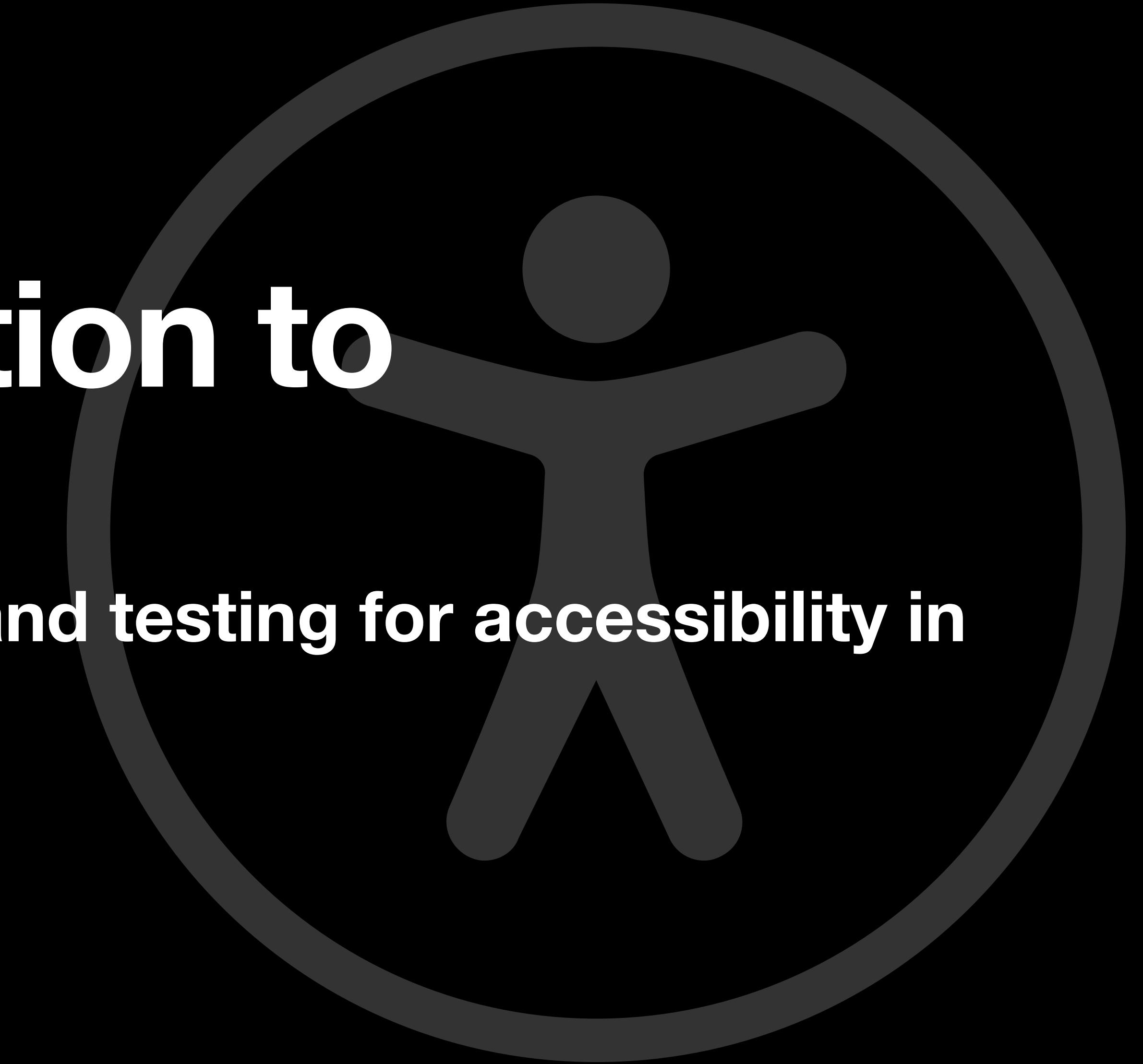


A Swift Introduction to Accessibility

Learn the basics behind making and testing for accessibility in your application

Kim Arnett

@kaydancode



What is accessibility?

Digital accessibility ensures technologies are designed and developed so that people with disabilities can use them.



Why Should You Care?



1. It's the right thing to do
2. 25% of Americans have reported living with a disability
3. Avoid Legal Issues (4000 digital accessibility lawsuits in 2021, average 15% increase rate each year.)

Why Should You Care?



4. Do it right the first time, reduce cost in rework: planning, development, testing, etc.
5. Would you ship without security and follow up fixing it later on?
6. Ensure anyone can use your product

Why Should You Care?



