Building Inclusive

free and open-source software

Company Name

MM/DD/YYYY

The “Building Inclusive Open-Source Software” training

“Despite being deeply based on values of sharing and empowerment, Free and Open-Source Software (FOSS) projects often lack inclusiveness and accessibility.”

This project aims at providing accessibility and inclusiveness knowledge for organizations and individuals who build open-source software.

These modules should not be seen as holding the truth regarding the subjects tackled, but rather as a condensed introduction which goal is to help teams get started with accessibility and inclusiveness for their product.

Navigating the training

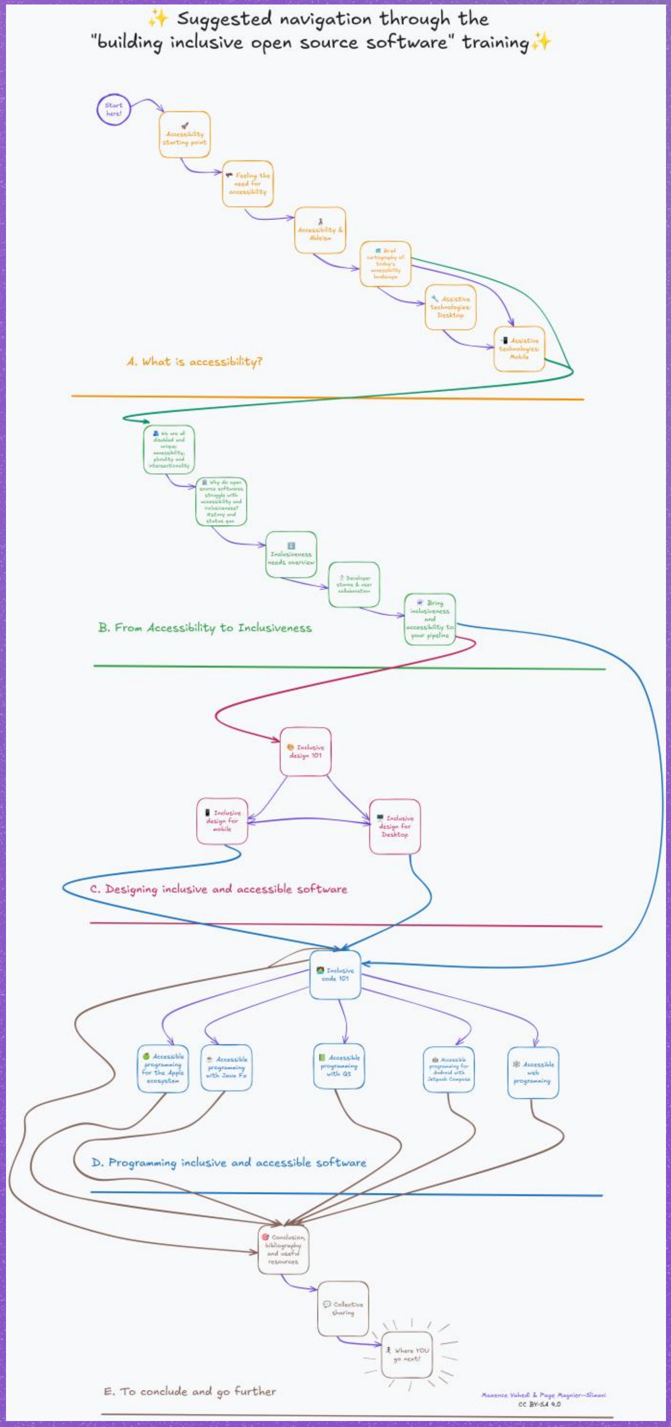
- **A. What is accessibility?**
  - 🚀 Accessibility starting point
  - 🫏 Feeling the need for accessibility
  - 🧑‍🦽 Accessibility & Ableism
  - 🗺️ Brief cartography of today's accessibility landscape
  - 🔧 Assistive technologies – Desktop
  - 📲 Assistive technologies – Mobile

- **B. From accessibility to inclusiveness**
  - 🔭 Inclusiveness needs overview
  - 🫂 We are all disabled and unique
  - 🏛️ Why does FOSS struggle with inclusiveness?
  - 🤺 Developer stance & user collaboration
  - ⚗️ Bring inclusiveness and accessibility to your pipeline

- **C. Designing inclusive and accessible software**
  - 🎨 Inclusive design 101
  - 🖥️ Inclusive design for Desktop
  - 📱 Inclusive design for Mobile

- **D. Programming inclusive and accessible software**
  - 🧑‍💻 Inclusive code 101
  - 🍎 Inclusiveness features and programming for Apple platforms
  - 📗 Inclusive programming with Qt
  - ☕ Inclusive programming with Java Fx
  - 🤖 Inclusive programming for Android
  - 🕸️ Inclusive web programming

