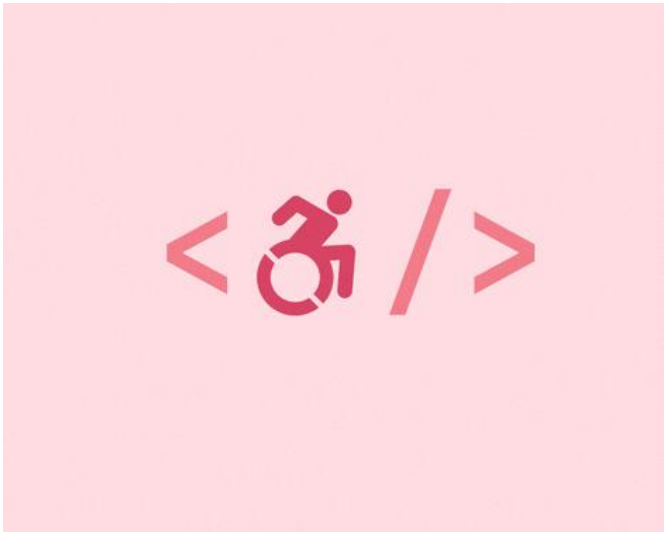
Web Fundamentals

Session-08

Accessibility & Best practices

Agenda



01 Accessibility Intro

02 Accessibility Tools

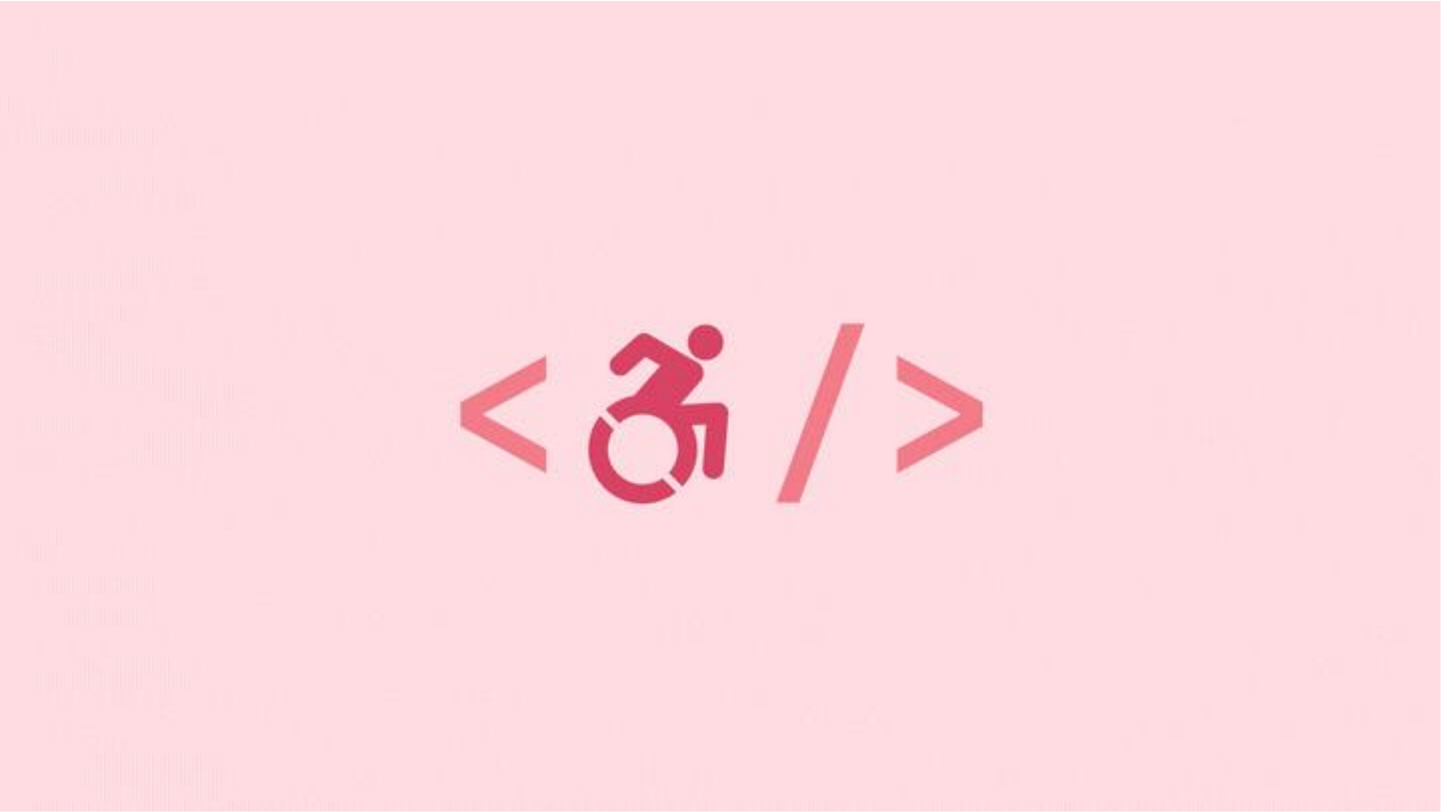
03 Labels, Aria Roles, Alt, Tab Index

04 Performance, Security

05 HTML, CSS Best Practices

06 Hands-On, Summary & Resources

What is Accessibility?



Accessibility

- **Visually impaired** users will generally use a ***screen reader*** that describes the page using speech.
- **Low vision** users will generally use large fonts or something called a ***screen magnifier***.