7. Better Usability For EVERYONE!

Principals of Accessibility

1. Perceivable
2. Operable
3. Understandable
4. Robust



@kaydancode

Disability Personas

Blind

Low Vision

Color-blind

Deaf

Deaf-blind

Motor



@kaydacode

Disability Personas

Blind

Low Vision

Color-blind

Deaf

Deaf-blind

Motor

Speech

Cognitive

Reading

Seizures

Multiple



@kaydacode

Tools Available in iOS

Voice Over (Screen Reader)

Voice Control

Magnifier / Zoom

Color Adjustments

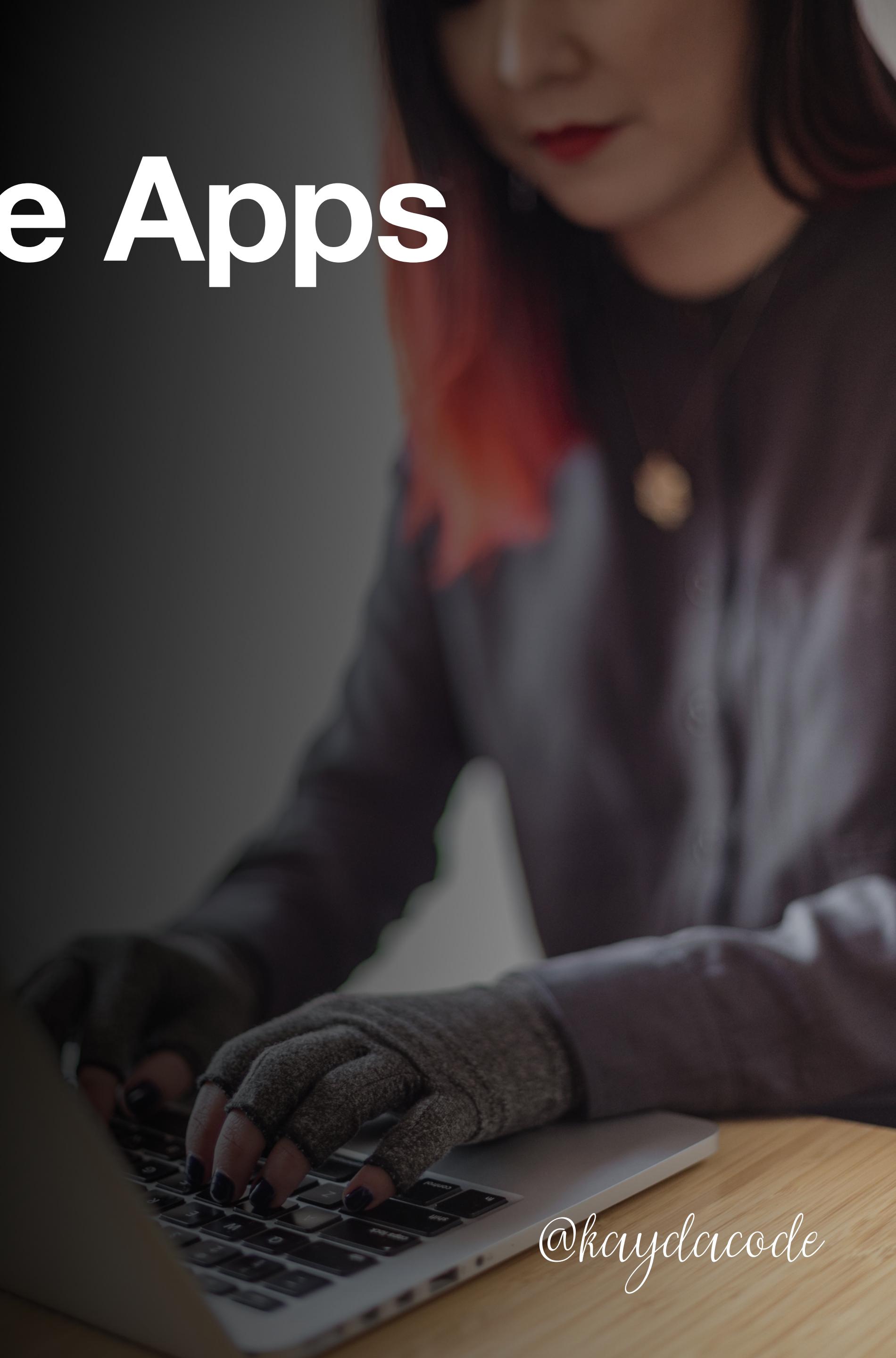
Switch Control / Keyboard

Live Captions

&& so many more!

