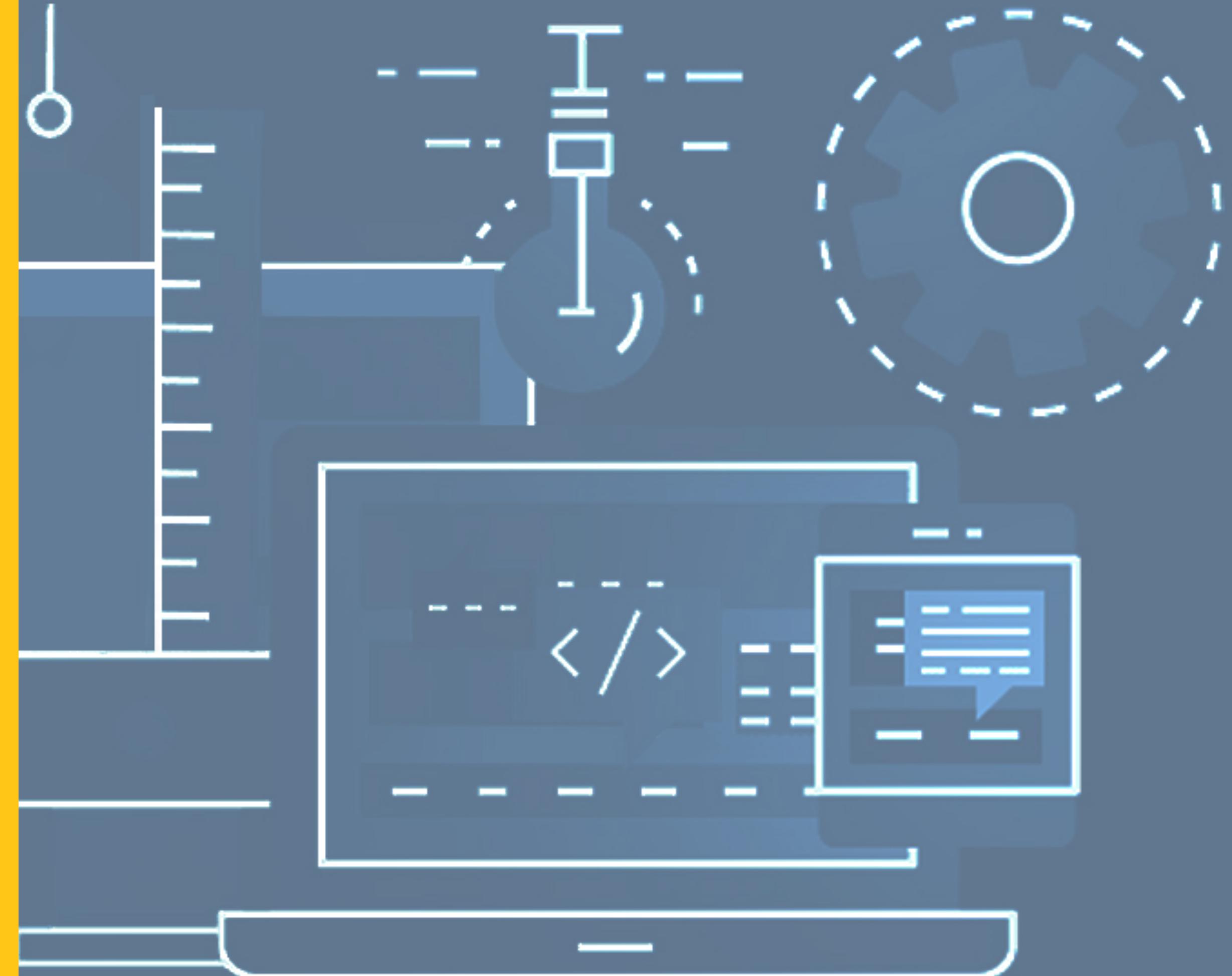


**SHELLY GRAHAM, 07/06/2020**

# **WEB DEV 4**

# **SUMMER 2020**

**Week 2: Accessibility**



# **WEEK 2: ACCESSIBILITY**



# SHIT IS HARD...

So make sure as many people as possible are able to use your enjoy your hard work!

—

# WHY EVEN BOTHER?

- Because everybody uses the internet!
- Code for as many people as possible **INCLUDING** users with permanent and/or temporary disabilities
- Accessibly helps to elevate the overall user experience
- Depending on industry you are in, can be mandated by the government



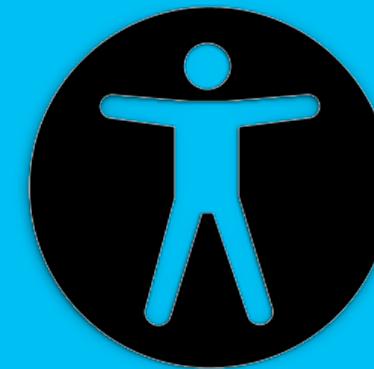
# EXAMPLES

- **Disabled users use screen readers and voice over technology to navigate websites and apps**
- **People with different native languages may use closed captions during shows or movies to better understand the context**
- **A strained or broken wrist can hinder you from using a mouse for a short period of time**



# STANDARDS

- WCAG = Web Content Accessibility Guidelines
- WebAim Checklist = Web Accessibility in Mind
- The A11Y Project = The Accessibility Project
- Government guidelines like Section508



The A11Y Project



# TOOLS

- 1. Code Validators for HTML, CSS or JS**
- 2. Screen Readers - External or Build-In Software**
- 3. VoiceOver / ChromeVox**
- 4. Browser Extension like Wave**

# CODE VALIDATORS

- Code validators are often the first step to check if your website is as accessible as it can be

ACHECKER®

- Good HTML validators:
  - [AChecker](#)
  - [Nu Validator](#)
- Good CSS validator:
  - [CodeBeautify](#)
- Good JS validator:
  - [jsHint](#)

