

# Best Practices for Making a Wagtail Site Accessible

Scott Cranfill (he/him)  
DjangoCon US 2023

# Best Practices for Making a Wagtail Site ~~Accessible~~ *as Accessible as Possible*

Scott Cranfill (he/him)

DjangoCon US 2023

# Who am I?

- Full Stack Web Developer at NASA's Jet Propulsion Laboratory
- Resident of Rochester, New York
- Member of the Wagtail Core Team



BLM ROC logo credit: [www.dunwoode.design](https://www.dunwoode.design)

# What is Wagtail?

- The leading open-source Python CMS
- Powered by Django
- Free for everyone
- Used by Google, Mozilla, NHS, CFPB

<https://wagtail.org>

<https://github.com/wagtail/wagtail>

# Target audience

- Developers like me who work on Wagtail-powered websites
- Those who might be considering Wagtail

# Introduction to accessibility

- Ensuring our web content is able to be consumed by the widest possible spectrum of people
- A legal responsibility for many
  - Domino's Pizza ADA lawsuit<sup>1</sup>
  - Section 508<sup>2</sup>
- A moral responsibility for all

<sup>1</sup> <https://www.boia.org/blog/the-robles-v.-dominos-settlement-and-why-it-matters>

<sup>2</sup> <https://section508.gov>

# Wagtail's accessibility initiatives

- Accessibility subteam
- Accessibility statement<sup>1</sup>
- Two prongs:
  - Accessibility of Wagtail itself
  - Accessibility of sites it produces
- Google Summer of Code and Outreachy projects

<sup>1</sup> <https://wagtail.org/accessibility>

# GSOC and Outreachy projects

- Forced colors support (high contrast mode)
  - GSOC participant: Anuja Raj Verma
  - Mentors: Jane Hughes, Scott Cranfill, Thibaud Colas
- Accessibility checker
  - Outreachy intern: Albina Starykova
  - Mentors: Sage Abdullah, Joshua Munn, Thibaud Colas



# Accessibility checker demo

**There is still a lot that developers  
need to do to set their editors up  
for success**

# What is still up to developers

- Semantic markup in templates
- Configuring Wagtail's accessibility checker
- Enforcing proper heading hierarchy
- Promoting good alt text
- Using ARIA labels, where appropriate
- Helping editors help themselves

# Semantic markup in templates

- Your first responsibility as a developer
- Ensures the ability to navigate around the site in a predictable manner
- Primary concern in a Wagtail context: **landmarks**
- Take advantage of implicit roles<sup>1</sup>

<sup>1</sup> <https://www.a11yproject.com/posts/aria-landmark-roles/#html5-implicit-mappings-of-landmark-roles>

# Configuring the accessibility checker

- Default is to only show errors that an editor can address
- Bakery demo is set up to show all errors to any user
- Recommendation: configure to show all errors to superusers<sup>1</sup>

<sup>1</sup> [https://docs.wagtail.org/en/stable/advanced\\_topics/accessibility\\_considerations.html#built-in-accessibility-checker](https://docs.wagtail.org/en/stable/advanced_topics/accessibility_considerations.html#built-in-accessibility-checker)

# Heading hierarchy

# Heading hierarchy

- Sometimes a developer responsibility
- Sometimes an editor responsibility
- Use custom validation to guide editors
  - Heading blocks and `RichTextBlocks` (override StreamField's `clean()`)
  - `RichTextFields` (subclass it and override `clean()`)
  - Both (override page model's `clean()`)

# Alt text



```

```



# Alt text in Wagtail

- Default image model does not provide a field for alt text
- Alt text is automatically set to the image's title field when using `{% image %}` template tag
- Early advice: create a custom image model to add alt field<sup>1</sup>
- Current thinking: **don't** add to model because alt text should be *contextual*

<sup>1</sup> [https://docs.wagtail.org/en/stable/advanced\\_topics/images/custom\\_image\\_model.html](https://docs.wagtail.org/en/stable/advanced_topics/images/custom_image_model.html)

# Current recommended approach to alt text

- Create your own Streamfield image block with an alt field for maximum flexibility
- Keep an eye out for "contextual alt text"<sup>1</sup>

<sup>1</sup> <https://github.com/wagtail/rfcs/pull/51>

# ARIA labels

```
<dt>2023 Annual Report</dt>
```

```
<dd>
```

```
  <a href="annual-report-2023.pdf">Download</a>
```

```
</dd>
```

```
<dt>2022 Annual Report</dt>
```

```
<dd>
```

```
  <a href="annual-report-2022.pdf">Download</a>
```

```
</dd>
```

