

Front End Development

Session 11



Hackathon Collaboration With *GitHub*



Learning Goals

In today's session, we will:

01

Review and recap all the awesome things we've learned

02

Practice resolving merge conflicts with GitHub

03

Demonstrate how to use Javascript Objects

04

Discuss our Hackathon goals and timeline—then get started!



Q. What kind of code are we learning in this course?

Q. Can you name a role, job, or industry that uses this kind of code?

Q. What three “languages” do we use to create websites?

Q. What does "HTML" stand for?

Q. How can we examine website code on our computers?

Q. In HTML, what's the difference between the <head> and the <body>?

What is the difference between Git and Github?

What does the CLI stand for? What would we use it for?

How do you format a link in HTML?

How do we save our work with Git and Github?

What does a <div> tag do?

What does CSS stand for? Why do we use CSS?

What is the difference between *block* and *inline*?

What does “float” do?

What are the five positioning properties?

What is the difference between *block* and *inline*?

What does “float” do?

What are the five positioning properties?

What is a framework?

What does “open-source” mean?

What is responsive design?

What is a variable? How do we declare a variable in JS?

What is a data type? What JS data types have we learned so far?

What does state refer to? Why is this useful?

What's the difference between alerts, console.log, and document.write()?

What is a browser event?

How do we generate a random number in Javascript?

What function do we append to round up or round down?

What is pseudocode and why is it useful?

What is a **FOR Loop**?

What does **control flow** refer to?

What is the **DOM** and *why is it useful?*

How might we describe the *relationship between objects* in our HTML?

What is a ***function*** in Javascript?

What does it mean to *invoke* a function?

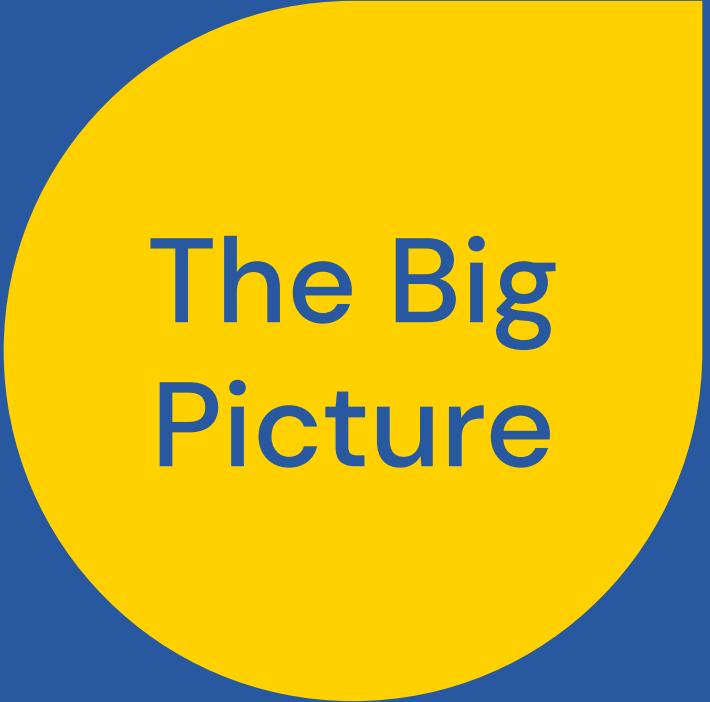
What is an **argument** and *where* are they used?

What are the three most important Git & Github commands?

What is a Javascript library and why is it useful?

What is jQuery? What is it used for?

What's an event listener and how might we use one?



The Big
Picture



WE ❤ CODING

Our mission has been to help you understand, engage with, and **enter the world of code!**

Our three week crash course in **front-end web development** has helped you learn about **some of the many skills and opportunities available in tech.**

Throughout the program, you've built your own **professional portfolio and a ton of cool stuff** that you can use for future applications, development, or fun!

So What Have We Learned?

Take a moment to think about all that you've learned so far!

- How to plan website development with wireframes and pseudocode
- How to build, layout, and style web pages with HTML and CSS
- How to code fundamental programming concepts in JavaScript
- How to create responsive, interactive web pages with Bootstrap and jQuery
- How to create and customize interactive web-based games
- How to collaborate with professional development tools and share your portfolio online!

Woah, You've Come So Far!

