Welcome to Intro to DOM!

We will begin shortly.

As you're coming in...

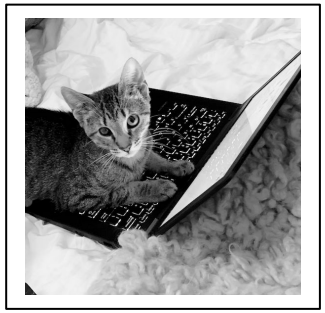
Please have your camera on, if possible and change your name on Zoom to include your pronouns

Drop into the Zoom chat: where are you calling in from?

Hi, I'm Tara! she/hers

Director of Curriculum
@ Kode With Klossy

Cat Mom
@ Cyclops



Coding facts at-a-glance:

- First love: javascript
- Gave me the most scars: java
- Favorite: python
- Controversial CS opinion: I like lightmode
- Non-controversial CS opinion: We need equity in the tech industry

Fun facts about me:

- I once accidentally hacked Target.com in a school project
- I'm a trivia host and ultra-marathoner in-training

Hi, I'm Hannah!

she/hers

Frontend Developer @ The Office of Experience

Denver, CO

Recently adopted @ Gemma

Coding facts at-a-glance:

- **I have been in the tech industry for about 5 years. Half of that time has been spent in education!**
- **JavaScript is great, but I really enjoy working with CSS. Micro-animations are my favorite!**

Fun facts about me:

- **I am a scuba diver, and have plans to learn cave diving**
- **I collect postcards from friends and have at least 300!**

Fast Friends: Round 1

- We will be going into breakout rooms of 2-3 people for 3 minutes
- When you get into the room, start by introducing yourself:
  - Your name, what your current role is
- Then, answer the this or that questions!
- If you have time leftover, dive deeper into your answers. Tell us why you like one thing or the other!

ROUND 1:

🐶😻 **Dogs or cats?**

🕹️🎲 **Video games or board games?**

🍬🍭 **M&M's or skittles?**

Welcome back!

Fast Friends: Round 2

ROUND 2:

 **Morning or night?**

 **Reading or writing?**

 **Donuts or bagels?**

Welcome back!

Agenda:

- <10 min> Introductions
- <5 min> Norms + Course Sequence
- <30 min> Introduction to the DOM
- <5 min> Break
- <30 min> Events
- <30 min> Beginning Our To-Do List
- <10 min> Close-Out

Norms and Course Sequence

What are our norms in this space?

- **You get out what you put in.**
- **You never need to share in a conversation if you don't feel comfortable, but we hope to make this a safe place**
- **Videos on when possible,** but it's ok if you are really exhausted from screens or have a lot of stuff going on to do no video when you need it.
- **People are in different places.** We will try to tailor supports, but also know that the same lesson will be old news to some and totally novel to others. Don't feel like you are way behind
- **Support each other!**
- **Proactive communication.** Feel free to ask questions whenever!

How do we approach teaching and learning?

- Teaching self-reliant skills
- Project-based learning
- Uplifting student voice

Course Objectives

The series will cover:

- Learning about DOM events and event listeners
- Responding to user interactions on our web pages
- Adding elements to the DOM to create a to-do list app
- Querying and updating a page after it's loaded
- Using browser (Chrome/Firefox) Developer Tools

Course Sequence

Day 1 - Tuesday

- Introduction to the course
- Introduction to DOM
- Learning about DOM events and event listeners
- Using browser (Chrome/Firefox) Developer Tools
- Wireframing and coding the HTML & CSS of a to-do list app

Day 2 - Wednesday

- Responding to user interactions on our web pages
- Adding elements to the DOM to create a to-do list app

Day 3 - Friday

- Querying and updating a page after it’s loaded
- Adding finishing touches to our to-do list app
- Final presentations

Course Resources

Please head to https://bit.ly/dom-resources to see our course Links & Resources!

Introduction to the DOM

Warm-Up

1. Think of something you want to buy!
2. Head to the website and write down 5 steps or ways you interacted with the website:
    i. Prompted me to type in my email to sign up for the newsletter
    ii. I typed in my email and pressed ‘Submit’
    iii. I clicked on the sweater I want
    iv. I chose different images to see different views of the sweater