@kaydacode

Developing Accessible Apps

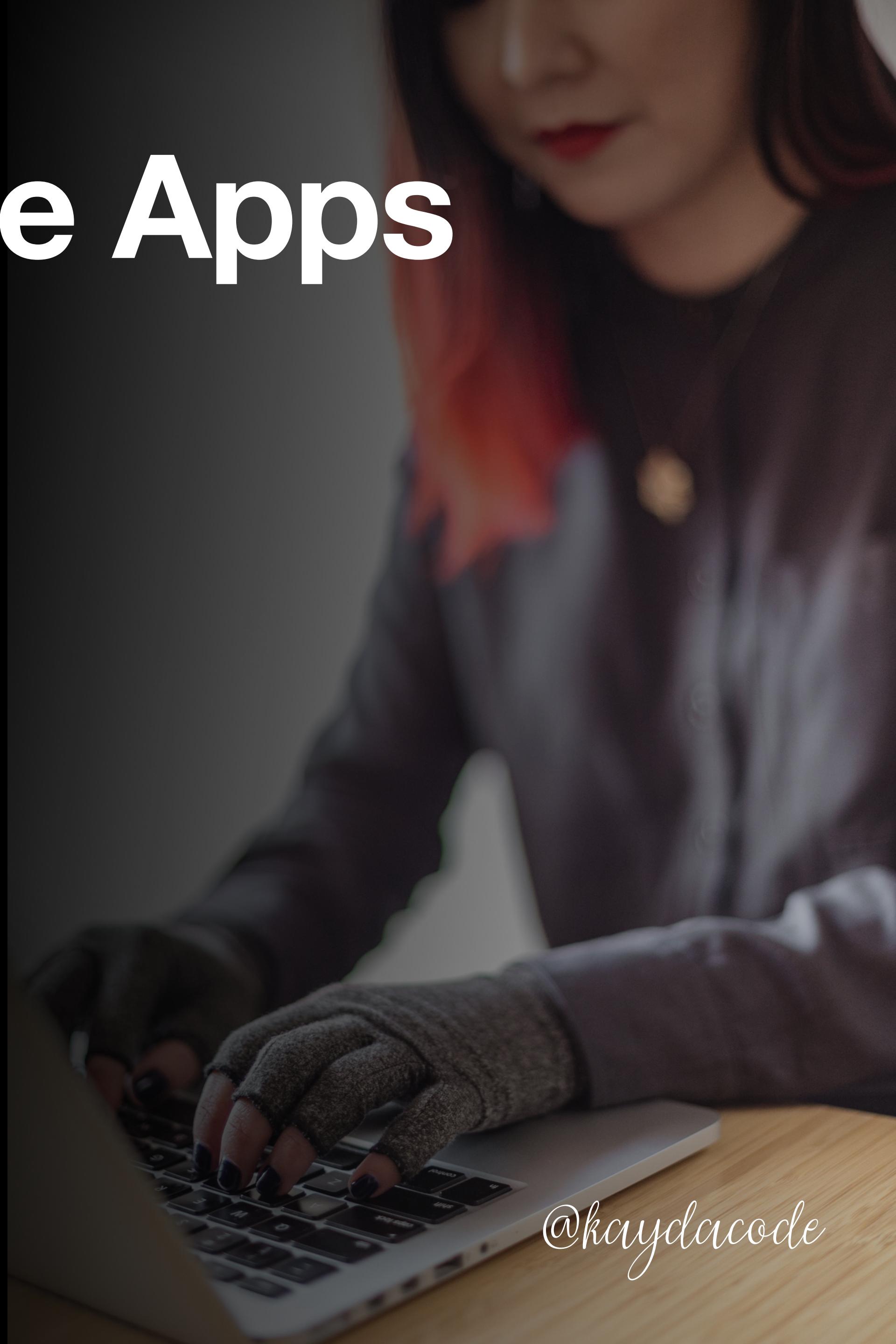


@kaydancode

Developing Accessible Apps

Accessibility Identifier - NOT utilized by accessibility tools. IS used for accessing views in testing frameworks.

Accessibility Label - IS utilized by accessibility tools to convey information to people