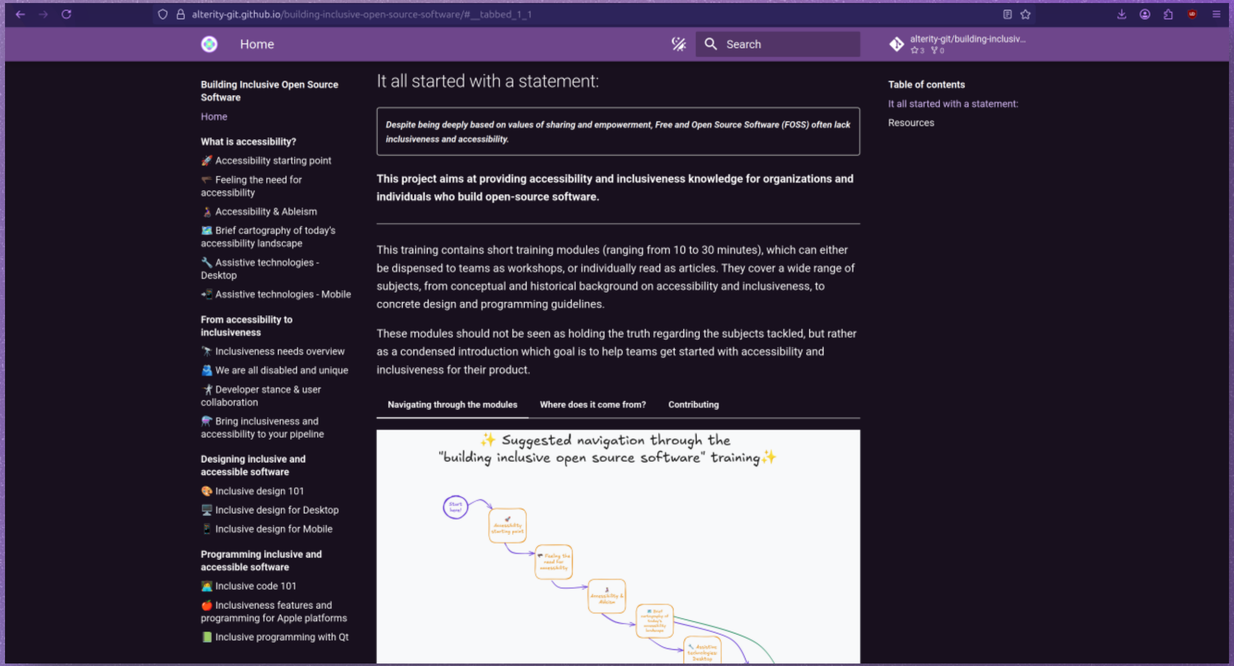
- **E. To conclude and go further:**
  - 🎯 Conclusion, bibliography and useful resources
  - 🏃 ➡️ Where YOU go next!
  - 💬 Collective sharing


✨ Suggested navigation through the "building inclusive open source software" training✨
Start here!
Accessibility starting point
Feeling the need for accessibility
Accessibility & Ableism
Brief cartography of today's accessibility landscape
Assistive technologies: Desktop
Assistive technologies: Mobile
A. What is accessibility?
We are all disabled and unique: accessibility, plurality and intersectionality
Why do open source softwares struggle with accessibility and inclusiveness? History and status quo
Inclusiveness needs overview
Developer stance & user collaboration
Bring inclusiveness and accessibility to your pipeline
B. From Accessibility to Inclusiveness
Inclusive design 101
Inclusive design for mobile
Inclusive design for Desktop
C. Designing inclusive and accessible software
Inclusive code 101
Accessible programming for the Apple ecosystem
Accessible programming with Java Fx
Accessible programming with Qt
Accessible programming for Android with Jetpack Compose
Accessible web programming
D. Programming inclusive and accessible software
Conclusion, bibliography and useful resources
Collective sharing
Where YOU go next!
E. To conclude and go further
Maxence Vahedi & Paye Maynier--Slimani
CC BY-SA 4.0

Where does it come from?

A collaboration between
Page Magnier--Slimani (Jami/Savoir-Faire Linux) and Maxence Vahedi (Olvid)

Using this project's content



alterity-git.github.io/building-inclusive-open-source-software/#__tabbed_1_1

Home

Search

alterity-git/building-inclusiv...

Building Inclusive Open Source Software

Home

What is accessibility?

- 🚀 Accessibility starting point
- Feeling the need for accessibility
- Accessibility & Ableism
- 🗺️ Brief cartography of today's accessibility landscape
- 🔧 Assistive technologies - Desktop
- 📲 Assistive technologies - Mobile

From accessibility to inclusiveness

- 🔭 Inclusiveness needs overview
- We are all disabled and unique
- Developer stance & user collaboration
- ⚗️ Bring inclusiveness and accessibility to your pipeline

Designing inclusive and accessible software

- 🎨 Inclusive design 101
- 🖥️ Inclusive design for Desktop
- 📱 Inclusive design for Mobile

Programming inclusive and accessible software

- Inclusive code 101
- 🍎 Inclusiveness features and programming for Apple platforms
- 📗 Inclusive programming with Qt

It all started with a statement:

> ***Despite being deeply based on values of sharing and empowerment, Free and Open Source Software (FOSS) often lack inclusiveness and accessibility.***

This project aims at providing accessibility and inclusiveness knowledge for organizations and individuals who build open-source software.

This training contains short training modules (ranging from 10 to 30 minutes), which can either be dispensed to teams as workshops, or individually read as articles. They cover a wide range of subjects, from conceptual and historical background on accessibility and inclusiveness, to concrete design and programming guidelines.

