



Week 1 Workshop

Web Development
Fundamentals



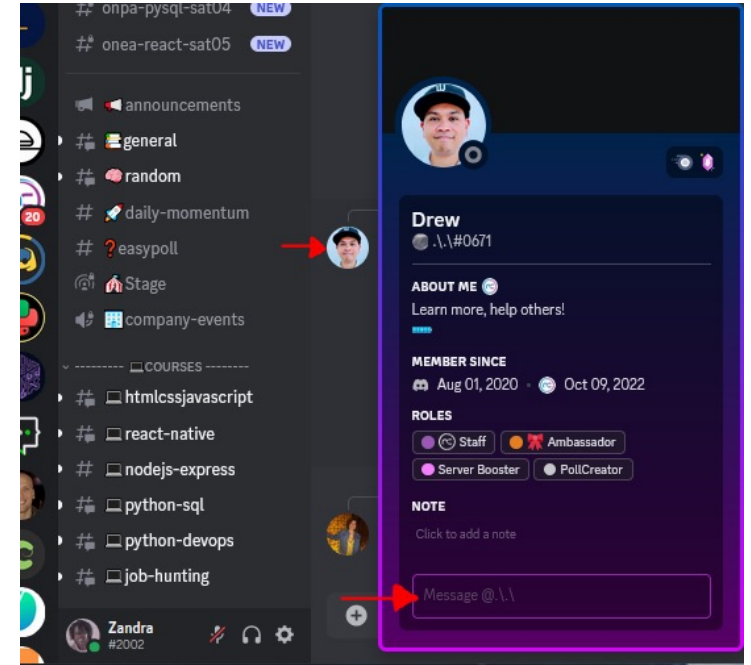
Agenda

Activity	Estimated Duration
Welcome & Housekeeping	5 mins
Instructor & Student Introductions	20 mins
Your Bootcamp Overview	20 mins
Week 1 Review	1 hour
Break	10 mins
Workshop Introduction & Demo	5 mins
Workshop Assignment	1 ½ hours
Profile Show and Tell	20 mins
Project introduction and Workshop Wrap-up	10 mins

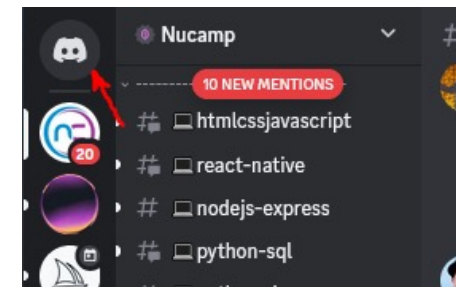


Using Discord

- **Make sure you are in the following Discord channels:**
 - #web-fundamentals
 - Our class channel (e.g. #onpa-htmlcssbt-sat03)
- **Complete the following tasks right now in our class channel:**
 - Post a message in the class channel and tag me or a classmate
 - Start a thread on someone else's message
 - Upload a code file or attachment
 - Send me a DM and say hello



Send a DM: Click your instructor's image



Your DMs are under the top-left Discord icon



Asking For Help in Discord Channels

- Post to the class channel or the main #web-fundamentals channel (not a DM)
- Instead of saying "Can anyone help me with a problem?"
 1. State which exercise you are working on,
 2. **ask a direct question about your problem/issue,**
 3. include what you have already tried,
 4. attach your relevant code file(s), and
 5. if you are seeing error messages, attach a screenshot of the errors
- Mention me with **@name** in your post so I am aware of your question, even if I am unable to respond right away



Your Bootcamp Learning Routine

- **Daily tasks every day (2+ hours):**
 - ✓ Watch videos and code along
 - ✓ Complete exercises and challenges
 - ✓ Work on your portfolio project
 - ✓ Use any extra time for additional study and practice
 - ✓ Don't get stuck ... ask for help!
 - ✓ Follow the "20-minute rule"
 - ✓ Help other students in Discord or form a study group
- **4-hour workshop every weekend**
 - ✓ Be ready to code
 - ✓ Be prepared to discuss
 - ✓ Read the instructions ahead of time
 - ✓ Work in small groups to complete the assignment
 - ✓ Turn in your work on time
- Next week's content is unlocked after you complete all tasks for the week and submit your feedback



**IT WILL BE HARD
IT WILL IMPACT
YOUR ROUTINE**



The 20-minute rule

If you ask for help too soon:

- You will not learn how to tackle problems or remove obstacles, which is core to coding.
- Remember the process or path that resolved the issue is more important than the solution.