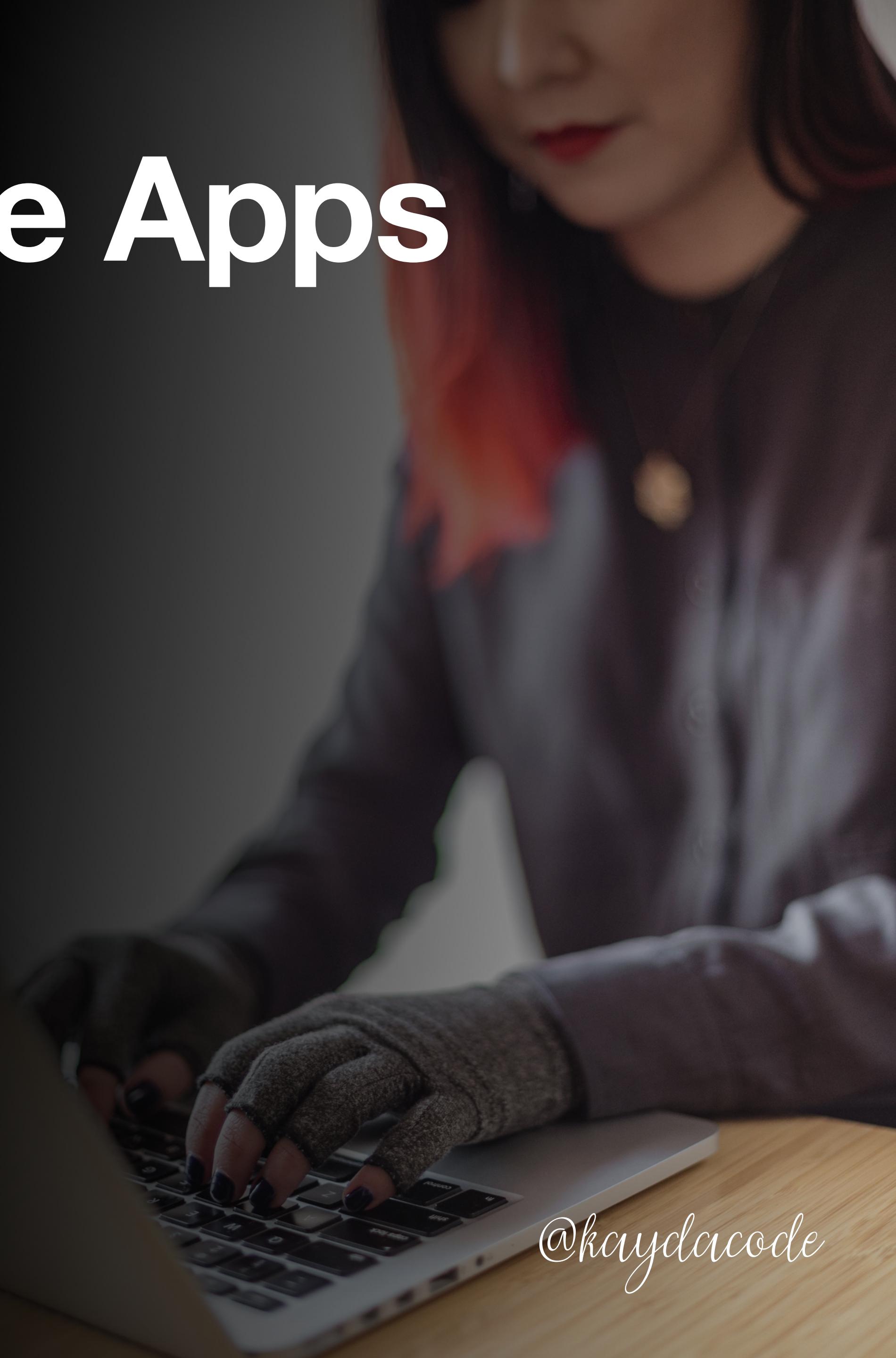
@kaydacode

Developing Accessible Apps

Controls

Controls must have a name, role and state conveyed to users in order to be accessible.

(Utilize native components for the best accessibility experience.)



