WEB DEV INTRO

CODEJAM WORKSHOP 1

CODEJAM 15

NOVEMBER 14–16, 2025

TODAY'S PLAN

1. Introductions
2. Housekeeping Items
3. Common Web Definitions
4. How the Web Works
5. Popular Web Dev Stacks
6. Demo & Set-Up

MEET THE TEAM

ASK ANY OF OUR TEAM MEMBERS IF YOU HAVE A QUESTION!

PREPARE FOR CODEJAM15!

WORKSHOP SCHEDULE

Intro to Web-Dev

Date: October 23rd, 2025
Location: Wong 1020
Time: 6:00pm-8:00pm

Intro to Front-End

Date: October 30th, 2025
Location: MacDonald 280
Time: 6:00pm-8:00pm

Intro to Back-End

Date: November 6th, 2025
Location:
Time: 6:00pm-8:00pm

Intro to Docker

Date: November 13th, 2025
Location:
Time: 6:00pm-8:00pm

All slides and materials can be found on our GitHub
https://github.com/mcgillcodejam/CodeJam_Resources

BEFORE WE GET STARTED:

Please take a quick minute to fill out this form!

BEFORE WE GET STARTED:

- Make sure you have VS Code installed: https://code.visualstudio.com/download
- Make sure you can access the Set-Up Guide:
- Ask a mentor if you need help installing or don’t have access!

IF YOU'RE CONFUSED AT ANY POINT, PLEASE ASK QUESTIONS!

This is a workshop, not a lecture! We are here to help you

WEB DEV DEFINITIONS

BREAKING DOWN COMMON TERMS YOU'LL USE

CODEJAM 15

NOVEMBER 14–16, 2025

WHAT IS WEB DEV?

Web Dev = Website Development

The process of building & maintaining websites and applications that run online a browser.

Encompasses everything from creating the user-facing design to managing the server-side logic and databases.

PARTS OF WEB DEV

FRONT-END VS BACK-END

What is Front-End Development?

- The process of designing and building the user-facing side of a website or browser.
- Includes everything a user sees and interacts with directly in the browser or interface.

What is Back-End Development?

- The process of building and maintaining the server-side logic of a website or applications.
- Includes databases, servers, and the APIs that connect them.
- “Behind the scenes” work.

Key Aspects of Front-End:

- User Interface (UI): Creating the visual elements such as text, images, buttons, navigation menus, layout, etc.
- User Experience (UX): Focuses on ensuring the application is intuitive, easy to navigate, and provides a positive and engaging experience for the user.

Key Aspects of Back-End:

- Server-side Logic: Developers create the code that runs on the server, handling user requestions and processing data.
- Databases: Manage how the data is stored, organized, and retrieved.
- APIs: Retrieve data to allow software components to communicate with each other.

WEBSITE VS WEB APP

A website is generally more static and is used to provide information.

A web app is more dynamic and generally involves complex interactions.

The key difference is in the intended use:

- Website focus on displaying info
- Web apps are built around user interaction

Note: Websites & Web Apps are not completely mutually exclusive

HOW THE WEB WORKS

AND HOW DO WE USE THE WEB?

CODEJAM 15

NOVEMBER 14–16, 2025

HOW WE MAKE WEBSITES

Fundamental languages:

1. HTML
2. CSS
3. JAVASCRIPT

Making a website involves writing these 3 languages.
This will be covered in the upcoming workshops.

Common steps:

1. Planning and designing: Defining your purpose, audience, and functionality, and then sketching visual representations of the layouts.
2. Development: Choosing your technologies including front-end, back-end, databases and then using those to bring your plan to life.
3. Deployment and Maintenance: testing the functionality, publishing your web app to a browser, and continuously monitoring any bugs or updates.

HOW THE WEB WORKS

2 Fundamental players in any web interaction:

1. The Web Browser
2. The Web Server

Web Browser (Client): Sends requests to servers and displays the content for users (e.g. Chrome, Firefox).

