



A practitioner's accessibility handbook

For product teams

 **thoughtworks**

Foreword

1. Introducing our accessibility newbies	5
2. What is accessibility (a11y)?	7
What kinds of disabilities are we looking at?	7
3. Why is accessibility important?	10
How to convince stakeholders about the importance of accessibility (a11y)	10
The legal aspect of accessibility	13
4. Accessibility Standards	15
5. How to build accessible products (a11y ops)	19
Set clear standards and guidelines	19
Include accessibility criteria in existing ways of working	22
Shift accessibility left	25

6. Overview of tools that disabled people use	26
Screen readers	26
Screen readers by platform	27
Keyboard navigation	27
List of keyboard shortcuts	28
Screen magnifiers	30
Built-in screen magnifiers by platform	30
7. Integration in daily workflows	31
Design	32
Development	36
How to build an HTML Page	36
When to use ARIA	37
Testing strategy	38
Tools for automated testing	38
Debugging	39
References	39
Issue prevention opportunity	39
8. Keep learning and exploring!	42
Glossary of terms	43
Acknowledgements	44
Authors	44

Foreword

In the summer of 2022, a small group of designers and engineers in Germany came together to answer the question, “What would help fellow Thoughtworkers –and other practitioners– take their first steps in accessibility?”

In answer to this question, a few members of the group created this guide, a key slice in the multi-faceted pie being baked by the Accessibility Trainers–Thoughtworkers who are committed to increasing awareness, expertise, and excellence in the area of accessibility and inclusive design.

In collaboration with allies from Australia, the UK, North America and other regions, Sophie Kano, Andree Müller, Larissa Günther, and Kylie Sy have created an engaging entrée into the world of accessibility through the eyes of three characters: Sally, Ben, and Ayla take us through their own step-by-step exploration, sharing their discoveries and resources along the way.

In the end, they bring invaluable insights to their clients, stakeholders, and teams–while making products the best they can be for all users.

We hope that you will enjoy following their path of learning.

May this playbook be an invaluable resource to you on your journey.

Minette Mangahas

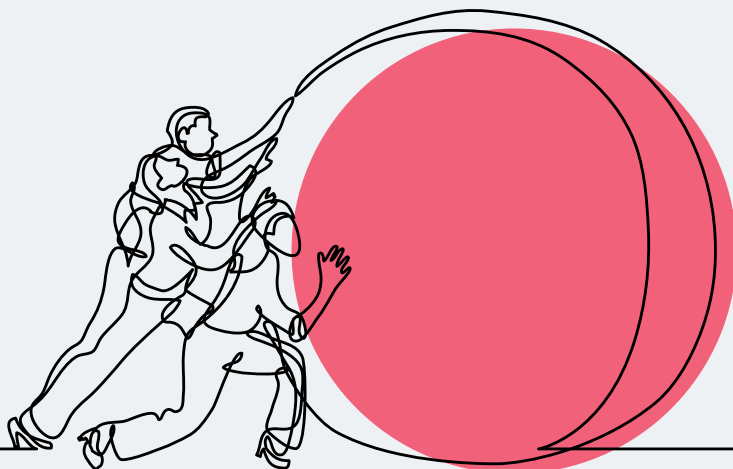
Principal Experience Designer

Berlin

1. Our accessibility newbies

Meet Sally, Ben and Ayla!

They are three professionals who do not know much about accessibility yet, but want to learn more about it. Making sure digital products are accessible will require different skills from each of them. Follow them on their journey on becoming accessibility specialists! Please note that this handbook is focusing mainly on web accessibility and doesn't cover mobile accessibility.





Sally is a full-stack **developer** and since she found out a friend of hers is red-green colorblind she's become more interested in how she could develop more accessible web applications.



Ben is a **UX Designer** and he knows the very basics of accessibility, like checking color contrasts and choosing appropriate font sizes. He's aware that this is just the very tip of the iceberg and wants to learn more about it — especially how to collaborate with his developer colleagues and how to implement accessibility standards from the start.



Ayla is a **business analyst**, passionate about lean product development and delivering real value to customers. Lately, she's noticed that more stakeholders want to follow accessibility standards because "they don't want to get sued." She thinks this is a great opportunity to embed accessibility into products from scratch and is eager to learn more.

2. What is accessibility (a11y)?

As accessibility newbies, Sally, Ben and Ayla ask themselves what is meant by the term accessibility and what are the expectations behind it when a team sets the goal of developing an accessible product.

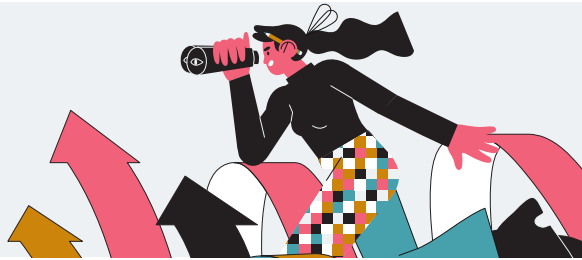
Accessibility is freedom. In a broad sense, accessibility means that environments, things, and processes can be used or followed by all people. For digital content, accessible means that it's usable by anyone on any device.