These modules should not be seen as holding the truth regarding the subjects tackled, but rather as a condensed introduction which goal is to help teams get started with accessibility and inclusiveness for their product.

Navigating through the modules | **Where does it come from?** | **Contributing**

✨ Suggested navigation through the "building inclusive open source software" training ✨

Table of contents

It all started with a statement:

Resources

Start here!
Accessibility starting point
Feeling the need for accessibility
Accessibility & Ableism
Brief cartography of today's accessibility landscape
Assistive technologies Desktop

Online documentation

You can contribute on this project's repository.

Accessibility starting point

What is accessibility?

Accessibility starting point

On a scale from one to ten, how do you evaluate your knowledge about accessibility?

How much time do you typically spend on accessibility issues during a week?

Is there something preventing you from dealing with the topic as much as you would ideally like to?

Feeling the need for accessibility

What is accessibility?

A few numbers about accessibility

27% of all Canadians*, 28% of 15+ French** have at least one disability

16% of the world's population experience significant disability.***

* Word health Organisation: Disability
**Le handicap en chiffres (DREES)
*** New data on disability in Canada, 2022

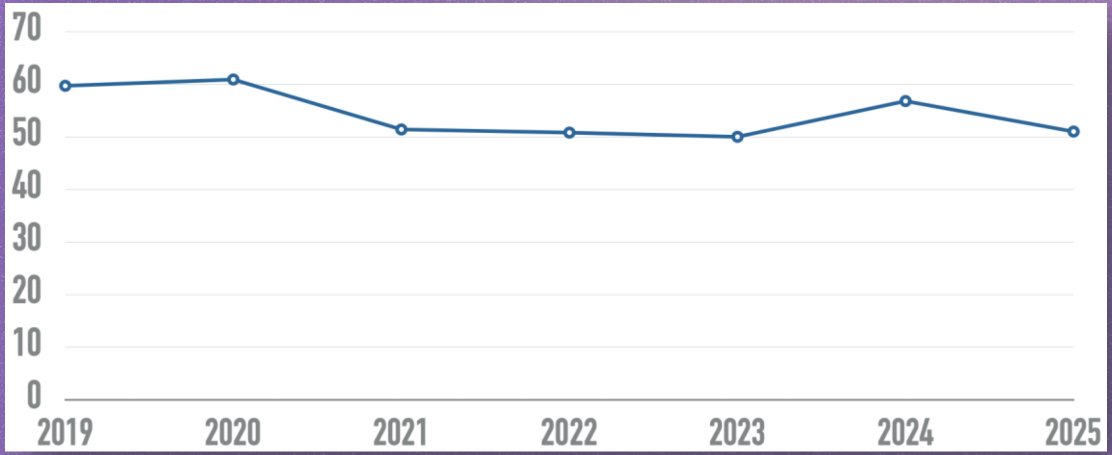
Activity

Let's try your app with a screen reader!

How does it feel? What are the blockers?

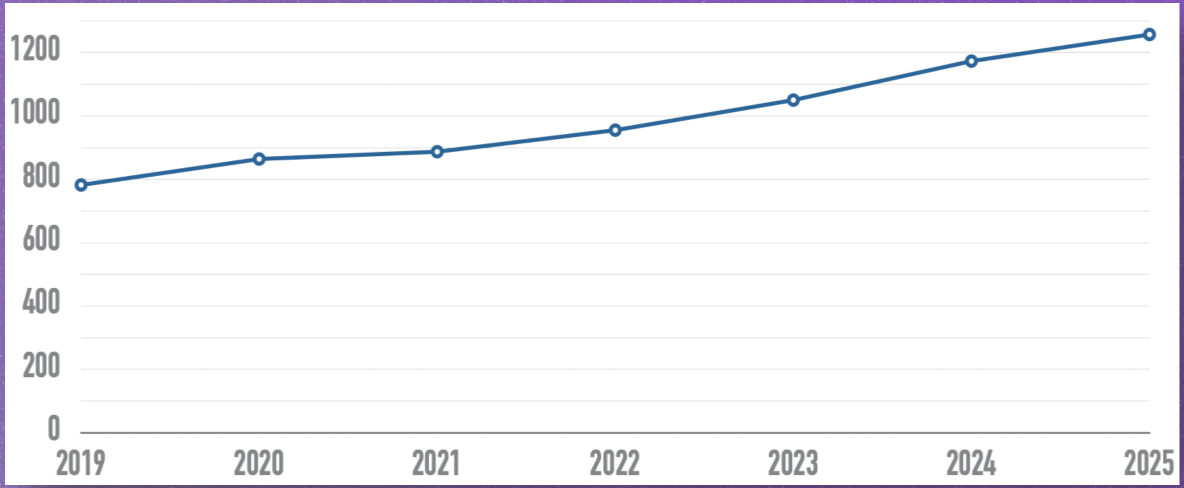
The systemic failure of software accessibility