Code Beautify

JS Hint

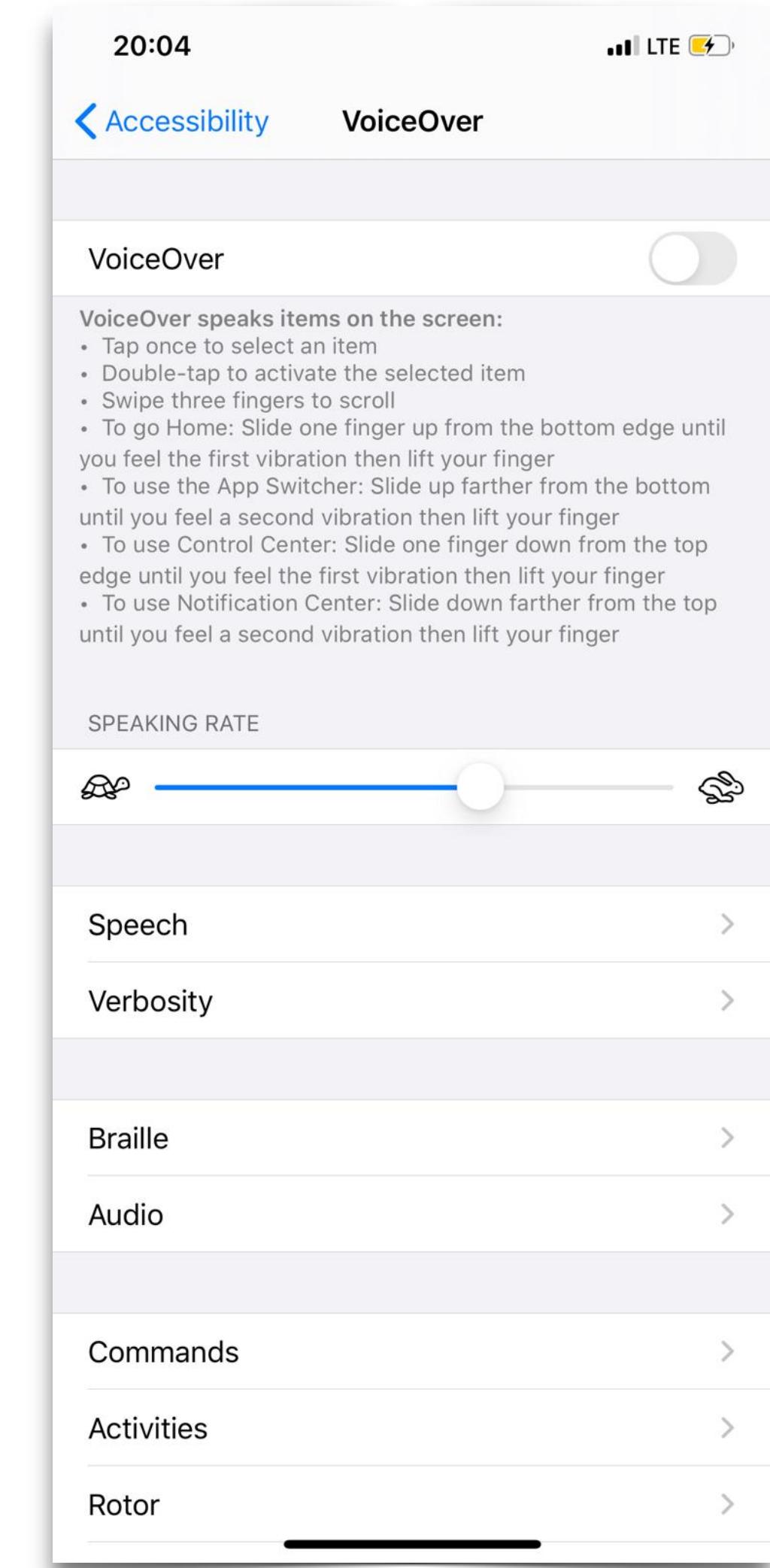
# SCREEN READERS

- A screen reader is a form of assistive technology that renders text and image content as speech or braille output.
- Screen Readers can be external machines, browser extensions or built-in machines like in mobile phones
- You can test every website simply by pressing the “tab” key.
- Popular screen reader: JAWS