Accessibility is often shortened to **a11y** — as in, “a”, then 11 characters, and then “y”. Pronounced as “ally”. You will see this abbreviation being used in this handbook, too.

We traditionally think of this as being about people with disabilities, but the practice of making digital products accessible also benefits other groups such as those using mobile devices, or those with slow network connections.



What kinds of disabilities are we looking at?



One big myth about accessibility is that it only benefits users with disabilities. Whenever we hear the term disability, we may think of people with visual impairments, hearing impairments, mobility impairments, or cognitive impairments. Disability is actually universal and dynamic, most people experience disabilities daily. Disabilities can be categorized as permanent, temporary, and situational.

Permanent disabilities are disabilities that will not change or get better over a period of time. They also don't change if you're in a different space or environment. With a permanent disability, when and where doesn't change the actual disability. It remains the same.

Temporary disabilities are disabilities that will change or get better over time. You can be anywhere in the world, but the only thing that will help is time, such as recovering from a sports injury.

Situational disabilities are a little more difficult to explain, and hence more easily overlooked. These are disabilities that wouldn't normally be called disability, but they are scenarios where one needs similar accommodations as someone with a permanent or temporary disability. In this case, the disability doesn't have anything to do with time, it is only relative to particular scenarios.

2. What is accessibility (a11y)?

Wow! Sally, Ben and Ayla are surprised by this information. Until now, they had never thought about the fact that they could also be affected by disabilities. The three of them feel an increased motivation and conviction that accessibility is also important for their team. However, they cannot provide the right arguments to convince their team yet.



Additional resource

If you would like to learn more about the different forms of disabilities, check out the [types of disabilities here](#)

by **Anne Gibson**

Pro tip

Don't use simulations to promote disability awareness. Ayla asks herself how she can build empathy for different forms of disabilities to build better digital products? In a conversation with a Thoughtworks accessibility trainer, she learned that simulating disabilities tends to create a sense of negative empathy, but does less to truly understand what it's like to live with a disability. The trainer points out that instead of using simulations, it is better to work directly with people with disabilities.

3. Why is accessibility important?

Ben would like to make his team aware of why accessibility standards are important and incorporate accessibility practices more into their product and workflow. But he can already hear his colleagues say something like:

“Isn’t the percentage of people with disabilities very small? I’m not sure if it would make a big difference...”

“How much more time and effort will this take? We have tight deadlines - can’t we take care of it in the next quarter?”



Sounds familiar? If you’re in the same situation as Ben and want to convince colleagues or stakeholders why accessibility is important, here are some facts to get you prepared to persuade them.

How to convince stakeholders about the importance of accessibility (a11y)

Are you a product manager, product owner, project manager or in a similar role directing or working with multidisciplinary

3. Why is accessibility important?

teams? Then this section will be interesting for you. Here are some hard facts and numbers that might be helpful to create strong arguments:

- **Two in every 10 people around you are experiencing a disability.** That's 1,580,000,000 people, or 20% worldwide. So chances are high that these people are using your product
- The spending power of disabled households in which at least one of the members has a disability is **13 trillion USD per year** (elderly people not included). So, if your company decides on making a digital product not accessible you're likely to miss out on this huge market share
- If a product is more accessible, 86% of people with disabilities would spend more on it
- In the UK alone, the click-away pound is worth £17.1 billion per year (that means the sales lost made when somebody can't complete a transaction because of missing accessibility)
- We often think of disabilities as something inherited, but **anyone can acquire a disability.** In fact, everybody tends to lose physical abilities as they age: seeing, hearing, mobility, cognition, and more. This means by building products accessible is an investment for yourself in the future, even if it doesn't seem like it right now
- What makes a website good to use for people with disabilities **will also benefit people without disabilities** on mobile devices, different kinds of browsers or computers. In fact, a lot of innovations that we take for granted today were created for disabled people, e.g. automated captions on video content, created for deaf people, or text to speech, created for the blind

3. Why is accessibility important?

- Not only people with disabilities benefit from accessible websites! **Search engines can index and catalog better** when a web page is accessible
- A lawsuit for **not complying with accessibility** standards will incur **extraordinarily high costs**. Costs that will be much higher than the additional time invested to build accessibility in from the start
- The **negative publicity** a company might receive from a lawsuit can be even more expensive. In the worst case scenario, a company decides to fight against the lawsuit - essentially deciding not making products accessible at all costs. Being **known as a company that intentionally discriminates against people with disabilities** can be the worst publicity possible and will undermine positive things the company may be doing

Sources: dequeuniversity.com, Matthew Johnston 2022



Additional resource

Did you get curious now about how many people who visit your website might have disabilities? Take a look at this page to find out: <https://how-many.herokuapp.com/> It gives estimates based on the UK national average figures. Try it out!

3. Why is accessibility important?

Ben is happy, this helps! He will go ahead and share these facts with Ayla to build up an alliance with her to convince their stakeholders.

The legal aspect of accessibility



Ayla is confident that with all the facts Ben shared with her they will be able to convince their stakeholders that accessibility is important for the product they are building. But before starting the conversation with their stakeholders, she would also like to understand the current legal situation regarding accessibility better, so she can consult with her clients on what they need in order to be compliant with the law.