based on a WebAIM Million 2025 report



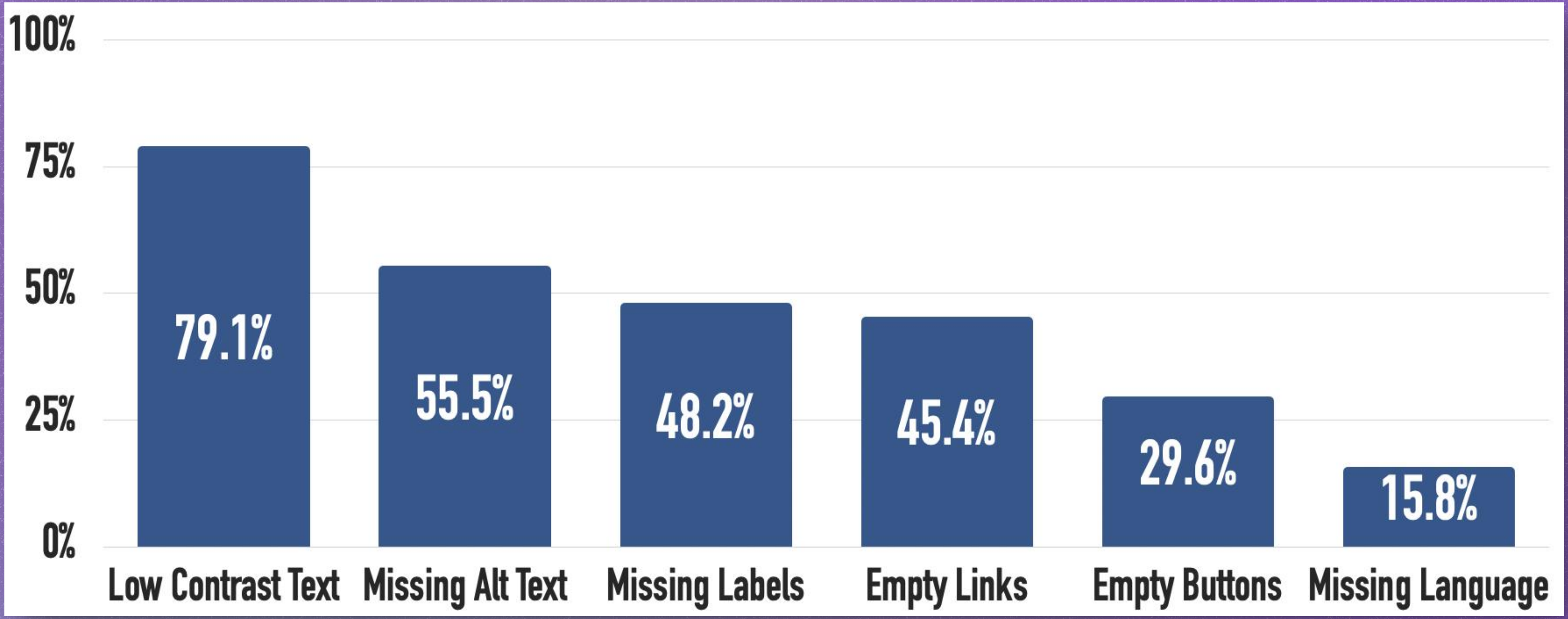
The WebAIM collective states that “**Across the one million home pages, 50,960,288 distinct accessibility errors were detected—an average of 51 errors per page**”.

Home pages tend to get more and more complex. While some websites are getting more accessible, those who don’t tackle the issue tend to get even worse. Examples of greatly accessible websites are rare. Indeed, **“94.8% of home pages had detected WCAG 2 failures.”**



The systemic failure of software accessibility

based on a WebAIM Million 2025 report



Most common accessibility shortcomings detected on websites

Feeling the need for accessibility

Despite being a crucial need for at least 20% of all users, accessibility as it is implemented on common websites and applications are overwhelmingly insufficient.

With that said, accessibility isn’t about lowering those stats of detected failures, it’s about making sure that we are not contributing in isolating and marginalizing whole communities.

Accessibility & Ableism

What is accessibility?

Let’s define accessibility

Formal definitions of accessibility

From a product design standpoint;

Accessibility is the design of products, devices, services, vehicles, or environments so as to be usable by disabled people.

The concept of accessible design and practice of accessible developments ensures both "direct access" (i.e. unassisted) and "indirect access" meaning compatibility with a person's assistive technology (for example, computer screen readers).

*Wikipedia

Formal definitions of accessibility

From a legal standpoint

CA Canada

- The overarching goal of the ACA is to realize a barrier-free Canada by 2040. The legislation benefits all Canadians, specially persons with disabilities, through the proactive dentification, removal and prevention of barriers to accessibility in 7 priority areas:
  - employment
  - the built environment
  - information and communication technologies (ICT)
  - communication other than ICT
  - the design and delivery of programs and services
  - the procurement of goods, services and facilities
  - transportation

F France

Allowing the autonomy and participation of people with disabilities, by reducing or even removing the gap between their abilities, needs and will - and the physical, organizational and cultural components of their environment.

EU European Union

- The European Accessibility Act states that accessibility "should be achieved by the systematic removal and prevention of barriers, preferably through a universal design or 'design for all' approach, which contributes to ensuring access for persons with disabilities on an equal basis with others."
- Since June of 2025, almost all businesses (excluding microenterprises) are required to comply with the EAA requirements.

