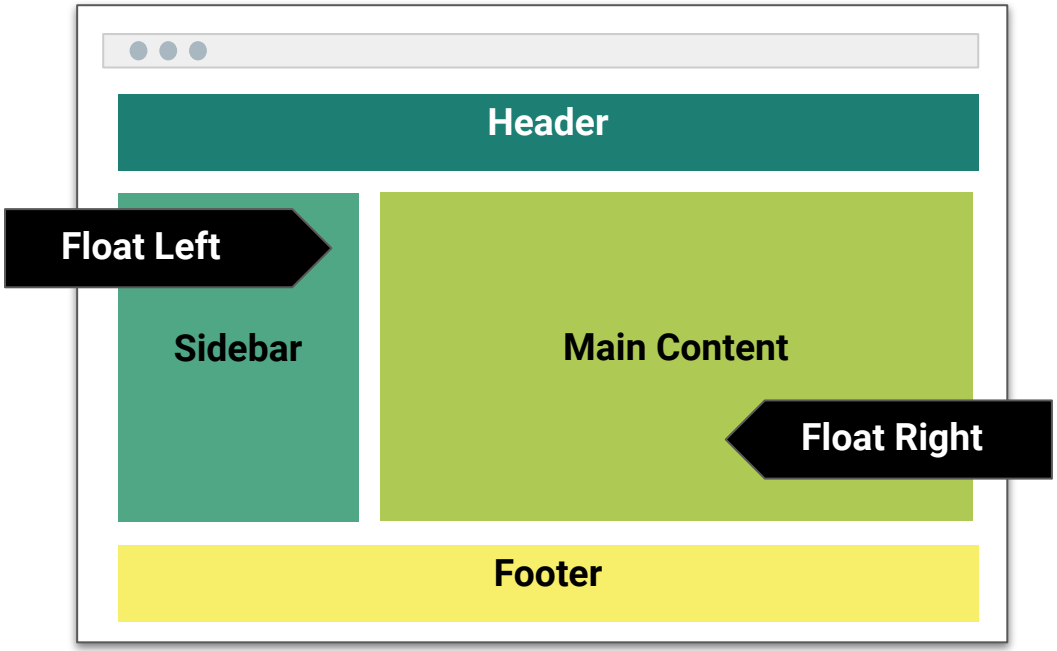
The **float CSS** property places an element on the left or right side of its container, allowing text and inline elements to wrap around it.

Floats were designed to wrap text around images, but can also be used to make an element move to the farthest left or right in its container.



Float CSS

Float can be used to build containers that line up next each other but it can be tricky.



CSS

```
#sidebar {
    float: left;
}
#main-content {
    float: right;
}
```


Time For a Quick Video
Floats in CSS

CSS Display Property

CSS Display Property

The display property is the most important CSS property for controlling layout.

As a beginner, **most problems you encounter** early on will be caused by the display property and trying to position elements in and around each other.

The display property specifies if/how an element is displayed.

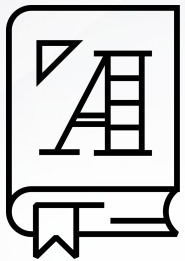
Each behaves differently from others and understanding the differences will save you both time and frustration.

There are many different display properties. Here are the basic ones we will be working with first.

We will be covering the more advanced display properties next week.

```
display: block; // Sets the element to display block.
display: inline; // Sets the element to display as a inline element.
display: none; // hides a element on your html page.
display: inline-block; // sets the element to display as a inline-block element.
```

HTML:
Block-Level Elements



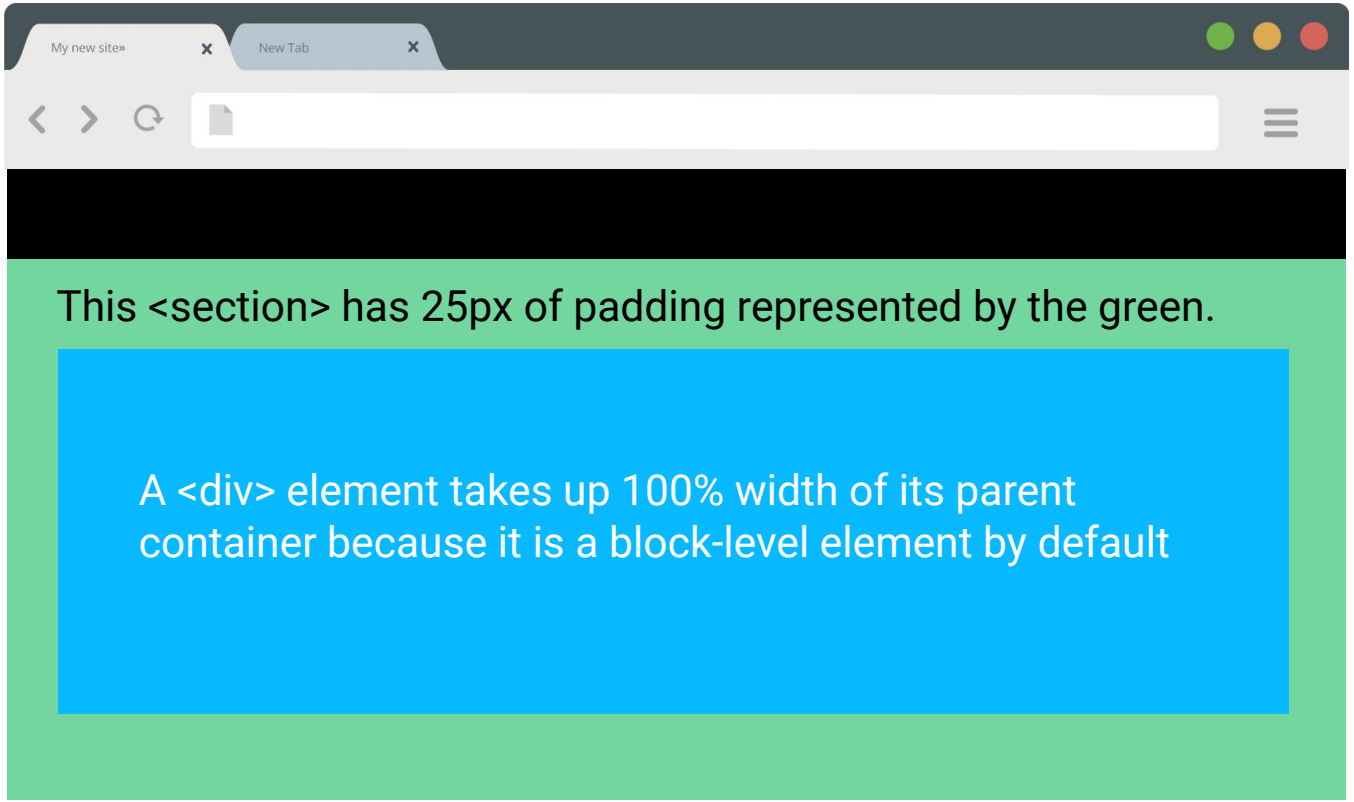
HTML: Block-level elements always start on a new line and take up the full width of its parent container (stretches out to the left and right as far as it can).

Block-level elements = `<div>`, `<section>`, `<p>`, `<h1-h6>`, `<ol>`, `<ul>`, `<form>`, `<blockquote>`, etc.
List of block-level elements: https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements

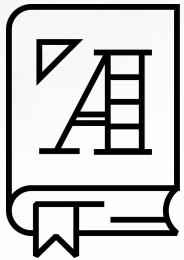
Block-Level Elements

Block-level elements respect the margin and padding of their parent containers.

Block-level elements are used to contain inline elements.



HTML:
Inline Elements



HTML: Inline elements only occupy the space bounded by the tags **defining** the **element**, instead of breaking the flow of the content.

Inline HTML elements examples = `<button>`, `<input>`, `<label>`, `<img>`, `<textarea>`, `<span>`, `etc.`
List of inline elements: https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements

Inline Elements

Inline elements only take up as much space as needed and will sit beside other inline elements. Below, we see an inline element, a span, side by side with text (also inline).

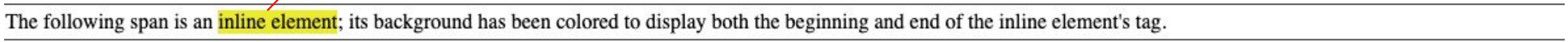
HTML

```
<div>The following span is an <span class="highlight">inline
element</span>;
its background has been colored to display both the beginning
and end of
the inline element's tag.</div>
```

CSS

```
body {
  margin: 0;
  padding: 4px;
  margin-top: 15px;
  border: 1px solid #333;
}

.highlight {
  background-color:#ee3;
}
```



https://codepen.io/TrilogyEdu/pen/NZyemG

HTML and CSS:
Inline-Block Elements

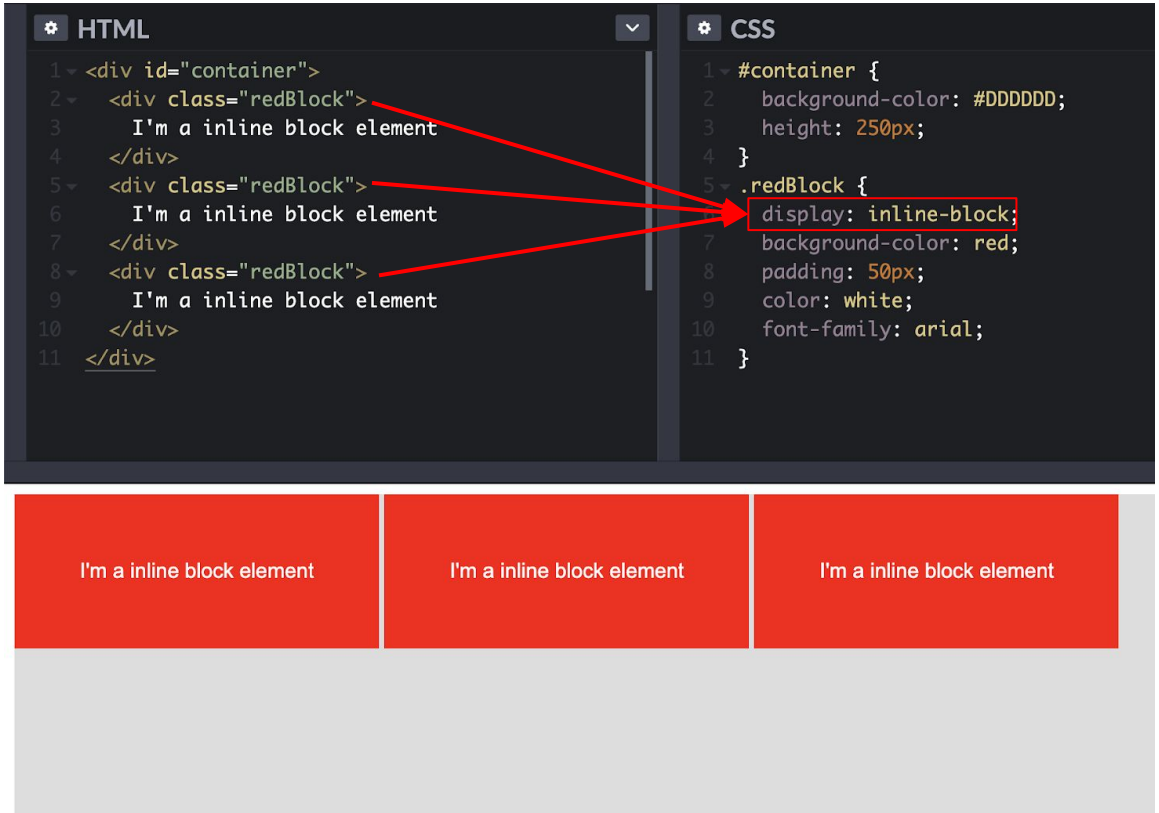
display: inline-block; sets an HTML element to display as an inline-block container via CSS class or ID.

The element itself is formatted as an inline element, but you can apply height and width values.

display: inline-block Elements

inline-block elements display as inline so they line up next to each other.

This could be a navigation bar or a series of cards containing content.



Inline-Block Elements

Inline-block elements can be used to create two-column layouts.

Be warned! With this method, you will not be able to have the two elements fill 100% of a screen.

There will always be a small gap between.

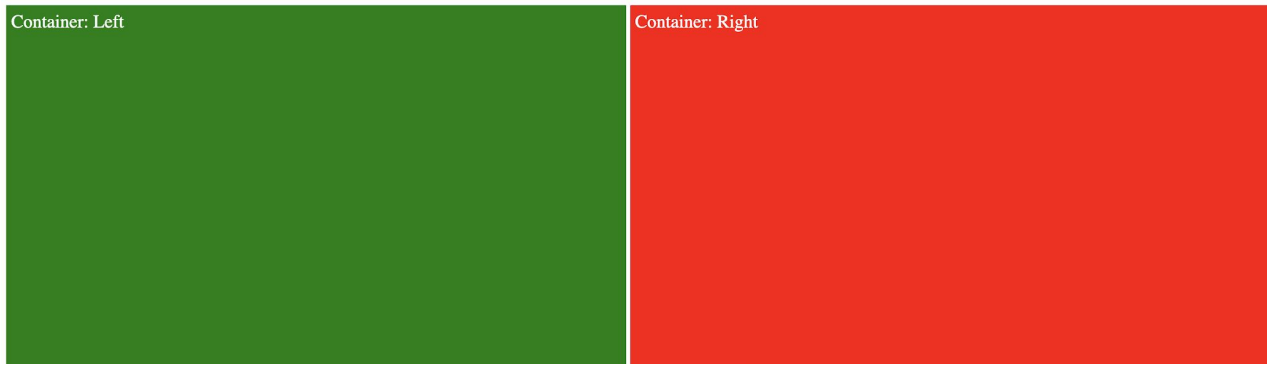
HTML

```
<section id="container">
  <div class="containerBlock left">
    Container: Left
  </div>
  <div class="containerBlock right">
    Container: Right
  </div>
</section>
```

CSS

```
.containerBlock {
  display: inline-block;
  height: 500px;
  font-size: 18px;
  padding: 5px;
}
.left {
  width: 40%;
  background-color: green;
  color: white;
}
.right {
  width: 58%;
  background-color: red;
```

JS



https://codepen.io/crbecker1/pen/VweJPYW

CSS Display Review

CSS Display Recap

HTML
Block-level elements fill 100% of the width of their containers.