Do you also feel like outlining legal requirements would help make the accessibility case for your product? Then below you can find an overview of the current legal situation in Europe.

While the EU Web Accessibility Directive has required all public sector websites and mobile apps adhere to accessibility requirements since 2020, the EU private sector has been largely unregulated when it comes to accessibility. This is now changing with the [European Accessibility Act \(EEA\)](#), which legally requires EU member states to enact accessibility requirements via their national law. The EEA is expected to have a knock-on effect around the globe, like the General Data Protection Regulation (GDPR), as it will require many private companies that do business in the EU to adhere to the EEA or face potential legal consequences.

3. Why is accessibility important?

The EEA applies to private companies that provide products and services sold or used **within the EU**, including

- Computer software (e.g. operating systems, applications, e-books)
- Websites (e.g. banking services)
- Apps
- e-commerce (e.g. shopping websites and apps)
- Hardware (e.g. computers, smartphones, tablets, ticketing and check-In machines, banking equipment)

According to the EEA these products and services should from now on follow a **universal design approach**: “the design of products, environments, programs and services to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design”. While the EEA establishes a set of common accessibility rules for private sector companies to follow, it is still up to the individual EU member states how to enact this in their national laws - in Germany the EEA is, for example, enacted with the Barrierefreiheitsstärkungsgesetz.

The EEA must be enforced from July 2025. Products and services that don't adhere to the EEA accessibility rules after that may be subject to fines or legal action.



Further reading

Right now, this guide only covers the legal aspects in Europe in detail. Check this [Global Directory](#) to see how legislation looks in other countries.

4. Web accessibility standards

Ayla, Ben and Sally are all convinced now that building a product accessible to all users is the right thing to do. They already have a good overview of different forms of disabilities for which they want to ensure their product is usable. But they are still a bit unsure when exactly they can call their product fully accessible.

Sally is asking the others:

“Can we come up with some kind of accessibility standards or criteria so we are confident our product is accessible when it meets these criteria?”



Luckily, you don't have to come up with those standards all by yourself. They are already out there: The Web Content Accessibility Guidelines (WCAG) are a set of recommendations for making web content more accessible, published by the Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C) - the main international standards organization for the world wide web.

Although the WCAG are not legally binding, they are internationally established and widely accepted. Many public and private companies and national laws use WCAG as a basis

4. Web accessibility standards

for their accessibility standards. The United States, the European Union, the UK, Canada, Australia and Israel have established laws to make it legally binding for a lot of websites to comply with the WCAG.

Currently, there are three versions of WCAG that build upon each other:

- WCAG 2.0 - published December 2008
- WCAG 2.1 - published June 2018
- WCAG 2.2 (Draft) - not published yet

WCAG 2.0 and WCAG 2.1 are both existing standards. WCAG 2.1 does not deprecate WCAG 2.0 but does introduce additional accessibility criteria – especially considering mobile devices. The W3C recommends always using the most recent version of WCAG. Accordingly, WCAG 2.1 is currently the most common standard for accessibility used by organizations and governments.

The WCAG standards are organized under four principles, abbreviated as **POUR**:

- **P**erceivable - information and user interface components must be presentable to users in ways they can perceive
- **O**perable - user interface components and navigation must be operable
- **U**nderstandable - information and the operation of the user interface must be understandable
- **R**obust - content must be robust enough that it can be interpreted by a wide variety of user agents, including assistive technologies

4. Web accessibility standards

Each of these principles has a set of guidelines, with testable success criteria. These success criteria determine the level of conformance of the WCAG: they are basically a checklist – in order to meet WCAG the product you are building needs to meet the success criteria. One success criterion would be, for example, “All non-text content that is presented to the user has a text alternative that serves the equivalent purpose.” (Also known as “alt-text”).

As you can see, it’s easy to determine if your product conforms or not: whenever you have non-text content—like an image—which does not have alt-text, then you don’t conform.

The success criteria are at three levels:

- **A** - lowest standard
- **AA** - medium standard
- **AAA** - highest standard

The levels build on each other – conformance at higher levels indicates conformance at lower levels. Typically, products strive for Level AA conformance as it is also the standard by which most legal requirements are judged.

Level A is the bare minimum and usually considered unacceptable for actual accessibility. Level AAA on the other hand might not be feasible to achieve across a whole product. While many products have some AAA level features, it is often considered too high of a bar to have Level AAA across the whole product. This is because AAA guidelines are stringent in adding additional features such as minimum seven to one contrast ratio, the optional removal of background noise in audio and

4. Web accessibility standards

video, limited flashing and animation, and the inclusion of sign language interpretation for all prerecorded audio content in synchronized media.

In summary, products usually strive for AA level conformance for the whole product.

5. How to build accessible digital products (a11y ops)

Ayla, Ben and Sally have a good understanding of why accessibility is important, about the existing standards and are eager to get started hands on.



Ayla is wondering now what a strategy could look like that business, design and development could follow. If she just goes ahead and sets “accessibility” as a common goal for the product, she’s afraid her colleagues won’t even understand what to do with it or what this means exactly for their work. She would like to make it as clear as possible how to reach the “accessibility” goal, so the team will be willing to get started with it.