If you ask for help too late:

- You will get frustrated and tired.
- You will miss an opportunity to go deeper on the same topic (time is limited).

10-minute rule during workshops:

- During the week, go by the 20-minute rule.
- During workshops, go by a 10-minute rule – try to solve the issue yourself (or with your classmate, if working together) for 10 minutes before asking your instructor for help.



20 minutes, then ask for help!



Next 4 Weeks Overview

- **Week 1:** This week! Introduction to HTML
- **Week 2:** Introduction to CSS
- **Week 3:** Introduction to Bootstrap
- **Week 4:** Bootstrap JS Components, Build, and Deploy



Your Learning Goals

- How to **WRITE** good, clean code
- How to **READ** code and documentation
- How to **TALK ABOUT** your code with others
- How to **DEBUG** your code
- How to **SEARCH** for answers

James Pritchett - Instructor

I tell every class that the key takeaway from every class is not an expert level knowledge of how to do everything.

The key takeaway should be having the knowledge that something is possible and that there's a function/pattern/library that enables it.

I don't know the syntax of everything react or everything javascript, but i know how to search for it. That's good enough.



Week 1 Review

Introduction to HTML



Most important to know about HTML:

HTML Elements

- What is an element? What is the syntax?

HTML Attributes

- What is an attribute? What is the syntax?

HTML Document Structure

- What elements and attributes are required?
- How should a document be structured?

We will discuss all of these on the following slides.



HTML Element Syntax

Opening tag

Content

Closing tag

`<h1>`Hello World!`</h1>`



Void Elements



Void elements have only one tag

- Called "void" or "empty" because it doesn't hold content between start and end tags
- The tag for a void element is called a **self-closing** tag because it's both an opening and closing tag in one.
- They can optionally end with `/>` but this is not required:
- Void elements can have attributes.

Examples: ``
``



Semantic HTML Elements

Semantic describes what it is:

```
<header>
  <h1>This is a title</h1>
  <p>This is a subtitle</p>
</header>
<article>
  <h2>This is a section heading</h2>
  <p>Here is some long text</p>
  <p>Here is more text</p>
</article>
```

Non-semantic could be anything:

```
<div>
  <span>This is some text</span>
  <span>This is more text</span>
</div>
<div>
  <strong>This is some more text</strong>
  <span>Some text here</span>
  <span>Some more text here</span>
</div>
```



Block vs. Inline Elements

Block Elements:

- Takes the full width of a page
- Block other elements
- Examples:
 - `<h1>`
 - `<p>`
 - ``
 - `
`
 - `<div>`

Inline Elements:

- Takes only as much width as it needs
- Stays in the same line as other inline elements
- Examples:
 - ``
 - `<a>`
 - ``
 - ``



HTML Attributes

Opening tag

Attribute

Content

Closing tag

`<section id="about">About Me</section>`

key

value

Attributes provide additional context for the HTML element!



HTML Attributes

- Many elements have a set of applicable **attributes**.
- Attributes are placed **in the start tag**.
- Separate multiple attributes with a space
- Order does not matter
- Every attribute is either:
 - a **key-value pair** with quotes **key="value"**
 - or a **boolean** -- no value, just the attribute name such as **(controls)**

`<video src="myVideo.mp4" loop controls>` is the same as `<video src="myVideo.mp4" controls loop>`



Attributes: Image element

The `` element has several attributes!

- **src** -- this is required!
- **alt** -- not required but you should always have one
- **width** -- optional
- **height** -- optional

```