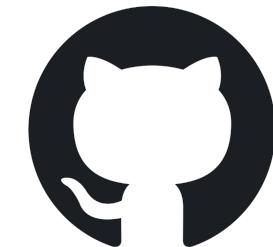
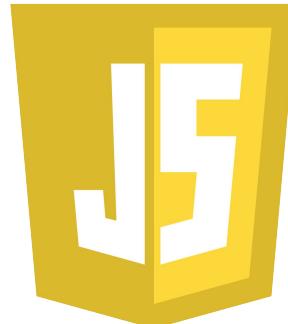
HTML



CSS



JS



GitHub



Visual Studio Code



git



Bootstrap



jQuery

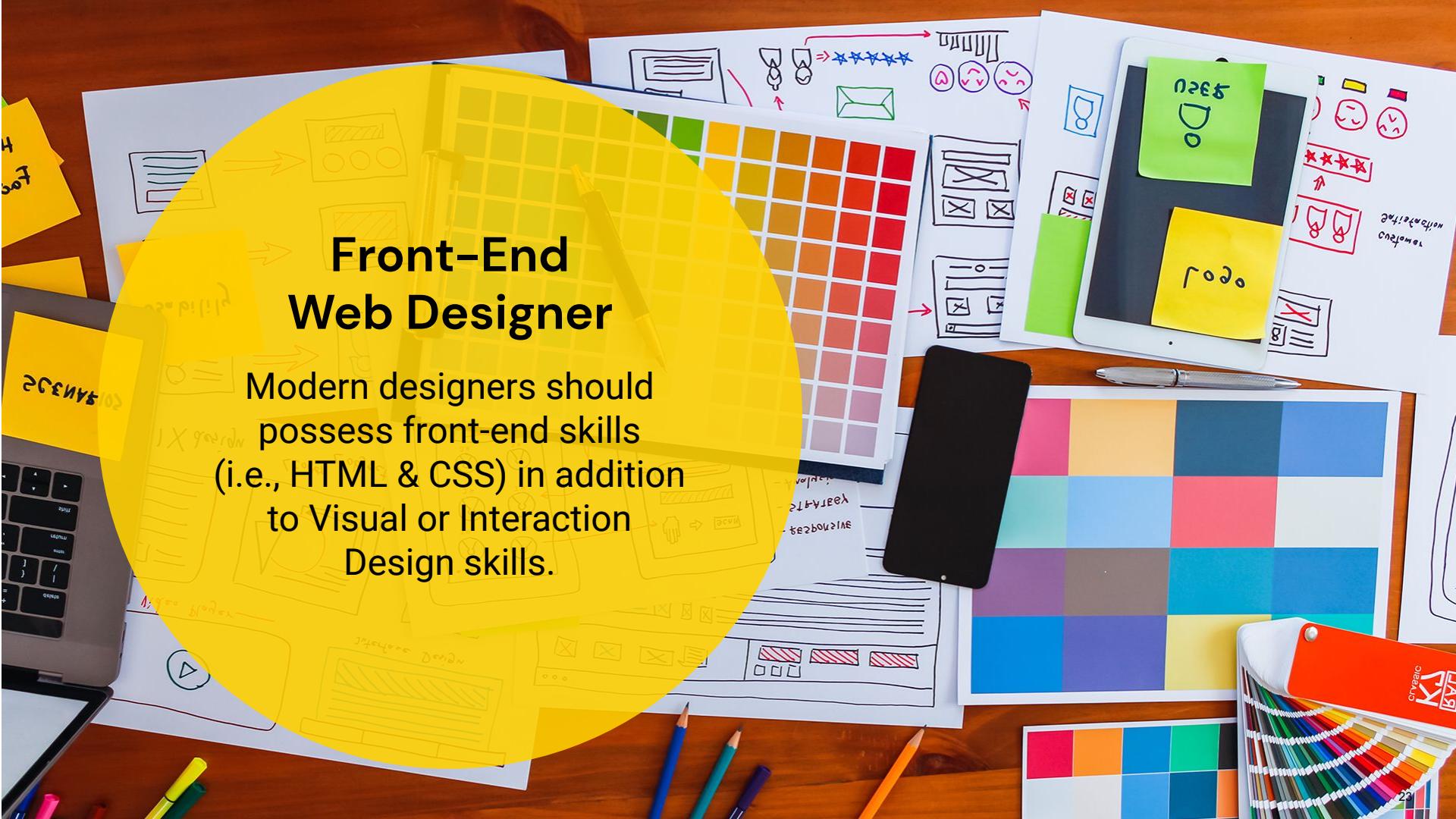
Front-End Developer

A developer who is skilled at HTML, CSS, and JavaScript and can implement these technologies across the web.



Front-End Web Designer

Modern designers should possess front-end skills (i.e., HTML & CSS) in addition to Visual or Interaction Design skills.