Set clear standards and guidelines

In the ideal case, you should have a simple list of criteria your product should comply with to be accessible —so teams can easily evaluate where they are and what they still need to do. Even though this list might not include all possible cases (there will be issues you only find through proper usability testing),

clear standards can be a great help for teams getting started with accessibility, so they don't feel lost on this complex topic.

You can create an accessibility criteria list for your product in three steps:

Step 1: agree on a standard

Try to agree with your team or even the whole project group to have one common standard that you strive to be compliant to long term, as a baseline for your accessibility work. Maybe the company you're currently working in already has some standard for accessibility defined? Then you should definitely follow that one.

The most commonly used standard for web accessibility is the Web Content Accessibility Guidelines (WCAG). It is a set of internationally accepted success criteria for making the web accessible and is used by a lot of companies and national laws as a standard. As described in the previous chapter "Accessibility Standards".

You can use WCAG as a base to set up the cross-functional requirements that the project is going to follow. Cross-functional requirements define how a system is supposed to be, for example, accessible, secure and auditable, etc. By defining an accessibility cross-functional requirement, you can establish a standard that products must meet. For example: "The system shall be accessible to comply with WCAG 2.1 standards."

Step 2: fit it to your product

The WCAG are very extensive and detailed, which is great, but can be at the same time quite overwhelming - especially for accessibility beginners. You can prioritize according to what

you're building - not based on what disabilities you think you need to consider because you should consider them all - but based on the features you are providing.

An example would be that you're currently building a web app that does not contain any kind of video or audio data - accordingly, you don't have to consider the criterion in the WCAG to make video and audio data in the web accessible. In the WCAG 2.1 AA, this already means five less success criteria.

Step 3: create a checklist

Put all the criteria that you identify as relevant for your product in a hands-on checklist that is measurable (e.g. percentage of criteria that are fulfilled) and usable in different phases of the product process:

- Planning
- Designing
- Developing
- Testing

Teams should be able to use this checklist for doing accessibility audits on built features and for making sure they bake in accessibility for features they are currently building. It's a good idea to include recommendations on how to test for these criteria, when to consider this in the product process and which role would be the most suitable to complete the checks. It's recommended to assign ownership, like having an accessibility champion in a team who keeps tabs on the effort overall and flag topics like proper delegation of tasks.

Include accessibility criteria in existing ways of working



“Choose a standard, fit it to my product and create or use an existing checklist. Okay, I can work with that!” Ayla thinks.

The only thing she is still worried about is how she is going to convince her colleagues about the extra work. She is confident that she can convince them to start working with accessibility standards once they plan it in for the next quarter. But they should get started with it right away.

It can often be hard to get the work on accessibility prioritized as an individual workstream and it might also not be very sustainable to reach compliance with your accessibility criteria in a one time effort aside from your regular work. It’s great if you can assign all forces to only accessibility for some time – but what happens afterward? Accessibility is not a one time effort but similar to security or usability, something you have to continuously work on to be successful.

It’s best to find ways to include accessibility in the existing workflows for your team so it 1) does not seem like extra work and 2) the efforts for doing it are realistic to include in a normal working day. The closer accessibility is aligned with your current way of working the easier it will be for your team to adhere to it.

Look into what your team is already doing well and think about ways to include accessibility.

Here are some ideas:

Include accessibility checks in acceptance criteria

When you're working with stories with acceptance criteria, it can be a good (and easy) practice to include accessibility checks

Example acceptance criteria:

"Given I have added two ideas with scores, when I click the "Rank" button, then ideas are sorted with the top-scoring ideas at the top."

The term "click" refers to the ability of using a mouse, but you can add a second acceptance criteria for the same thing, for keyboard-only users.

"Given I have added two ideas with scores, when I tab through the page and toggle the Rank button (with ENTER or SPACE key), then ideas are sorted with the top-scoring ideas at the top."

You can extend this even further for screen reader users:

"Given I have added two ideas with scores, when I navigate with my screen reader through the page, then the Rank button is read out to me so I can understand its purpose and when I toggle it, the ideas are read out to me, sorted with the top-scoring ideas at the top."

With just tweaking the wording a bit for your "default" acceptance criteria, you've already covered most of the

accessibility criteria for this feature. You also make teams aware of what to look out for while building this feature and what to test for when evaluating if this story is done.

by default in stories. Often acceptance criteria are limited to the “default” interaction pattern of being able to use the mouse and see the screen - but they can easily be extended to comply with accessibility criteria:

Create personas with disabilities

If you’re regularly looking at personas to design new features and plan properly for the needs of your users, personas with disabilities can help your team keep accessibility in mind and avoid accessibility issues right from the start.