About an Accessible Canada
Notion d'accessibilité numérique - RGAA (France)
Loi accessibilité : cadre légal et obligations

Formal definitions of accessibility

From a software standpoint

Can all your users, **with the abilities and senses that they possess, <u>perceive</u> the information** your application presents to them?

Can your users, **with their specific input device or assistive technology, <u>operate</u> all the controls** within your application's user interface?

Can your users **<u>understand</u>** the information and the user interface controls?

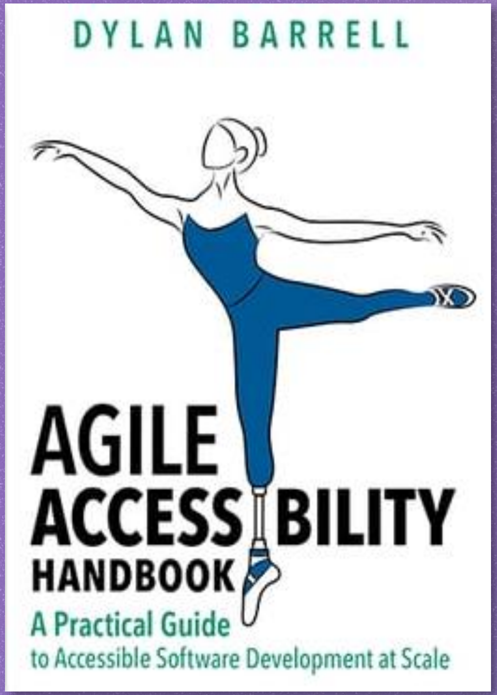
Barrell Dylan. *Agile Accessibility Handbook*

Formal definitions of accessibility

From a political standpoint

Striving to fight back against the structural tendency to exclusion of disabled people and attempting to provide them with a place (virtual or not) where they feel at ease and free to express themselves.

From a business standpoint


DYLAN BARRELL
AGILE ACCESSIBILITY HANDBOOK
A Practical Guide
to Accessible Software Development at Scale

1. **The business opportunity represented by households with a disability. The numbers that are of interest here are:**
   - 20% of people in the United States have a disability, and improvements in usability (or falling behind the competition) represent a major portion of the market;
   - $490 million is the after-tax disposable income of adults with a disability in the United States (as of 2019). This is comparable in size to the African American or Hispanic market segments; and
   - $10.3 billion is the e-commerce market size for accessibility.

2. The number of disability-related lawsuits increased by 181% in 2019 alone, and the cost of responding to a lawsuit (independent of the settlement costs) is $350,000. This does not take into account the cost of brand damage or future lawsuits; and

3. Accessibility is a human right, it is the right thing to do, and it probably aligns with the organization's values and its desire to improve digital user experiences.

Barrell, Dylan. *Agile Accessibility Handbook*. Amplify Publishing, 2020.

Understanding ableism

Ableism is "policies, behaviors, rules, etc. that result in unfair or harmful treatment of disabled people* and in a continued unfair advantage to people who are not disabled".

*people who have an illness, injury, or condition that makes it difficult for them to do things that most other people can do

Cambridge Dictionary. Definition of ableism

Understanding ableism

People who are qualified of “disabled” are simply working with a different set of abilities, which is typically more marginal, yet not inferior.

If they can’t access your app, it means that your design is broken - not them.

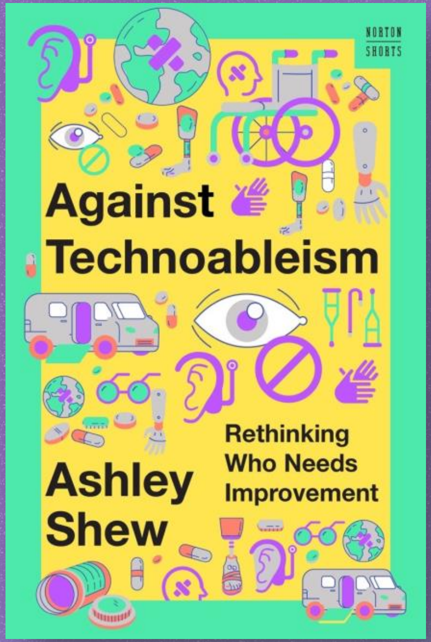
Cambridge Dictionary. Definition of ableism

Understanding ableism

Environment - and the way it is designed - plays an equal role as the body when it comes to disability.

What is techno-ableism?

According to Ashley Shew, in Against Technoableism, Rethinking Who Needs Improvement

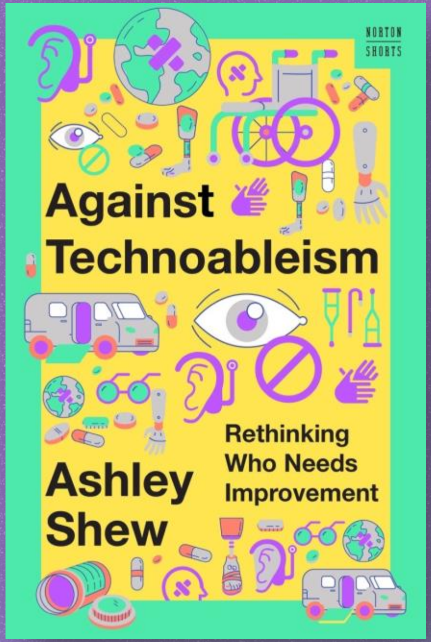


"Technoableism is a particular type of ableism, one that is highly visible in media and entertainment and omnipresent in the ways most people casually talk about technologies aimed at disability. **Technoableism is a belief in the power of technology that considers the elimination of disability a good thing, something we should strive for.** It's a classic form of ableism—bias against disabled people, bias in favor of nondisabled ways of life. **Technoableism is the use of technologies to reassert those biases, often under the guise of empowerment."**

Put simply, it's believing that disabled people need to be fixed by technology.