CSS/HTML Developer

A developer who excels at using HTML and CSS to create newsletters and websites. They may collaborate with visual designers to inform specifications or convert designs into real working websites!

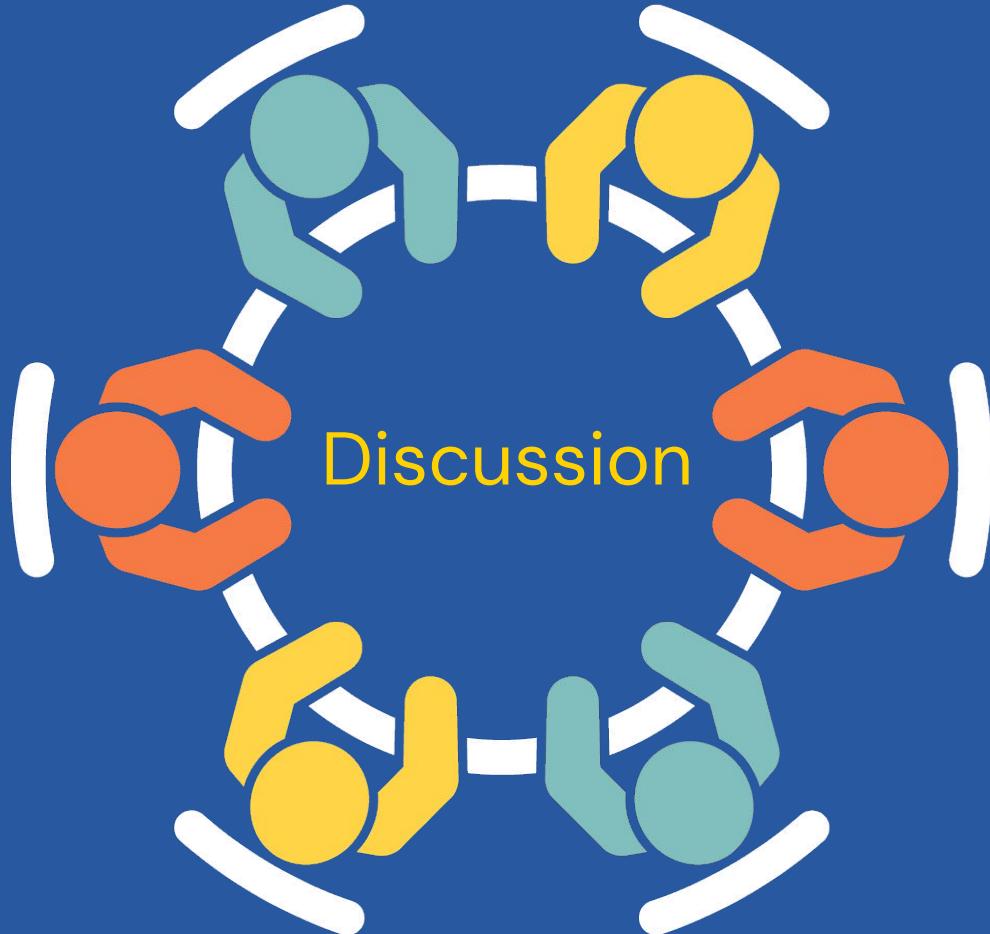
```
1<div class="wrap">
2  <div id="content">
3    <header id="topnav">
4      <nav>
5        <ul>
6          <li class="active"><a class="scroll" href="#">Home</a></li>
7          <li><a class="scroll" href="#service">Service</a></li>
8          <li><a class="scroll" href="#product">Products</a></li>
9          <li><a class="scroll" href="#portfolio">Portfolio</a></li>
10         <li><a class="scroll" href="#team">Team</a></li>
11         <li><a class="scroll" href="#contact">Contact</a></li>
12       <div class="clear"></div>
13     </ul>
14   </nav>
15   <div class="logo">
16     <a href="/">
17       
18     </a>
19   </div>
20 </div>
21 </div>
```



Mobile Front-End Developer

A developer who has experience creating sites for mobile and tablet devices (either natively or in the browser). And there's a large market for developers who can help convert existing sites to modern accessibility design standards!





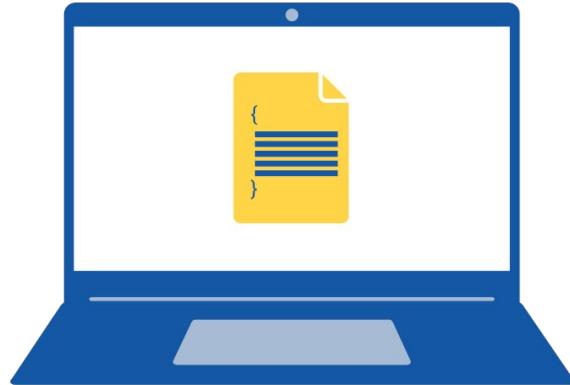
Questions?