Some examples for such personas:

- Disability user profiles: UK Government
- Stories of web users: Web Accessibility Initiative



Integrate automated accessibility in your pipeline

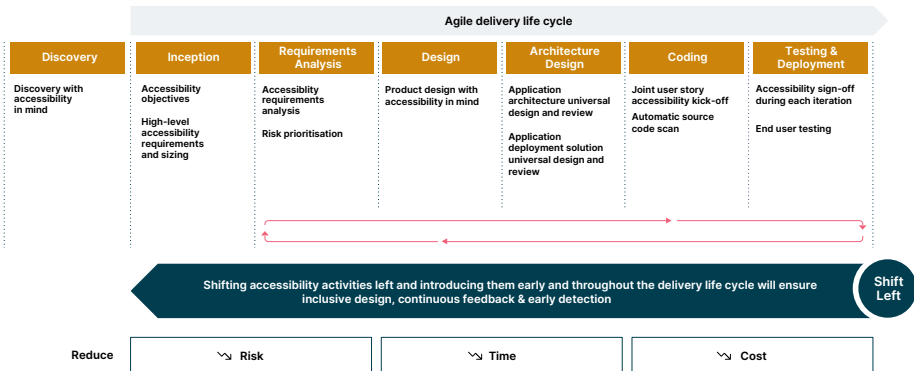
If your team has a focus on automating things, including accessibility checks in your pipeline can reduce the manual workload of testing for accessibility (see the Testing strategy chapter).

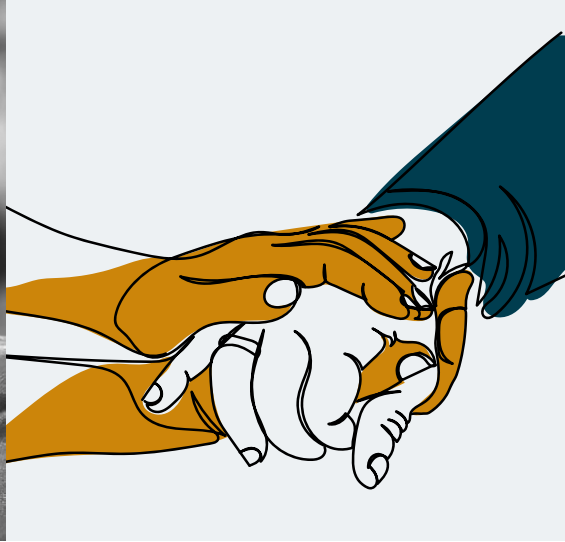
These are just some ideas on how to utilize already existing ways of working to include accessibility in your daily tasks. It is important that you include accessibility work in a way that your team can make it a continuous practice.

“This is what we need to do!” Ayla thinks. She is thrilled and is going to take these points to Ben and Sally, her Design and Development colleagues, to integrate these practices into their existing workflows.

Shift accessibility left

Generally, it is better to incorporate accessibility practices right from the start. If you have the opportunity to work on a new digital product, try to do so as early as possible. Unfortunately, we often find ourselves in a project that has been going on for quite some time or working on a product that has been developed for years. In this case, it is a good practice to shift accessibility activities left on the delivery life cycle. The graphic below gives an overview of common phases and which activities should be incorporated when.





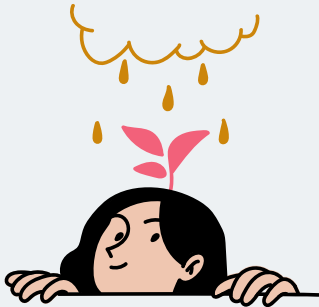
This is a cultural shift for whole organizations, so it's okay if it takes time. It's also a mindset shift - "if you don't intentionally and deliberately include accessibility practices you will unintentionally exclude people with disabilities."

Matthew Johnston

Global head of disability inclusion, Thoughtworks

6. Overview of tools that disabled people use

Screen readers



Sally hasn't worked with screen readers or other assistive technologies before and wants to get familiar with them now. She goes ahead and downloads one on her machine. After five minutes, she is startled:

"Wow. I had no idea what screen readers are about – it is basically reading every bit of content off the screen for people with vision impairment. If a website is not developed considering how screen readers work this can be incredibly frustrating!"

Sally's experience is quite common – tools for disabled people work very differently. It can be a very humbling experience and we recommend trying out one or two tools to actually understand what it's like to use these kinds of tools.

As a quick start, check out this video and see how a blind person uses a screen reader: [How A Blind Person Uses A Computer](#)

Screen readers by platform

Use these for a start, but be aware of limitations.

Windows

- [NVDA](#)
- [JAWS](#) (commercial)
- Narrator (built-in)

Mac

- VoiceOver (built-in)

Linux

- [Orca](#) (Fedora, Ubuntu, and more)

Mobile

- TalkBack on Android (built-in)
- VoiceOver on iOS (built-in)



Further reading

[Most used screen readers survey](#)

[Keyboard shortcuts for screen readers](#)

