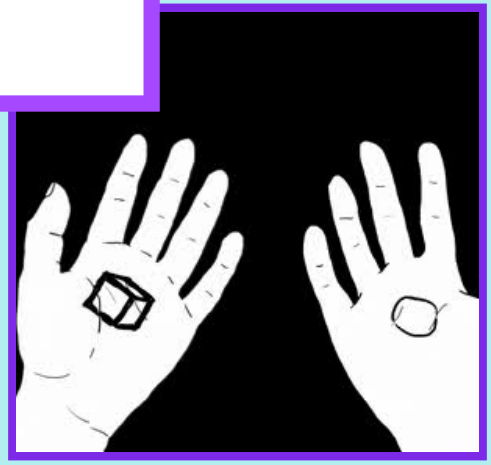
Accessibility Part 3

Presented by
Mike Nam-Lee



3. Understandable

Information and the operation of user interface must be understandable. (the content or operation cannot be beyond their understanding)

One part of understandability is that the language is defined.

It makes a big difference to screen readers and voice assistants.

```
<html lang="en">
<!-- Or other language
your page is on -->
```

Predictable

</>

This one is also really easy to satisfy. A big theme of this section is to not do inaccessible things. So to keep things predictable, don't cause page changes or form submissions on focus or input.

Also, to aid accessibility tools, keep IDs consistent and unless the content changes, don't change the underlying HTML.

To aid low-bandwidth users, content should make as much sense as possible without CSS.

Site navigation

Last lecture, we covered how to navigate within a page but it's also important to navigate within a site. We need to know where we are and where links go. Be succinct and unique.

This also helps for search engines as site crawlers and screen readers do similar things.

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <title>Understandable -
      Accessibility Part 3</title>
      <meta name="description"
      content="COMP 6080 lecture on how
      understandable content can make
      your site more accessible."/>
   </head>
...
```

Link purpose

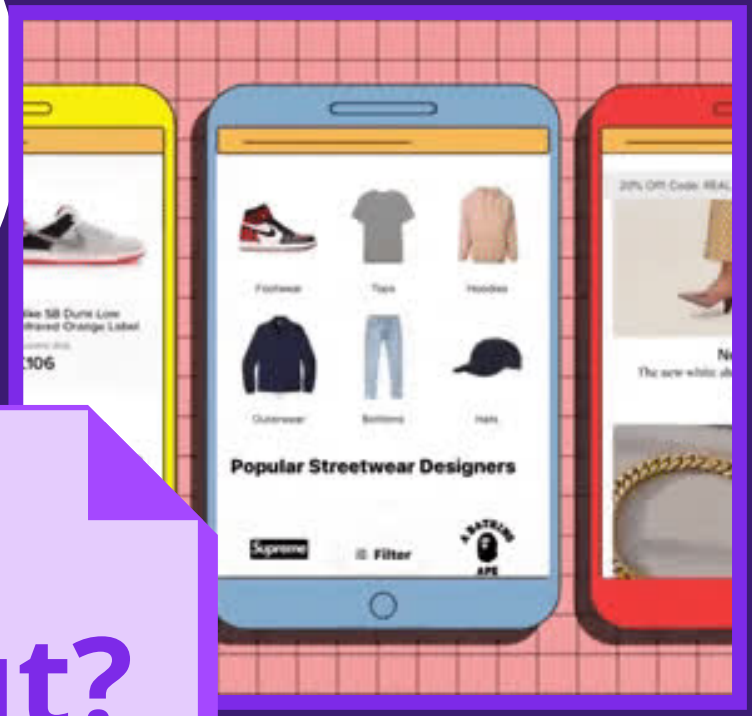
For links, a Level A requirement is that it should understandable in context of the sentence and a Level AAA requirement is that it's understandable with the link text alone.

Also, don't forget any interactive item that sends you to a location (internal or external) should use a link tag.