What hurts disabled people?

According to Ashley Shew in Against Technoableism, Rethinking Who Needs Improvement



"Sometimes technology is seen as redeeming our lives: non-disabled people believe—and expect us to believe—that technology will "solve" the problem of our disability and save us, or those like us, in the future. Yet these expectations often don't match our circumstances. They confine us. When people assume that one device will "fix" us, they don't pay attention to the host of other concerns around disability technology—the bad planning and design, the need for constant ongoing maintenance, the problem of money [...], and the staggering lack of social support for disability accommodations [...]. These are all forms of ableism."

The tech tendency to saviorism

According to Ashley Shew in Against Technoableism, Rethinking Who Needs Improvement

“THE TECH GIVING PEOPLE POWER TO DEAL WITH DISABILITY” —BBC NEWS

“HOW TECHNOLOGY WILL CHANGE THE LIVES OF PEOPLE WITH DISABILITIES” —FORBES

“FIRST PROSTHETIC LIMB DESIGNED FOR WOMEN: ‘I FEEL LIBERATED’” —BBC2

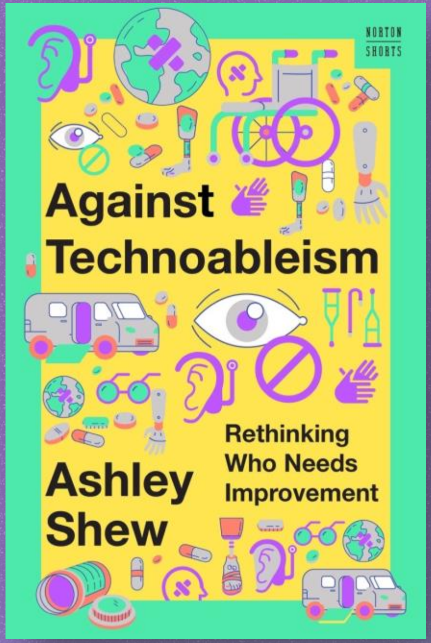
“ROBOTIC EXOSKELETONS ARE HERE, AND THEY’RE CHANGING LIVES” —POPSCI

What should be our goal?

Our goal is to try and provide a safe place rather than fix or save people.

How can we become safe spaces?

According to Ashley Shew in Against Technoableism, Rethinking Who Needs Improvement



"Awni mentioned Discord as a social platform “easily adapted by autistic users to facilitate autistic-styled communication due to its flexibility both with custom emotes and for purpose-centric server organization.” Gardiner pointed to the importance of letting autistic people lead tech conversations with what they want and how they want to do things. This doesn’t mean jumping in to teach autistic people to play games in the way allistic people do but rather letting the “nothing about us without us” lesson of the disability rights movement carry into this space: to learn from autistic people how to use spaces autistically."

Brief cartography of today's accessibility landscape

What is accessibility?

Countries & Institutions

Countries

Countries plays an increasing role in the accessibility landscape.

As awareness rises, the law evolves to guarantee access to technology as a fundamental human right.

Non-profit orgs

Non-profit organizations are at the core of producing resources about accessibility and inclusiveness, especially from a design and platform agnostic perspective.

Consulting firms

There are many consulting firms about accessibility and inclusiveness. The quality of their services vary greatly in quality.