- **Motor-impaired** users, who can't use a mouse, might use some kind of ***special keyboard*** or even a ***voice control interface*** to interact with your app.

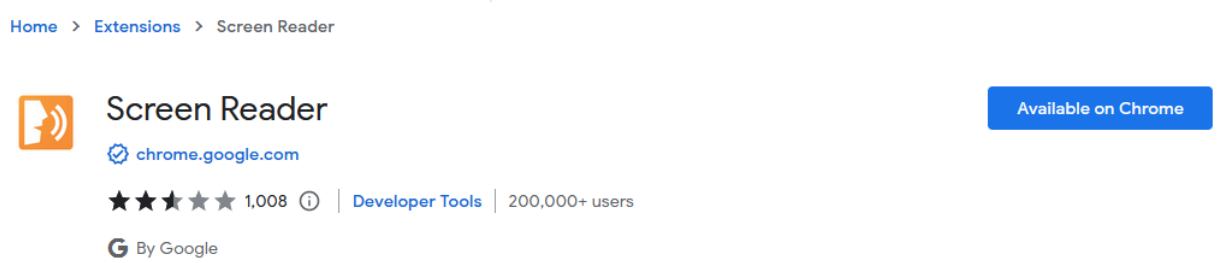
Why Accessibility?

- Improved search engine optimization (SEO)
- Compliance with laws and regulations – Many countries have rules that websites need to be accessible
- Better User Experience

Accessibility Tools

Screen Readers
Talkbacks

Demo
Walkthrough



Caret Browsing
Navigate the web using keyboard instead of touchpad or mouse.

Dark Reader
Apply dark themes for websites.

JustRead
Create distraction free reading environments.

High Contrast
Adjust colour contrast in webpages .