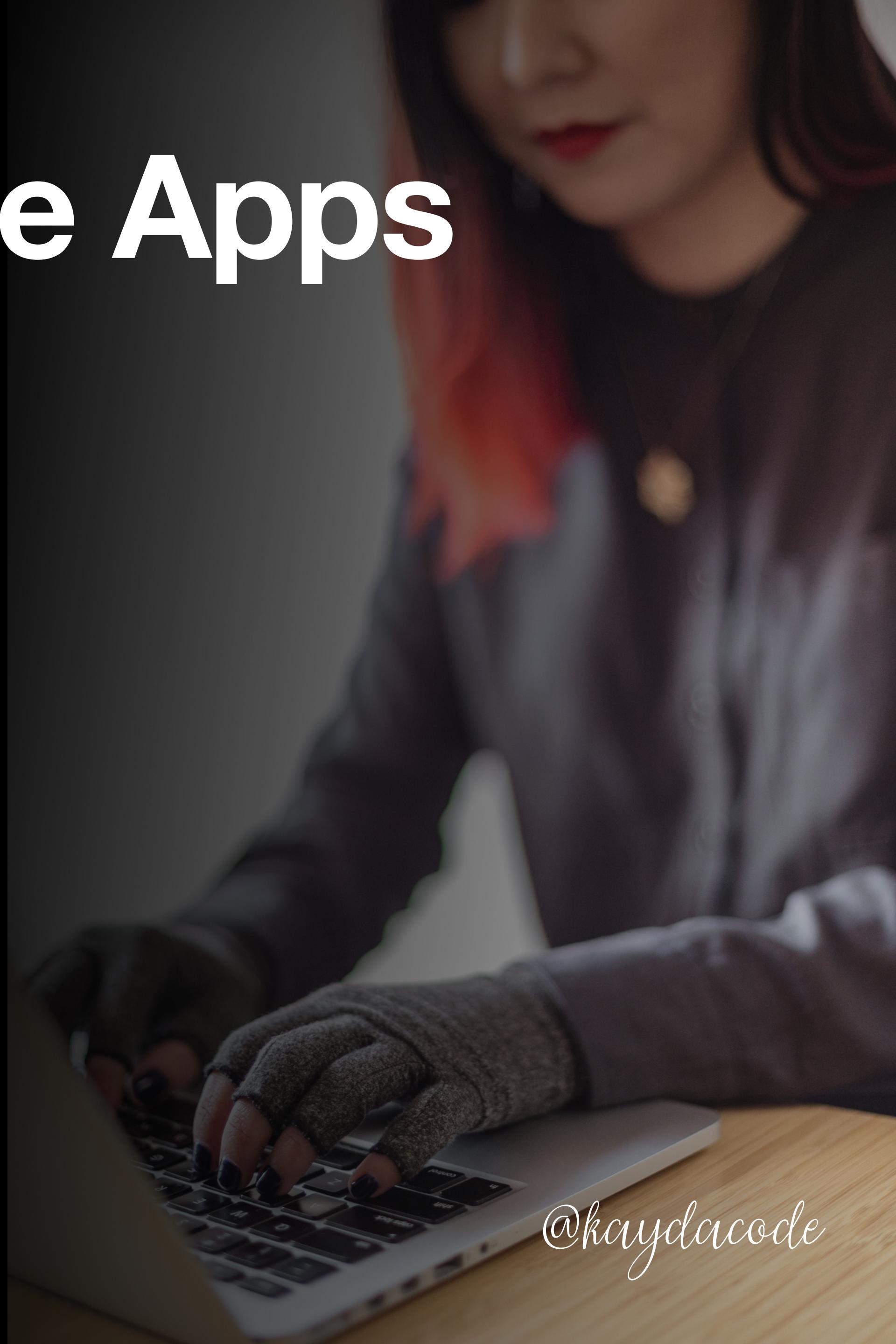
@kaydacode

Developing Accessible Apps

Controls

Name

Ensure controls have visible text, or an accessibility label set conveying the required information