Block Element

Block Element

Block Element

HTML
Inline elements only take up the space they need and line up next to other inline elements. You cannot give inline elements height or width.

Inline elements display next to each other. This span sits next to other elements and doesn't take up space.

HTML/CSS
Inline-block elements line up next to each other like inline elements.

You can give inline-block elements heights and widths.

Inline-Block

Inline-Block

Inline-Block

CSS Width-Based Percentages

CSS Width-Based Percentages Using Relative Units

We have referenced content that fills 100% of the width of its container. We can apply percentage-based widths and margins to block-level elements.

It's worth noting, however, how the box model calculates widths on a page, as setting them correctly is necessary for successful layouts.

HTML

```
<div class="width">100% Width Issues</div>
```

CSS

```
div {
  width: 100%;
  margin: 5%;
  padding: 1%;
  background-color: red;
  height: 250px;
  color: white;
  font-family: arial, helvetica;
  font-size: 22px;
}
```

https://codepen.io/TrilogyEdu/pen/zVWpgE

CSS Width-Based Percentages Explained

Our content is breaking out of its container because we have it set to 100% width and 5% margin (left and right) and 1% padding.

```
div {
  width: 100%;
  margin: 5%;
  padding: 1%;
  background-color: red;
  height: 250px;
  color: white;
  font-family: arial, helvetica;
  font-size: 22px;
}
```

CSS Width-Based Percentages Explained

Margin and padding applies itself to the left, right, top, and bottom of an element.

Our 5% margin really takes up 5% on the left and 5% on the right (10% total), causing our div to break. The same applies for padding.

```
div {
  width: 100%;
  margin: 5%;
  padding: 1%;
  background-color: red;
  height: 250px;
  color: white;
  font-family: arial, helvetica;
  font-size: 22px;
}
```

CSS Width-Based Percentages Explained

To get our layout to fill up 100% of its width, we need to make sure our box model equals 100%.

We set our width to take up 88% of our layout.

We set our margin to be 5%. Because the margin includes the left and the right side, a margin of 5% is really equal to 10% of the total width.

Lastly, we have 1% padding set as well. The left and the right of our div take up 2% of the space with padding.

88% width + 10% margin + 2% padding = 100% width

HTML

```
<div class="width">100%
Width Issues</div>
```

CSS

```
div {
  width: 88%;
  margin: 5%;
  padding: 1%;
  background-color: red;
  height: 250px;
  color: white;
  font-family: arial, helvetica;
  font-size: 22px;
}
```

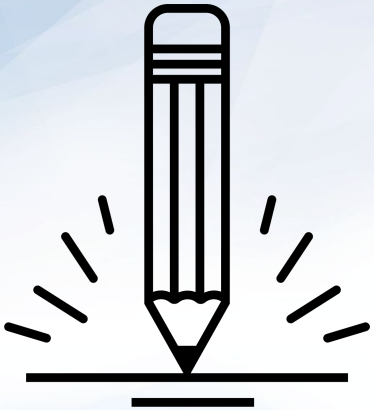
Let's Review: Layout Positioning

What is an HTML inline element?

What is a HTML block element?

What is an HTML/CSS inline-block element?

What is a CSS float used for?



Activity:

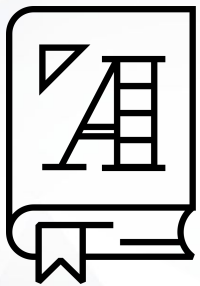
Practice with CSS Positioning

(Instructions sent via Slack)

Suggested Time:
20 minutes

Time's Up! Let's Review.

CSS Position: Five Properties



In CSS, the **position property** is used to create complex layouts where you need items placed in a specific area.

CSS — Position: Static

We position the element according to the normal flow of the document. Top, right, bottom, left and z-index have no effect on elements that are not positioned.

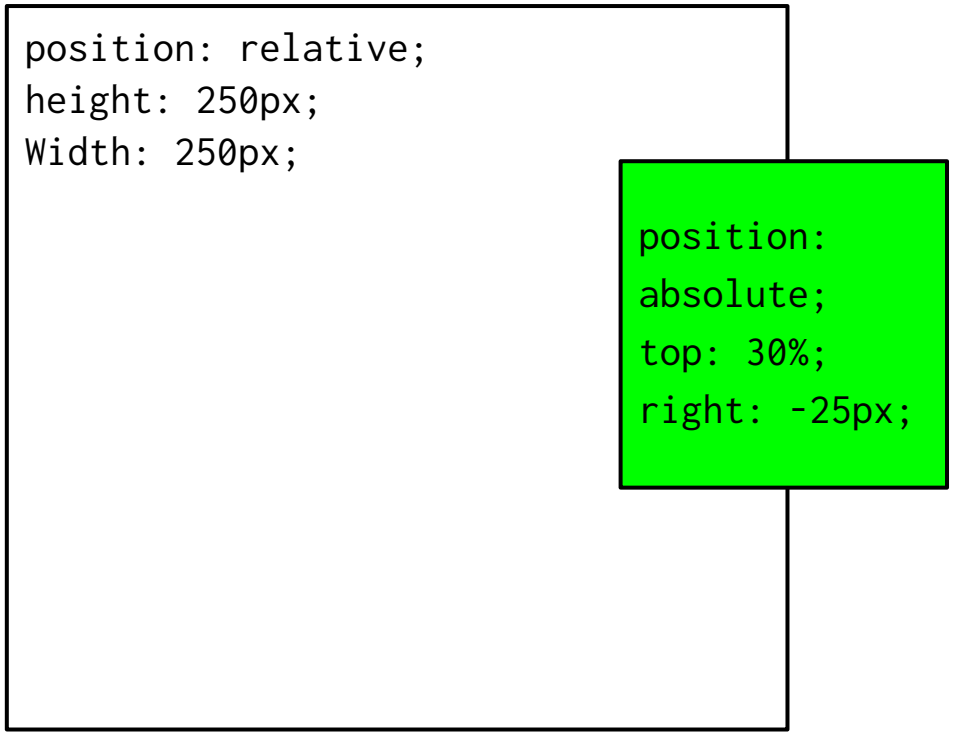
Static elements are the default value of all elements.

CSS — Position: Relative

Relative positioning allows you to control the position of your element using top, left, right, and bottom.

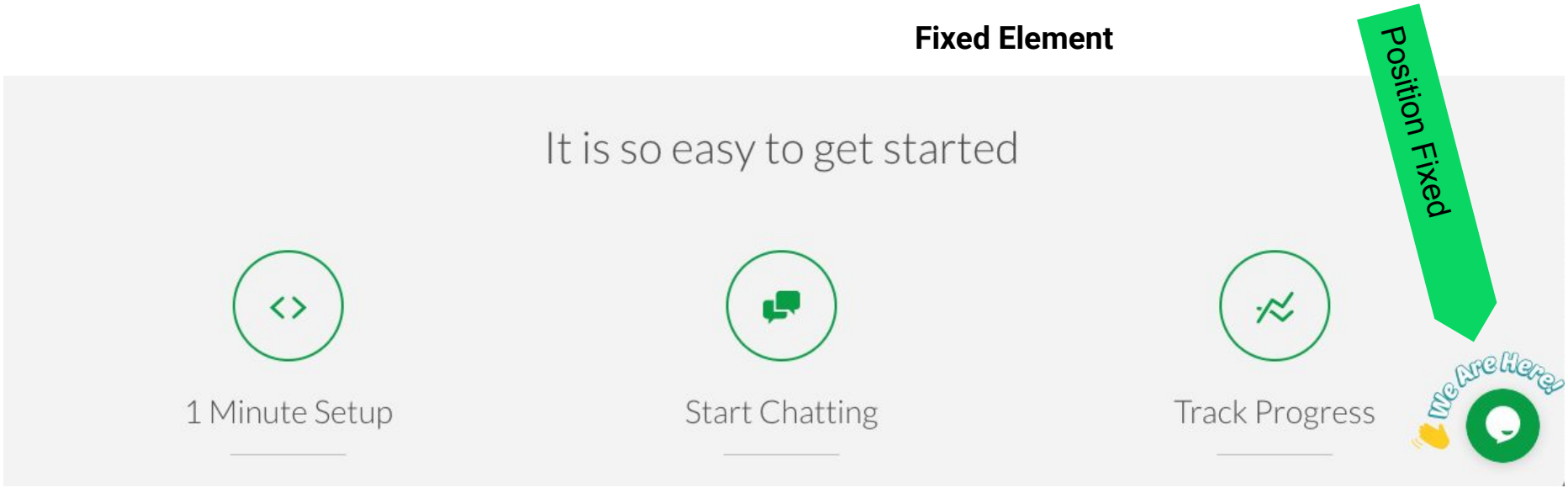
More important, relative positioning is used to contain absolutely positioned elements.

(Absolutely positioned elements are positioned according to their nearest positioned container. More on this later.)



CSS — Position: Fixed

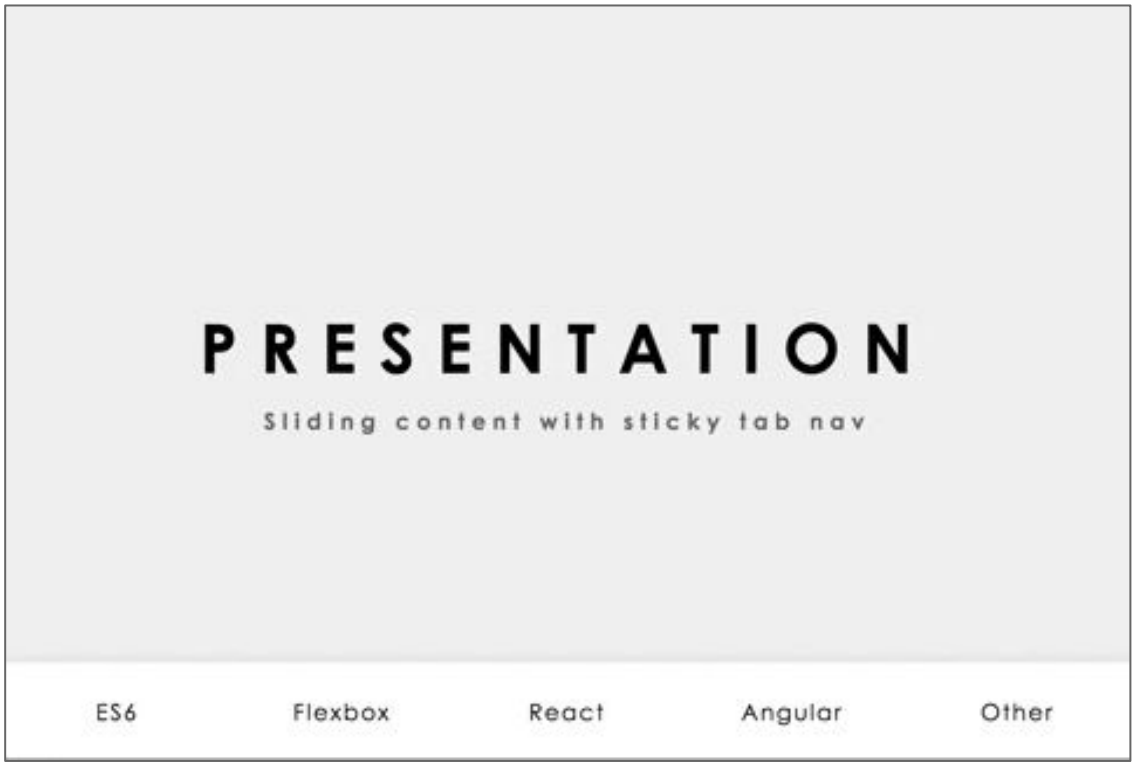
The element is removed from the normal document flow. It is positioned to the viewport of the browser and will always stay where you put it even if the user scrolls down the page.



CSS — Position: Sticky

This is the newest position property.

An element is positioned according to the normal flow of the document, and then sticks to the top of browser when a certain scroll height is met.