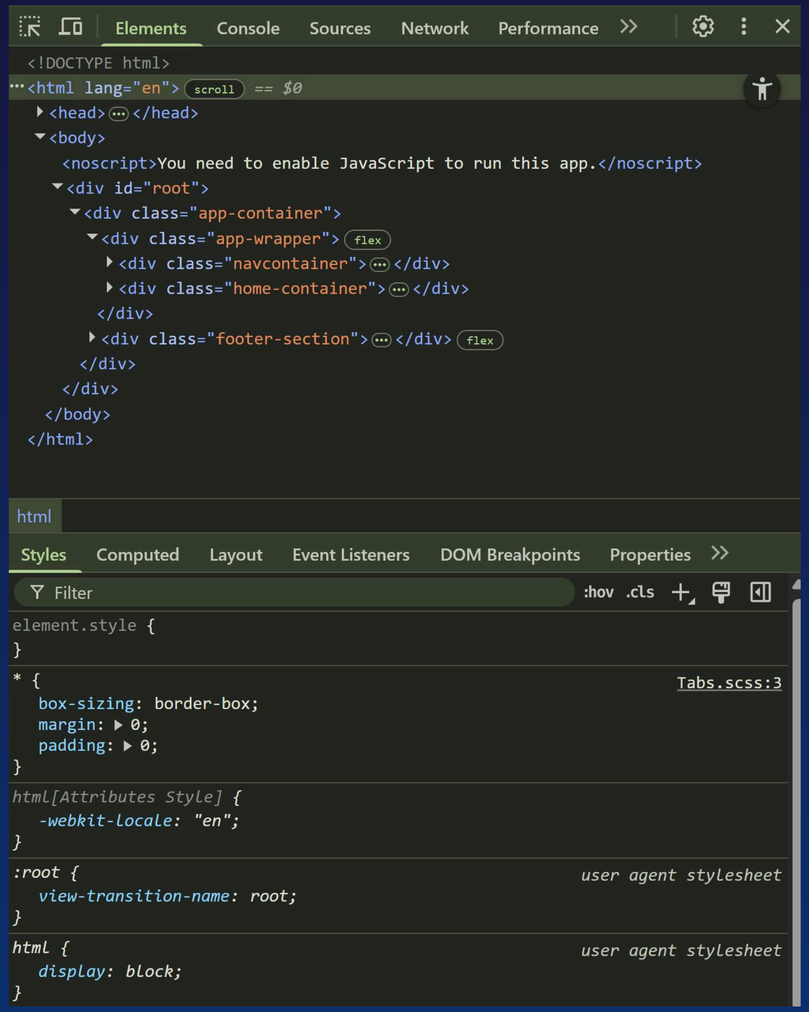
Web Server: Receives requests, processes them, and sends back responses—usually web pages or data.

Browser (Client) → Request (HTTP) → Server
Server → Response (HTML/CSS/JS) → Browser

CONNECTING THIS TO YOUR CHILDHOOD

EXAMPLE

- We now know that whenever you visit a website, the server sends you a copy of the code that makes up that website.
- We all remember messing around with ‘inspect element’ in middle school, where we would go on a website, right-click and then be able to directly modify the code for the website.
- What you were really doing is just editing your local copy of that website's code. This is why whenever you refreshed the page your changes would disappear, since your browser would send a new request to the server and get a fresh copy.



CHECKPOINT!

Ask a mentor if you have any questions so far

POPULAR TECH STACKS

WHAT TOOLS AND TECHNOLOGIES TO USE FOR WEB DEV

CODEJAM 15

NOVEMBER 14–16, 2025

FRONT-END VS BACK-END

Front-End Tools:

UI and Design:
- Figma
- Canva

Frameworks & Libraries:
- React
- Angular
- Vue.js

Styling:
- CSS
- Bootstrap
- Sass

Back-End Tools:

Programming Languages:
- Python
- Java
- JavaScript

Frameworks:
- Django
- Express.js
- Spring

Databases:
- PostreSQL
- MySQL
- MongoDB

WHAT IS A TECH STACK?

A tech stack is the combination of technologies used to build and run a software application, including the front end, back end, and database

MERN

M: MongoDB → Database
E: Express.js → Backend web app framework
R: React.js → Frontend JS library for UI
N: Node.js → JS runtime environment for server-side development.

Pros:

- JavaScript everywhere so you only need to know 1 language.
- High performance and scalability.

Cons:

- When working with large apps it can be tougher to manage the data flow.
- Not ideal for heavy relational data.

XAMPP

X: Cross Platform → Works on all OS

A: Apache → Web server software that handles HTTP requests

M: MySQL → Relational database

P: PHP → Server-side scripting language

P: Perl → General-purpose programming language

Pros:

- Easy to install and set up → ideal for local dev & testing
- Includes everything you need in 1 package

Cons:

- Less suited for large-scale or modern full-stack applications.
- Not recommended for production environments because of limited security.

MEAN

M: MongoDB → Database
E: Express.js → Backend web app framework
A: Angular → Frontend framework
N: Node.js → JS runtime environment for server-side development

Pros:

- Also uses JavaScript everywhere
- High performance and scalability with Node + Express.

Cons:

- Angular has a larger learning curve
- State and data flow in large apps can be complex to manage

Side note here

JAM

J: JavaScript → Frontend logic

A: APIs → Reusable services

M: Markup → Prebuilt HTML generated at build time

Pros:

- Fast by default: static assets served
- Strong security from minimal server surface area.
- Scales easily and cheaply

Cons:

- Long build times for large sites
- Complex dynamic features may require stitching multiple servers or functions
- Cache/invalidations can be tricky

Side note here

CHECKPOINT!

Ask a mentor if you have any questions so far

DEMO TIME

INSTALLATION GUIDE

SCAN HERE OR VISIT OUR GITHUB FOR THE INSTALLATION STEPS:

PLEASE FILL OUT THIS FEEDBACK FORM!

COMING UP:

- CodeJam applications close tomorrow night!!
  Apply ASAP
- Next Thursday is the Intro to Front End Workshop!

Check out our Instagram for workshop details: @mcgillcodejam