```
<!-- Doesn't pass Level A -->
<p>I accept the terms of service.</p>

<a href="/privacy">Click here</a>

<!-- Good but doesn't pass Level AAA -->
<p>I accept the terms of service <a
href="/privacy">here</a>.</p>

<!-- Passes all levels -->
<p>I accept the <a href="/privacy">terms
of service</a>.</p>
```

Let's make forms understandable

Q



How can you label an input?

This is the most ideal way to label an input. It has the most support.

```
<label for={someInput}>Name</label>
<input id={someInput} />
```

This is also acceptable but this has some issues with custom component inputs, say if the custom input has text, it'll be treated as a label.

```
<label>
   Name
   <input />
</label>
```

This should not be used in this way. Placeholder text has pretty bad usability with screen readers. Even if you use a label as well, the placeholder should be an example rather than another label.

```
<input placeholder="Name" />
```

This is the inverse of the first way. This is most useful if there are two or more elements that should have the same label as long as they are related, connected, and adjacent.

```
<label id={someLabel}>Name</label>
<input aria-labelledby={someLabel} />
```

If you have an input with an invisible label (this is most common with search bars), use aria-label.

```
<input aria-label="Name" />
```

As a supplemental addition, you can add a description to the input. This should not replace the label and it should also not be too long.

This is useful for when there is more context that would be relevant to the user (e.g. if the field needs to be in a particular format). This is also where you should add any custom error alerts on forms.

```
<label for={someInput}>Name</label>
<input
   id={someInput}
   aria-describedby={someDesc}
/>
<p id={someDesc} role="alert">
   Please use your full name as this
   will be printed on the invitation
</p>
```

Additional attributes

You may be familiar with required and type. These will help with accessibility as the error can be identified immediately rather than on submit.

You should also use aria-invalid to indicate if an input field needs amendment. This will specifically be useful for custom error validation as HTML will validate according to the specified type by default. This should be true only after onBlur, not by default.

Where possible, it's better to fix incorrect data formats in code (if the field contains enough data).

```
<!-- No aria-invalid necessary for
empty inputs -->
<input type="email" required {...}/>

<!-- No aria-invalid necessary for
negative numbers -->
<input type="number" min={0} {...}/>

<!-- No aria-invalid necessary for
short inputs -->
<input type="password" minLength={8}
{...}/>

<!-- Aria-invalid helps here -->
<input type="date" aria-invalid={true}
aria-describedby={err} {...}/>
<p id={err}>Bookings are only accepted
on weekdays.</p>
```

Error messaging

Error messages should be succinct and specific. Saying "Start date is invalid" isn't as helpful as "Start date must be on a Monday". It should be short as long messages are annoying to screen reader users.

Technically, they should have a role="alert" to signify its importance and it is common to focus on the first invalid element. Adding the alert role interrupts the screen reader to alert the user of the error.

Announcing Roles

I mentioned using the alert role for error messaging. There are a few other roles I'd like to introduce:

- "progressbar": use this for important loading states, say when uploading a file
- "status": use this for global statuses, say any updates to your shopping cart
- "log": use this for new chat messages as they enter
- "timer": use this for countdowns or stopwatch readouts

By default, if the text gets updated, it'll only announce the change. If you want the whole text to be reannounced (most useful for timers), use aria-atomic={true}.

```
<div role="status">
   New email from Ari
</div>
```

Announcements

If you need to make an announcement and none of the roles fit, use aria-live.

Aria-live can be added to most elements and is either "polite" or "assertive", though if you're trying to use "assertive", you probably want to use the alert role.

Use aria-live and announcing roles very sparingly as they are a disruptive experience.

Be careful: there is a browser quirk where adding both aria-live and an announcing role can lead to the same message being announced twice. Prefer to use the role instead.

```
<div aria-live="polite">
   This site will undergo
   routine maintenance in
   30 minutes. Please
   save your data.
</div>
```

See you
next time!