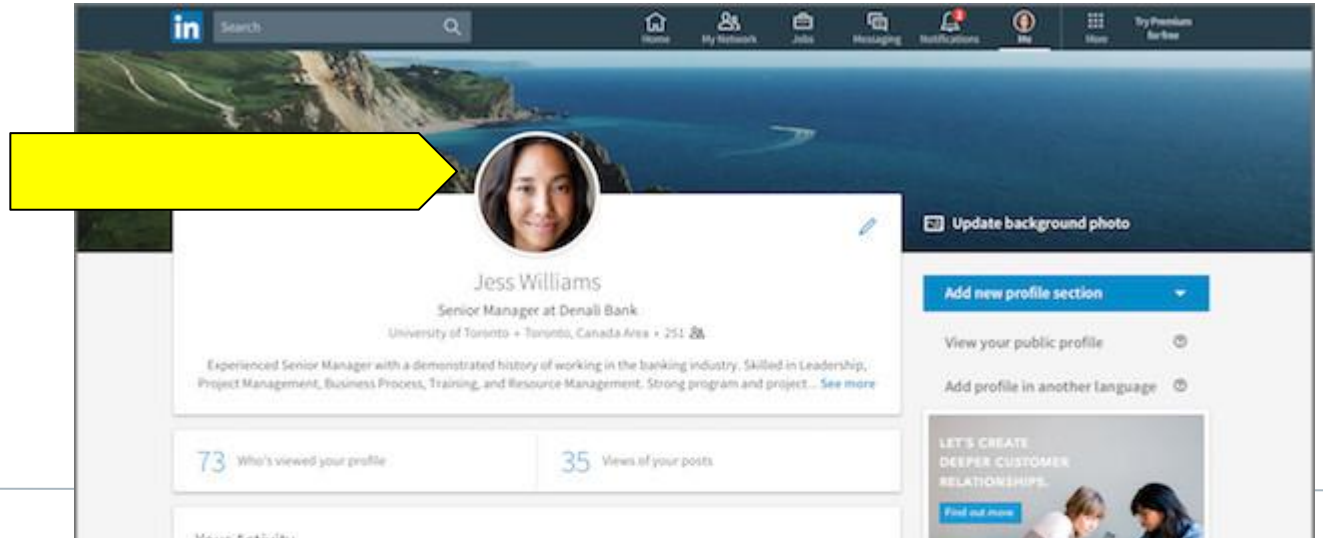


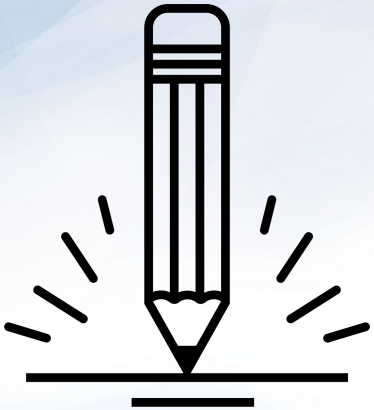
CSS — Position: Absolute

Absolute elements are removed from the document flow and are positioned relative to their nearest-positioned ancestor.

If a parent container has any type of position property applied to it (except static), then absolutely positioned elements will be contained and positioned inside it. If no element has a stated position, then it will be positioned to the browser viewport or window.

This element is absolutely positioned, so it can be pushed above the div to create this look.





Activity:
CSS Positioning With Relative, Absolute, and Fixed Positioning

(Instructions sent via Slack)

Suggested Time:
20 minutes

Time's Up! Let's Review.


Break

Challenge:

Build a One-Page Website

You will practice building a basic webpage and then upload it to GitHub Pages.

(Instructions sent via Slack)

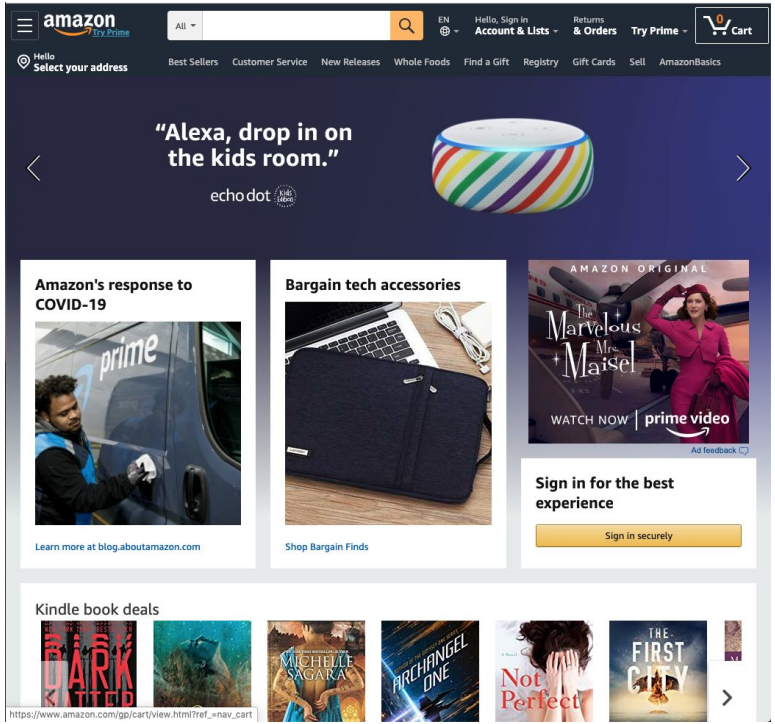
Suggested Time:
50 minutes

Time's Up! Let's Review.

Webpage vs Website

Webpage vs. Website

vs.



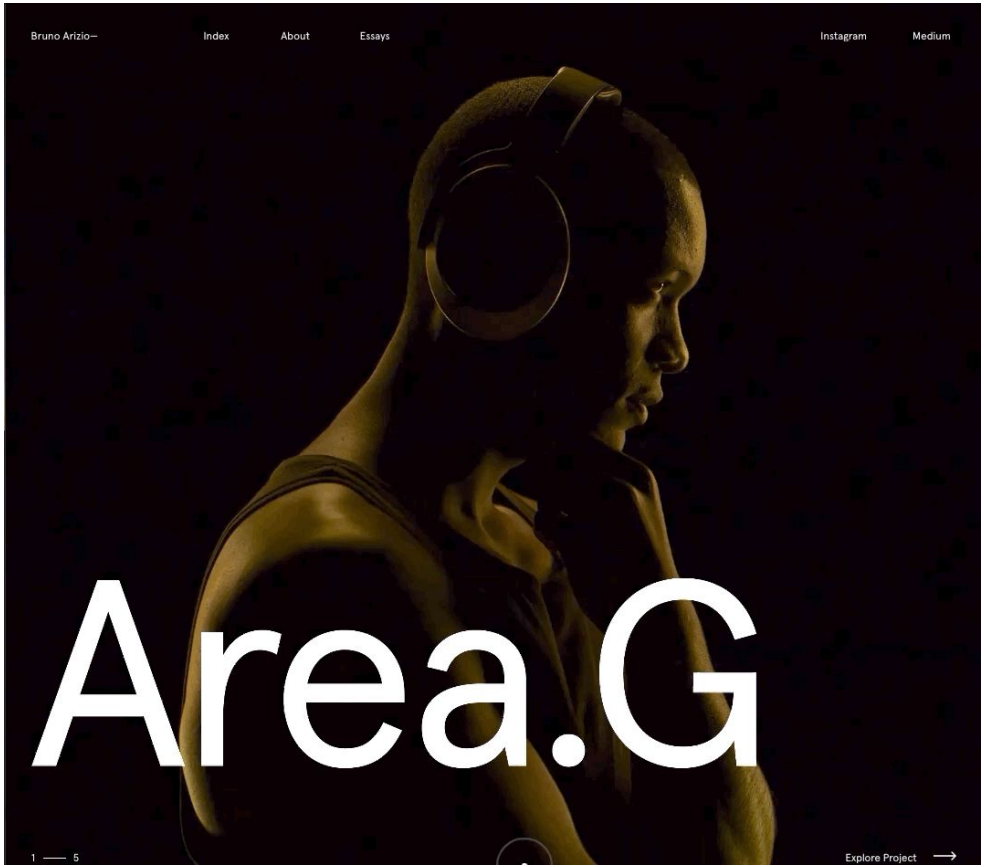
https://ourstoriesin.com/ and https://www.amazon.com

A Webpage Is ...

A domain that is connected to the internet that displays a single HTML document that can be viewed in a web browser.

https://brunoarizio.com/ > homepage (index.html)

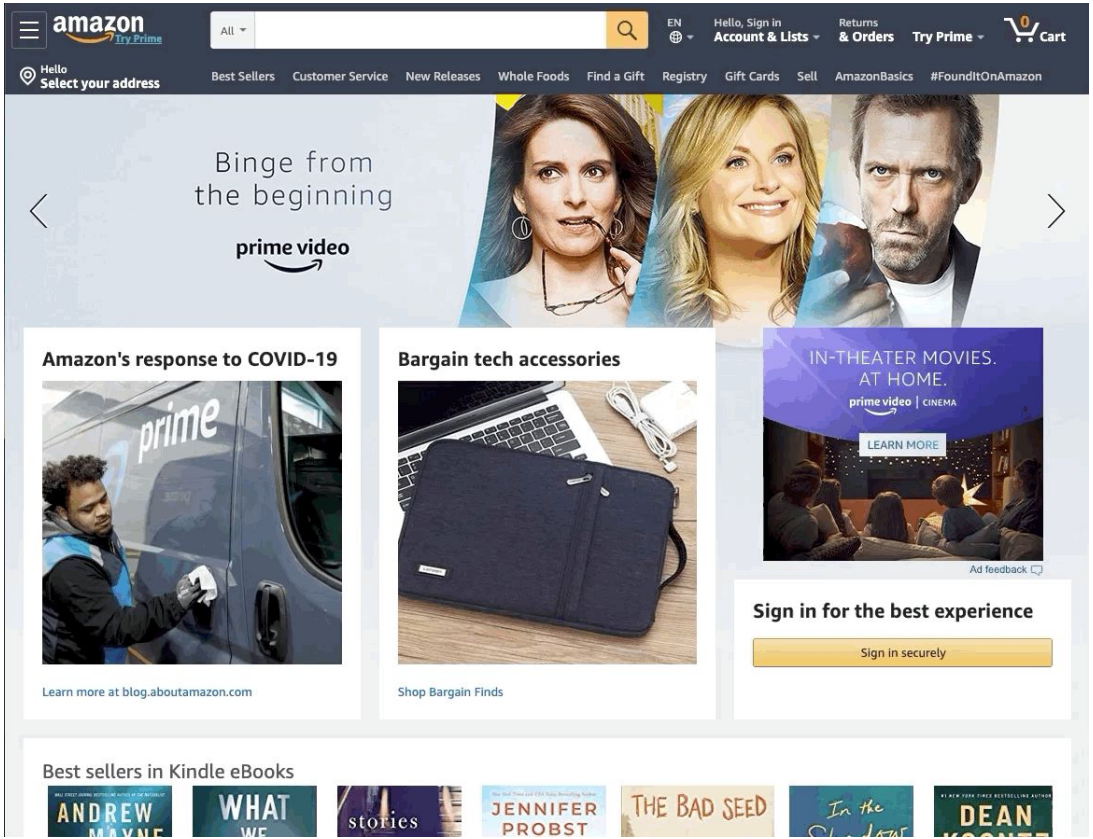
- https://brunoarizio.com/about > about page (about.html)
- https://brunoarizio.com/essays > essays page (essays.html)



A Website is ...

A domain connected to the internet that maintains one or more pages on the World Wide Web.

For example,
Amazon's website has 1000's of webpages.



https://www.amazon.com

GitHub and Version Control

Git/GitHub is a distributed **version control** system for tracking changes in source code during software development.

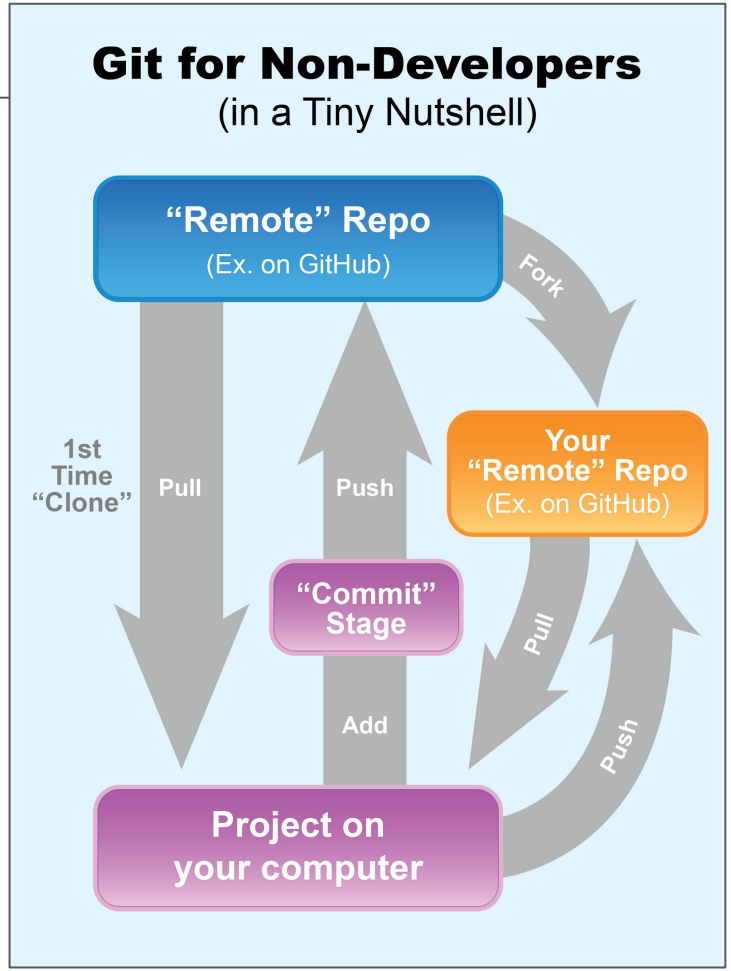
It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.

https://github.com/

Why Git?

It is a version control system that allows team members (developers, UX, UI) to collaborate without overriding each other's work.