<|>esto

No <Div> Soup

No Salad

Create meaningful HTML structure

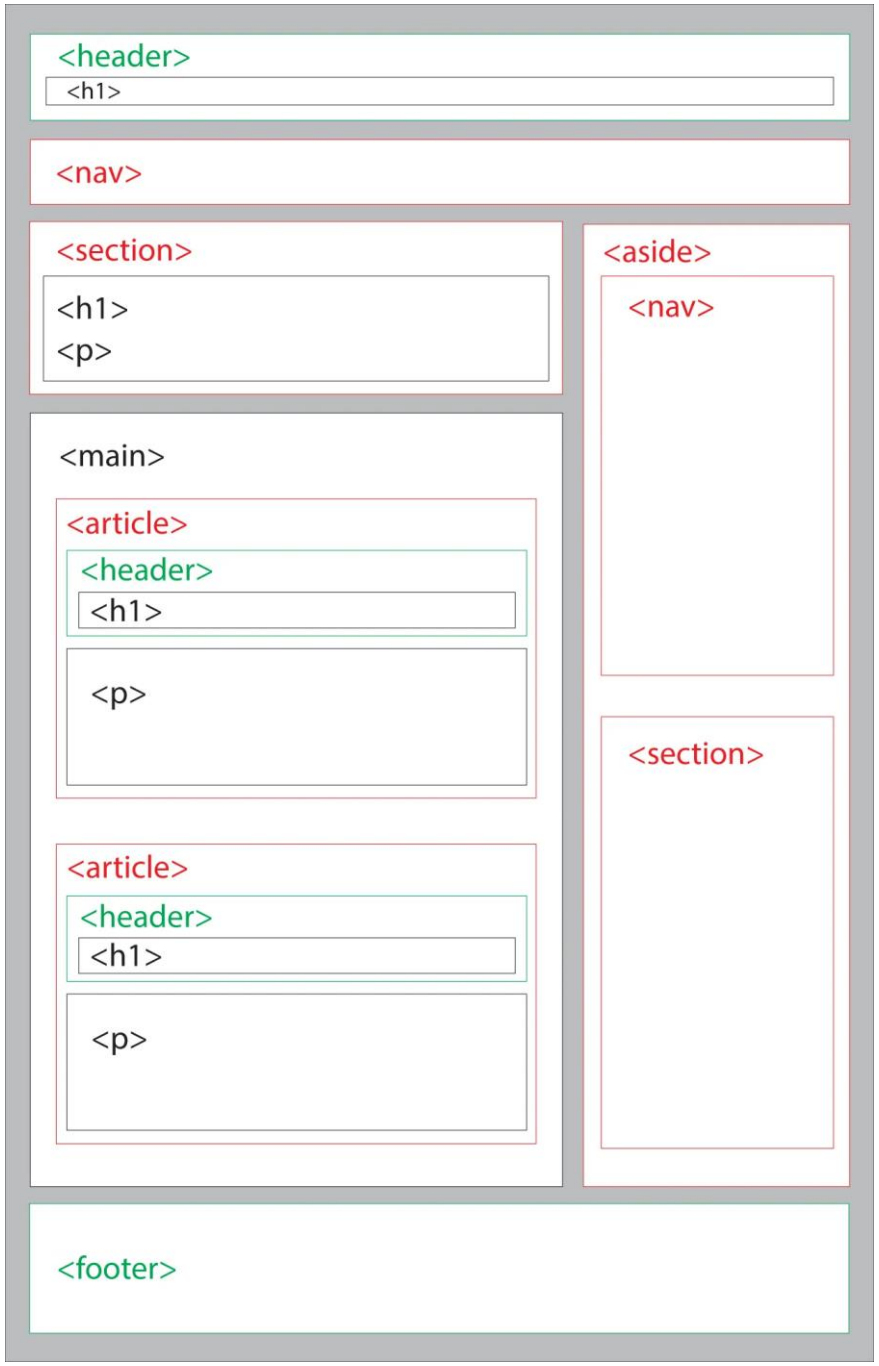
Screen Readers Can’t Identify Divs & Spans
- Not focusable
- Not activated by keyboard

```
// Instead of 'span' use 'a' tag for onClick event
<span onclick=""></span>
<a href="#"></a>

// Instead of 'div' use 'button' tag for onClick event
<div onclick="">Click Me</div>
<button onclick="">Click Me</button>
```

<|>esto

Accessibility #1
- Semantic HTML



Accessibility #2

All images should have the alt attribute

All Form controls should have labels

```
<label for="firstname">First name:</label>
<input type="text" id="firstname"/>
```

```
<img src=”chart.png” alt=”[description]”>
```

Accessibility #3

For Keyboard only users use “TabIndex” to focus

Order in which elements receive focus when the user navigates

```
// Natural tab order
<div onclick="" tabindex="0"> Clicky 1</div>

//Custom tab order
<div onclick="" tabindex="7">Clicky 2</div>

//Focusable but not in tab order. This is useful when hiding focusable controls f
<div onclick="" tabindex="-1">Clicky 3</div>
```

Accessibility #4

***Role*: Purpose of HTML element**

```
<!-- Widget Roles -->
<div role="button" tabindex="0">Click me!</div>

<input type="text" role="textbox" aria-label="Type your name here">

<div role="slider" aria-valuemin="0" aria-valuemax="100" aria-valuenow="5
```