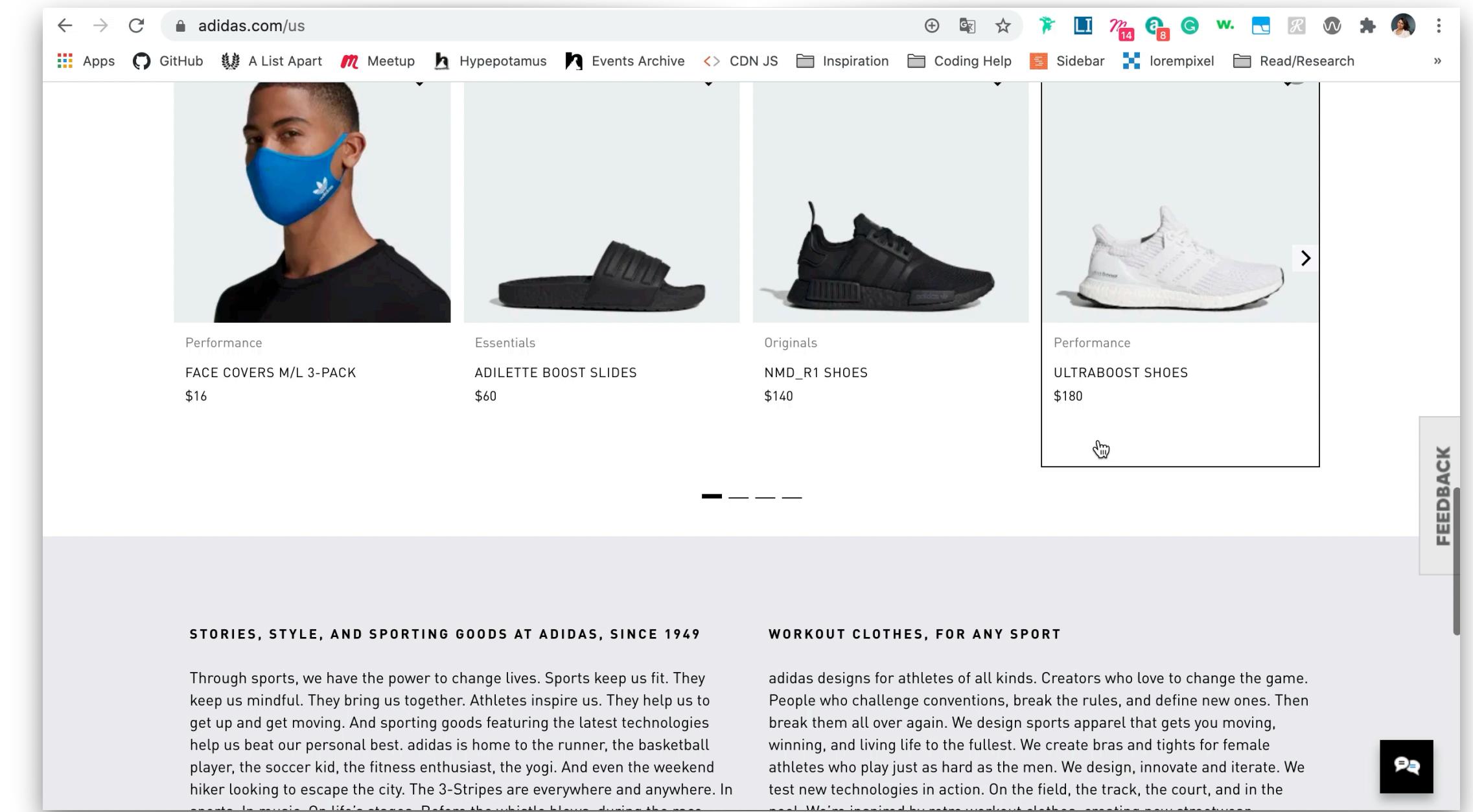
# VOICE OVER

- VoiceOver Software will read all HTML elements with focused out loud
- The software will read the elements out based the order in the DOM tree
- You can change the speed of the spoken word



# BROWSER EXTENSIONS

- Use browser extensions to help with accessibility
- Good extensions:
- WAVE can be used as website as well as browser extension
- Click play on video to see how to use WAVE



# 4 WAYS TO IMPLEMENT

- 1. Focus**
- 2. Semantic HTML Elements**
- 3. Styles**
- 4. ARIA**

# 1. FOCUS

- Focus is the currently selected element on a page when using a keyboard to navigate
- Most HTML elements are focusable by default
  - Links
  - Form fields with input fields
  - Buttons
- HTML elements that the user does not interact with are not accessible by default
- Order is determined by DOM order - which you can change!

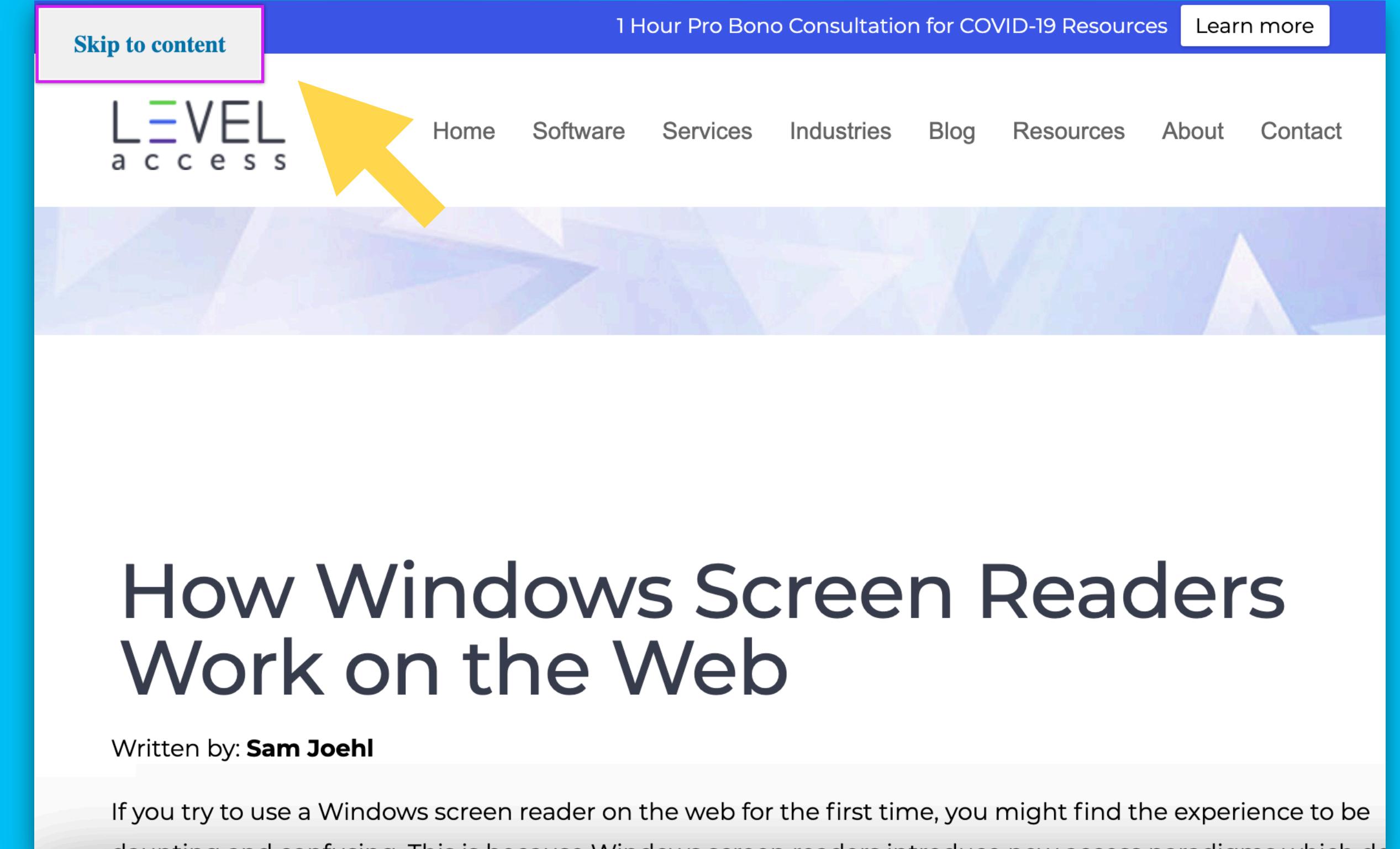
Hover me: |

Without Focus: |

submit

# 1. FOCUS CONTINUED

- Skip-to-content-button
- Remember to include tabindex="-1" on the target of the skip link. IE and Chrome both require this attribute to be present to make the skip link work consistently!
- Source
- **Tab Order: Structure your HTML in the same way you would expect the user to ready the website, top to bottom, left to right**
- **Tab-Index-Attribute: Use to set focus order when you can't change the HTML order, use sparingly**