Git keeps a timeline and allows your code repo to be protected and fixed quickly.



https://git-scm.com/book/en/v2/Git-Internals-Git-References

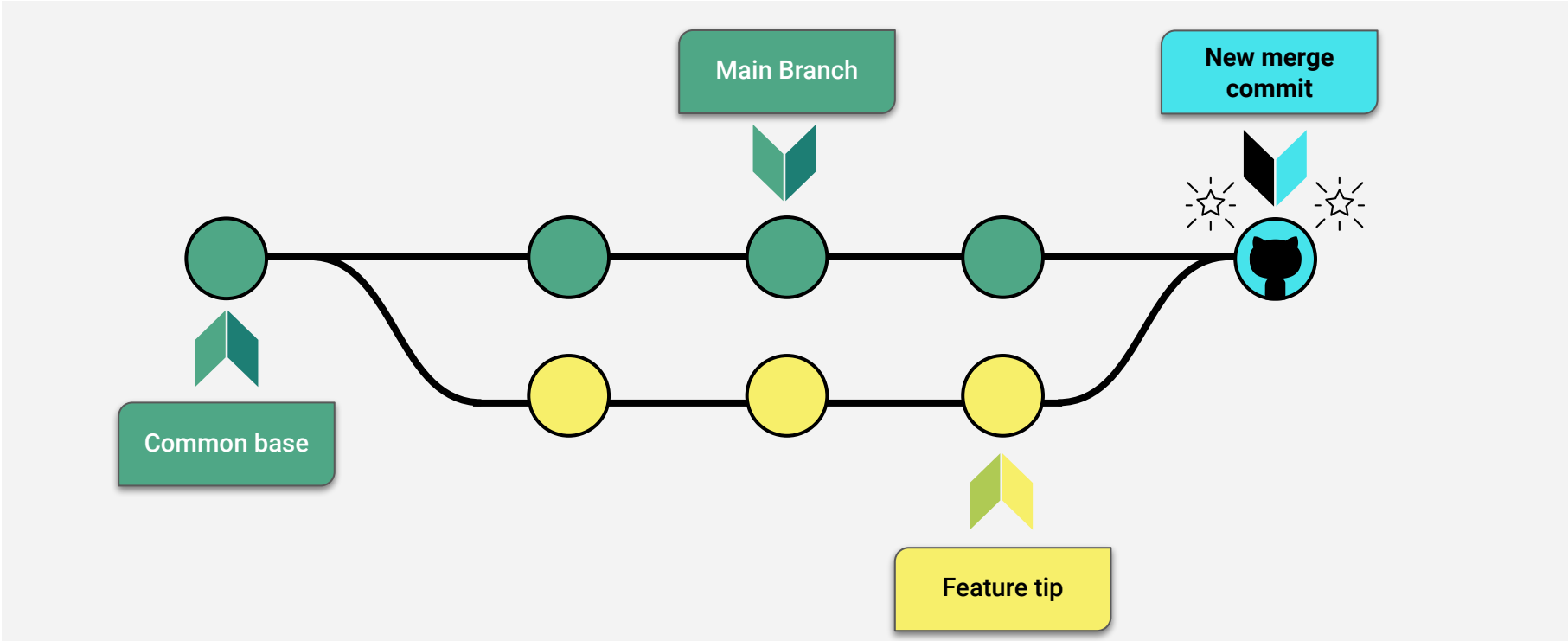
https://www.c-sharpcorner.com/article/what-is-version-control-git-vs-tfs/

https://medium.com/shyp-design/managing-style-guides-at-shyp-c217116c8126

https://www.atlassian.com/git/tutorials/why-git

What Is Version Control?

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

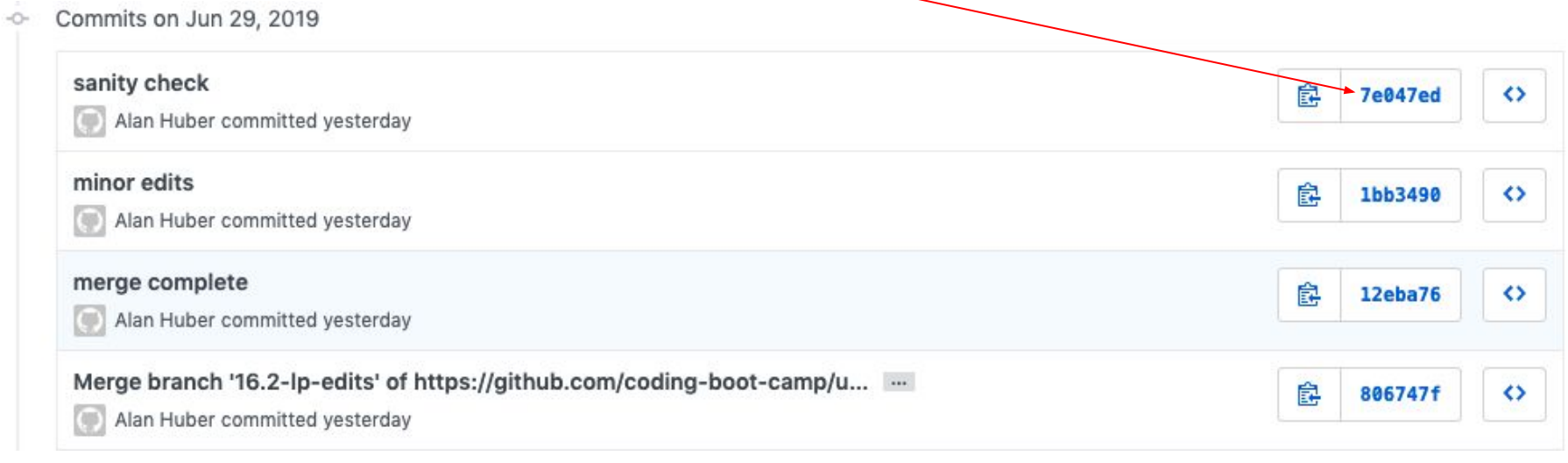


What Is Version Control?

Git tracks changes to any file that is being tracked by Git.

Saved versions of these changes are called commits. They represent individual changes in a file that you then pushed to GitHub.

Commits are tracked with a number



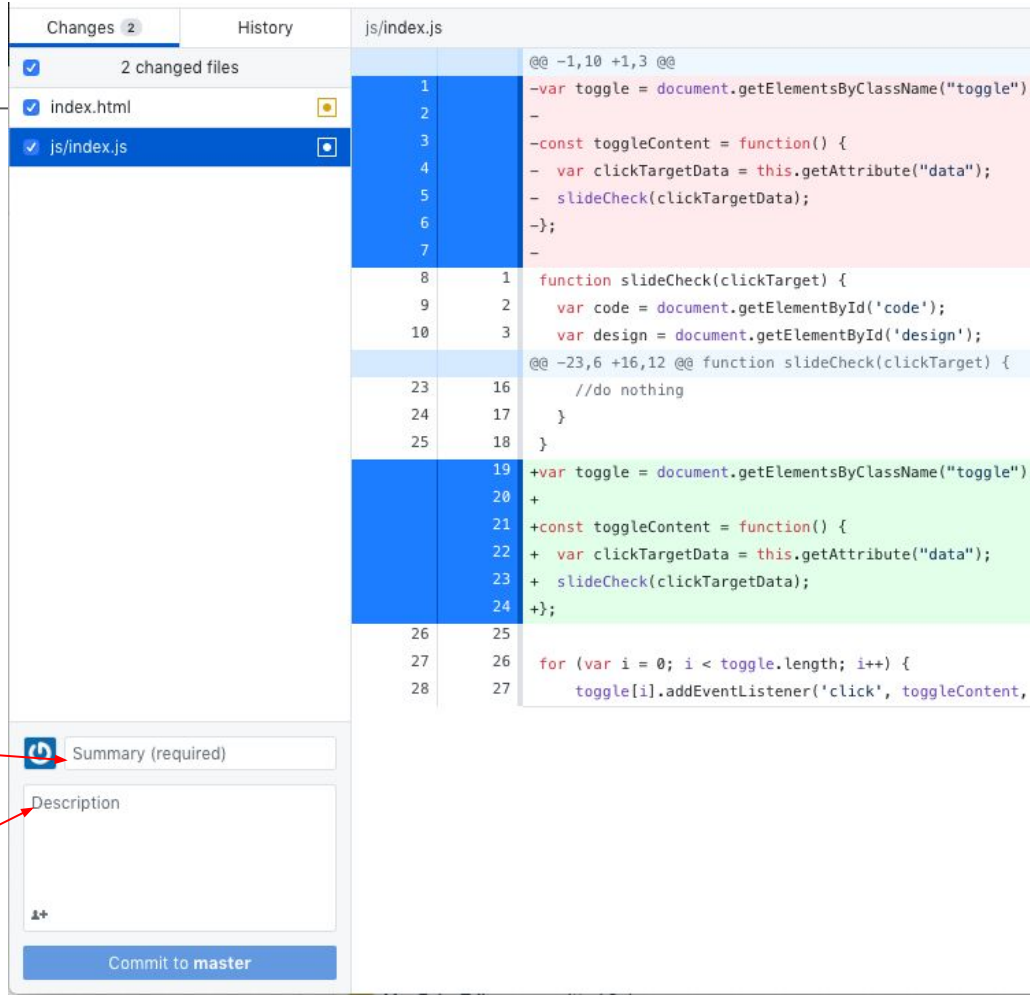
What Is Version Control?

GitHub/GitHub Desktop tracks individual changes in your file and allows you to roll back to previous commits if you make a mistake.

Git is an industry standard for any kind of development job. But it is gaining popularity in other industries as well.

When **commits** are created, they are given descriptions so you know what you changed.

Give them descriptive names!



UI Developer

Web Advanced—Irvine CA, 92618
$70,000—89,000 (DOE) a year Full-time,

Apply Now

Required

- Expert in Adobe XD and Photoshop
- Expert in at least one wireframing/prototyping software such as InVision
- 4 years experience in providing UX across large-scale websites
- Excellent writing and communication skills
- Conceptual understanding of the technical implications of UX/UI decisions
- Proficient in Google Analytics

Desired

Development Software Experience

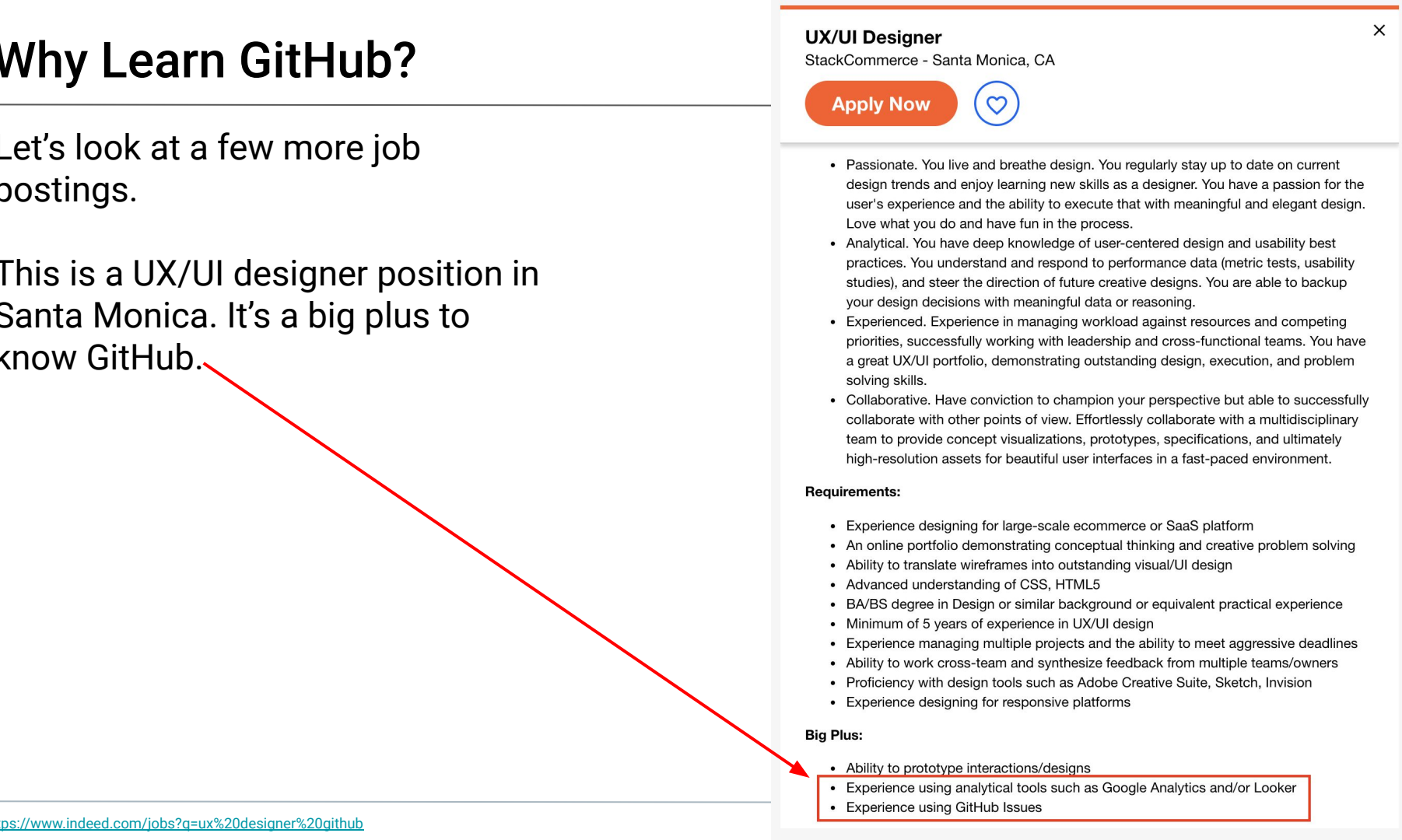
- Expertise with Git + GitHub Version control.
- Experience with tools such as HotJar, UserTesting.co, Visual Website Optimizer
- Experience working with clients on a variety of projects
- Experience as part of a team that including front and back end developers
- Proficiency in design review softwares such as Zeplin, and project management softwares such as Jira.
- Some HTML/CSS knowledge/coding ability

https://www.indeed.com/jobs?q=ux%20designer%20github

Why Learn GitHub?

