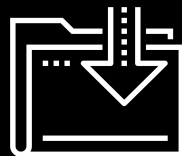




# Intro to HTML

Data Boot Camp  
Lesson 11.1



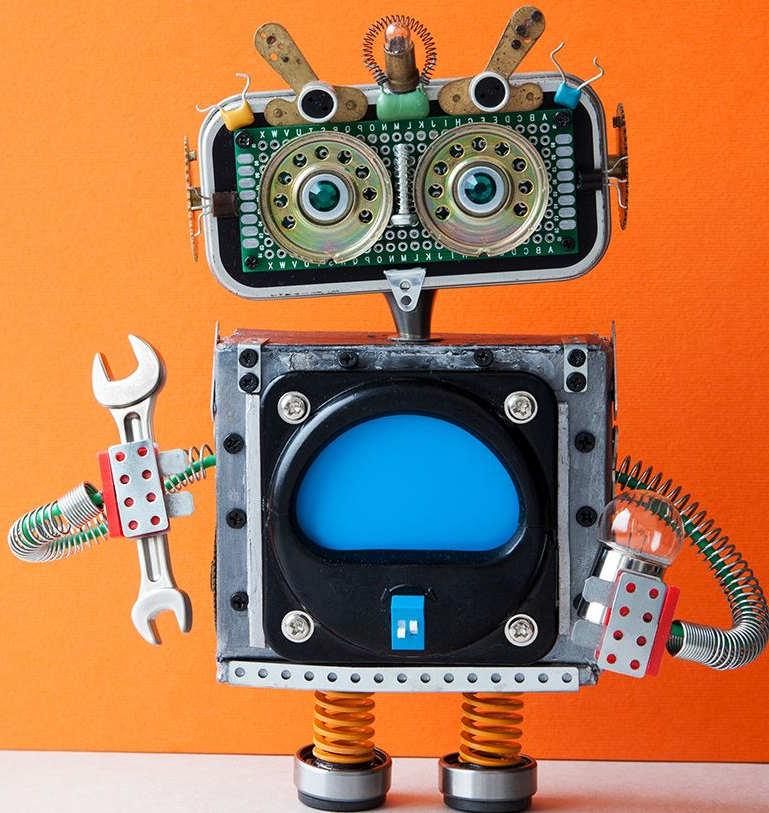


# All the Technologies!

---

This week we will cover the following:

- HTML
- CSS
- Bootstrap CSS
- JavaScript
- JavaScript Charting Libraries
- And more!