# ARIA labels

```
<dt>2023 Annual Report</dt>
```

```
<dd>
```

```
  <a href="annual-report-2023.pdf"
```

```
    aria-label="Download 2023 Annual Report">
```

```
    Download
```

```
  </a>
```

```
</dd>
```

# ARIA label hardcoded in a template

```
<dl class="download-list">
  {% for document in value %}
    <dt>{{ document.title }}</dt>
    <dd>
      <a href="{{ document.url }}"
        aria-label="Download {{ document.title }}">
        Download
      </a>
    </dd>
  {% endfor %}
</dl>
```

# ARIA labels entered by editors

```
class LinkBlock(StructBlock):
    text = CharBlock()
    page = PageChooserBlock(required=False)
    external_url = URLBlock(required=False)
    aria_label = CharBlock(required=False)

class Meta:
    value_class = LinkStructValue
    template = "blocks/link_block.html"
```

[https://docs.wagtail.org/en/stable/advanced\\_topics/customisation/streamfield\\_blocks.html](https://docs.wagtail.org/en/stable/advanced_topics/customisation/streamfield_blocks.html)

# ARIA labels entered by editors

```
<a href="{{ value.url }}"
  {% if value.aria_label %}
    aria-label="{{ value.aria_label }}"
  {% endif %}>
  {{ value.text }}
</a>
```

# Helping editors help themselves



## help\_text

- Django's built-in tool for providing hints when editing a field

# Help text on a heading block

```
size = ChoiceBlock(
    choices=[ ... ], blank=True, required=False,
    help_text=mark_safe(
        'Please ensure that you do not skip heading
        levels. For example, the next heading after an H2
        should only be either an H3 or another H2. <a
        href="https://www.a11yproject.com/posts/how-to-
        accessible-heading-structure/">Learn more about
        heading structure</a>'
    )
)
```



## Heading block



Heading text \*

## Heading level 2

### Size

Please ensure that you do not skip heading levels. For example, the next heading after an H2 should only be either an H3 or another H2. [Learn more about heading structure](#)

H2



# Help text on an alt text field

```
alt_text = CharBlock(  
    required=False,  
    help_text=mark_safe(  
        'If this image is not purely decorative, enter  
a text alternative to be displayed if images fail to  
load or read by screen reader software. <a  
href="https://www.a11yproject.com/posts/alt-  
text/">Learn more about how to write alt text</a>'  
    )  
)
```



## Image block



### Image \*



Baking Soda ...

### Alt text

If this image is not purely decorative, enter a text alternative to be displayed if images fail to load or read by screen reader software. [Learn more about how to write alt text](#)

A small porcelain bowl of baking soda with a carved wooden scoop

### Caption

Baking Soda

### Attribution

Creative Commons

# Help text on an ARIA label field

```
aria_label = CharBlock(  
    required=False,  
    label="ARIA label"  
    help_text=mark_safe('If your link text is generic,  
like "Read more" or "Download", enter something more  
descriptive to be read by screen readers instead, like  
"Read more about our commitment to accessibility". <a  
href="https://a11y-101.com/development/aria-  
label">Learn more about ARIA labels</a>')  
)
```



## Link block



Text \*

Page



Choose a page

External URL

ARIA label

If your link text is generic, like "Read more" or "Download", enter something more descriptive to be read by screen readers instead, like "Read more about our commitment to accessibility". [Learn more about ARIA labels](#)

## HelpPanel

- Use for page-level guidance
- Editor panel with no field; just displays any HTML content you define<sup>1</sup>
- Cleaner alternative to help text on every field for things like image-heavy pages

<sup>1</sup> <https://docs.wagtail.org/en/stable/reference/pages/panels.html#helppanel>



# Recap

- Use semantic markup in templates
- Configure Wagtail's accessibility checker to show developers all errors
- Enforce proper heading hierarchy
- Add alt text fields and promote better alt text
- Add ARIA label fields and educate editors on when to use them
- Provide timely help to editors as they work on their content

# Further reading

[https://docs.wagtail.org/en/stable/advanced\\_topics/  
accessibility\\_considerations.html](https://docs.wagtail.org/en/stable/advanced_topics/accessibility_considerations.html)

# Thank you!

<https://github.com/Scotchester/djangocon-wagtail-accessibility>

Questions? Find me in the hallway or on Slack!