```

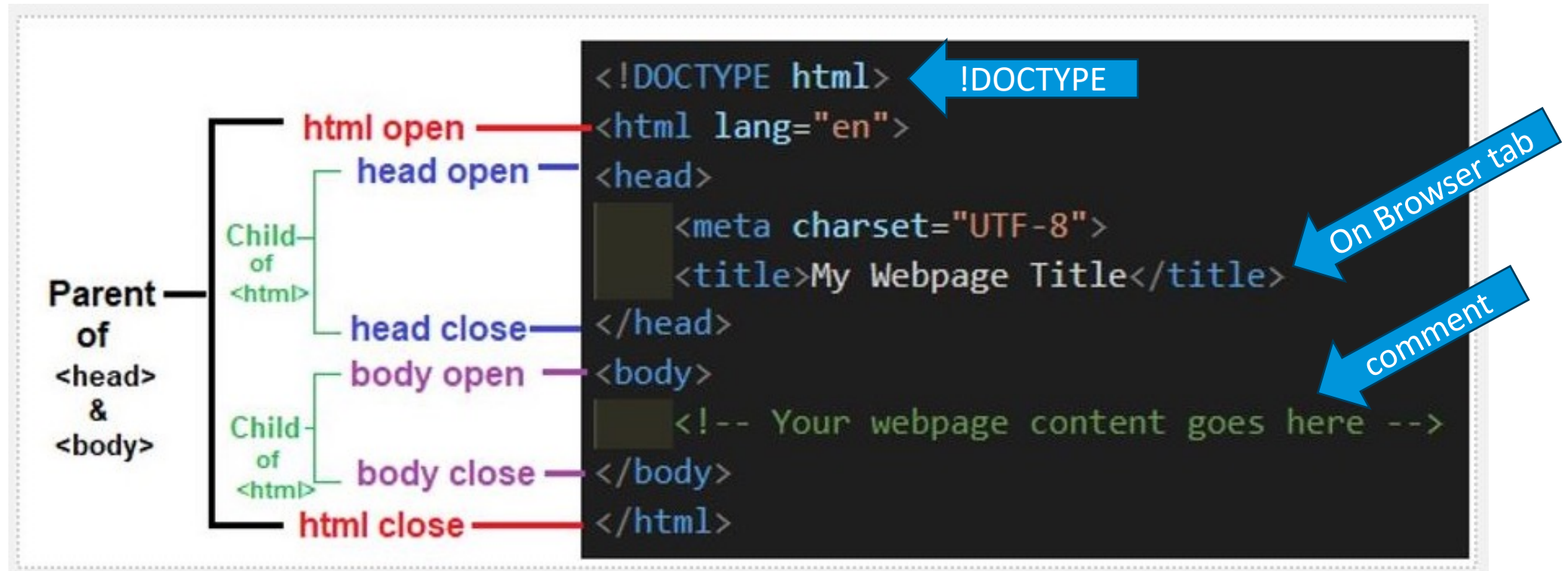


Attributes: Anchors (hyperlinks)

- The `<a>` tag is used to make a link to...
 - Someplace on the interwebs (URL)
 - Another page on the same website (relative link)
 - Another spot on the same page
- Discuss:
 - What **attributes** are required for this element?
 - What goes in the **href** attribute?
 - Absolute vs. relative paths?
 - How do you link to an email address or phone?
 - How do you link to another element on the same page?



Basic HTML5 Document Structure



- When an HTML element "opens" it is considered the **parent** of the elements directly inside of it prior to its "closing"
 - e.g. In the above code, the **head** element is the parent of the **meta** and **title** elements
- You'll sometimes hear the term "container" or "wrapper", this just means the parent element
 - e.g. "wrap an `<a>` element around an `` element"

```
<a href="#">
  </img>
</a>
```



<!-- Comments in HTML -->

- Comments are not visible in the browser (ignored when the page is rendered)
- Three reasons to use comments:
 1. Leave a helpful note for yourself
 2. Leave notes for other developers
 3. "Comment out" code that you don't want to be rendered, temporarily



HTML Tables

- Used to display tabular data – not for page layout
- Simplest example:

```
<table>
  <tr>
    <td>Row 1 Cell 1</td>
    <td>Row 1 Cell 2</td>
  </tr>
</table>
```

- Use `<thead>` and `<tbody>` as containers for table header and body rows
- Use `<th>` to create header cells with bold, centered text content
- Which of the above elements accept text content?