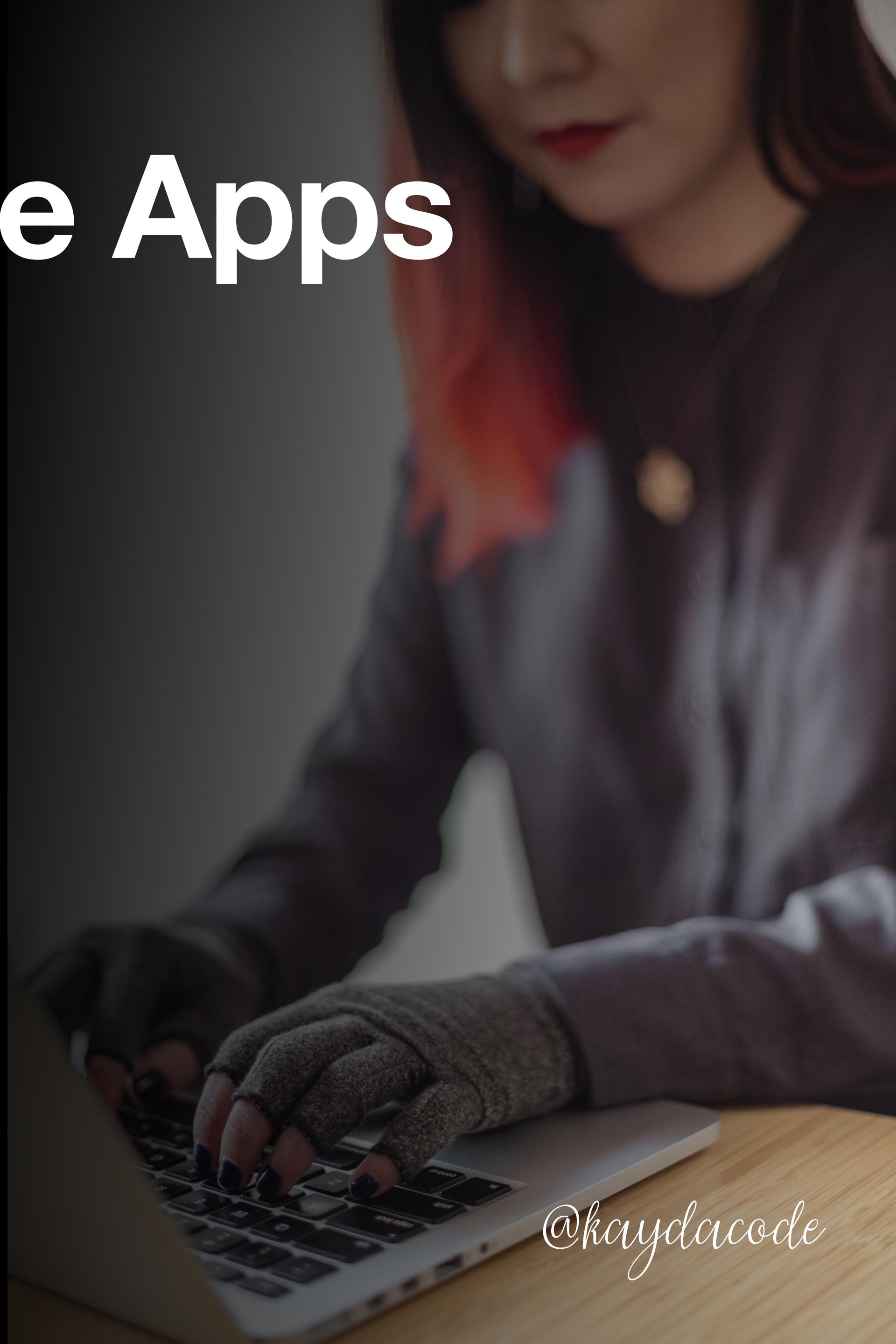
@kaydacode

Developing Accessible Apps

Controls

Role

For custom controls, set the element's accessibility trait to convey role and behavior.



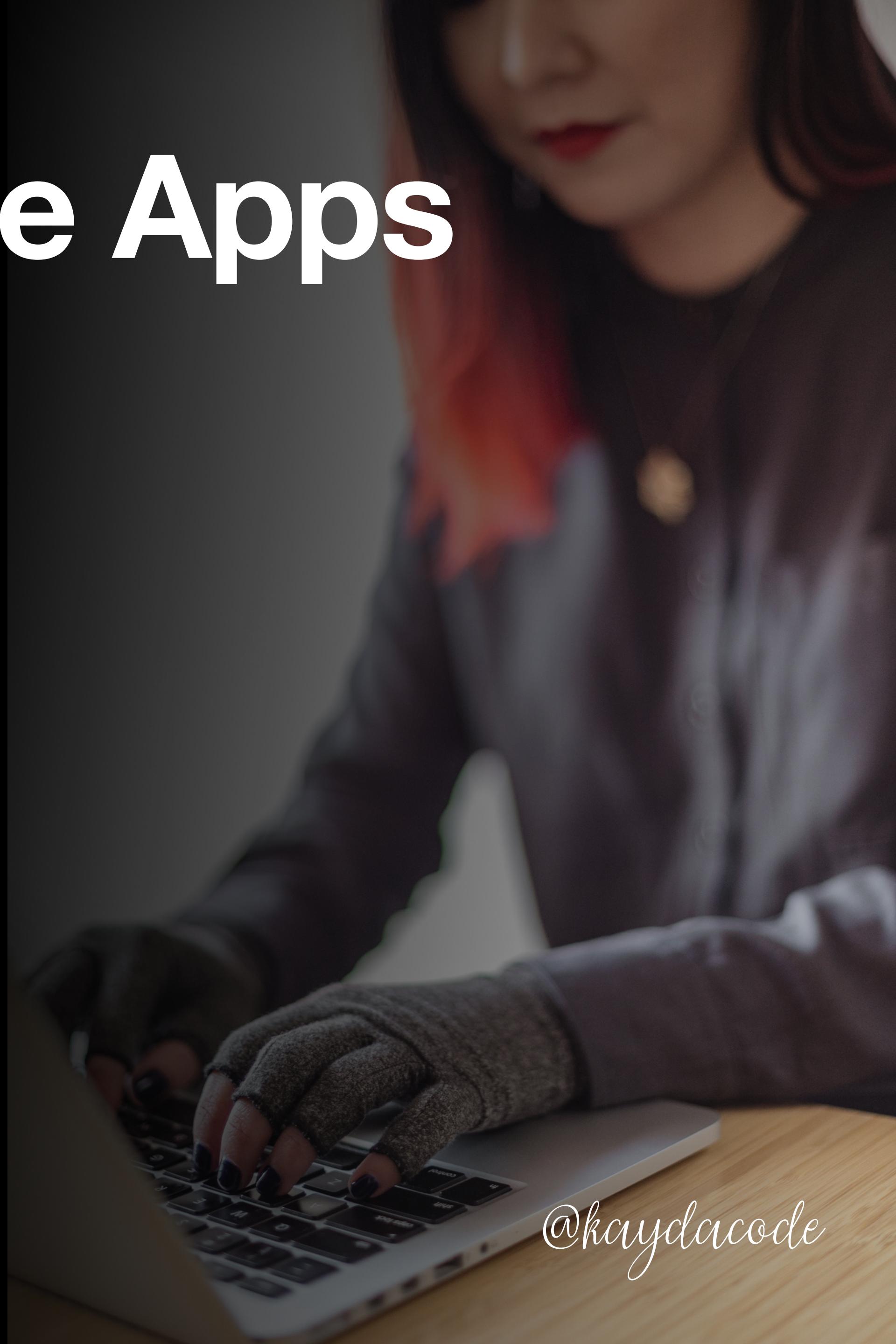
@kaydacode

Developing Accessible Apps

Controls

State

Use isEnabled to convey the state of the control properly.