Let's look at a few more job postings.

This is a UX/UI designer position in Santa Monica. It's a big plus to know GitHub.



UX/UI Designer
StackCommerce - Santa Monica, CA

Apply Now

- Passionate. You live and breathe design. You regularly stay up to date on current design trends and enjoy learning new skills as a designer. You have a passion for the user's experience and the ability to execute that with meaningful and elegant design. Love what you do and have fun in the process.
- Analytical. You have deep knowledge of user-centered design and usability best practices. You understand and respond to performance data (metric tests, usability studies), and steer the direction of future creative designs. You are able to backup your design decisions with meaningful data or reasoning.
- Experienced. Experience in managing workload against resources and competing priorities, successfully working with leadership and cross-functional teams. You have a great UX/UI portfolio, demonstrating outstanding design, execution, and problem solving skills.
- Collaborative. Have conviction to champion your perspective but able to successfully collaborate with other points of view. Effortlessly collaborate with a multidisciplinary team to provide concept visualizations, prototypes, specifications, and ultimately high-resolution assets for beautiful user interfaces in a fast-paced environment.

Requirements:

- Experience designing for large-scale ecommerce or SaaS platform
- An online portfolio demonstrating conceptual thinking and creative problem solving
- Ability to translate wireframes into outstanding visual/UI design
- Advanced understanding of CSS, HTML5
- BA/BS degree in Design or similar background or equivalent practical experience
- Minimum of 5 years of experience in UX/UI design
- Experience managing multiple projects and the ability to meet aggressive deadlines
- Ability to work cross-team and synthesize feedback from multiple teams/owners
- Proficiency with design tools such as Adobe Creative Suite, Sketch, Invision
- Experience designing for responsive platforms

Big Plus:

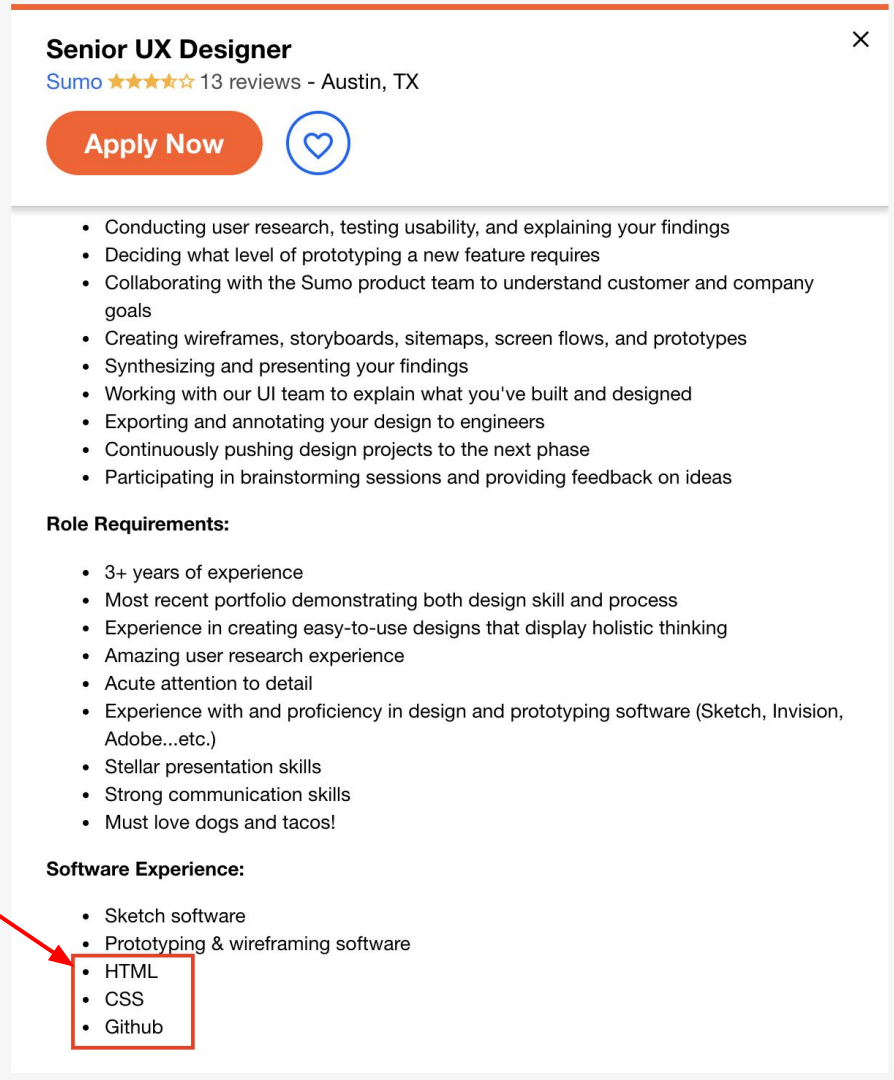
- Ability to prototype interactions/designs
- Experience using analytical tools such as Google Analytics and/or Looker
- Experience using GitHub Issues

https://www.indeed.com/jobs?q=ux%20designer%20github

Why Learn GitHub?

Here is a senior UX designer role. Once again, they specifically ask for knowledge of HTML, CSS, and use of GitHub.

Many UX jobs are for websites, making understanding the tools essential to excelling at the job.



Senior UX Designer

Sumo ★★★★☆ 13 reviews - Austin, TX

Apply Now

- Conducting user research, testing usability, and explaining your findings
- Deciding what level of prototyping a new feature requires
- Collaborating with the Sumo product team to understand customer and company goals
- Creating wireframes, storyboards, sitemaps, screen flows, and prototypes
- Synthesizing and presenting your findings
- Working with our UI team to explain what you've built and designed
- Exporting and annotating your design to engineers
- Continuously pushing design projects to the next phase
- Participating in brainstorming sessions and providing feedback on ideas

Role Requirements:

- 3+ years of experience
- Most recent portfolio demonstrating both design skill and process
- Experience in creating easy-to-use designs that display holistic thinking
- Amazing user research experience
- Acute attention to detail
- Experience with and proficiency in design and prototyping software (Sketch, Invision, Adobe...etc.)
- Stellar presentation skills
- Strong communication skills
- Must love dogs and tacos!

Software Experience:

- Sketch software
- Prototyping & wireframing software
- HTML
- CSS
- Github

https://www.indeed.com/jobs?q=ux%20designer%20github

Git: Client Graphical Software

Git Client Software options:

A Git Client GUI that offers a visual representation of your repositories.

Sourcetree

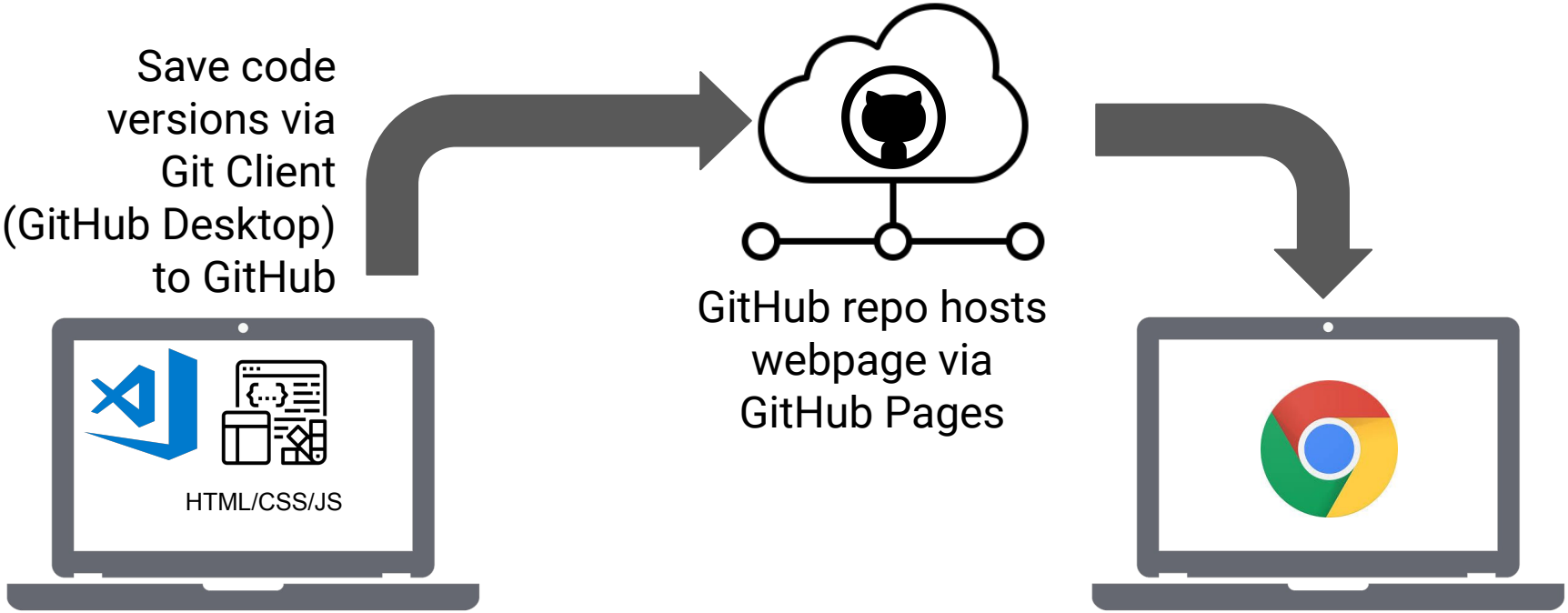
axosoft GitKraken

Fork

UX/UI Boot Camp will use:

Code Environment

Save code versions via Git Client (GitHub Desktop) to GitHub



GitHub repo hosts webpage via GitHub Pages

Code in Visual Studio Code (Local)

Review/test via Chrome developer tools (Online)

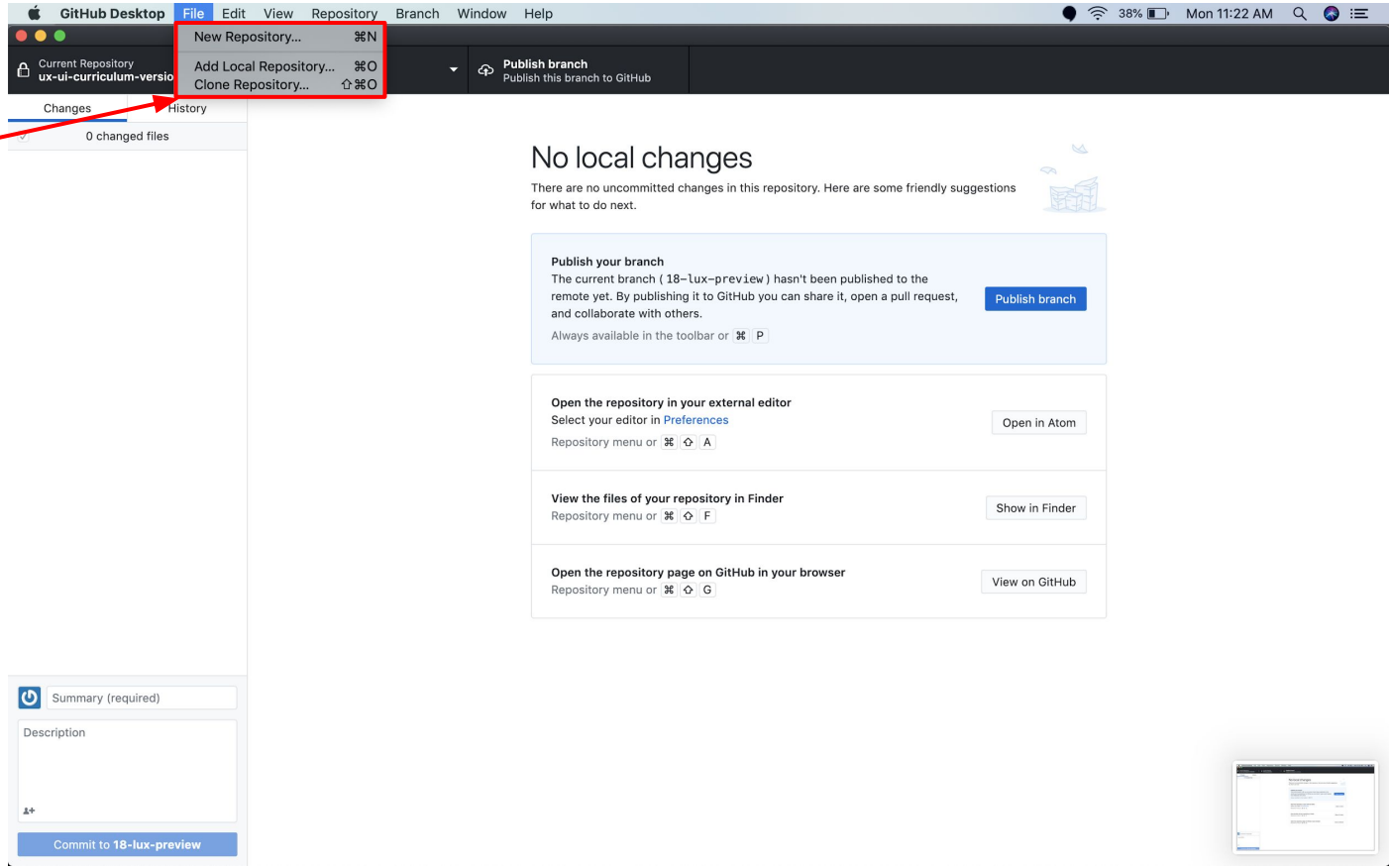
GitHub Desktop Functions

GitHub Desktop Functions

New Repository:
Button creates a new repo on your desktop.