Let's Talk
Github

Git + GitHub = Awesome



Github = Community



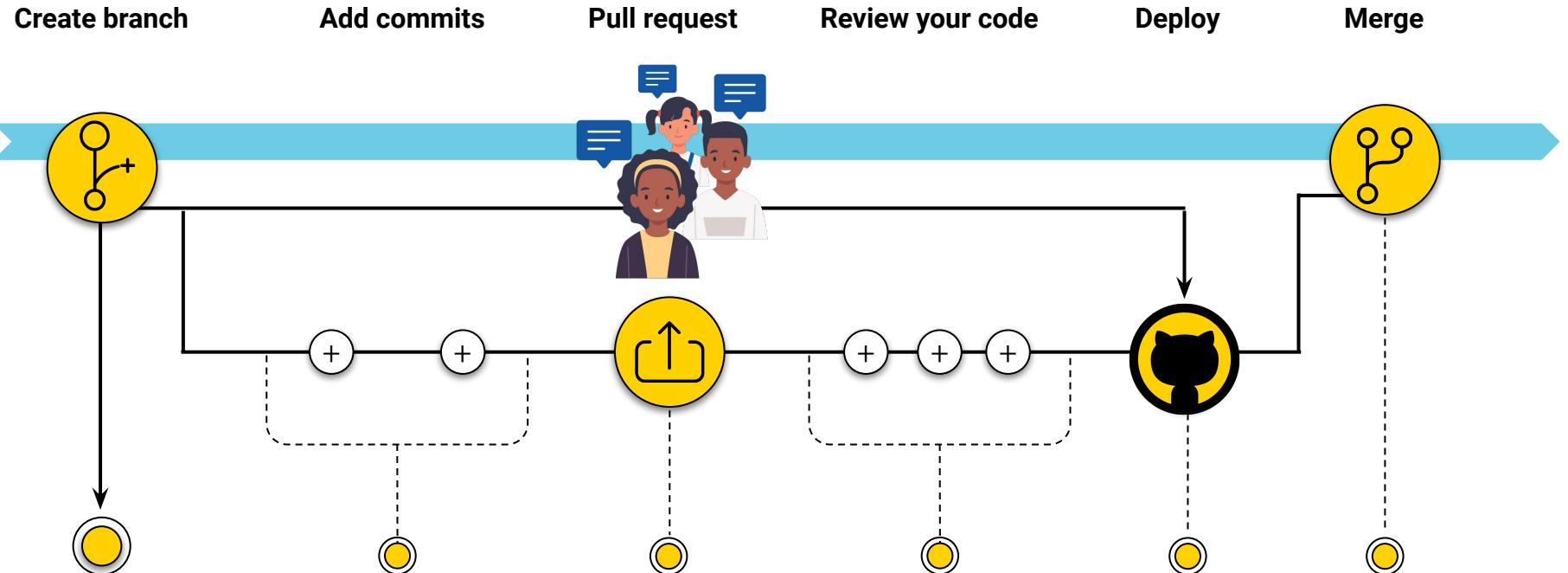
Document Your Work!



