

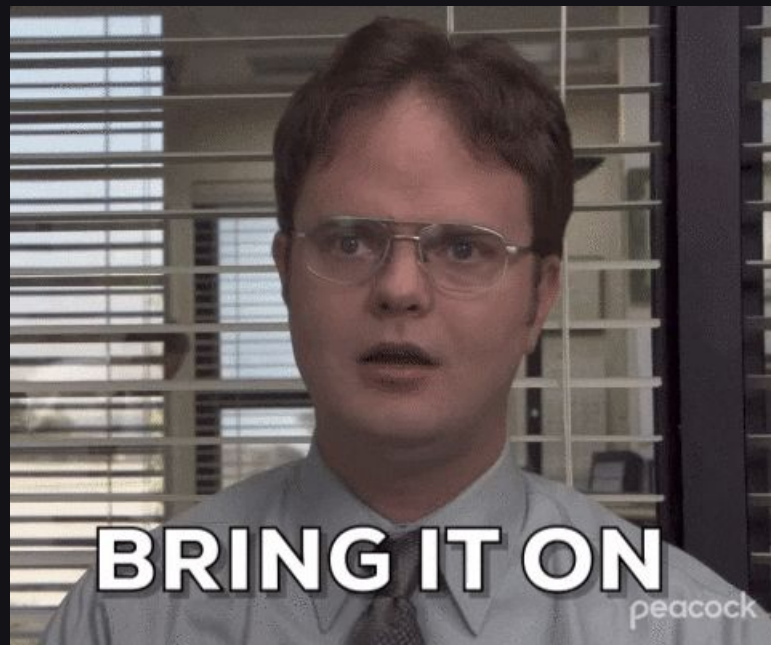
The background is a dark navy blue. It is decorated with various abstract geometric shapes: a large orange circle on the left, a medium blue circle on the top right, a small yellow semi-circle at the top center, a small orange square at the top right, a small yellow semi-circle inside an orange square on the left, a small yellow circle and a larger yellow circle outline at the bottom left, and several other small orange and yellow geometric shapes scattered throughout.

# The Keys To An Accessibility Mindset

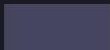
DANIEL YUSCHICK

# Table Of Contents

- 1 Relating To Accessibility
- 2 The POUR Principles Of Accessibility
- 3 The Keys To An Accessibility Mindset
- 4 Questions & Resources



# Relating To Accessibility



# Accessible Design Is Good Design



peacock













# The Benefits Of Responsive Design Are The Benefits Of Accessible Design

- Requires cross-team collaboration
  - Is never *'done'*
  - Benefits an ever-expanding audience
  - Directly impacts conversion and sales
  - A person expects their hardware to be supported
- 
- Products only require responsive design

Tell Me Why



# Web Accessibility is Fundamentally Misunderstood

	Permanent	Temporary	Situational
Touch	 One arm	 Arm injury	 New parent
See	 Blind	 Cataract	 Distracted driver
Hear	 Deaf	 Ear infection	 Bartender
Speak	 Non-verbal	 Laryngitis	 Heavy accent

# The POUR Principles Of Accessibility





Perceivable



Operable



Understandable



Robust

# The POUR Principles Of Accessibility

P

## Perceivable

Providing content that can be interpreted in layers by multiple senses.

- Images with descriptive alternative text
- Videos with captions and/or subtitles
- Indicating state with more than just color

O

## Operable

Ensuring content and experiences are interactive for multiple device types.

- All functionality available by mouse must be available by keyboard
- Visible and consistent focus states on all interactive elements
- All pages have clear titles with descriptive, sequential headings

# The POUR Principles Of Accessibility

U

## Understandable

An experience that can be reliably navigated with simple content written for the audience.

- Defining content language(s)
- Consistent landmarks, like navigations
- Error messages are descriptive and, when possible, actionable

R

## Robust

Creating content that can be accessed by a wide set of devices and platforms.

- Validating the rendered HTML to ensure it can be reliably interpreted by devices
- Using markup to assign names and roles to non-native elements

# The Keys To An Accessibility Mindset

# Markup Is Functional

## Not Visual

### Markup Is Functional, Not Visual

**Radio Group**

A wonderful thing about design is its ability to transform rudimentary tasks and elements into appealing and adaptive design languages. However, the way something looks is not always indicative to how it works.