HTML Forms

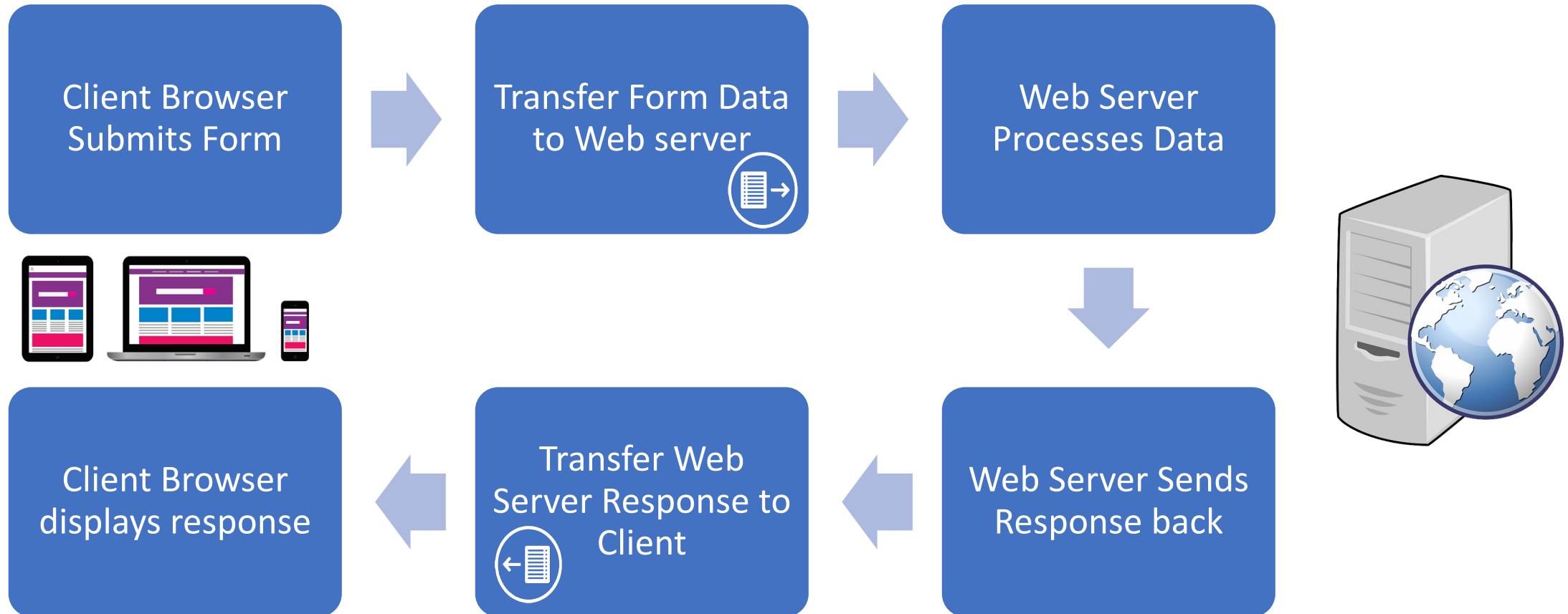
- `<form>` element
 - `action` attribute
 - `method` attribute “post” or “get”
 - `enctype` attribute for file transfer
- `<input>` element
 - `type` attribute
 - `name` attribute
 - `id` attribute
- `<label>` element
 - `for` attribute
- Other form elements:
 - `<fieldset>`
 - `<legend>`
 - `<textarea>`
 - `<select>`
 - `<option>`

Discuss:

- How do labels get linked to inputs, and why does that matter?
- Why is the “name” attribute so important, and what does it have to do with key-value pairs?
- What are some other input types? How do these look on different devices?



Form Submission Data Flow





Workshop Assignment

- Create a **profile.html** page with information about yourself
- Required (minimum):
 - proper document structure
 - 2 photos
 - 2 anchor elements
 - 1 audio element
 - 2 unordered lists
- Optional: add more content!

Profile - Minae Lee

▶ 0:00 / 2:39 ———— 🔊 ⋮



About Me

- I live in Bellingham, Washington
- I was born in Seoul, South Korea
- My current occupation is an instructor and curriculum developer for [Nucamp](#).
- I started learning to code in 1990.
- One goal I'd like to accomplish within my lifetime is to write a novel.

Interests

- Painting and drawing
- Playing music (guitar, ukulele, singing)
- [The game Go](#)



- Camping and hiking



Portfolio Project Overview

- Over the next few weeks of this course, you will be working on a personal project alongside your lessons
- You will build a professional portfolio (online resume) webpage:
 - Similar to today's assignment, but showcasing your professional work
 - You will add future projects to your portfolio page
- Your webpage must use HTML, CSS, and Bootstrap
- See instructions in the course after the workshop



Wrap-up

- Submit your workshop project:
 - Create a screenshot of your **profile.html** page as shown in Chrome or Firefox. See the "Submission" section at the bottom of the written instructions to find out how to create a full-page screenshot.
 - Submit both the **screenshot** and your **profile.html** page at the bottom of the assignment page in the learning portal.
- Answer the Week 1 Feedback survey
- Start working on your project



**Happy
Learning!**