Save

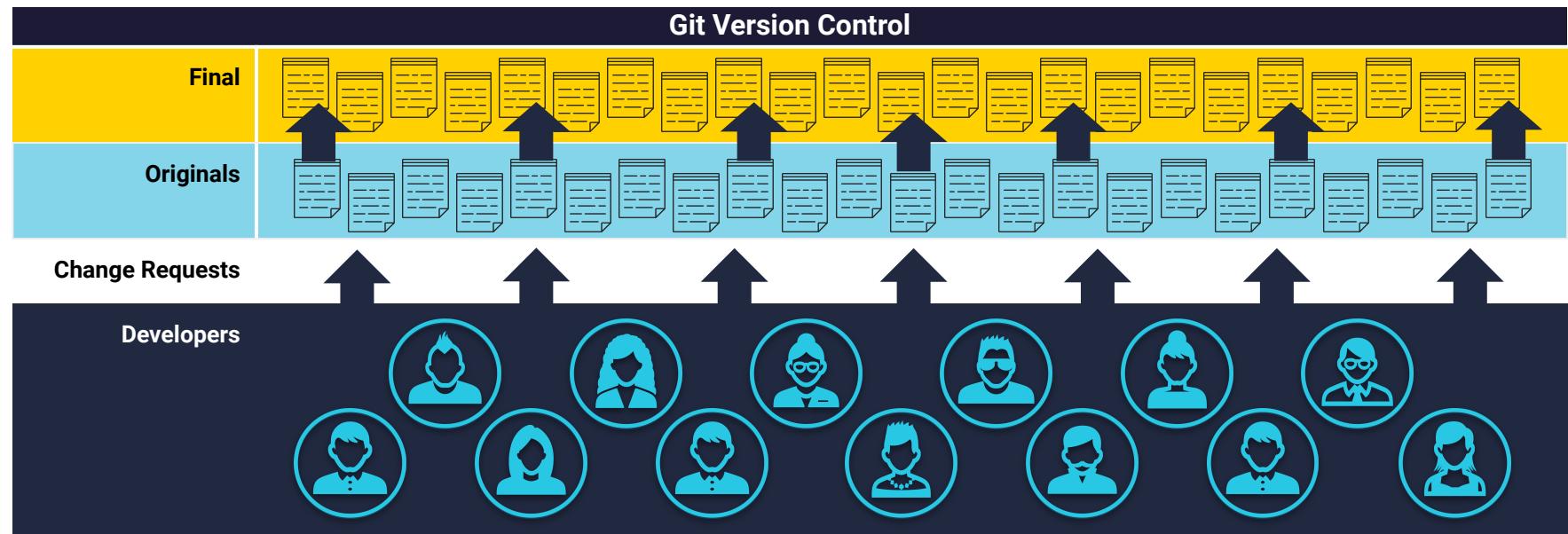
Share 

More Ways to Collaborate with Github



Teamwork with Git & Github

Assign roles and responsibilities when collaborating on complex projects!



Stage, Save and Share Your Work

Our Three Favorite Commands!