Accessibility #5

***Aria-Label*: label or name for an HTML element**

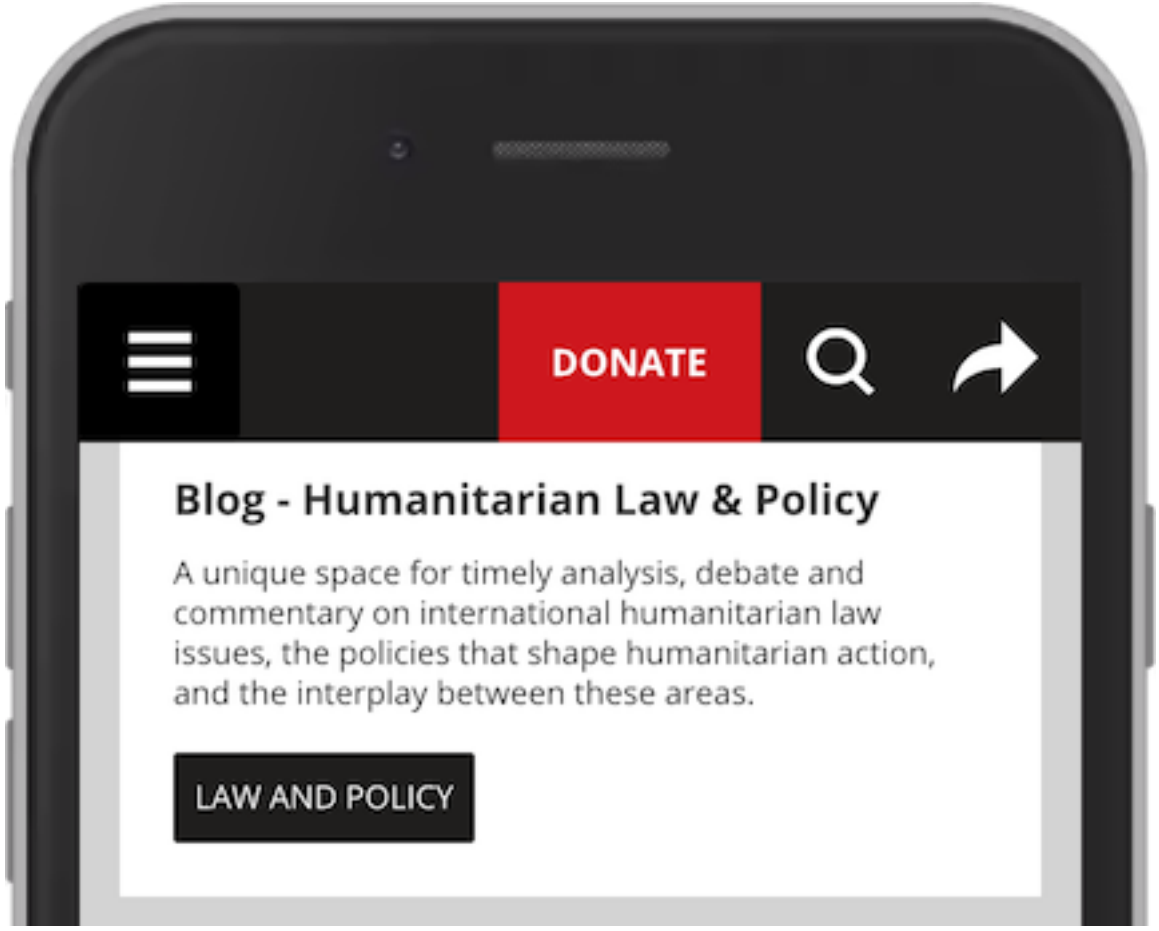
```
<!-- Widget Roles -->
<div role="button" tabindex="0">Click me!</div>

<input type="text" role="textbox" aria-label="Type your name here">

<div role="slider" aria-valuemin="0" aria-valuemax="100" aria-valuenow="5
```