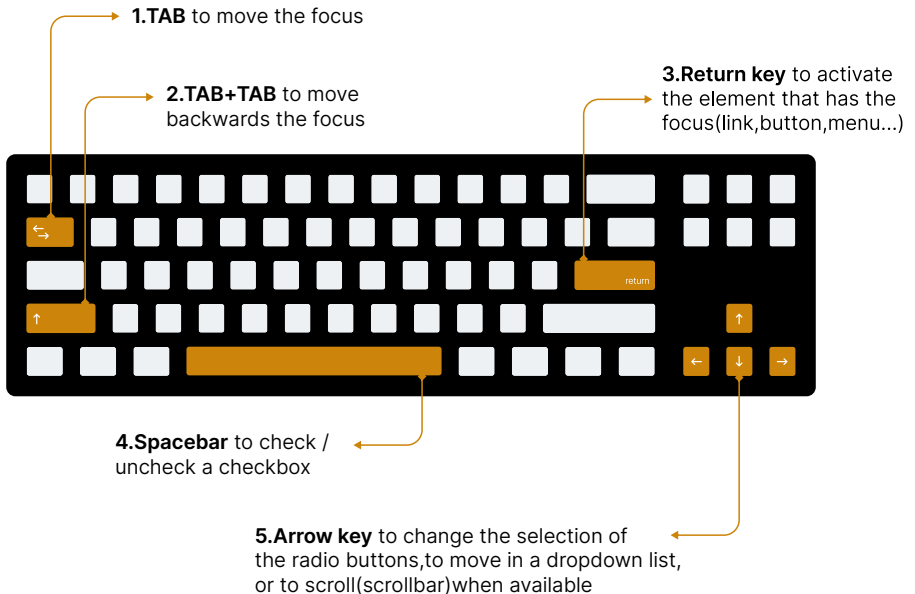
Keyboard navigation

There are countless different assistive technologies, but they share a common base, which is that most of them base their functionality on keyboard access. For example, voice control software is based on keyboard accessibility. So if you ensure keyboard accessibility, you'll naturally ensure accessibility for most of these other technologies and devices.

To test if your service is accessible using the keyboard, you can try to navigate without your mouse. All available features must be accessible.

Note: the focus must be visible enough on all focusable elements.

List of keyboard shortcuts



6. Overview of tools that disabled people use

Interactions	Shortcuts
To move the focus	Tab
To move backwards the focus	Shift + Tab
To activate the element that has the focus (link, button, menu...)	Return key
To check / uncheck a checkbox	Spacebar
To select of the radio buttons into a radio group	arrow keys Up / Down or Left / Right
To move in a drop-down list	arrow keys Up / Down, then Spacebar
To scroll(scrollbar)	arrow keys Up / Down for a vertical scrollbar, arrow keys Left / Right for a horizontal scrollbar

Note: on Mac, keyboard navigation must be enabled first

In Safari: Edit > Preferences... > Advanced

Then check the box *“Press tab to highlight each item on a webpage”*.

Screen magnifiers

A screen magnifier is a piece of software that enlarges a certain part of your screen. People with low vision use it. The following video shows an example on how these are used [Screen readers and screen magnification - Getting interested in technology.](#)

Built-in screen magnifiers by platform

- [Windows Magnifier](#)
- [Mac Zoom](#)

7. Integration in daily workflows

In chapter five, How to build accessible products (a11y ops) it was mentioned that it's key to include accessibility into existing ways of working. There's no need to reinvent the wheel – build on what you already do.



Ben wants to start with incorporating accessibility practices in his design work. But he is not sure where to begin.

“How do I make sure that in the future my designs are accessible? Where do I start?”

To get him going, here's a series of topics to check off when designing. Make sure to check some points with your front-end developer colleagues.

Please note that the following list is web/browser focused – but most of the points about visual design should also be valuable for mobile focused development.

Design

Page title

- The browser page title (the title that is displayed in the browser or tab; not the main content heading or <h1> describe the topic or purpose of the page?

Headings

- Are all headings marked as such? The heading levels should communicate the hierarchical order and not only visual styling

Navigation

- Is it possible to skip repetitive navigation and move to the main content? This includes providing e.g. header, navigation, main, footer (known as HTML/ARIA landmarks) and “skip navigation” links
- Is the content’s reading order logical and intuitive? And is the code order following this order?

Links

- Does the link text describe the purpose or destination of the link clearly?

Color

- Is information communicated with means other than color? Using additional text labels to indicate an error is good practice
- Using different shapes in diagrams will also help people with dyschromatopsia identify differences

Contrast

- Does the color contrast of all text against its background pass with a rating of at least AA? You can download several plugins for your prototyping tool. For Figma and Sketch the Stark plugin is a good choice
- Are link texts differentiable from non-link text by more than just color?
- Do actionable elements have clear and visible focus when people use tab or arrow keys?
- Do all non-text elements, like user interface components or graphical objects that are important for accessing and understanding content, have sufficient color contrast that passes with an AA rating?

Magnification and responsive design

- Is magnifying or zooming in on the content in the browser on any device, including desktop and mobile, possible without problems?
- Is the design optimized for all zoom states? It's recommended to simplify the design as much as possible, especially eliminating horizontal scrolling
- Can users use the website with their device in their preferred orientation (landscape/portrait)?

Images

- Does the alternative text (alt-text) for images provide the same information as the image? Are complex infographics fully described in the page content or with a short alternative text description?

7. Integration in daily workflows

- Does the alternative text for actionable images as image links or image maps make it clear what the destination link is or what the button purpose is?
- Are decorative images identified as not requiring alternative text? Are these images really necessary?
- Is plain text used instead of embedded text in images? (Text in logos or decorative text in images can be an exception)

Tables

- Does the table have a caption (name or title), and are columns and rows properly identified in the markup?
- Are tables simplified to minimize or eliminate the need for compound column or row headers?