@kaydancode

Developing Accessible Apps

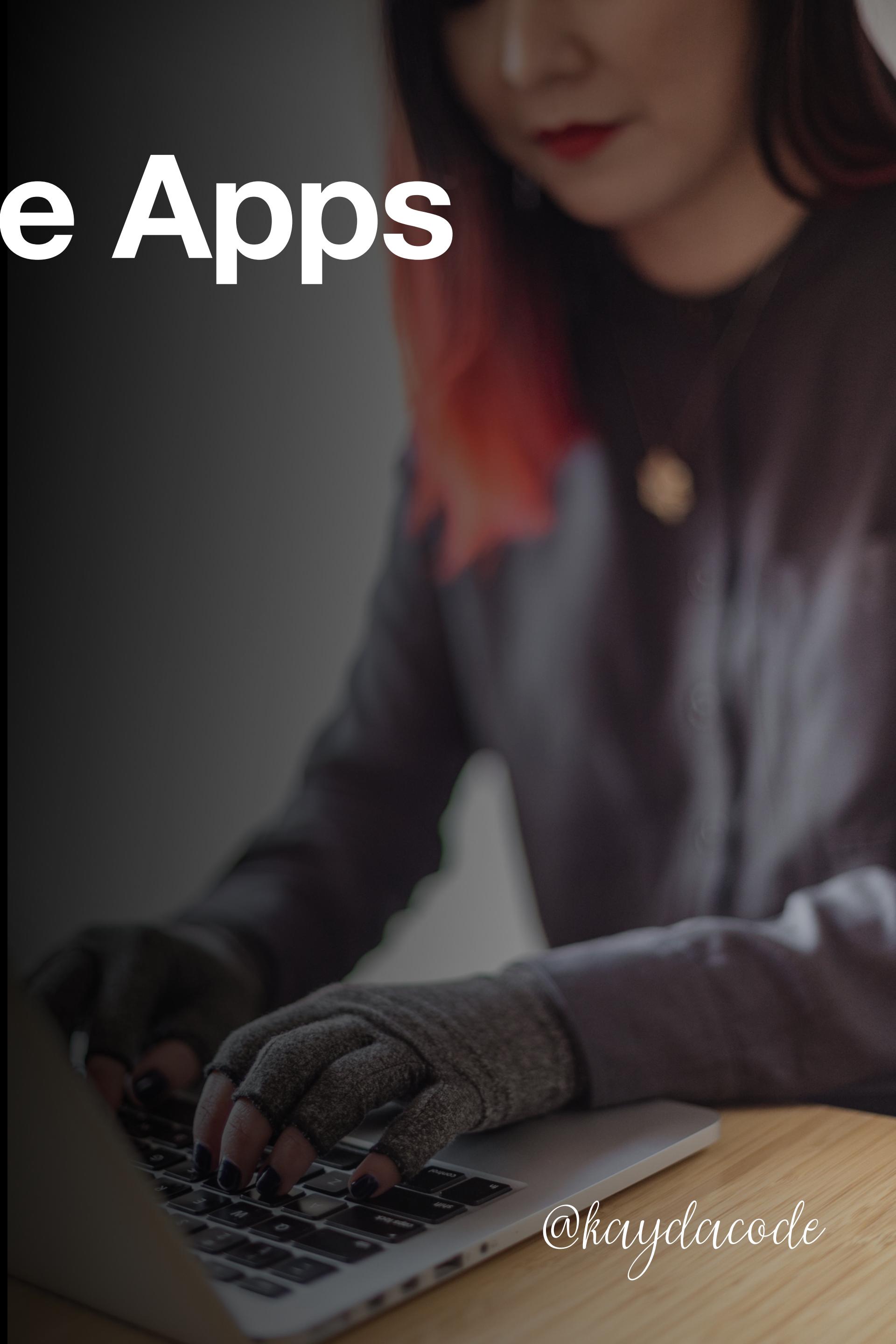
Controls

Sliders / Steppers

Accessibility label must match visible label

Need to convey value associated with the control

Implement `accessibilityIncrement` and `accessibilityDecrement` for adjusting value with assistive technology