Corporate actors

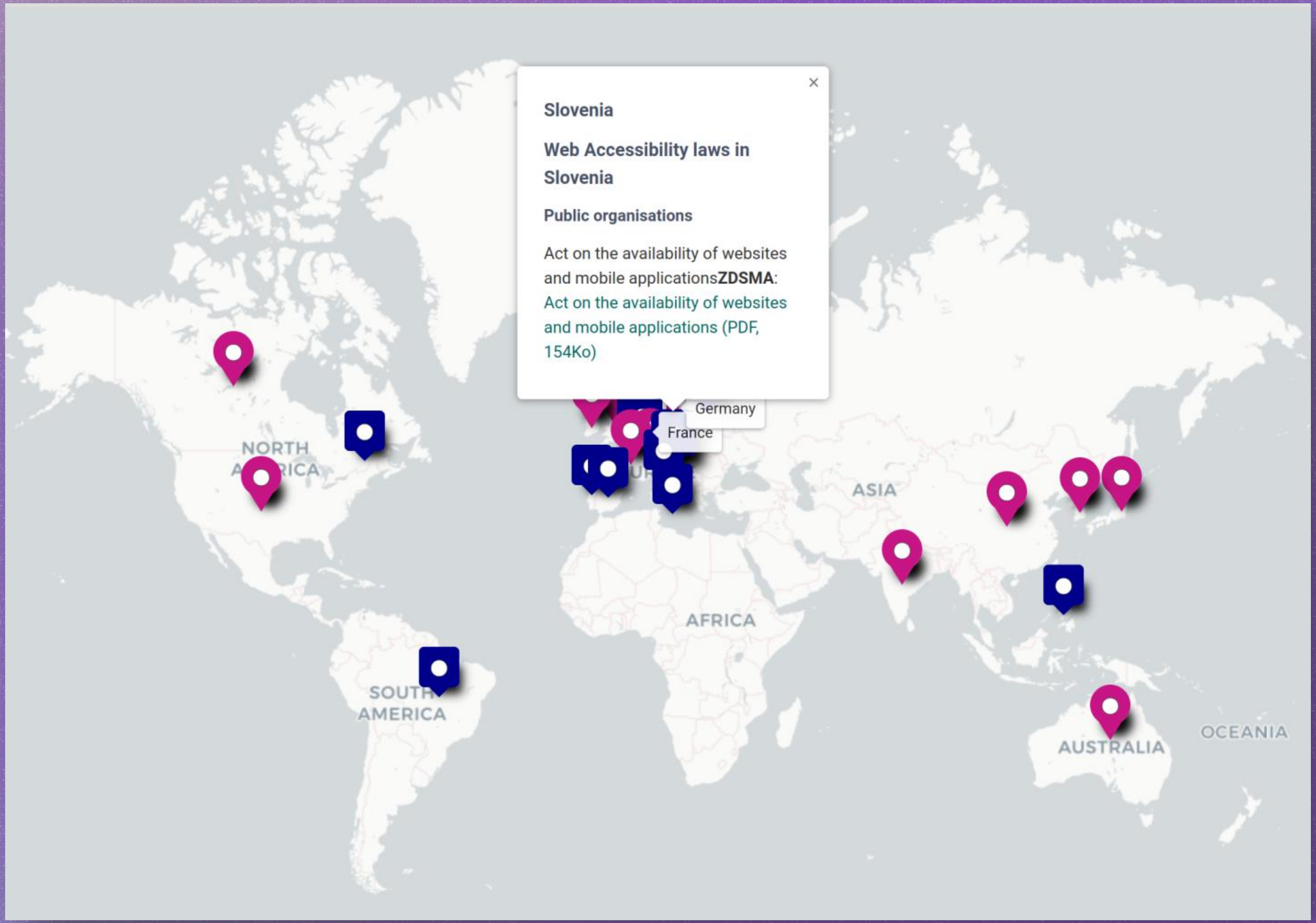
Corporate actors and especially operating system providers can take an important role in accessibility and inclusiveness measures, either by their good or their bad will.

Countries & Institutions

Countries

Countries plays an increasing role in the accessibility landscape.

As awareness rises, the law evolves to guarantee access to technology as a fundamental human right.



Slovenia

Web Accessibility laws in Slovenia

Public organisations

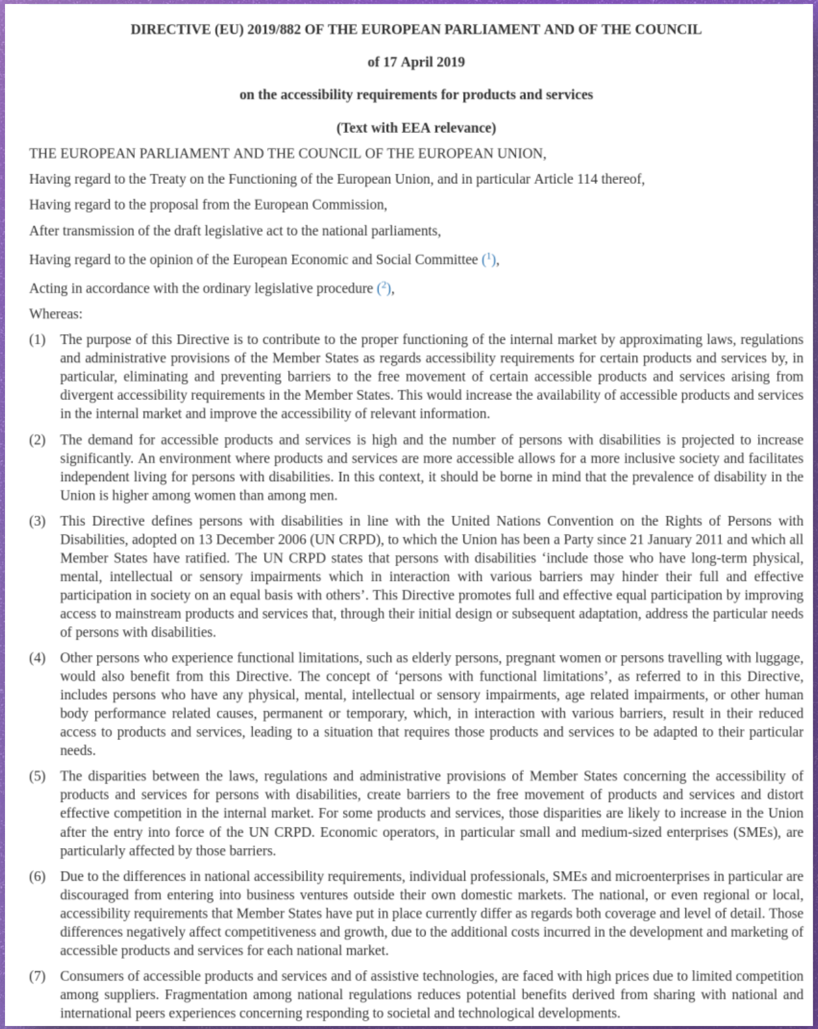
Act on the availability of websites and mobile applications**ZDSMA**: Act on the availability of websites and mobile applications (PDF, 154Ko)