The screenshot shows the header of the LEVEL access website. At the top left is a purple "Skip to content" button. To its right is a blue bar with the text "1 Hour Pro Bono Consultation for COVID-19 Resources" and a "Learn more" button. The main navigation menu below the bar includes links for Home, Software, Services, Industries, Blog, Resources, About, and Contact. A yellow arrow points from the text "Remember to include tabindex=-1" in the list above to the "Home" menu item in the header. The background of the header features a light blue geometric pattern.

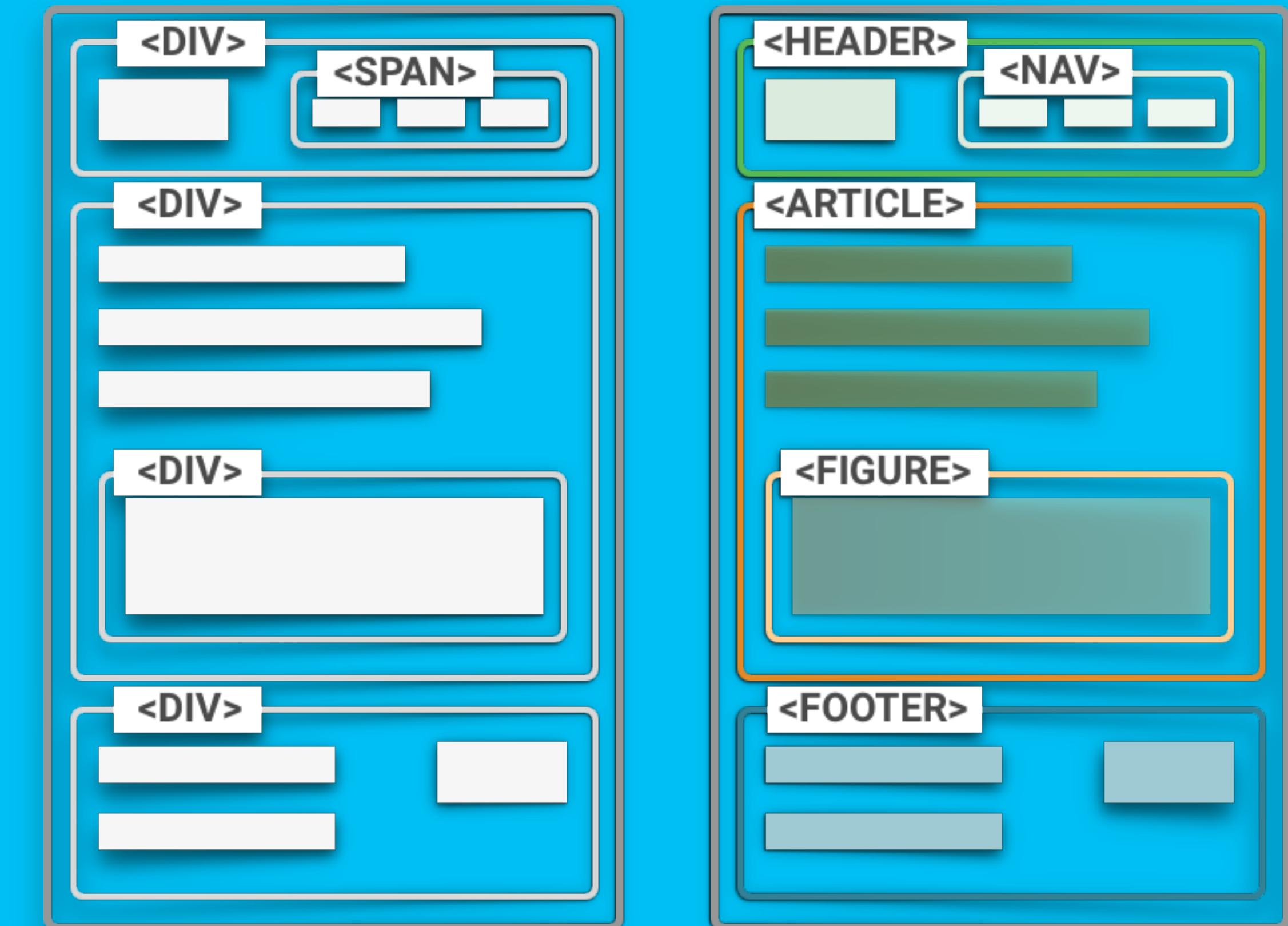
## How Windows Screen Readers Work on the Web

Written by: **Sam Joehl**

If you try to use a Windows screen reader on the web for the first time, you might find the experience to be daunting and confusing. This is because Windows screen readers introduce new access paradigms which do