# Class Objectives

---

**By the end of today's class you will be able to:**



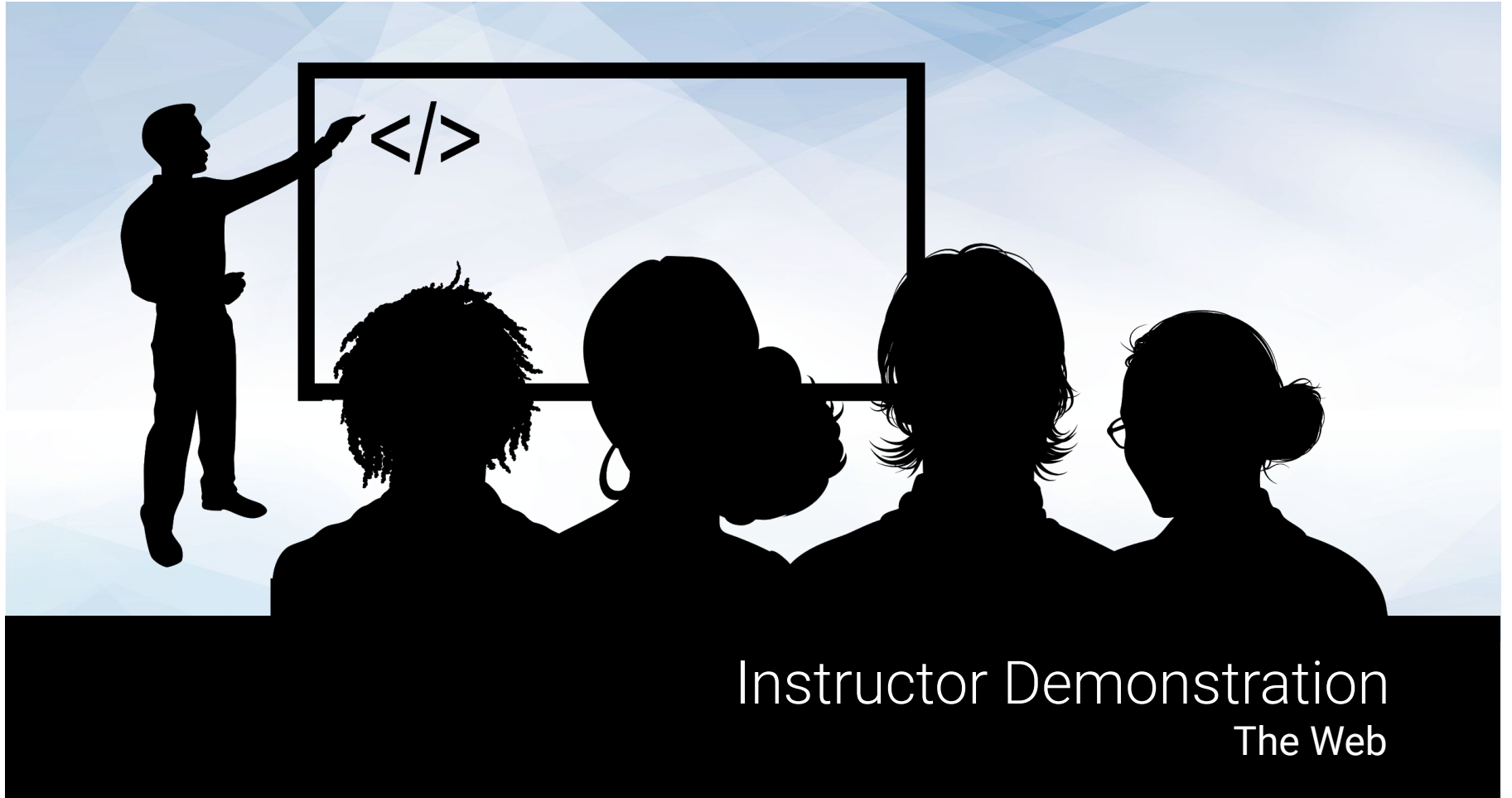
Fully grasp the mechanics of HTML, CSS and JavaScript as their role building websites.



Comprehensive understanding of the basic parts of an HTML web page and how to create one from scratch.



Learn to cover and utilize some of the most common HTML tags and selectors.