01

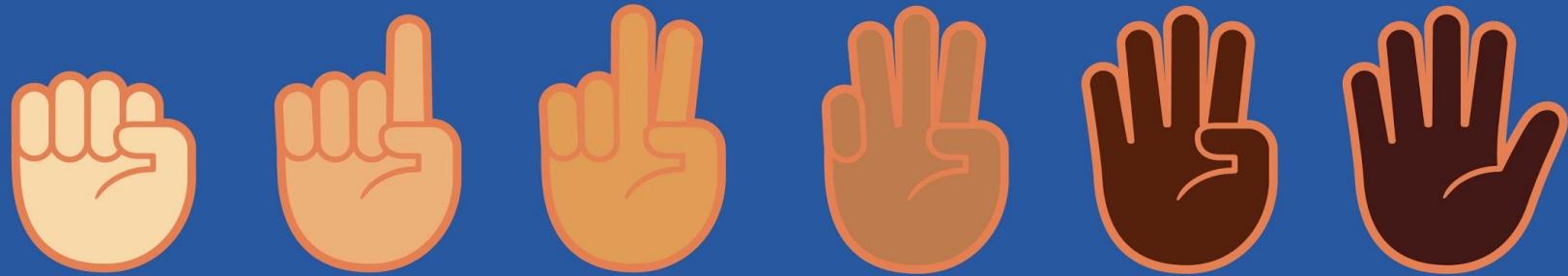
ADD

02

COMMIT

03

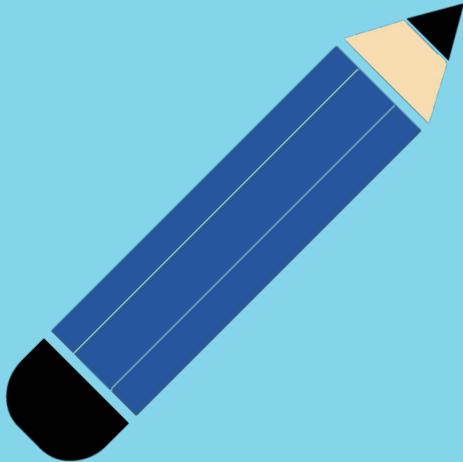
PUSH



Fist to five

Let's





Activity: Resolving Merge Conflicts

1. Open instructions and GitHub
2. Team up!
3. Follow instructions
4. Resolve those conflicts!

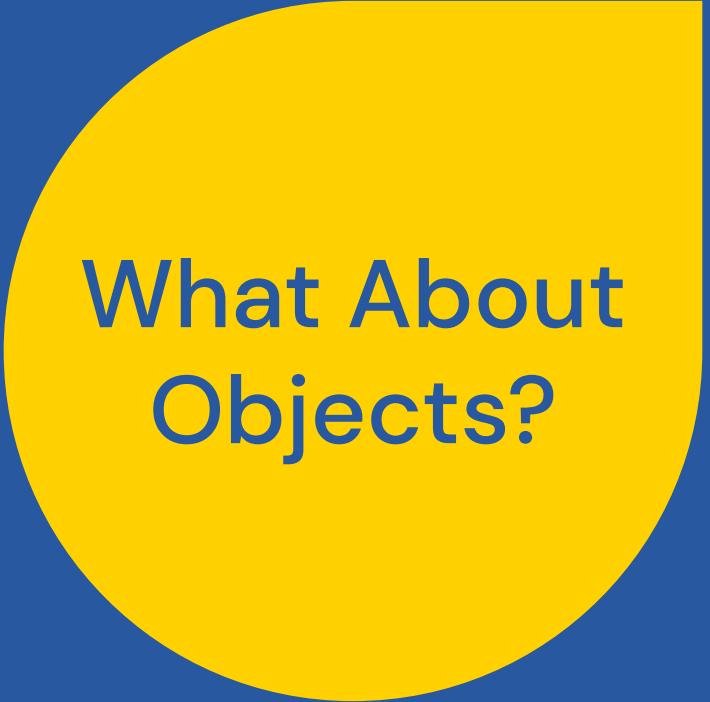
Suggested Time:
15 minutes





Questions?





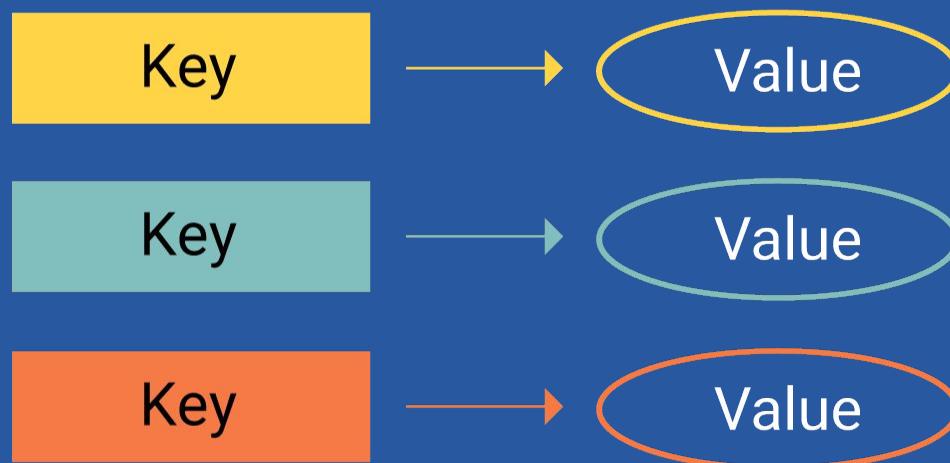
What About
Objects?

Objects are a type of *data structure* that can store **data** and **functions** using **key/value** pairs.

In other words, objects are a **variable** that can **store multiple values**, just like an array.

Objects in Javascript

Values stored in an object can be collected by using a reference known as a key.



Object Syntax

Objects are declared by using **curly brackets**, with properties defined using the **property:value** syntax.

```
var car = {type:"Fiat", model:"500", color:"white"};
```



Instructor Demonstration:

Let's Look At Some Objects In Javascript!

Suggested Time:



Set up an object called PlayerInfo

```
var playerInfo = {  
  
    name: "Tim the Tyrannosaurus",  
    health: 100,  
    stats: {  
        attack: 10,  
        defense: 5  
    }  
    money: 10,  
    inventory: ["teeth", "claws", "tiny arms"]  
};
```

Collect a value from our object

```
var playerInfo = {  
  name: "Tim the Tyrannosaurus",  
  health: 100,  
  stats: {  
    attack: 10,  
    defense: 5  
  }  
  money: 10,  
  inventory: ["teeth", "claws", "tiny arms"]  
};
```

