

Keeping Sites Accessible With Types

Fotis Papadogeorgopoulos (@isfotis)

Accessibility

On The Web

Accessibility on The Web

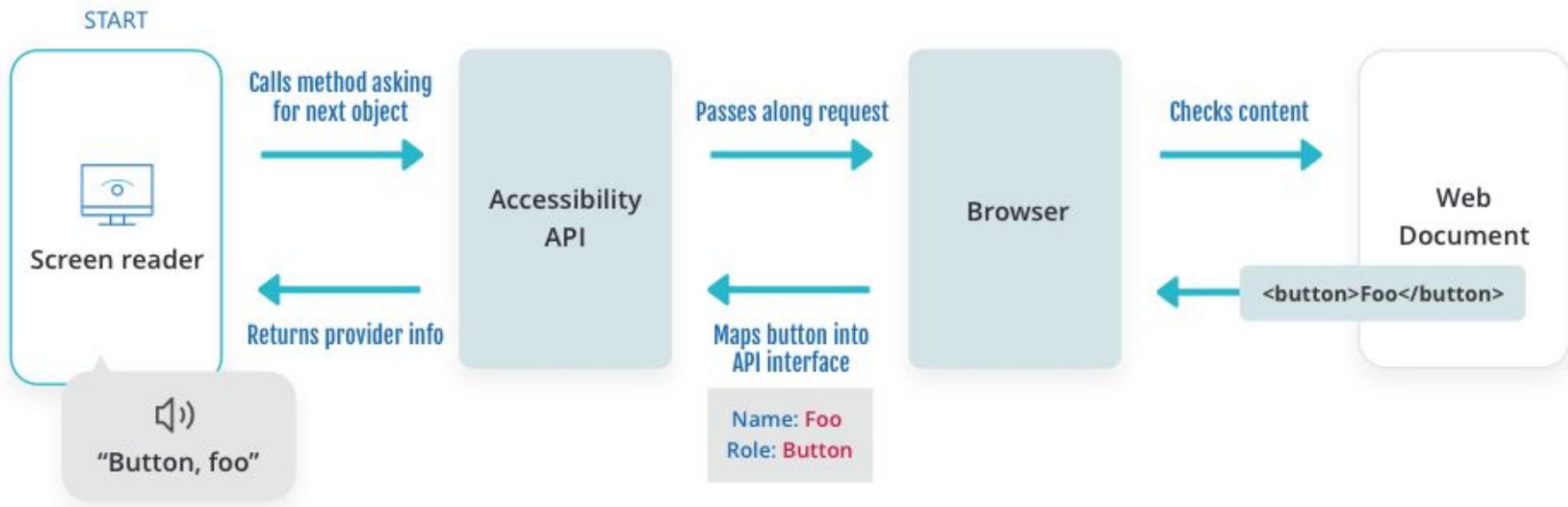
Communicate information to users

Name + Role + Value (+ Interaction)

“Book tickets, button, press enter to activate”

Covered in [Web Content Accessibility Guidelines \(WCAG\)](#)

Flow 1. Screen Reader wants to present next object

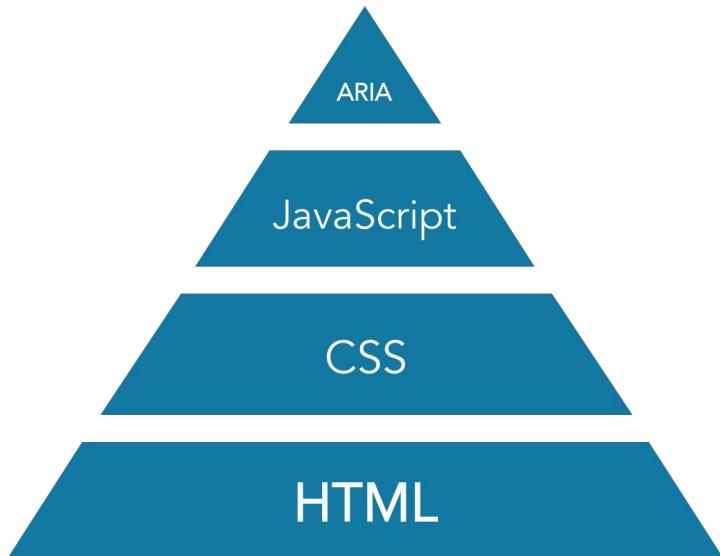


Source:

Semantics to Screen Readers, by **Melanie Richards**
alistapart.com/article/semantics-to-screen-readers

@isfotis

The Accessibility Cake



Markup + Styling + Interactivity

Source:

<http://simplyaccessible.com/article/the-accessibility-stack/>

Common Issues

Heading Order

Headings provide the outline for the document.