# Instructor Demonstration

## The Web

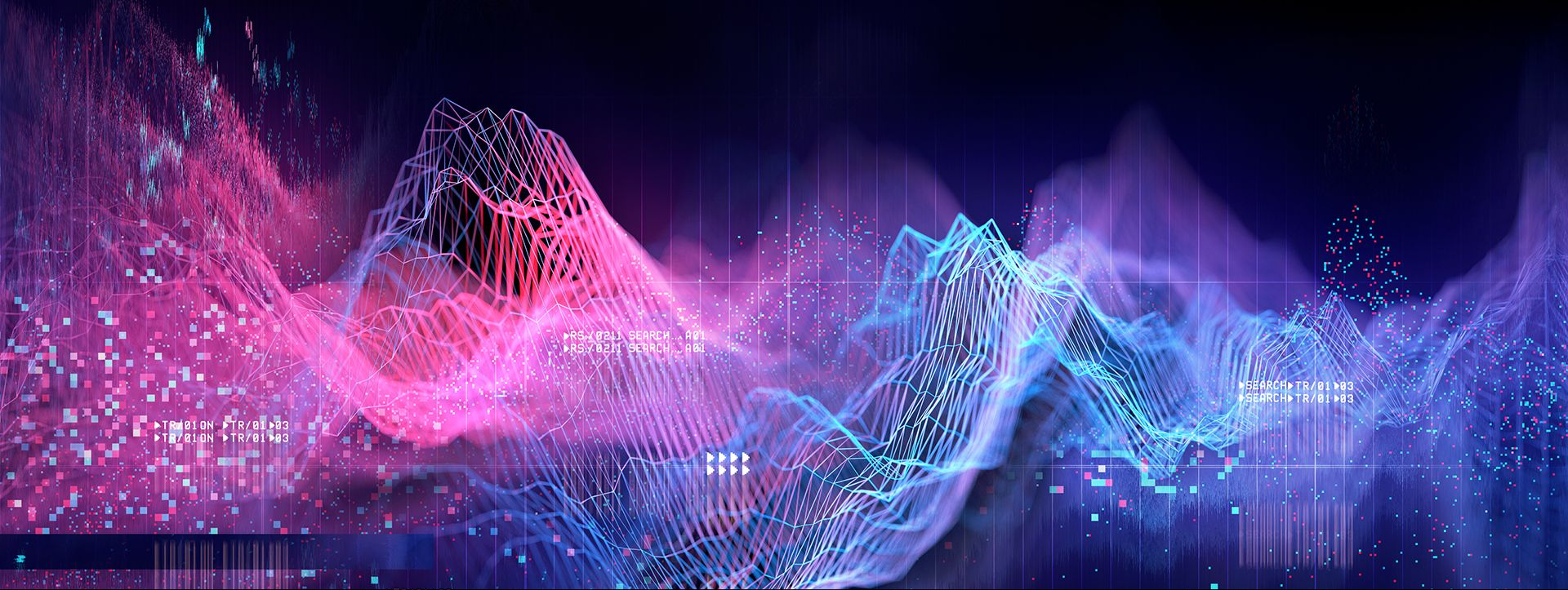


# The Web

## What's the Web?

Web developers build them into a special type of document: a webpage. These webpages are then accessible to other people on the internet.





Data analysts can use the web to  
**tell stories with data**  
and reach millions of people.

# The core Three Front-End Web Technologies

## The Web

---

**HTML**



**JS**



**CSS**







# Instructor Demonstration

## Hello HTML

# `<title>Intro to HTML</title>` Hello HTML

---

## HTML5

- HTML is one of the three base languages behind every website.
- It defines all the basic content and a bit of formatting.



# Periodic Table of HTML5 Elements

## Hello HTML

html																col	table
head	span										fieldset	form	body	h1	section	colgroup	tr
title	a										meter	select	aside	h2	header	caption	td
meta	rt	dfn	em	i	small	ins	hr	p	div	blockquote	legend	optgroup	address	h3	nav	menu	th
base	rp	abbr	time	b	strong	del	br	figcaption	ol	di	label	option	datalist	h4	article	command	tbody
link	noscript	q	var	sub	mark	kdb	wbr	figure	ul	dt	input	output	keygen	h5	footer	summary	thead
style	script	cite	samp	sup	ruby	bdo	code	pre	li	dd	textarea	button	progress	h6	hgroup	details	tfoot
					device	video	audio	track	canvas	iframe	source	param	object	embed	map	area	img

# Hello HTML

HTML elements are rendered by the browser as visible parts of a webpage





# Hello HTML

---

## HTML Syntax (Basic)



# Hello HTML

---

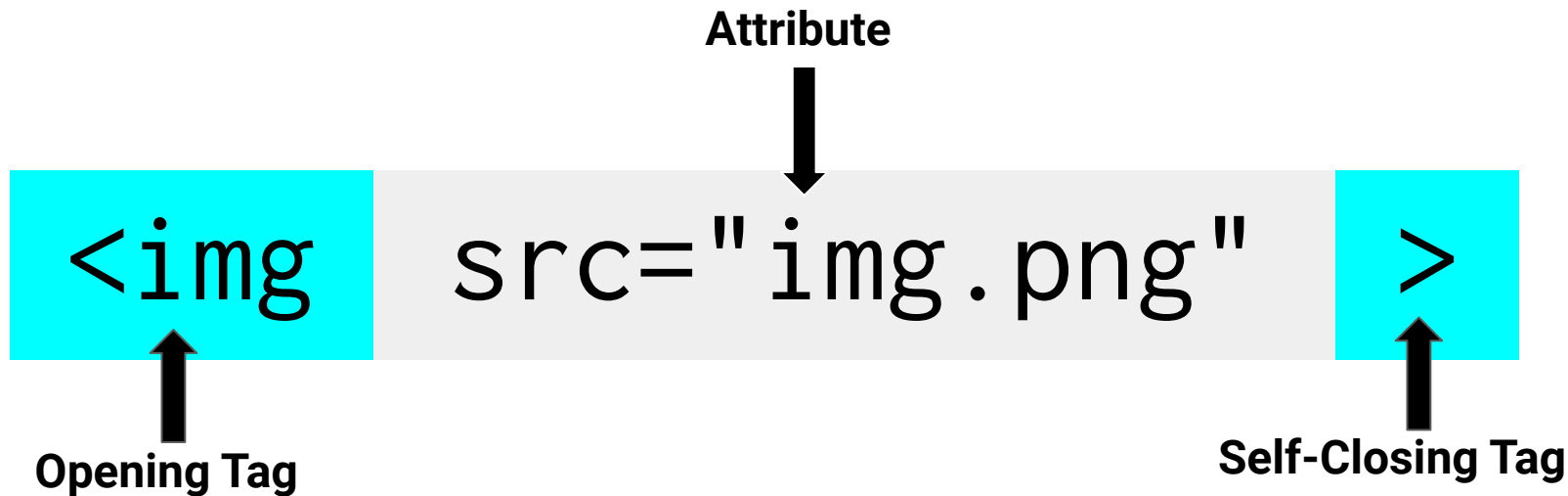
## HTML Syntax (with Attribute)