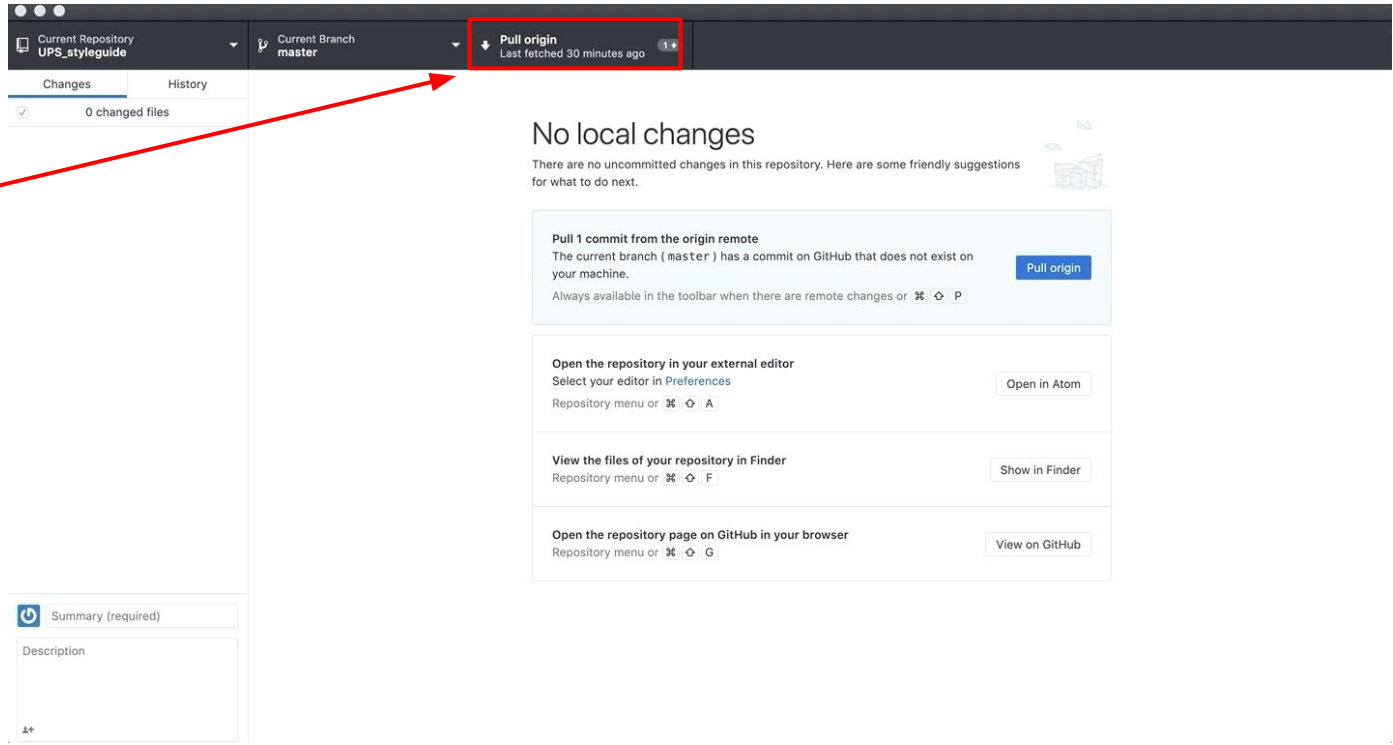
Add Local Repository:
Adds an already existing git repo to GitHub Desktop.

Git Clone:
Clones a git repository onto the local file system.



GitHub Desktop Functions

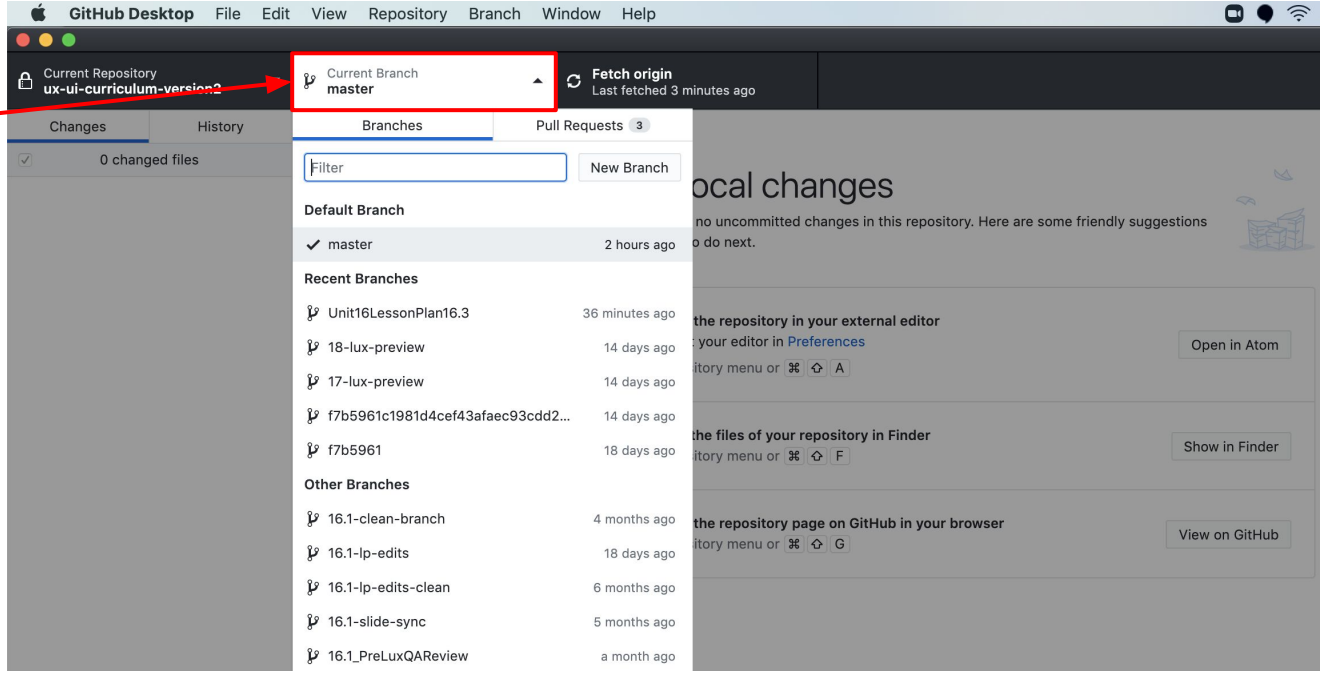
Git Pull:
Used to update your local repository with changes to local branches.



GitHub Desktop Functions

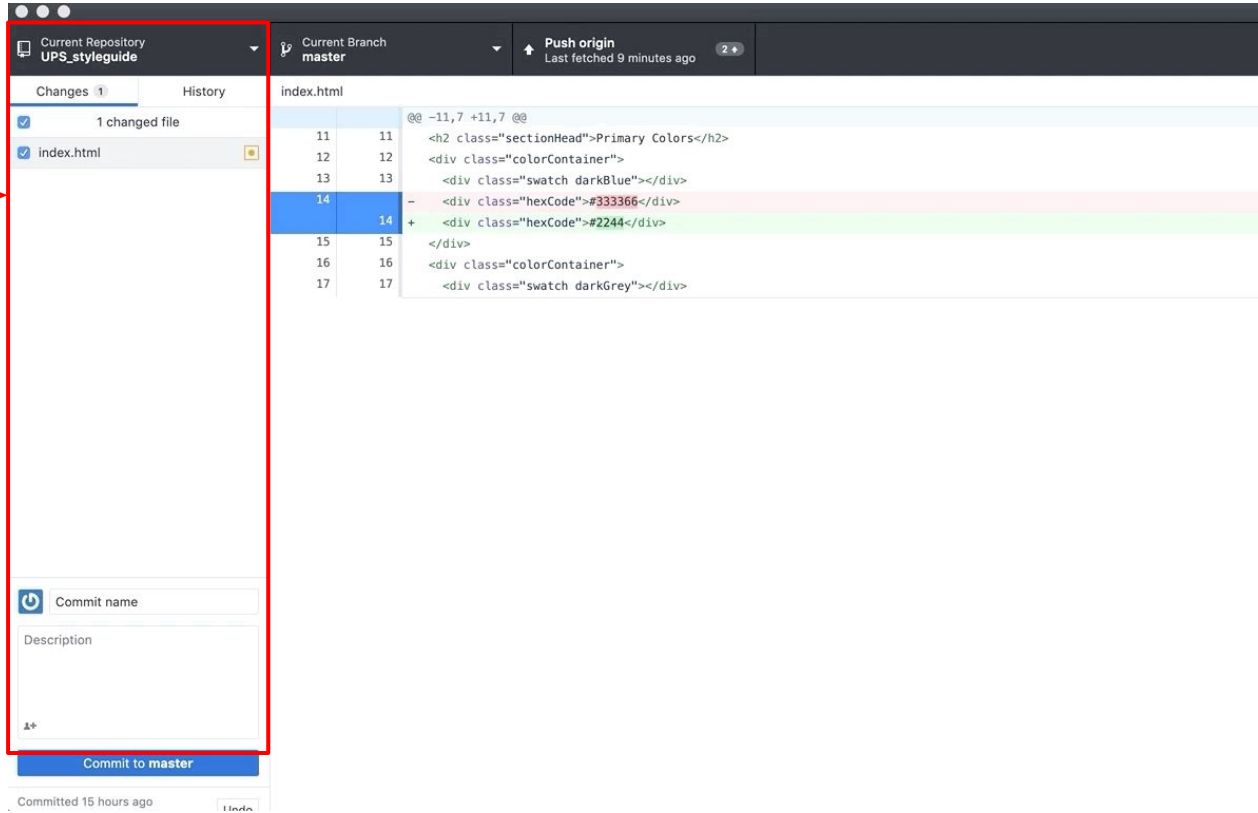
Branches:
You can create and manage branches via GitHub Desktop.

Branches are local copies of repositories that allow you to work separately from the main repository.



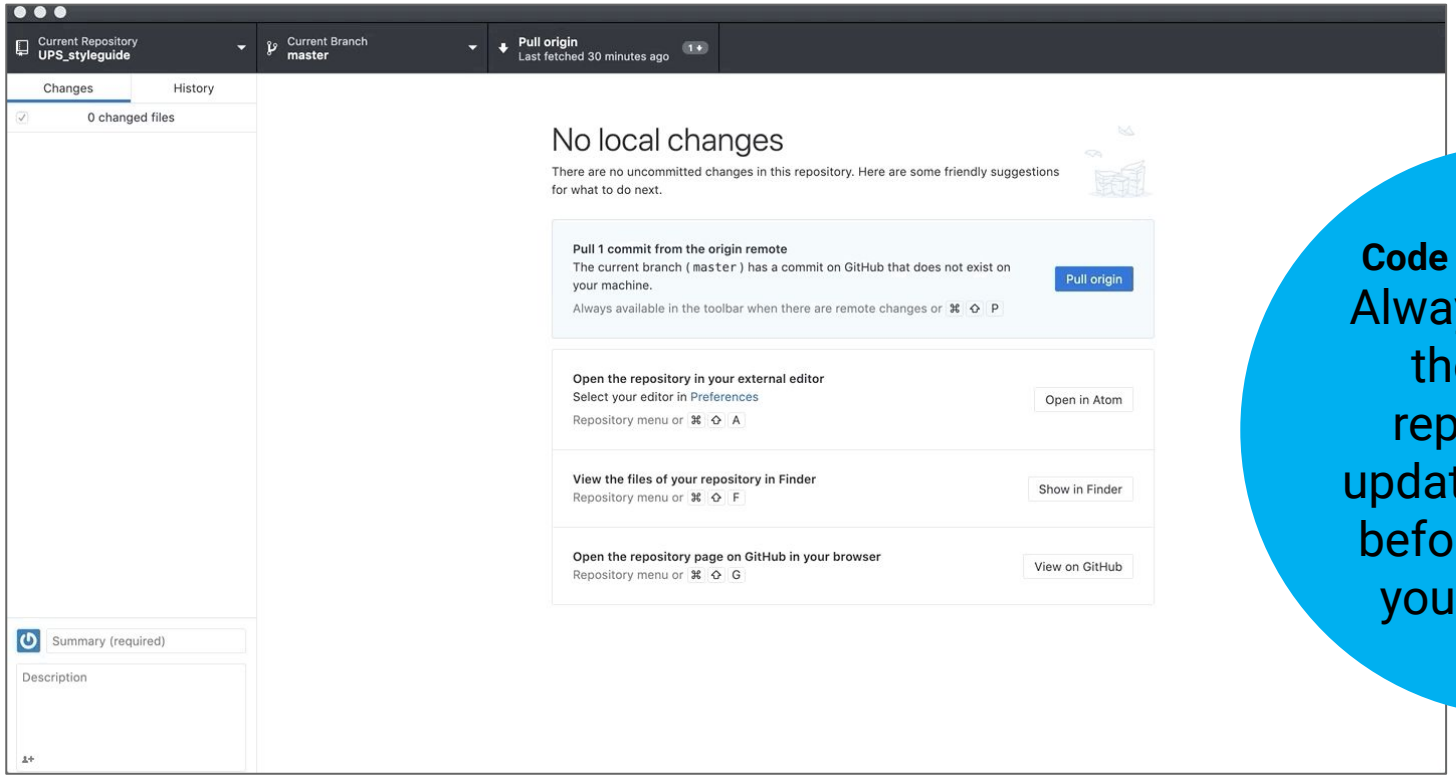
GitHub Desktop Functions

Git Push:
When changes are made to your local Git repository, GitHub Desktop tracks your changes and allows you to update the cloud repository.