Forms

- Do all form fields have a label that is always visible?
- Are all form labels descriptive and instructive? Is all the information the user needs to fill out on the form visible and not hidden while entering inputs?
- Are all form labels and instructions near to their form element so that users (e.g. using screen magnification) can easily connect the form element with its label and/or instructions?
- Are all controls close to the content they are controlling? For example, are Edit and Delete buttons next to the content they modify?
- Do error messages provide enough and clear information without jargon so users can correct their errors?
- Are forms written so they can be used with Autofill?

Dynamic content

- Are users made aware of content that is dynamically inserted on a page (e.g. error messages after an input)? Or does the new content come right after the element that caused it to appear, in the logical reading order of the page?
- Do all keyboard-only and touch screen interactions follow expected patterns so users know how to interact with all widgets on the page?
- Is success and failure feedback incorporated into all interactions? E.g. when users activate scripted functionality (buttons, form submissions, etc.), or the obvious activation of a feature (e.g. a video starts to play after the user activates the “play” button), etc. The feedback must be available to sighted users, screen reader users, and all other user categories

Custom widgets

- Does the design use standard HTML widgets (links, buttons, form elements, controls, etc.) whenever possible? Native widgets have built-in accessibility capabilities. Custom widgets do not
- If you do have any custom widgets, have they been created with full keyboard support, and are they compliant with WAI-ARIA authoring practices?

Touch devices

- Are the touch target size of main links and buttons large enough and far enough away from each other to activate easily with a finger? A minimum size of 44 px in height and width is recommended.

- Is there an alternative way to activate any custom swipe actions or gestures? Note that when a screen reader is activated on a touch device, it overrides all custom swipe actions and gestures.

Development



Sally wants to upskill her fellow developers. She did a deep dive into accessible development and compiled some basic strategies on how to build an accessible website.

“There’s a lot of material out there. How can we build accessible HTML by default? Are there tools to automate our workflow?”

How to build an HTML page

- To achieve bare minimum accessibility, using the correct HTML is essential. When there is already an equivalent native element, it is always better to use that, as it is already built with assistive technologies in mind. This is called semantic HTML. For example, instead of
`<div>Close</div>`
use
`<button>Close</button>`

It is focusable via keyboard Tab key and is also discoverable by screen readers, which prompts the users that it is something that can be interacted with

- Make sure to structure your text content, so that the flow is easily understandable. This includes using `<h1>` for headings and `<p>` for paragraphs. See the [Design section](#) on how to connect with the designer on these aspects
- For more information, check out [HTML accessibility best practices from MDN](#)

When to use ARIA

ARIA stands for Accessible Rich Internet Applications and is a W3C standard specification for additional markup that can be applied to languages like HTML to add accessibility information that is not possible with standard HTML.

- The first [rule of ARIA](#) is to not use it when a native HTML equivalent is available. For example, a `<button>` element is better than using `<div role="button">`, as the latter loses default functionalities, such as keyboard focus and screen reader interactivity
- When using semantic HTML is not enough to communicate meaning to the users of assistive technologies, this is the right time to use ARIA. [This example of a flyout menu](#) using aria-expanded tells screen reader users when a menu is opened
- Make sure to check if your code is supported with various assistive technologies. A useful resource is: <https://a11ysupport.io/>

Testing strategy

According to [Deque's study](#), 57.38% of accessibility issues can be caught by automated testing. This enables us to fix issues immediately. However, reaching a 100% pass on automated tools does not always mean that your page is accessible. The [limitations](#) can be solved by creating a mindset that "...accessibility doesn't automatically mean usability for people with disabilities". This requires constant collaboration with UX designers.

Tools for automated testing

- [Testing Library](#) has the guiding principle of: "The more your tests resemble the way your software is used, the more confidence they can give you." You can have a rule that if we cannot find the element without **using `getByTestId`**, then it is not an accessible element. The following is an example of a test with keyboard and screen reader:
 - » You are building a menu element. You know that this is an interactive button that opens a list. We can use `getByRole("button", {name: Menu})`. Screen reader reads it as a button, and a keyboard user can navigate to it
- Adding accessibility tests such as part of your unit testing framework using [Axe-chai](#)
- Using static code analysis such as [Eslint plugin for JSX](#) to check if you are using the correct markup
- For a full page analysis that you can also integrate in your CI pipeline:

7. Integration in daily workflows

- » [Pa11y](#)
- » [Lighthouse](#)
- » [Axe](#)

Debugging

Tools that are already available by default in your browser:

- [Accessibility features in Microsoft Edge Dev Tools](#)
- [Accessibility features in Google Chrome Dev Tools](#)
- [Accessibility features in Firefox's Developer Tools](#)

In general, these are used for:

- Checking correct markup and the accessibility tree
- Checking contrast
- Simulating color blindness and low vision

References

[How to bake layers of accessibility testing into your process](#)

Issue prevention opportunity

We can often prevent web accessibility issues from occurring by addressing the accessibility criteria early in the design and development process. This is called proactive rather than reactive. Being reactive means waiting until some situation occurs and then responding to it. To be proactive means anticipating, planning and confronting problems before they reach crisis points. Becoming proactive puts you in control of the situation.