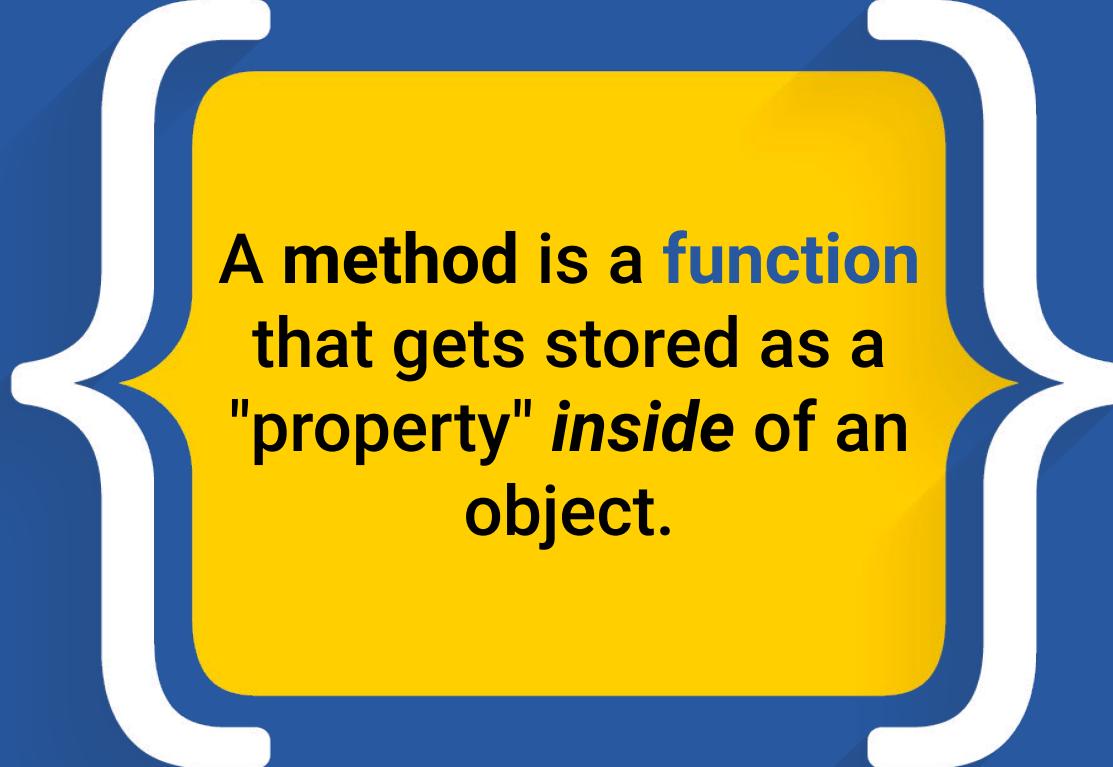
```
console.log(playerInfo.name)
```

This will print "Tim the Tyrannosaurus" to our console!



Now we know a few different ways to store data: *arrays* and *objects*.

But *data isn't the only thing* we can store in our objects!



A **method** is a **function** that gets stored as a "property" *inside* of an object.

Methods are **helpful** when **working with the data stored in your object**.

By storing built-in *methods* that correspond with your data, your code becomes more **organized and reusable!**

Add a method to the playerInfo object

```
var playerInfo = {  
  name: "Tim the Tyrannosaurus",  
  health: 100,  
  stats: {  
    attack: 10,  
    defense: 5  
  },  
  money: 10,  
  inventory: ["teeth", "claws", "tiny arms"]
```

```
strikeEnemy: function(){  
  console.log(this.name + " attacks the enemy for" + this.attack +  
  "damage")}  
};
```

Now let's call our method!

```
var playerInfo = {  
  name: "Tim the Tyrannosaurus",  
  health: 100,  
  stats: {  
    attack: 10,  
    defense: 5  
  }  
}
```

```
strikeEnemy: function(){  
  console.log(this.name + " attacks the enemy for" + this.attack +  
  "damage")  
}  
;  
  
playerInfo.strikeEnemy()
```

This will call the
`strikeEnemy()` function
within the `playerInfo` object





Pro Tip: Use the “this” command
to quickly refer to the “owner”
of your method!

```
var playerInfo =  
    name: "Tim the Tyrannosaurus",  
    health: 100,  
  
    strikeEnemy: function(){  
        console.log(this.name + " attacks the  
enemy for" + this.attack + "damage")  
    } };
```

{“Object-Oriented” Code}

Objects make your code more efficient!

Questions?





Hello,
Hackathon

HACKATHON



A "hackathon" is a popular fast-paced event where programmers and designers work together to create fun and interesting projects!

Hackathon Goals: Get Ready for a Challenge!

For our Week 3 hackathon, you will need to leverage all of your amazing new skills to collaborate and build a functioning website or app from scratch!

01

Your challenge is to plan, design, build, test, and demo a working project by *the end of the week*.

02

On the last day of our program, your group will present your work and walk us through your code!