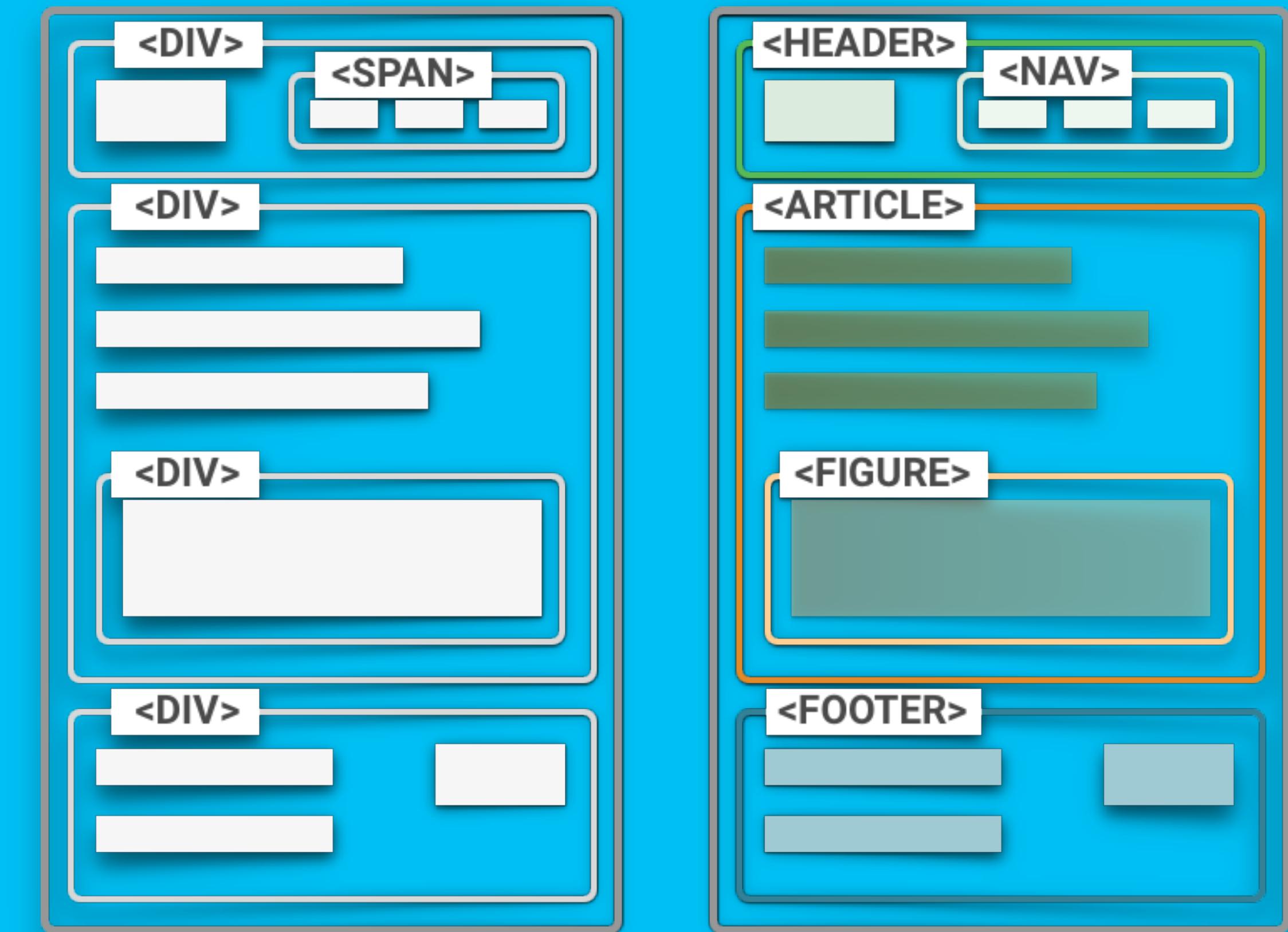
## 2. SEMANTIC HTML

- Not only important for SEO but also for accessibility
- Instead of using just divs, group your website into meaningful section
- General guideline: header with nav, main including sections, articles and possibly asides and lastly the footer



## 2. SEMANTIC HTML CONTINUED

- Use the form-tag for forms along with input fields and labels that describe the input field
- Always define the type of the input field. Don't use type: text for a password field!
- If no alt-text is used, screen readers will read the file name, which are often only symbols and number. Use alt-text on images so screen readers can read them out loud
- Use an empty alt-tag for decorative images alt="".



# 3. STYLES

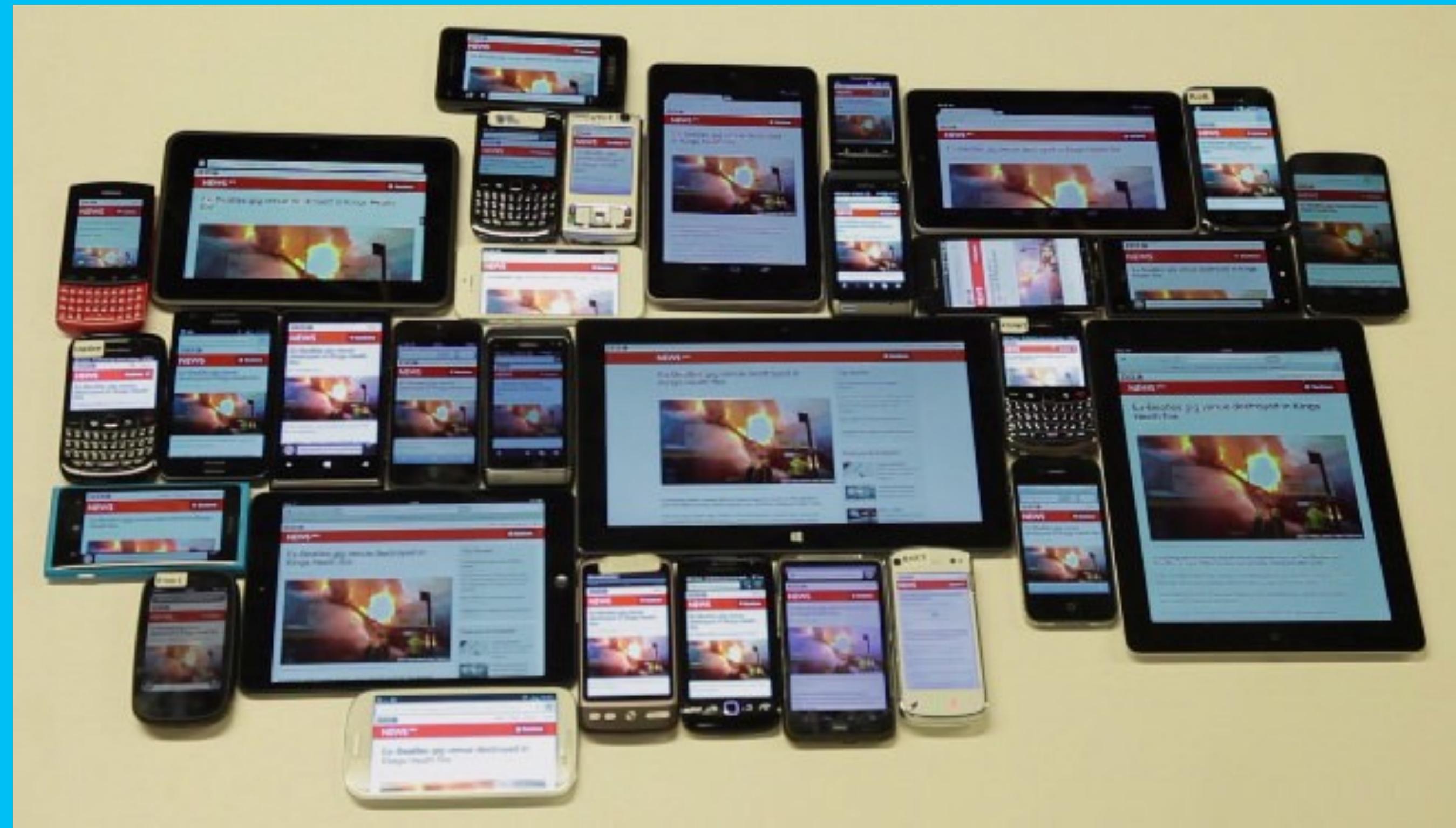
- Generally concern the responsiveness, size and contrast of the website