Accessibility #6

***Link Text*: A link text should explain clearly what information the reader will get**



Accessibility #7

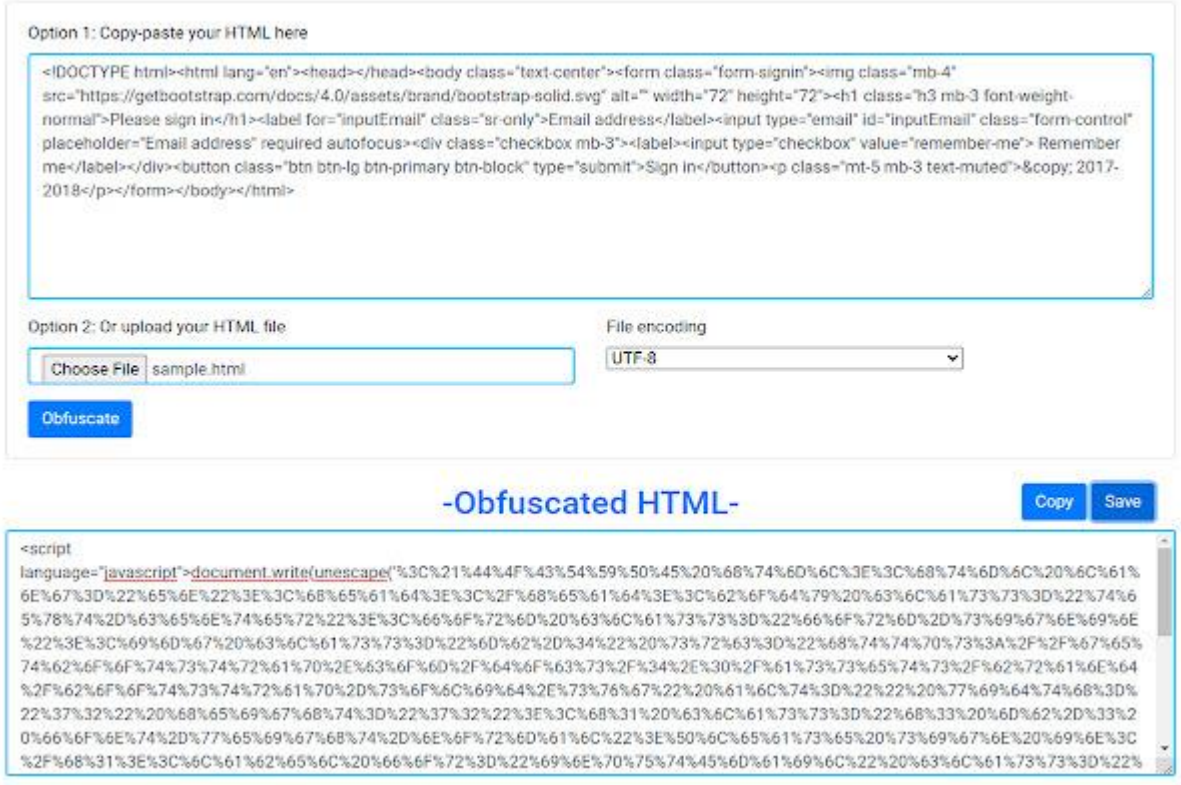
***Contrast Ratio*: web page need good contrast so that we make sure that it is perceivable for visually challenged users.**

Performance Best Practices:

- Minify HTML, CSS, JS
- Lazy load assets (images/files/css/js)
- Reduce size of DOM, Load only what is required

Security Best Practices:

- Encrypt/Obfuscate HTML, CSS, JS
- Use HTTPS for data transmission
- Always validate user inputs (interaction)