# Hello HTML

---

## Tricky Tags (Self-Closing)



# Hello HTML

```
1  <!DOCTYPE html>
2  <html lang="en-us">
3
4  <head>
5      <meta charset="UTF-8">
6      <title>My First Page</title>
7  </head>
8
9  <body>
10     <!-- Header -->
11     <h1>Hello World!</h1>
12
13     <!-- Image -->
14     
15     <br>
16
17     <!-- Link with New Tab -->
18     <a href="https://www.google.com" target="_blank">Opens new tab</a>
19     <br>
20
21     <!-- Bold Link -->
22     <strong><a href="https://www.youtube.com">This is a bold link</a></strong>
23     <br>
24
25     <!-- Placeholder link -->
26     <a href="#">This is placeholder link goes nowhere, for now</a>
27
28 </body>
29
30 </html>
31 |
```





## Activity: Inspect Hello, HTML

In this activity, you and your partner will spend a few minutes going over the example coded out in the last example and use Chrome Devtools to inspect it.

**Suggested Time:**  
5 Minutes



# Instructions: Activity: Inspect Hello, HTML

---

- Open the example from the last activity slacked out to you in Chrome and open Chrome Devtools. You can launch Chrome Devtools by doing any of the following:
  - Right click the rendered HTML document inside of Chrome and clicking the `inspect` option.
  - Pressing `F12` on Windows while in Chrome.
  - Pressing `command + option + i` on a Mac.
- Make sure the `Elements` tab is selected.
- Study the code here and see how it compares to the actual HTML document slacked out.
- Experiment with modifying the HTML document in VS Code and reloading the web page and inspector.
  - What happens when you remove an element's closing tag or misspell `head` or `html`?



**Time's Up!** Let's Review.



## Activity: Research HTML Tags

In this activity, you and your partner will create a list of HTML elements and their descriptions. You should create a valid, informational HTML document in which they use and describe as many of their researched elements as possible.

**Suggested Time:**  
10 Minutes





# Instructions: Activity: Inspect Hello, HTML

---

- With a partner, research and create a list of different HTML elements and their descriptions.
- This list should be a combination of elements we've gone over already and some new elements we haven't yet discussed.
- Try to list as many elements as possible.
  - Identify which elements are void elements (self-closing), and which are not.
  - Create a valid, informational HTML document where you use and describe and use as many of the different researched elements as possible.




## Everyone: Discuss HTML Tags

In this activity, everyone will discuss some of different HTML elements we were able to research in the last activity.

**Suggested Time:**  
10 Minutes



An overhead photograph of a group of people sitting around a dark wooden table. Several hands are visible, some holding smartphones and others typing on laptops. The scene suggests a collaborative work environment. The text is overlaid in the center of the image.

Different  
**HTML** elements  

---

we were exposed  
to so far!



## Activity: My First HTML

In this activity, you will create a simple HTML page.