## Focus State Ideas

Default	Border + Outer Shadow 1px stroke   Black 4 px, no blur	Thick Border 2px stroke   Black	Outer Outline 3px stroke   Black	Outer Outline 2px stroke   Glacier Blue
<input type="text" value="FIRST NAME *"/>				
<input checked="" type="radio"/> Selected <input type="radio"/> Not selected	<input checked="" type="radio"/> Selected <input type="radio"/> Not selected	<input checked="" type="radio"/> Selected <input type="radio"/> Not selected	<input checked="" type="radio"/> Selected <input type="radio"/> Not selected	<input checked="" type="radio"/> Selected <input type="radio"/> Not selected
<input checked="" type="checkbox"/> Selected <input type="checkbox"/> Not selected	<input checked="" type="checkbox"/> Selected <input type="checkbox"/> Not selected	<input checked="" type="checkbox"/> Selected <input type="checkbox"/> Not selected	<input checked="" type="checkbox"/> Selected <input type="checkbox"/> Not selected	<input checked="" type="checkbox"/> Selected <input type="checkbox"/> Not selected
<a href="#"> Download</a> <a href="#"> Save</a>				
<a href="#">This is a link</a>				
<input type="button" value="Select"/>				

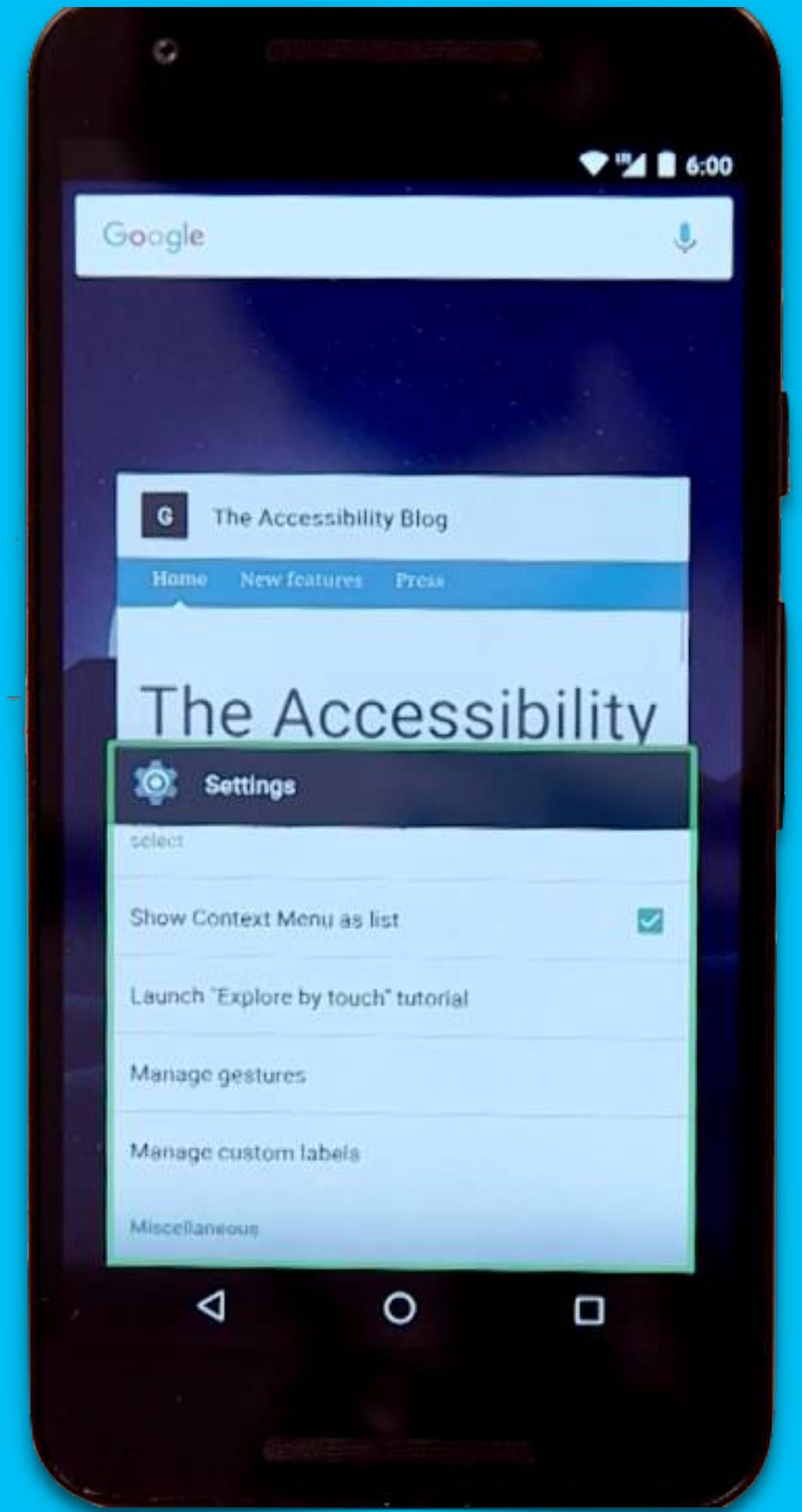
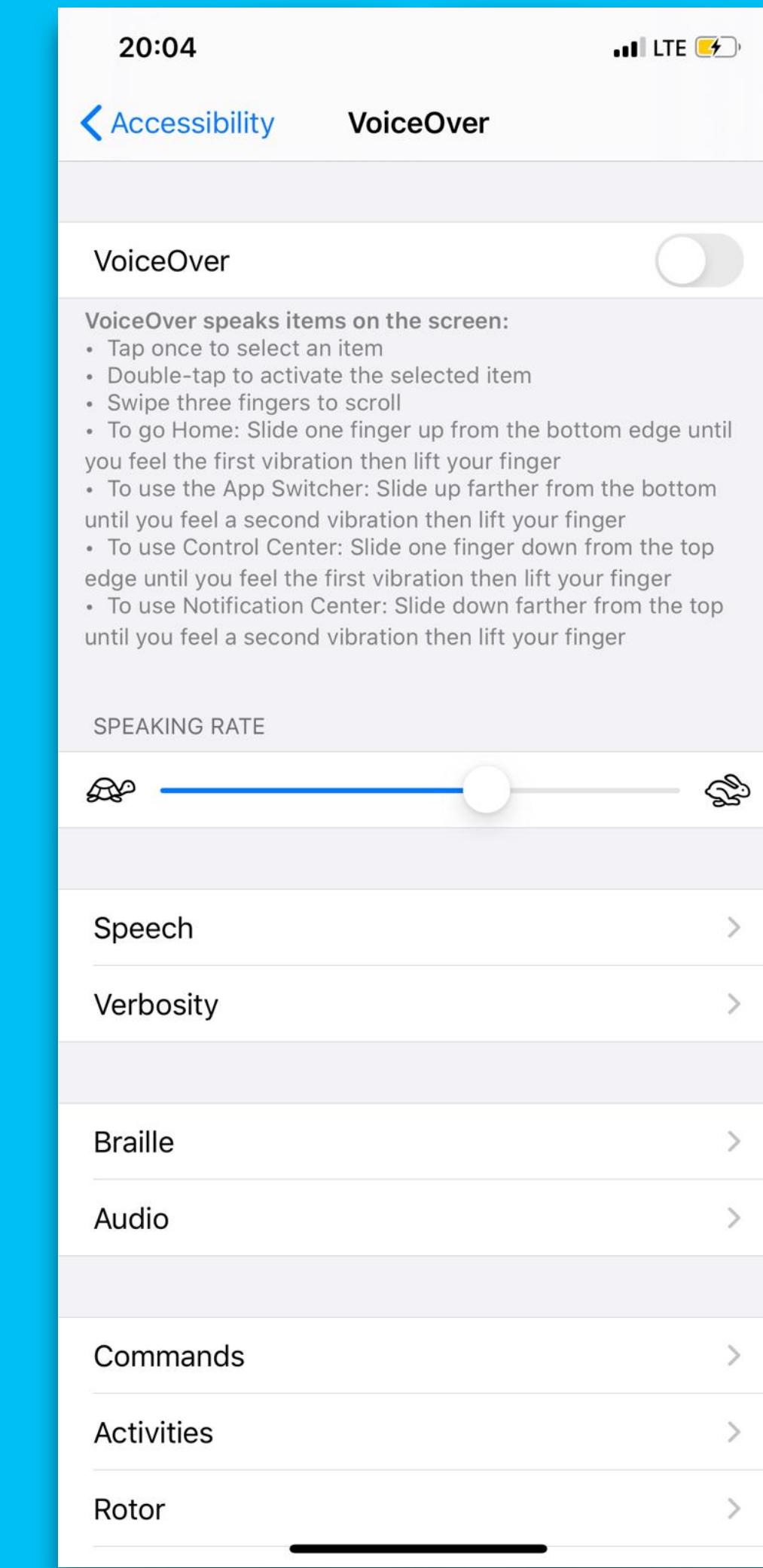
# 3. STYLES- RESPONSIVENESS

- Responsiveness is best tested on different devices with different operating systems
- Test minimum: iOS and Android
- For mobile devices: different operating systems have different built-in screen readers
- Don't rely on external hardware like bluetooth keyboards



# 3. STYLES- DESKTOP VS. MOBILE

- **82% of screen readers use a mobile device!**
- **Be sure to test on as many devices as possible**
- **The biggest difference in testing for mobile vs. desktop is the screen reader navigation - you can't use tab on mobile**
- **Learn essential finger commands**



## 3. STYLES- CONTRAST

- Set body size to a minimum of **16px!**
- Use rem and ems instead of pixels so typography can grow with the size of screen
- Use well-readable fonts and provide native alternatives (also, never use more than 2-3 fonts per page)
- Use contrast checker tool: [Contrast Checker](#)