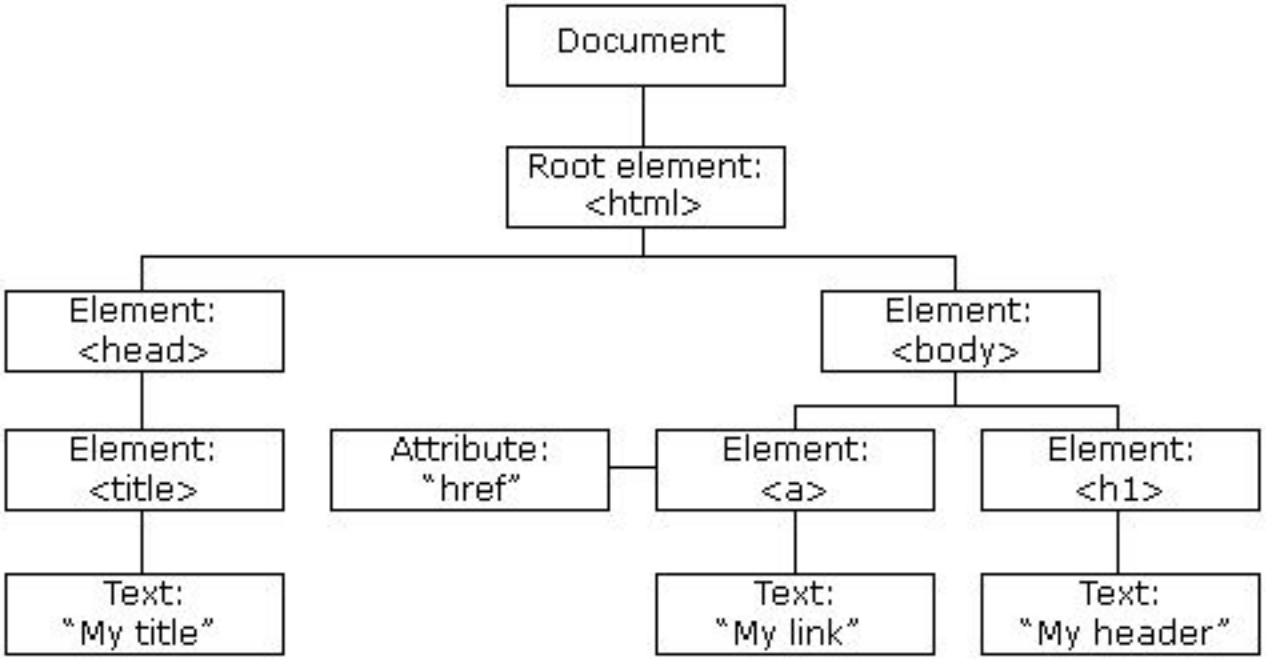
I clicked the button, and in response, JavaScript made that message appear.

In this lesson, we will learn how to change something on our site based on user interaction.

What is the DOM?

- The **Document Object Model (DOM)** is a data representation of the objects that comprise the structure and content of a document on the web
  - Represents the page so that programs can change the document structure, style and content
  - Represents the page as **nodes** and **objects**
- A web page is a document that can be either displayed in the browser or as the HTML source
  - The DOM representation allows it to be manipulated
  - Thus, it can be modified with a scripting language, such as JavaScript

The HTML DOM Tree of Objects



Why do we use the DOM?

With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

Let’s interact with the DOM!

- You’ll need Google Chrome for this exercise
  - Open the Google Developer console
  - Option+command+i to open the developer tools on Mac or right click and select Inspect

Your turn to manipulate the DOM!

1. Head to Wikipedia, Twitter, or any site you'd like and manipulate the DOM!
2. Take a screenshot and send it to a friend
3. Take another screenshot and share it with the other members here!
    a. Head to our Links & Resources document for the link to the slide deck

Woah! What just happened?

- HTML is a markup language used to display content in a browser. When we change the appearance of a web page, what we are really changing is the Document Object Model, which directly determines the appearance displayed in the browser.

- We can view and manipulate the Document Object Model by opening our developer tools, but when we do so the HTML is not changed.

DOM Methods

- The HTML DOM can be accessed with JavaScript
- **All HTML elements are defined as objects**
  a. HTML DOM **properties** are values (of HTML Elements) that you can get, set, or change
  b. HTML DOM **methods** are actions that you can perform (on HTML Elements)

Let’s take a look at an example!

- First, we’ll have to create a repl.it account
- Then, we’ll take a look at this code together!

In that example...

- `getElementById` was the method
- `.innerHTML` was the property

There are lots of methods and properties! (Check out the Links & Resources for more)

Your Turn: Intro to DOM Practice 1

- Without using copy and paste, your objective is to create another paragraph element with an ID of your choice to use the DOM to change the innerHTML to be your name!
- Once you've got it, head to the Links & Resources
  - Find the Intro to DOM Practice 1 Excel and drop in the answers to your code there
  - Make sure you copy and paste INTO the cell by typing into the function bar
- If you finish early, read the other changes you can do to the DOM with the link we drop into the chat!

Break

Events

Events

- Events are actions or occurrences that happen in the system you are programming, which the system tells you about so your code can react to them.
- For example:
  - The user selects, clicks, or hovers the cursor over a certain element.
  - The user chooses a key on the keyboard.

Blast From the Past!