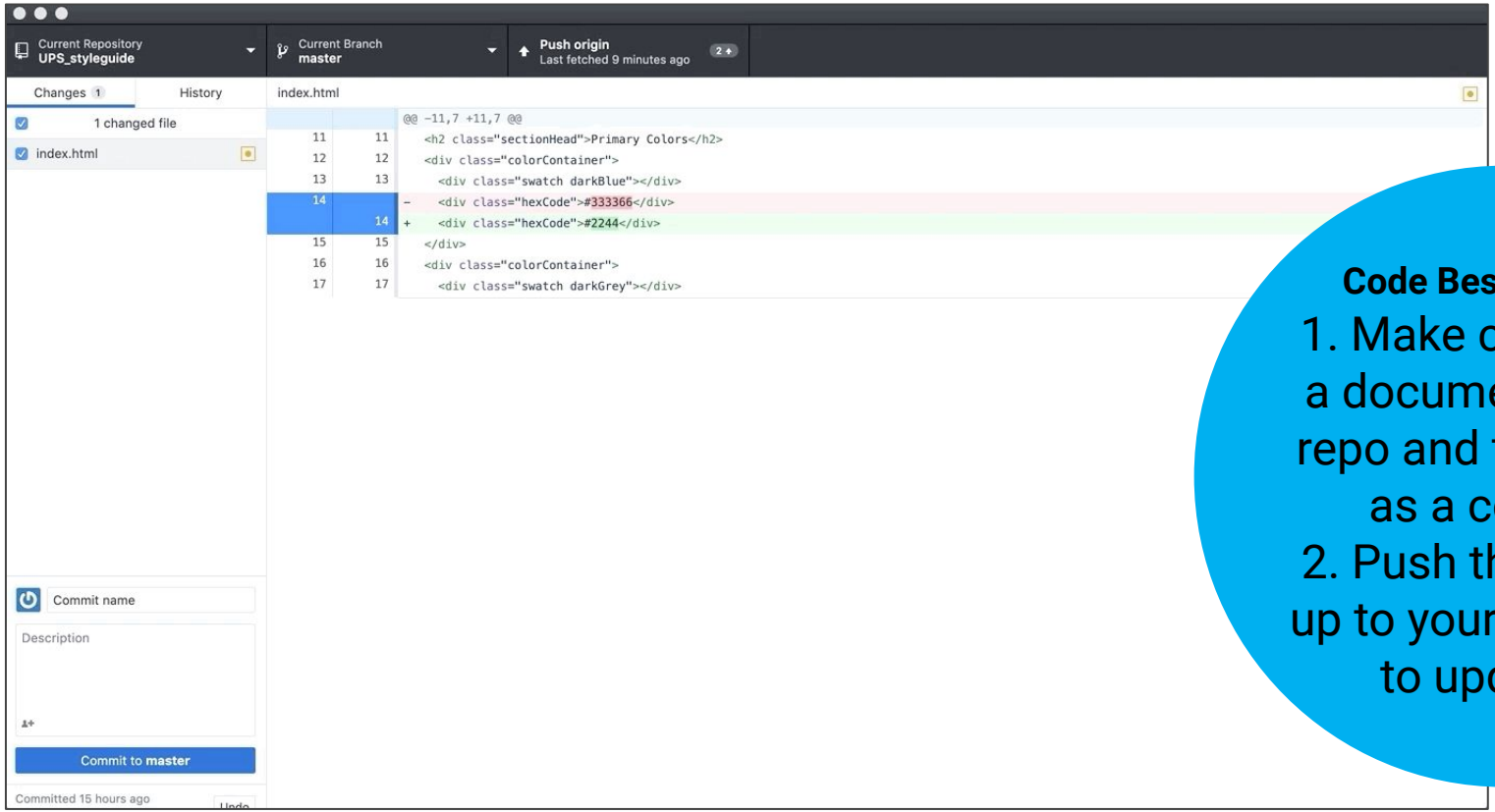
Common GitHub Desktop Workflows

Workflow 1: Pull From a Repository



Code Best Practice:
Always pull from the remote repository to update your code before you start your workday.

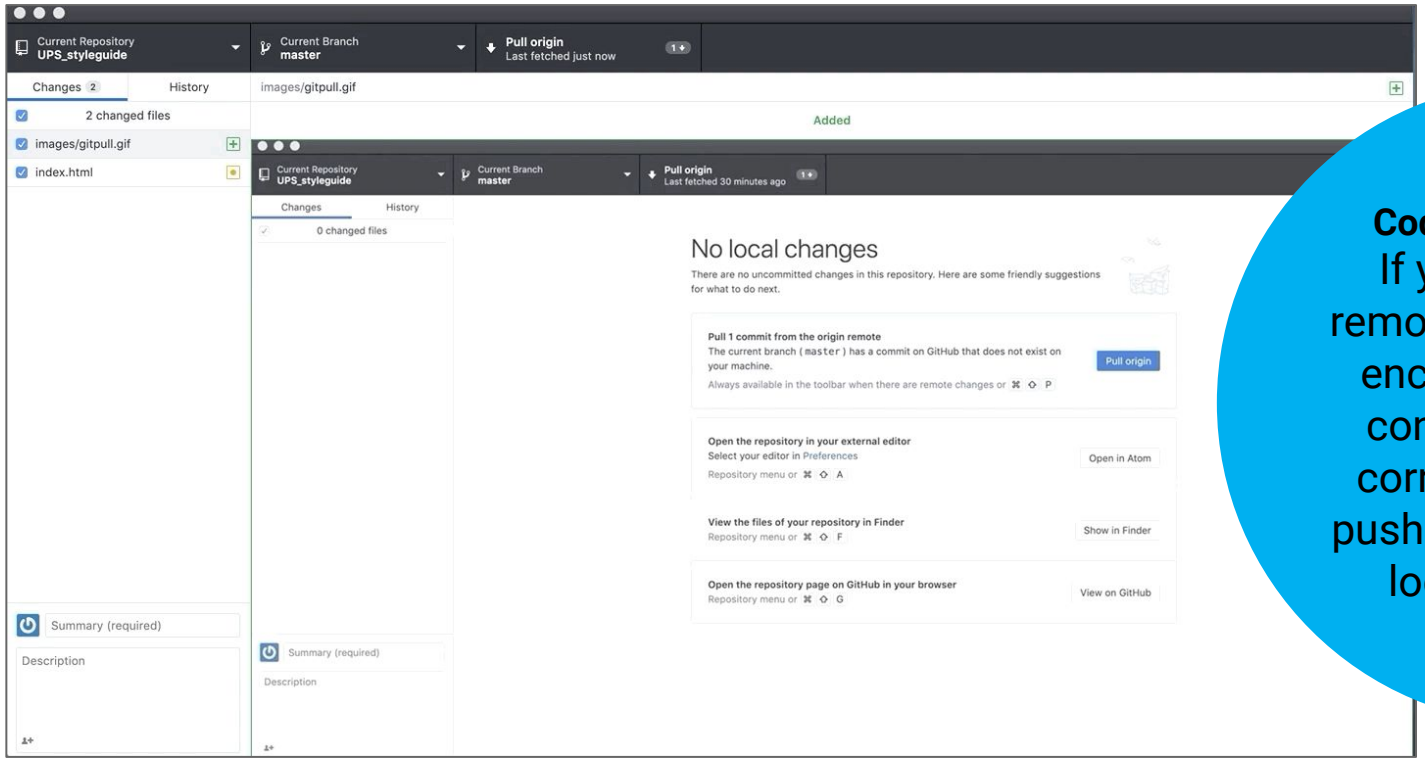
Workflow 2: Push Local Changes



Code Best Practice:

1. Make changes to a document in your repo and then add it as a commit.
2. Push the commit up to your repository to update it.

Workflow 3: Resolve Merge Conflicts

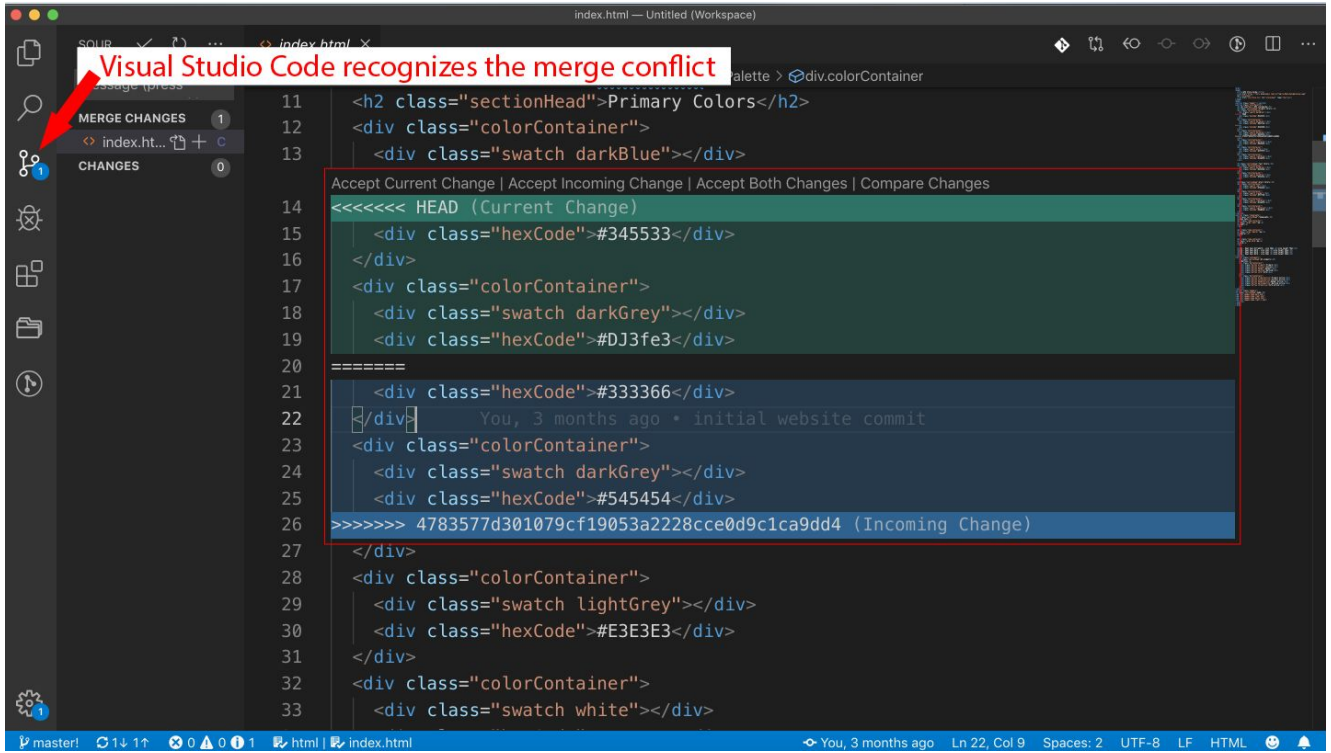


Code Best Practice:
If you pull from a remote repository and encounter a merge conflict, select the correct version and push the merge to the local repository.