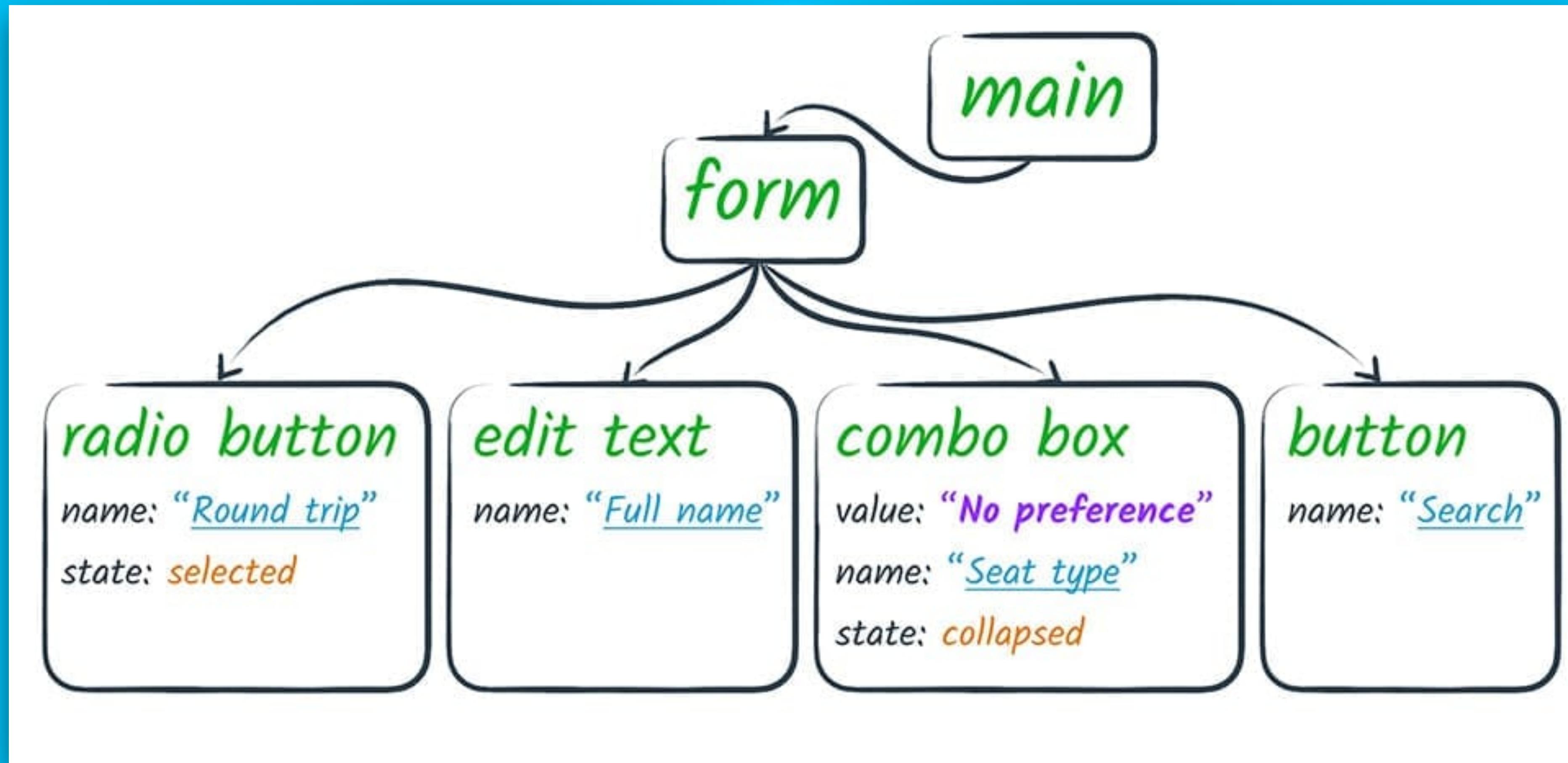
## 4. ARIA

- ARIA allows you to specify attributes on elements which then modify the way they are integrated into the accessibility tree
- Modify semantics
- Adds extra labels and description not available in regular HTML5
- DOES NOT change elements inherent behavior, e.g. make element focusable
- No ARIA is better than bad ARIA!

Web  
Accessive  
Initiative

Accessible  
Rich  
Internet  
Applications

WAI - ARIA



Or Accessibility Object Model (AOM) contains accessibility-related information for HTML elements which focus on:

- name
- description
- role
- state

—

# ACCESSIBILITY TREE

# ARIA TERMINOLOGY

1. **aria-label:** will overwrite native label mechanisms

```
<button class="hamburger" aria-label="menu">Button Image</button>
```

2. **aria-labelledby:** like label usage but reversed relationship:

—> label refers to input it labels, aria-labelledby refers to input its labelled by

—> can be used with multiple elements

3. **aria-owns:** alters the relationship between elements in DOM tree

e.g. siblings can now be coded as parent-child pair

4. **aria-hidden:** prevents screen readers from seeing elements and reading them out

5. **aria-live:** notify user when content is updated during session (e.g. score during sports game, price on a product)

```
<h3 class="connection-status" aria-live="polite">Reconnecting...</h3>
```

# 5. FORMS

- Should be one column
- Top align labels
- Group labels with their input
- Avoid call caps
- Show all selection options if under 6
- Resist using placeholder text as labels
- Make CallToActions descriptive
- Group related information

bigcartel

Tour Examples Buzz Pricing + Sign Up

## Sign up

Payment is month-to-month. Upgrade, downgrade, or cancel at any time.

1 Tell us about your store

Store name

Store address

Password

Password again

Store type Accessories

Country United States

Time zone (GMT-05:00) Eastern Time (US & Canada)

Currency U.S. Dollar (USD \$)

# RESOURCES

## Form Controls

QUESTIONS?

# HOMEWORK

- Expand your website from last weeks homework and add accessibility features. Also, add in feedback you've received during class.
- If not already responsive, make your site responsive and use semantic HTML elements wherever it makes sense.
- In addition to your landing page, add one subpage. The topic of the subpage is up to you. I'm suggesting a product page to be able to work with different links or a contact page, that includes an input field.
- At a minimum, your site should include:
  - A Name field
  - An Email field
  - At least one set of radio buttons
  - At least one checkbox
  - At least one expandable text area
  - All inputs should have an associated label.
  - The entire site should be navigable with keyboard only, and each field should be clearly described when read by a screen reader.
  - Your site must have no warnings or errors using accessibility validation chrome extension that we looked at in class.
- You will be graded on the correct usage of tools to help you achieve good usability.
- Bonus Points if you are able to implement a “Skip to Main Content” button.
- Submit a link to the git repository for this assignment.

**FIN**