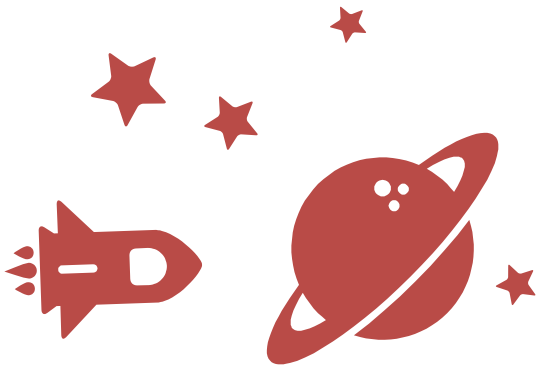
Warm-Up

1. Think of something you want to buy!
2. Head to the website and write down 5 steps or ways you interacted with the website:
    i. Prompted me to type in my email to sign up for the newsletter
    ii. I typed in my email and pressed 'Submit'
    iii. I clicked on the sweater I want
    iv. I chose different images to see different views of the sweater

16

Blast From the Past!

What other event listeners did you write down / other event listeners can you think of? Visit some of your favorite websites! Write them down and then we'll share in chat!



Events Examples

- For example:
  - The user selects, clicks, or hovers the cursor over a certain element.
  - The user chooses a key on the keyboard.
  - The user resizes or closes the browser window.
  - A web page finishes loading.
  - A form is submitted.
  - A video is played, paused, or finishes.
  - An error occurs.

Event Handlers

- To react to an event, you attach an event handler to it.
- This is a block of code (usually a JavaScript function that you as a programmer create) that runs when the event fires. When such a block of code is defined to run in response to an event, we say we are registering an event handler.

Event Handlers vs. Event Listener

- The minor difference between event handlers and event listeners is easy to understand:

  You can only have one event handler for a specific event type, but you can add multiple event listeners for it.

- **Generally, it is advised to use event listeners over event handlers because you can add multiple event listeners for the same event.**

Codealong: Let’s change the background color of a document when a button is clicked!

- Follow along with me!

You did it!

🎉

Now it’s your turn

- Using what we just did as starter code, you have three options:
  - OPTION 1: Change the event listener
  - OPTION 2: Change the outcome
  - OPTION 3: BOTH!
- Check out the resources available at the Links & Resources for inspiration!

Breakout Rooms

<12 minutes> Groups of 3-4

- **~ 3 minutes per person**
  - **Share your output/website**
  - **Share your code**
  - **Share about the process**
    - **What resources did you use?**
    - **What was a challenge that you overcame?**
    - **What did you learn?**
    - **What are you still curious about?**
  - **Answer any questions that other people in the group may have!**

Beginning Our To-Do List

Your Task

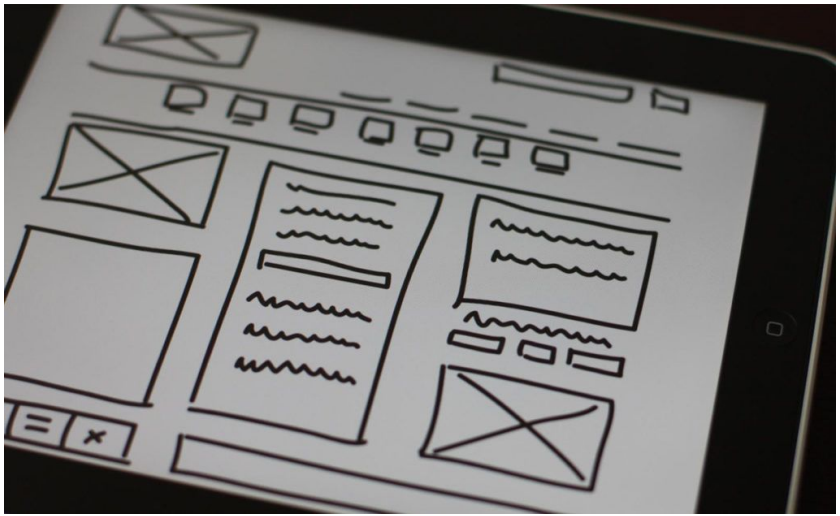
Build a to-do list using HTML, CSS, and JavaScript.

Your To-Do list will include:

- A space to type a to-do
- Add the item to an ongoing list
- Mark an item as complete

First Step: Wireframe!

Take 7 minutes and wireframe what your To-Do List website would look like.



Breakout Rooms

<3 minutes> In groups of 3-4, share your wireframe!

Second Step: Begin coding the HTML & CSS of your to-do list

- In breakout rooms of 4-5, you're going to begin coding the HTML & CSS of your to-do list
- Some elements that you may want to consider adding:
  - Form
  - Input
  - Div
  - Button
- Collaborate with each other!
- Tara and Hannah will stay in the main room, come back to the main room if you need help. We also may jump from room to room!

Closing Out

Course Sequence

Day 1 - Tuesday

- Introduction to the co
- Introduction to DOM
- Learning about DOM e event listeners
- Using browser (Chrom Developer Tools
- Wireframing and coding the HTML & CSS of a to-do list app

Day 2 - Wednesday

- Responding to user interactions on our web pages
- Adding elements to the DOM to create a to-do list app

Day 3 - Friday

- Querying and updating a page after it's loaded
- Adding finishing touches to our to-do list app
- Final presentations

Close-Out Padlet

Next Steps

1. Continue working on the HTML and CSS of your To-Do List App
2. Email Tara or Hannah if you have any questions!