7. Integration in daily workflows

The table shows which role(s) has the greatest influence on preventing each problem from arising.

WCAG 2.0 Success Criteria (A & AA)	Designing	Coding	Automated Testing
1.1.1 Non-text Content	Strong	Moderate	Moderate
1.3.1 Info and Relationships	Strong	Strong	Moderate
1.3.2 Meaningful Sequence	Strong	Strong	None
1.3.3 Sensory Characteristics	Strong	None	None
1.4.1 Use of Color	Strong	None	None
1.4.2 Audio Control	Strong	None	None
1.4.3 Contrast (Minimum)	Strong	None	Strong
1.4.4 Resize text	None	Strong	None
1.4.5 Images of Text	Strong	Moderate	None
2.1.1 Keyboard	Strong	Strong	Weak
2.1.2 No Keyboard Trap	None	Strong	None
2.2.1 Timing Adjustable	Strong	None	Weak
2.2.2 Pause, Stop, Hide	Strong	None	Weak
2.3.1 Three Flashes or Below Threshold	Strong	None	None
2.4.1 Bypass Blocks	Strong	Strong	Moderate
2.4.2 Page Titled	Strong	Moderate	Strong
2.4.3 Focus Order	Strong	Strong	None

7. Integration in daily workflows

WCAG 2.0 Success Criteria (A & AA)	Designing	Coding	Automated Testing
2.4.4 Link Purpose (In Context)	Moderate	None	None
2.4.5 Multiple Ways	Moderate	None	None
2.4.6 Headings and Labels	Moderate	Moderate	None
2.4.7 Focus Visible	Strong	Moderate	None
3.1.1 Language of Page	None	Strong	Strong
3.1.2 Language of Parts	None	Strong	Weak
3.2.1 On Focus	Strong	Moderate	None
3.2.2 On Input	Strong	Moderate	None
3.2.3 Consistent Navigation	Strong	None	None
3.2.4 Consistent Identification	Strong	None	None
3.3.1 Error Identification	Strong	Moderate	None
3.3.2 Labels or Instructions	Strong	Moderate	None
3.3.3 Error Suggestion	Strong	None	None
3.3.4 Error Prevention (Legal, Financial, Data)	Strong	None	None
4.1.1 Parsing	None	Strong	Strong
4.1.2 Name, Role, Value	Strong	Strong	Moderate

(Barrell 2020 , p. 133-137)

8. Keep learning and exploring!

Sally, Ben and Ayla have come a long way. From knowing almost nothing about accessibility to having a good foundation of what accessibility is about, what to consider and how to integrate methods into their daily workflows.

They are confident now that they're able to create more inclusive web applications that will do people with disabilities more justice. They know now how many people with disabilities struggle if applications aren't built accessibly and that small adjustments can make a big difference.

Ben: "This has been quite a ride! It's great that once you build accessible applications the whole society is going to benefit!"

Ayla: "Yeah, I agree – we learned so much – but I guess this is just the beginning..."

Sally: "You're right – let's keep learning and exploring!"

The trio is motivated and thrilled. They want to level up their accessibility game even further and spread their learnings and methodologies to their colleagues and teams so in the future the whole company will create accessible products and services only.

Join them!

Glossary of terms

This hand is for agile team members and people working in or managing the central accessibility team. This glossary translates common terms used by one of these groups for the benefit of the other.

ADA (Americans with Disabilities Act): US Federal legislation that aims to protect the rights of people with disabilities.

ARIA (Accessible Rich Internet Applications): a W3C standard specification for additional markup that can be applied to languages like HTML to add accessibility information that is not possible with standard HTML.

EEA (European Accessibility Act): The European accessibility act is a directive that aims to improve the functioning of the internal market for accessible products and services, by removing barriers created by divergent rules in Member States.

Persona: A fictitious character created by user experience designers to inspire the design of exceptional user experiences for the intended target user base. They are intended to allow the designers to really understand who they are designing products for.

WCAG (Web Content Accessibility Guidelines): a series of W3C standards that specify the requirements for applications to meet in order for these to meet certain accessibility levels. WCAG 2 is the most commonly used version of this standard.

Acknowledgements

- Matthew Johnston, Thoughtworks Global head of disability inclusion
- Katie Peterson, Developer, Luminary Scholarship, Thoughtworks Australia
- Thoughtworks CXPD Service Line, Germany
- Deque University
- Barrell, D. (2020) 'Agile Accessibility Handbook - A Practical Guide to Accessible Software Development at Scale', Herndon, VA: Amplify Publishing.
- Burgstahler, S. (unknown) 'What are Alternatives to Disability-Related Simulations to Promote Disability Awareness?'. Available at: [What are Alternatives to Disability-Related Simulations to Promote Disability Awareness? | DO-IT](#) (Accessed: 25 October 2022)

Authors

Larissa Günther

Kylie Sy

Sophie Kano

Andree Müller

Thank you for making it to the very end of the handbook!

Get in touch with us

accessibility@thoughtworks.com

Thoughtworks Deutschland GmbH

Revaler Straße 29

10245 Berlin

Germany

+49 (0)30 555 73 5890

 **thoughtworks**