They must be consistently nested in the source.

h1 Information About the Tour

Jump to a section

[Dunk](#) [Shootout](#) [Awards](#) [Prizes](#) [Handbook](#) [Media Services](#) [History](#)

h2 Dunk

h3 Dunk competition

FIBA 3x3 World Tour events are famous for their dunk competitions. Check out [the 2018 Dunk playlist](#) on FIBA 3x3 Youtube channel!



h3 The format of the competition

Rounds: Dunkers have 75 seconds and 3 attempts per round to complete a dunk. An attempt means the dunker had the intention to dunk and he or the ball touched the rim and/or the backstop unit. The only exception to the 75-second-rule is the third round of the final where there is no time limit. The 75 seconds start running

Form Labels

Labels provide visual guidance and hierarchy

Labels expose the accessible name of the input

Placeholders don't count!

Log In

*“Text input,
username or email,
empty”*

- Don't have a profile yet? [Sign up](#)
- [Forgot your password?](#)

Username or email

A text input field containing "user@example.com". To the right of the input field is a small icon showing three dots and the number "3", indicating character count or suggestions.

Password

A password input field containing "correcthorsebatterystapl". To the right of the input field is a small icon showing three dots and the number "3", indicating character count or suggestions.

LOGIN



> 50%

Of home pages on the web, don't have labels

Source: <https://webaim.org/projects/million/#wcag>

Missing Image Alt Text

Images, both as and <svg>:

- If they are content, need alternative text
- If they are decorative, should be marked as such

68%

Of home pages on the web, don't have alt text

Source: <https://webaim.org/projects/million/#wcag>

On a bad day, or tight
deadline, even the most
diligent person will get
things wrong

Pause and ponder,
with types

```
type HeadingLevel = H1 | H2 | H3 | H4 | H5 | H6

hX : HeadingLevel -> List (Attribute msg) -> List (Html msg) -> Html msg
hX level =
  case level of
    H1 ->
      h1

    H2 ->
      h2

    H3 ->
      h3

    H4 ->
      h4

    H5 ->
      h5

    H6 ->
      h6
```

```
heading : HeadingLevel -> List (Attribute msg) -> List (Html msg) -> Html msg
heading level attrs children =
    hX level (class "myHeading" :: attrs) children
```

```
subHeading : HeadingLevel -> List (Attribute msg) -> List (Html msg) -> Html msg
subHeading level attrs children =
    hX level (class "mySubHeading" :: attrs) children
```

```
view : Model -> Html ()
view model =
  div []
    [ heading H1 [] [ text "Heading Demo"
      , nestedSection H2 "Hello"
    ]
  ]

nestedSection : HeadingLevel -> String -> Html msg
nestedSection level demoData =
  div []
    [ subHeading level [] [ text "Data"
      , p [] [ text "This is a section about data." ]
    ]
    -- Propagate the heading level + 1
    , subHeading (incrementLevel level) [] [ text "Count"
      , p [] [ text demoData ]
    ]
  ]
```

Type Mismatch

Line 31, Column 25

The 1st argument to `nestedSection` is not what I expect:

```
31|     , nestedSection [] [text "Hello"]  
      ^^
```

This argument is a list of type:

List a

But `nestedSection` needs the 1st argument to be:

HeadingLevel

```
type
  InputLabelMethod
    -- Label contains the input + text
    = EmbeddedLabel String
      --Label is associated with a DOM id
    | LabelledById DomId
```

```
type DomId
  = DomId String
```

```
domIdToString (DomId idRef) =
  idRef
```

```
-- Creating an input
input : InputLabelMethod -> List (Attribute msg) -> List (Html msg) -> Html msg
input labelMethod attrs children =
  case labelMethod of
    EmbeddedLabel labelText ->
      label []
        [ text labelText
        , Html.input attrs children
        ]
    LabelledById (DomId idRef) ->
      Html.input
        (id idRef :: attrs)
        children

InputLabel : DomId -> List (Attribute msg) -> List (Html msg) -> Html msg
InputLabel idRef attrs children =
  label (for (domIdToString idRef) :: attrs) children
```

```
-- In use
nameLabelId =
  DomId " nameLabel"

view : Model -> Html ()
view model =
  div []
    [ input (EmbeddedLabel "Email") [ attribute "type" "email" ] []
    , inputLabel nameLabelId [] [ text "Name" ]
    , input (LabelledById nameLabelId) [ attribute "type" "text" ] []
    ]
```

The part where we talk
about things

Hey, I have an
error about
labels. What
do I pick?

The part where we talk about things

Hey, I have an
error about
labels. What
do I pick?

Embedded
labels might be
simpler

The part where we talk about things

Hey, I have an error about labels. What do I pick?

Embedded labels might be simpler

Ah, we also have more high-level ones

The part where we talk about things

Hey, I have an error about labels. What do I pick?

Embedded labels might be simpler

Ah, we also have more high-level ones

Let's look at it together

“In this codebase, we
care about form labels”

```
type Purpose
  = Decorative
  | Content String

img : Purpose -> List (Attribute msg) -> Html msg
img purpose attrs =
  case purpose of
    Decorative ->
      Html.img
        ((alt "") :: attrs) []
    Content description ->
      Html.img
        ((alt description) :: attrs) []
```

```
view =
  div []
    [ -- Ok
      img
        Decorative
        [ src "/kitten.jpg" ]
    , img
        (Content "The tops of colored buildings, lit by
sunset.") [ src "/sunset.jpg" ]
    , -- Error! Expected "Purpose"
      img [ src "/error.jpg" ]
    ]
```

“What is my purpose?”

```
-- A base-level SVG image
svgImg : Purpose -> List (Attribute msg) -> List (Svg msg) -> Html msg
svgImg purpose attrs content =
    -- NOTE: There are more ways to do this!
    case purpose of
        -- If Decorative, hide from Assistive Technologies
        Decorative ->
            svg ([ attribute "aria-hidden" "true" ] ++ attrs) content

        -- If Content, set the aria-label, and treat it as img (content)
        Content description ->
            svg
                ([ attribute "aria-label" description
                  , attribute "role" "img"
                  ]
                 ++ attrs
                )
            content
```

```
-- Creating a reusable image/icon
homeIcon : Purpose -> List (Attribute msg) -> Html msg
homeIcon purpose attrs =
    svgImg purpose attrs
        [ polyline [ points "21 8 21 21 21 3 21 3 8" ] []
        , rect [ x "1", y "3", width "22", height "5" ] []
        , line [ x1 "10", y1 "12", x2 "14", y2 "12" ] []
        ]
```

```
-- In use
view model =
  nav [] [
    a [href "/"] [homeIcon (Content "Home") []],
    a [href "/entries"] [
      homeIcon Decorative [],
      text "Home"
    ]
  ]
```

```
type Purpose =
  | {type: 'decorative';}
  | {type: 'standalone'; label: string;};

const Decorative = (): Purpose => ({type: 'decorative'})
const Standalone = (label: string): Purpose => ({type: 'standalone',
label})
interface IconProps extends SVGAttributes<SVGElement> {
  purpose: Purpose;
  color?: string;
  size?: string | number;
}
```

Types will not solve
everything...

There are layers of
expertise, visible and
invisible work

Types can help you ask
the right questions

Types can help make
the work visible

References

- [Tessa Kelly's `accessible-html` package for Elm](#)
- [Sara Soueidan on Accessible SVG icons](#)
- [Scott O' Hara on Contextually Marking up images](#)
- References to labels

Thanks!

Want to work with me?

Futurice is hiring (futurice.com/careers/)

Get in touch

@isfotis

fotis@fpapado.com

Let's chat in the breaks!

@isfotis

Thanks!

HELP :)