@kaydacode

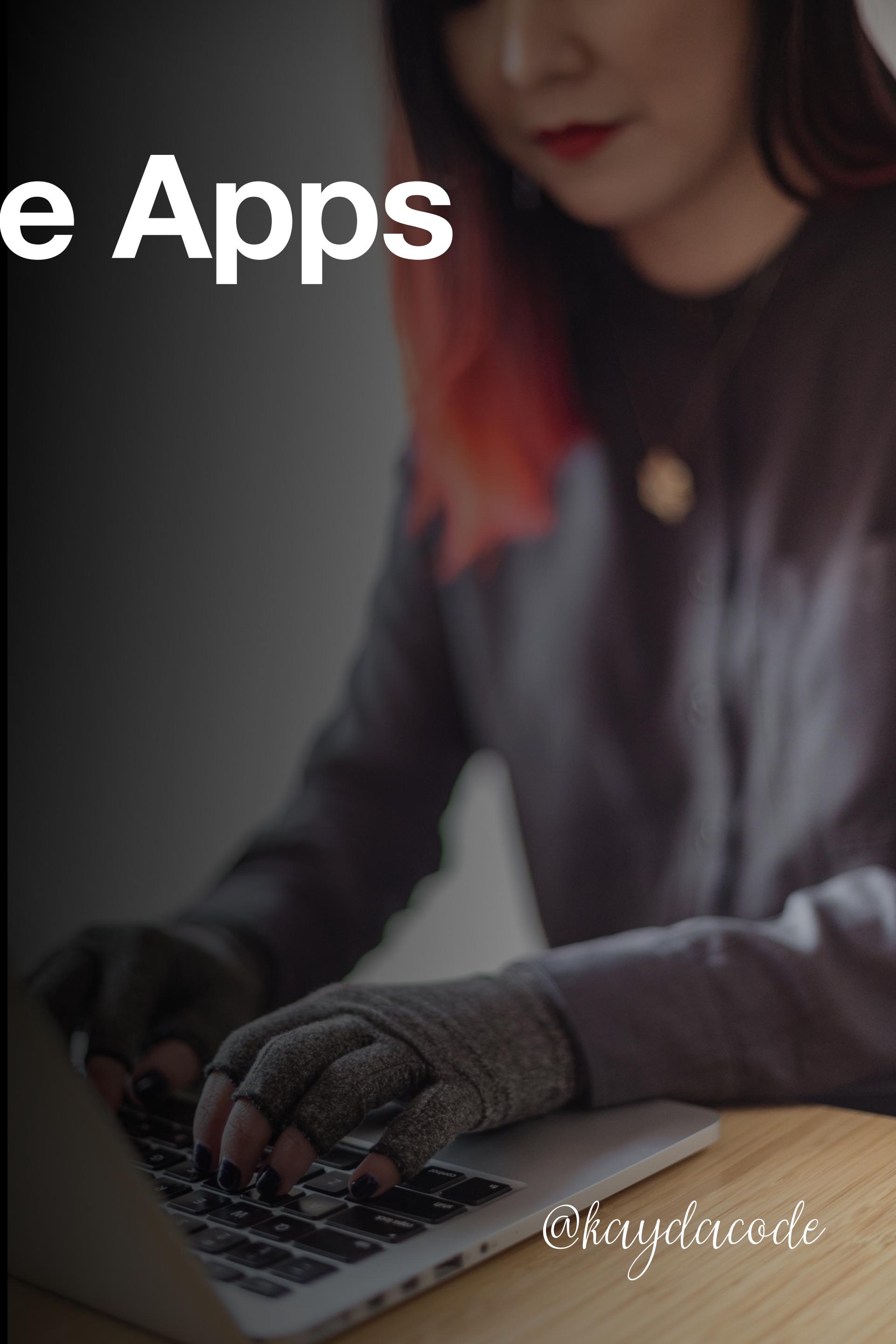
Developing Accessible Apps

Controls

Switches

Accessibility label must match visible label

Group visible text and switch so that they are announced together: assign an accessibility label, value, and trait to provide all needed information.



@kaydacode

Developing Accessible Apps

Images



Informative images are provided an accessibility label and focusable with `isAccessibilityElement = true`.

Decorative images are set to non focusable with `isAccessibilityElement = false` (Default).

Developing Accessible Apps

Text



Prepare for Dynamic Type:

Text is in a view that scrolls and expands so that all information is available no matter font size settings.

Text in Controls applies as well.

**Guideline is up to 200% its normal size*

@kaydacode

Developing Accessible Apps

Structure / Relationships



Each screen should have a clear title announced

Sections within a page are defined with a header label (accessibility trait heading) for quick navigation

Screen is able to rotate for devices in a fixed orientation

Developing Accessible Apps

Other Considerations / Best Practices



List-type elements

- Provide as much context as possible for specific item.
- Have accessibility label match the visible text.
- Utilizing accessibility hint as needed & ensure any action the element takes is communicated.
- ie: A list of clothing items where each has a ‘favorite’ button.

Developing Accessible Apps

Other Considerations / Best Practices



Tappable Elements

- 44pt by 44pt minimum for interaction
- minimum 24pt spacing radius between elements

Testing Accessible Apps



@kaydancode

Testing Accessible Apps

Accessibility Inspector

Utilizing Voice Over

Testing on Real Device

axe DevTools for Mobile by Deque



@kaydacode

Accessibility with deque

- axe DevTools Mobile by Deque
 - Automated & User-Initiated Testing
- Deque University
 - iOS Accessibility Techniques course
 - Mobile & Web courses available



14 Day Free Trial

@kaydacode

Resources

W3C : Working towards developing standards for the web

Section 508 Gov

Apple: iOS Accessibility Features



Images provided by Disabled and Here and nappy



@kaydacode

Thank You!



@kaydancode