Consider an option to change which selection a store's prices are shown. While a design may make these options look like buttons, they're technically working as a radio option where one item can be selected at a time. Using functional markup changes how the content is communicated and controlled.

**Without operable and perceivable markup**

USD EUR GBP

**With perceivable markup only**

USD EUR GBP

**With functional markup**

Prices Currency:

- ☒ USD
- ☐ EUR
- ☐ GBP

Key: To An Accessibility Mindset

@DanielYurchick

### Markup Is Functional, Not Visual

**Progress Bars**

A wonderful thing about design is its ability to transform rudimentary tasks and elements into appealing and adaptive design languages. However, the way something looks is not always indicative to how it works.

Some native elements, like **progress**, have limited styling capabilities and may require the use of non-standard markup. In these cases when markup is visual, markup must also communicate as clearly as its design.

**Without perceivable markup**

25%

**With perceivable markup**

75%

**With functional markup**

# If You Can Do It With A Mouse

## You Can Do It With A Keyboard

### Markup Is Functional, Not Visual

#### Links and Buttons

A wonderful thing about design is its ability to transform rudimentary tasks and elements into appealing and adaptive design languages. However, the way something looks is not always indicative to how it works.

#### Using an `anchor` to navigate

When navigating with a link, the element can be focused, the url can be previewed while hovering/focusing, the url can be copied and the page can be opened in a new tab or window.

< Back

#### Using a `button` to navigate

However, when navigating with a `button` that expected functionality is lost.

< Back

#### Using a `div` to navigate

Saving the worst for last. When navigating with a `div` all native and expected operability is lost. The element cannot natively be focused and the native cursor does not indicate interactivity. This is only operable by a traditional mouse click.

< Back

[Visit CodePen Demo](#)



# Markup Must Communicate

## As Clearly As Design

### Markup Must Communicate As Clearly As Design

#### Indicate The Current Page

When working from a design, it's common to build what we see. However, the visual design is only one part of creating perceivable content.

Because some people may not be able to see a visual cue indicating the current page in the navigation, use `aria-current="page"` to ensure the markup communicates the same as the UI.

Without `aria-current="page"`

Home **Products** Blog Contact

With `aria-current="page"`

Home **Products** Blog Contact

### Markup Must Communicate As Clearly As Design

#### Live Notifications

When working from a design, it's common to build what we see. However, the visual design is only one part of creating perceivable content.

Because some people may not be able to see notification circle, use screen-reader-specific markup to ensure it communicates the same as the UI.

Without perceivable markup



With perceivable markup



# How To Know When Content Needs Perceivable Markup

1

When design visually communicates state

*E.g. Active page, active tab, loading, disabled*

2

When information is outside the flow of content

*E.g. Toast message, dialogs, error messages, tooltips*

3

When content is dynamic or live

*E.g. Notifications, chat, tabs*

4

When recreating standard elements

*E.g. Progress bars, accordions, select menus*





# Use **More Than Color** To Indicate State

HEL

Helsinki

✈

KGL

Kigali

Time To Destination

2:37

Local Time at Origin

17:47

Altitude

12505m

Estimated Time of Arrival

19:29

Local Time at Destination

17:47

Ground Speed

898 km/h

Imperial

Metric

HEL

Helsinki

✈

KGL

Kigali

Time To Destination

2:37

Local Time at Origin

17:47

Altitude

12505m

Estimated Time of Arrival

19:29

Local Time at Destination

17:47

Ground Speed

898 km/h

Imperial

Metric

Recap

# Recap

1 Perceivable, Operable, Understandable, Robust

2 Markup Is Functional, Not Visual

3 If A Mouse Can Do It, So Must A Keyboard

4 Markup Must Communicate As Clearly As Design

5 Use More Than Color To Indicate State



# Questions & Resources

# Resources

- [CodePen Demos Collection](#)
- [W3 Understanding WCAG 2.2](#)
- [W3 Web Accessibility Principles](#)
- [Layman's Accessibility Checklist](#)
- [Accessible Error Messaging - Wix](#)
- [Microsoft Inclusive Design](#)
- [Axe Dev Tools](#)
- [Axe VSCode Linter](#)
- [A11y Color Contrast Figma Plugin](#)
- [Build & Test Accessible Web Apps Course - Marcy Sutton](#)
- [Learn Accessibility Course - Carie Fisher](#)
- [Practical Accessibility Course - Sara Soueidan](#)





Thank You!

@DANIELYUSCHICK