**Suggested Time:**  
15 Minutes





# Instructions: Activity: Discuss HTML Tags

---

- In a new HTML file, create the basic structure of an HTML document and include in it the following:
  - DOCTYPE declaration
  - head tag with nested title tag
  - h1 tag with a title of your choice
  - Embed an image
  - Create the following three links on your page:
    - One link that leads to the website of your choice. Give this link an attribute `target="_blank"` so that it opens a new tab when clicked.
    - Create a second link to a different website, but give this link bold text.
    - Make the third link a placeholder so it goes nowhere.
- **HINT:** You should be checking the rendered HTML in Chrome as you code to make sure you're going in the right direction.
- **Bonus:**
  - Create an ordered list of steps to make a sandwich.
  - Create an unordered list of 5 bands/musicians you like.
  - Use an alternate way of separating links without line breaks.
  - Embed a YouTube video of your favorite band/musician.





**Time's Up!** Let's Review.





Countdown timer

**15:00**

(with alarm)



## Activity: Fix the HTML

In this activity, you will be tasked with fixing mistakes in a HTML document.

**Suggested Time:**  
5 Minutes



# Instructions:

## Activity: Fix the HTML

---

- Open the `index.html` file inside the `Unsolved` folder.
- This HTML document has several errors. While it may seemingly load browser without many issues, this is not valid HTML.
- Correct the document and leave a comment above each error describing what you changed.
  - Refer to the [Mozilla HTML Documentation](#) to research any unfamiliar HTML tags.



**Time's Up!** Let's Review.



# Instructor Demonstration

## Working with tables



# The Function of each element that goes into making a table

## Working with tables

---

- **table**: Acts as a container for our rows and columns.
- **tr (Table Row)**: These elements serve as rows for our table. They contain a row of table cells.
- **th (Table Header)**: These are table cells that are for containing column titles. These will have bold text by default.
- **td (Table Data)**: These are table cells for holding data. These are not column titles.
- Optional table tags that can be used to give semantic meaning to our table. These include:
  - **thead (Table Head)**: Acts as a container for the row(s) holding our table headers.
  - **tbody (Table Body)**: Acts as a container for the rows holding our table data that is not part of the table head.
  - **tfoot (Table Footer)**: Acts as a container for bottom rows in a table which acts as a summary of the tables content. The tfoot element is rendered at the bottom of the table by default, regardless of which row(s) it's holding.



## Activity: Working with tables

In this activity, you will create an HTML5 table modeled from a spreadsheet. You will use a border and colspan properties, as well as make use of the `thead`, `tbody` and `tfoot` elements.

**Suggested Time:**  
15 Minutes



# Instructions: Working with Tables

---

- Open the [Resources](#) folder. Here we have a relatively simple spreadsheet.
- Create a new HTML document and create a table to represent the data in the spreadsheet.
- Try to get the formatting as close to the spreadsheet as possible.
- **Bonus:** Use table `colspan` and `border` attributes to make the table look even more similar to the spreadsheet.
  - Group the completed table using a `thead`, a `tbody`, and a `tfoot` element.
    - Check you **SLACK** for a good example of these semantic table elements being used.



**Time's Up!** Let's Review.



## Activity: More with Images and Links

In this activity, you will create a webpage using images downloaded from their Jupyter notes from previous project.

**Suggested Time:**  
15 Minutes



# Instructions: Activity: Fix the HTML

---

- Create a new project folder. Inside, create a new HTML file and name it `index.html`. Fill this document in with a basic HTML skeleton.
- Launch Jupyter Notebook, and open any previous project created using `matplotlib`. Download all images of graphs created with pyplot to your local project folder.
- At the top of the HTML page, add a H1 tag containing the name of your chosen project.
- For each downloaded image, add one image element to your `index.html` file. Set the `src` attribute of each image to the path to a different image in your project folder. Be sure to add a description for each image in the `alt` attribute.
- Add a horizontal rule between each image.
- Underneath each image, add a description of its graph in italicized text using the `<em>` element.
- Add functionality so that when an image is clicked, it will open a new tab to that image's URL.
- **Bonus:**
  - Take advantage of the `figure` and `figcaption` elements to have more semantic markup.



**Time's Up!** Let's Review.



**Not Done yet!!!  
Almost there!  
TAs please introduce the HW**



*The  
End*