Best Practices- HTML

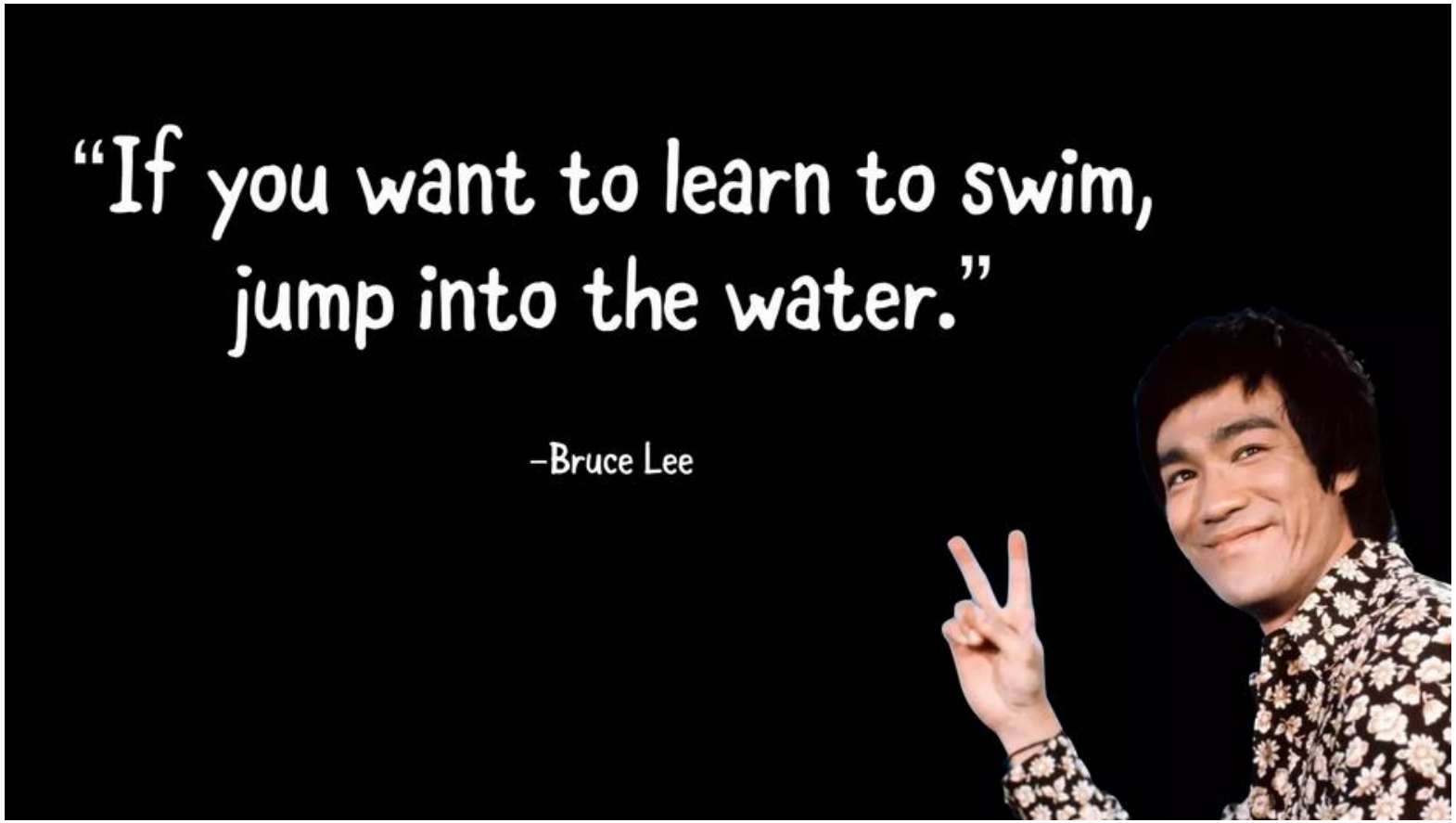
- Indent and format properly.
- Use semantic HTML.
- Optimize images for web.
- Test code on browsers.
- Use fallback fonts.
- Don’t keep empty ***“src”*** or ***“href”***
- Use accessibility best practices.
- Add meaningful comments to describe your code

Best Practices- CSS

- Indent and format properly.
- Avoid inline styles
- DRY- Do not repeat yourself, Reuse by using classes
- Use external CSS.
- Use CSS selectors efficiently.
- Keep CSS organized.
- Test code on browsers.
- Use responsive design.

Best Practices for your mini-project

- Minify CSS & JS
- Refactor your code
- Remove Duplicate code
- Avoid unnecessary tags in head tag
- Use Async & Defer wherever required


"If you want to learn to swim,
jump into the water."
-Bruce Lee

Q & A