

Marian Foley

Senior Content Designer

GOV.UK Accessibility and inclusion project

Government Digital Service

The GOV.UK Accessibility and Inclusion Project



It's not filthy in here!

Accessibility...

To find and fix WCAG fails caused by
content across GOV.UK

Inclusion...

To improve disability inclusion on
GOV.UK

**Making GOV.UK
more inclusive**

1. The language we use to talk about disability

Problem term	Inclusive term	Mainstream	GDS other	Other depts
deaf and dumb/deaf mute	deaf, user of British Sign Language (BSL), person with a hearing impairment	0	0	0
insane	person with a mental health condition	0	0	13
suffers/suffer/suffering from	Suggests discomfort and hopelessness. Consider a more positive expression.	25	20	1,513
attacks	seizures	7	31	3,742

2. The way we design content for disabled users

Dealing with HMRC if you have additional needs

Contents

- Help you can get
- [If you're deaf, hearing-impaired or have a speech impairment](#)
- [If you're blind or partially sighted](#)
- [If English isn't your first language](#)

Get help from HMRC if you need extra support

- Help you can get
- [If you cannot use a telephone and need a different way to contact HMRC](#)
- [If you need information in a different format](#)
- [If you need help filling in forms](#)
- [If you need more time because of your circumstances](#)
- [If you need information in another language](#)
- [If you need someone to talk to HMRC for you](#)

**Making GOV.UK
content more
accessible**

1. **Identifying** where content can fail WCAG standards
2. **Finding** those fails
3. **Fixing** them

Identify



All requirements ("success criteria") from 2.0 are included in 2.1. The 2.0 success criteria are exactly the same (verbatim, word-for-word) in 2.1.

There are
[What's New](#)

WCAG is HARD

ced in

Content that conforms to WCAG 2.1 also conforms to WCAG 2.0. (This is often called "backwards compatible".) A website that meets WCAG 2.1 should meet the requirements of policies that reference WCAG 2.0.

To put it another way: If you want to meet both WCAG 2.0 and WCAG 2.1, you can use the 2.1 resources and you don't need to bother looking at 2.0.

WCAG 2.0 and WCAG 2.1 are both existing standards. WCAG 2.1 does not deprecate or supersede WCAG 2.0. W3C encourages you to use the most recent version of WCAG when developing or updating content or accessibility policies.

We used Al Duggin's WCAG primer:

<https://alphagov.github.io/wcag-primer/>

Links

- multiple links with the same text (on same page) that point to different destinations **F**
- multiple links to the same destination (on same page) that use different link text **B**
- starting with verbs that don't lead to an action **B**
- top links to inaccessible external sites **B**
- including 'here', 'click' or 'more information' **B**
- one-word links **B**

Headings and titles

- heading structures that miss a level **F**
- pages on GOV.UK with the same title **F**
- pages with more than one H1 **B**

Language attributes

- pages set up with the wrong language attribute **F**

Style issues

- eg (screen readers may read this as ‘egg’) **B**
- ie **B**
- negative contractions **B**

Tables

- no relational information about cell or row headings **B**

Use a hash (#) character after the pipe if the cell is the row title.

```
govspeak
|
| ----- | Monthly direct debit | One-off payment |
| # Two-wheeled vehicle | £72 | £60 |
| # Three-wheeled vehicle | £132 | £120 |
```

This is what that looks like on GOV.UK:

	Monthly direct debit	One-off payment
Two-wheeled vehicle	£72	£60
Three-wheeled vehicle	£132	£120

Images

- no alternative text information **F**

Useful alt text guidance

[W3C alt text decision tree](#)

[GOV.UK guidance on alt text](#)

Find



There are 535,218
content items on
GOV.UK

Automagically

Jenkins

- specific task - keyword searches
- simple outputs telling us everything we needed to know about each content item
- built for content designers
- limited access (4 people in team)
- searches content only

Knowledge Graph

- keyword and relational searches (if X and Y are true but Z is false)
- uses cipher queries and node.js
- could potentially be used by everyone at GDS
- this project has optimised it for content designers

Actual magic



Reports in progress

- multiple links with the same text pointing to different destinations **F**
- multiple links to the same destination that use different link text **B**
- heading structures that miss a level **F**
- pages on GOV.UK with the same title **F**
- pages with more than one H1 **B**
- style guide issues, images and tables

Future reports:

- links including the words ‘here’, ‘click’ or ‘more information’ **B**
- one-word links **B**
- language attributes **F**
- links starting with verbs that don’t lead to an action **B**
- top links to inaccessible external sites **B**

Fix



The heading issues report has over
115,000 lines

Publishing app	Errors	Details	Action
Collections Publisher	341 (54% of app content)	Topic AZ and curated topic pages	Dev correction
Content Publisher	158 (18% of app content)	Human error caused by app design confusion	Department and app design correction
Manuals and HMRC Manuals	72,370 (85% of apps content)	2 x h1s	Report anomalies - investigate and confirm
Publisher (Mainstream)	777 (30% of content)	Guide chapters and possible human error	Dev correction and human correction - investigate
Travel advice publisher	225 (50% of content)	2 x h1s	Dev correction
Whitehall	c40,000	looks like human error	sift and tell depts

The duplicate page titles report
shows 80,000 errors across
GOV.UK

GDS is fixing

- template issues that cause content accessibility fails
- all the issues our reports show in GDS-owned content

But that's quite a small
percentage of the problems...

For example, there are 24,489 pages on GOV.UK with tables in them. Only 321 of them belong to GDS.

**We need other
departments on
board**

Please get excited - this is our
chance to really fix things!

Tell everyone you know....

We plan to send departments reports showing where their issues are over the next couple of months.

We'll send one report at a time
because otherwise the size of the
reports crashes everyone's
devices!



**Future
possibilities....**

The accessibility project:

- rerun the reports every six months to pick up any new errors and show how much we've fixed
- build failsafes into publishing apps

The knowledge graph:

- it's now optimised for content designers!
- add it to the content data tool
- give people outside GDS access
- find data on end-to-end user journeys

Questions?