Workflow 3.1: Resolve Merge Conflicts

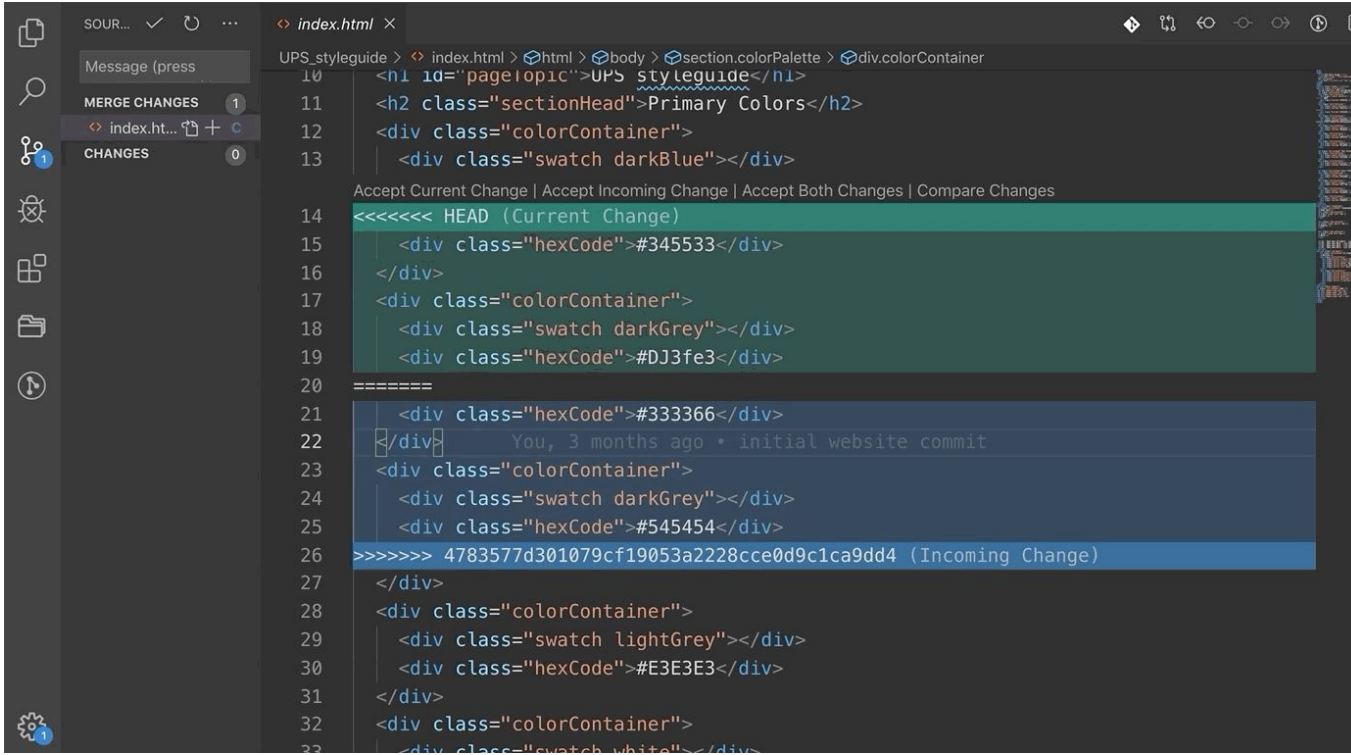
The next step is to open the conflicting file in Visual Studio Code.

Make sure you have this plugin installed for Visual Studio Code:
[Visual Studio Code Git Package]



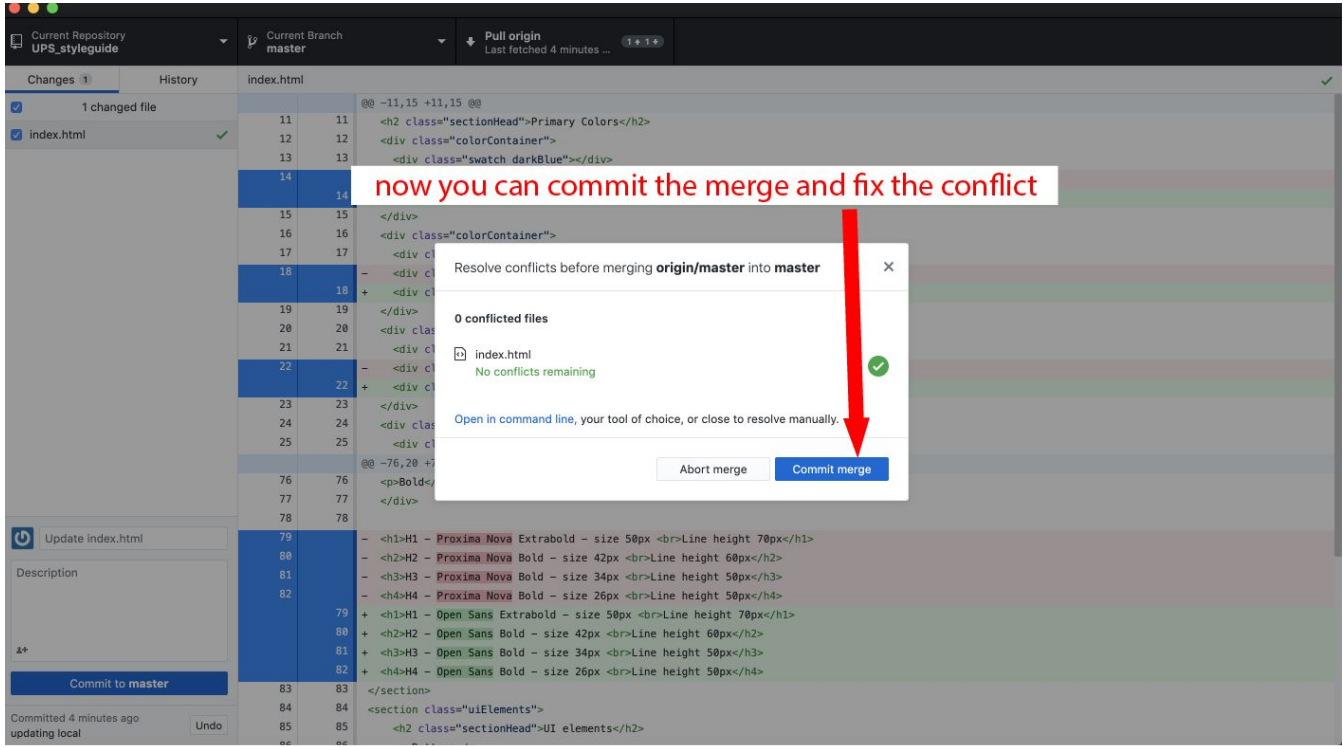
Workflow 3.2: Resolve Merge Conflicts

Select which change is the most current and up-to-date by clicking accept current change or accept incoming change.



Workflow 3.3: Resolve Merge Conflicts

After all the conflicting files have been fixed, we can merge our code and push it back to the repo, all in GitHub Desktop.



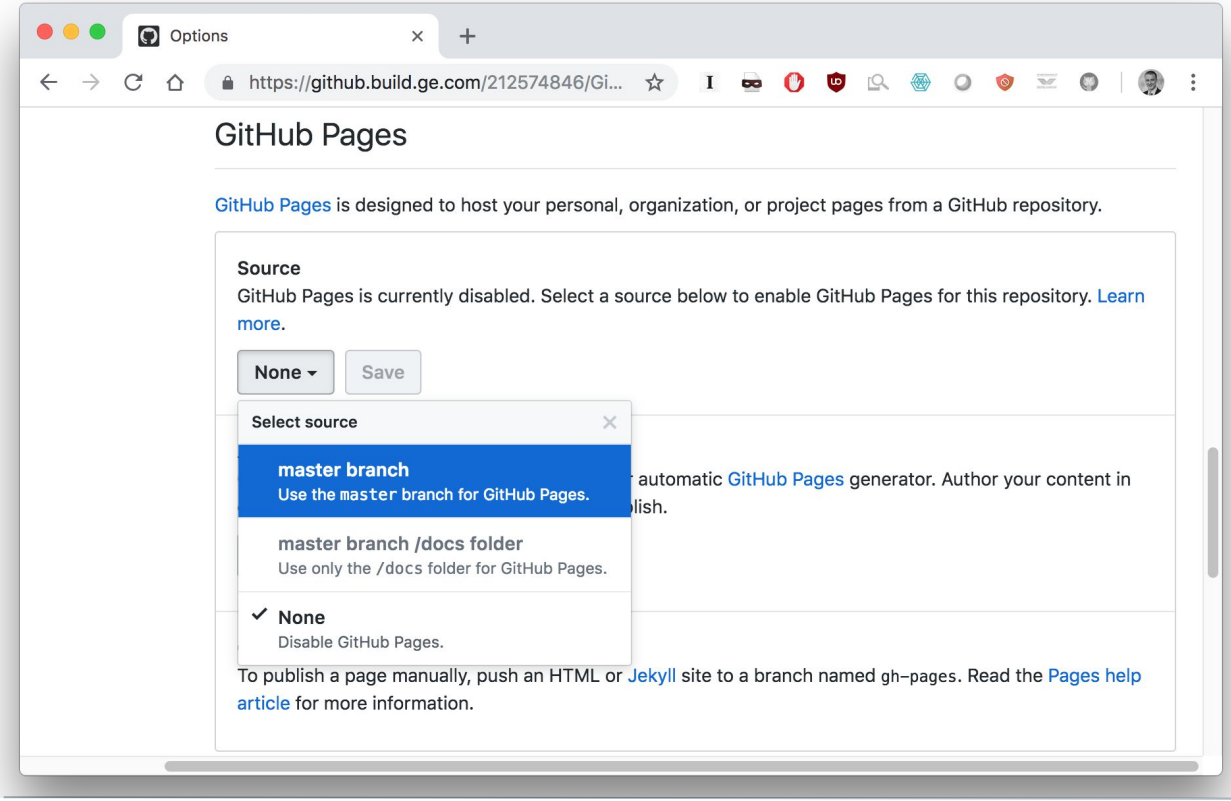
Hosting a Website With GitHub Pages

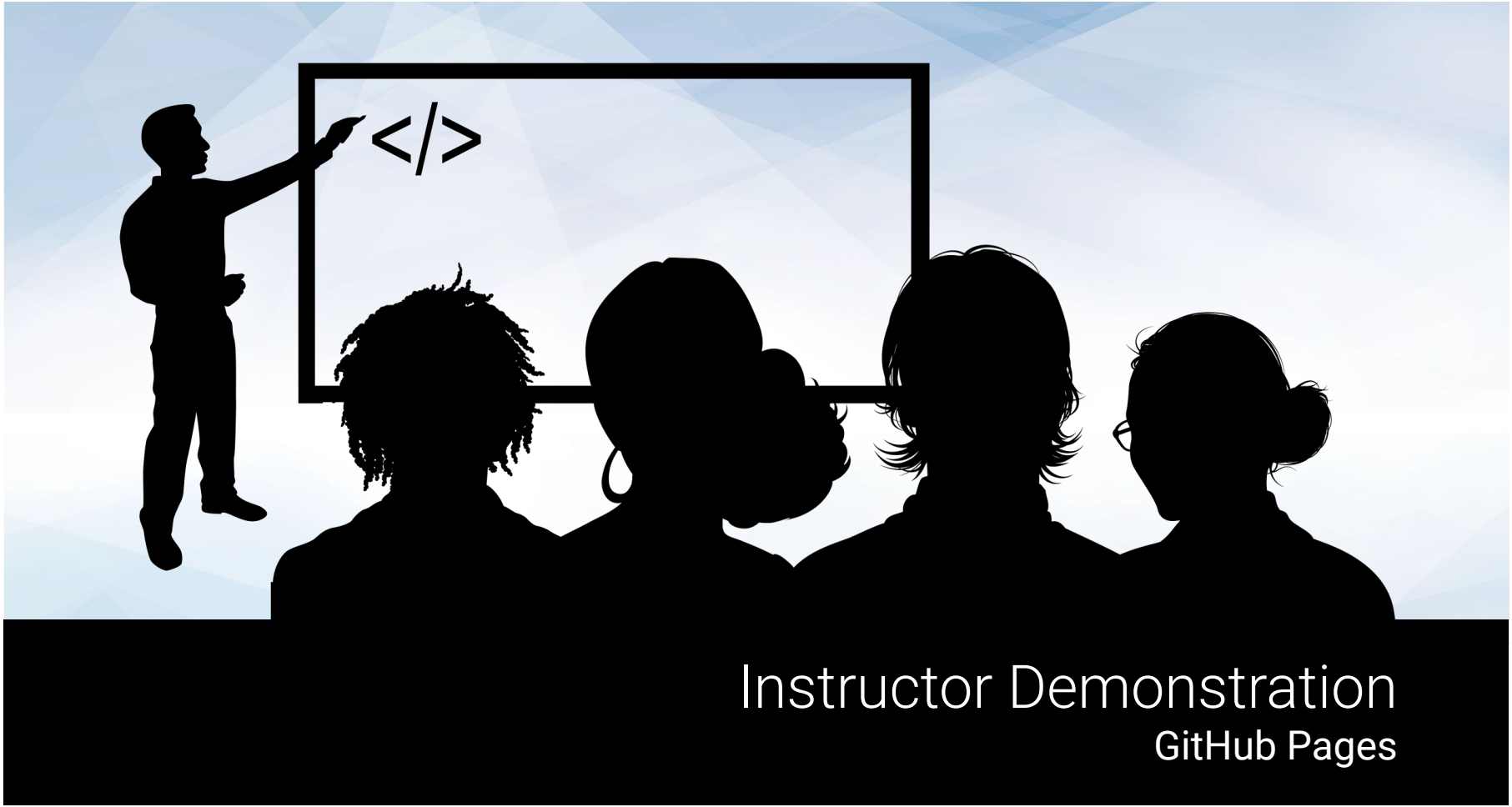
GitHub Pages is a static site-hosting service designed to host your personal, organization, or project pages directly from a GitHub repository.

https://pages.github.com/

GitHub Pages

A GitHub Page will host your web content for turning in assignments.





Instructor Demonstration

GitHub Pages

Challenge:

Upload Your First Webpage to the GitHub Cloud!

(Instructions sent via Slack)

Suggested Time:
20 minutes

Time's Up! Let's Review.

Congratulations! Recap

Today we learned:

01

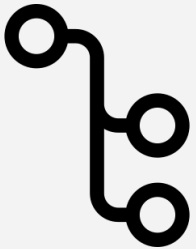
HTML Positioning of Elements
How to code html markup for webpage design.

02

HTML Structure and CSS Classes and IDs
How to code html and CSS together for element positioning.

03

GitHub and GitHub Pages
How to push pages to a version control cloud and publish webpages.



Questions?


The End

30 minutes