03

Your instructional team will be on-hand to support and check in with your group throughout the week.

Requirements: A Few Ground Rules

By the end of the week, your group will need to create:

01

A working, navigable website or web app prototype

02

Shareable code hosted and deployed via GitHub

03

Clearly commented code

04

A Presentation / Demo of your team's finished work!

What Will You Build?

During our program, you'll complete three different projects, providing you with a chance to practice and apply your new skills while building something awesome!

Option 1: Website

Create a website for a client!

- A Homepage
- About page
- Contact information
- News and Updates
- Additional features:
 - *Photo gallery*
 - *Calendar/Schedule*
 - *Embedded media*

Option 2: Games!

Build a web-based game!

- Fully functional and playable
- Require keyboard inputs or clickable buttons
- Rules of the game must be explained in the application

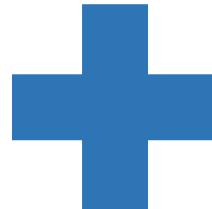
Option 3: You Decide!

Work on a topic of your choice—get creative!

- Help out a nonprofit
- Support a local business
- Redesign a government site
- Sell a product
- Anything goes! (*within reason*)

How Will We Spend Our Time This Week?

Monday through Thursday



Friday



Workshop Time:

Work with your groups to code something awesome!

Hackathon Presentations:

Share out a 5-10 minute demo of your work with the class!

Hackathon Demo: Sharing Your Work

Sharing Your Work

01

Introduce your topic

02

Describe your goal(s)

03

Share your code

04

Demo your final product!

05

Tell us about the process

06

Ask/answer any questions

A Brief Reminder

Graduation Requirements: In order to earn your certificate, you'll need to:

Create
something
awesome!



Don't
plagiarize,
please

Tips and Tricks, Part 1: How to Get Started

Working as a group is key to a successful Hackathon.
Here are a few successful strategies:



Discuss your goals!



Find some inspiration



Set a vision



Define requirements and roles



Sketch & Wireframe



Write Some Pseudocode!



Break tasks into smaller tasks



Collaborate with GitHub



Combine your skills!

Tips and Tricks, Part 2: How to Host a *Great* Presentation

Presentation Tips



Assign sections and give everyone a chance to speak



Use slides or visuals for your audience



Use notes to structure your time



Demo the final product!



Use Inspector or VS Code and let us see how it works!

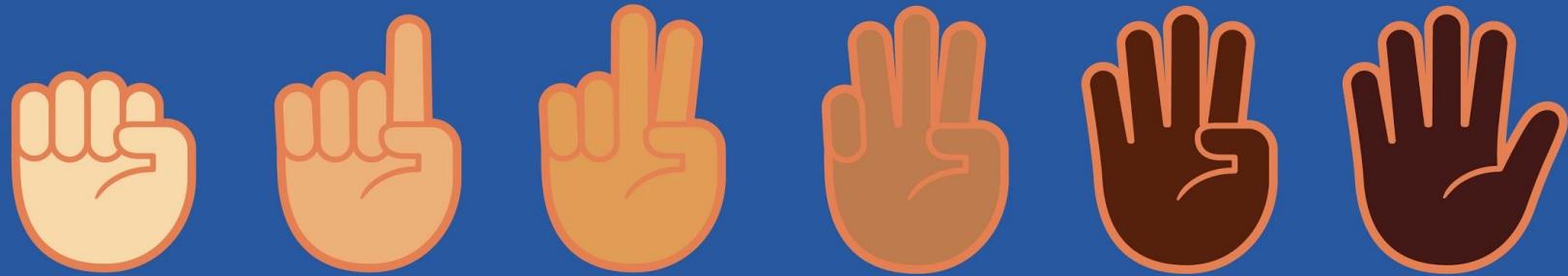
Questions?





Hackathon Time!

Workshop



Fist to five



A large yellow circle is centered on a dark blue background. The circle's diameter spans from approximately the 285 mark on the left axis to the 655 mark on the right axis.

**Time to
Recap**

Learning Goals

Our objectives for today's session were to:

01

Review and recap all the awesome things we've learned

02

Practice resolving merge conflicts with GitHub

03

Demonstrate how to use Javascript Objects

04

Discuss our Hackathon goals and timeline—then get started!

Reflection

What was your favorite part of today's session?

What was the most interesting thing we covered today?

What do you still have questions about?





Sneak Preview

- You'll be spending the rest of our sessions this week working on your group hackathon project!
- We'll be checking in with you throughout the week to help you prepare.
- Your goal is to create a working app or site built and ready to share by Friday!

Questions?



*The
End*