this map holds current laws for many countries

Countries & Institutions

EU

The European Union adopted in 2019 the European Accessibility Act (EAA), which requires digital service providers to make their product accessible.



DIRECTIVE (EU) 2019/882 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 17 April 2019

on the accessibility requirements for products and services

(Text with EEA relevance)

THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION,

Having regard to the Treaty on the Functioning of the European Union, and in particular Article 114 thereof,

Having regard to the proposal from the European Commission,

After transmission of the draft legislative act to the national parliaments,

Having regard to the opinion of the European Economic and Social Committee ([1]),

Acting in accordance with the ordinary legislative procedure ([2]),

Whereas:

(1) The purpose of this Directive is to contribute to the proper functioning of the internal market by approximating laws, regulations and administrative provisions of the Member States as regards accessibility requirements for certain products and services by, in particular, eliminating and preventing barriers to the free movement of certain accessible products and services arising from divergent accessibility requirements in the Member States. This would increase the availability of accessible products and services in the internal market and improve the accessibility of relevant information.

(2) The demand for accessible products and services is high and the number of persons with disabilities is projected to increase significantly. An environment where products and services are more accessible allows for a more inclusive society and facilitates independent living for persons with disabilities. In this context, it should be borne in mind that the prevalence of disability in the Union is higher among women than among men.

(3) This Directive defines persons with disabilities in line with the United Nations Convention on the Rights of Persons with Disabilities, adopted on 13 December 2006 (UN CRPD), to which the Union has been a Party since 21 January 2011 and which all Member States have ratified. The UN CRPD states that persons with disabilities 'include those who have long-term physical, mental, intellectual or sensory impairments which in interaction with various barriers may hinder their full and effective participation in society on an equal basis with others'. This Directive promotes full and effective equal participation by improving access to mainstream products and services that, through their initial design or subsequent adaptation, address the particular needs of persons with disabilities.

(4) Other persons who experience functional limitations, such as elderly persons, pregnant women or persons travelling with luggage, would also benefit from this Directive. The concept of 'persons with functional limitations', as referred to in this Directive, includes persons who have any physical, mental, intellectual or sensory impairments, age related impairments, or other human body performance related causes, permanent or temporary, which, in interaction with various barriers, result in their reduced access to products and services, leading to a situation that requires those products and services to be adapted to their particular needs.

(5) The disparities between the laws, regulations and administrative provisions of Member States concerning the accessibility of products and services for persons with disabilities, create barriers to the free movement of products and services and distort effective competition in the internal market. For some products and services, those disparities are likely to increase in the Union after the entry into force of the UN CRPD. Economic operators, in particular small and medium-sized enterprises (SMEs), are particularly affected by those barriers.

(6) Due to the differences in national accessibility requirements, individual professionals, SMEs and microenterprises in particular are discouraged from entering into business ventures outside their own domestic markets. The national, or even regional or local, accessibility requirements that Member States have put in place currently differ as regards both coverage and level of detail. Those differences negatively affect competitiveness and growth, due to the additional costs incurred in the development and marketing of accessible products and services for each national market.

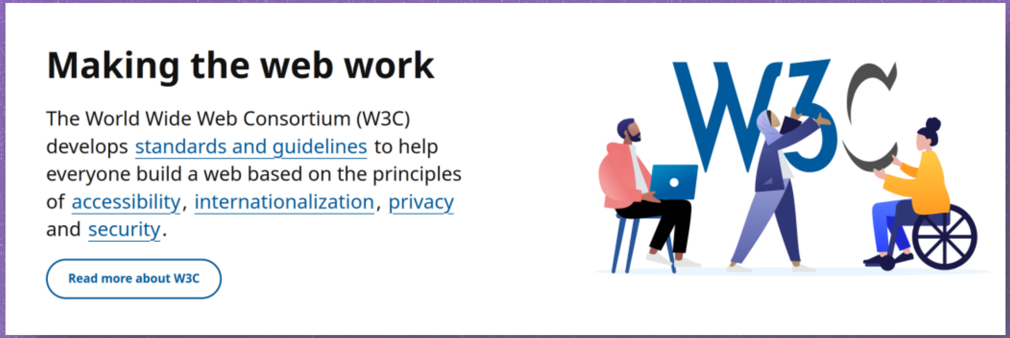
(7) Consumers of accessible products and services and of assistive technologies, are faced with high prices due to limited competition among suppliers. Fragmentation among national regulations reduces potential benefits derived from sharing with national and international peers experiences concerning responding to societal and technological developments.

Non-profit collectives

The World Wide Web Consortium (W3C)

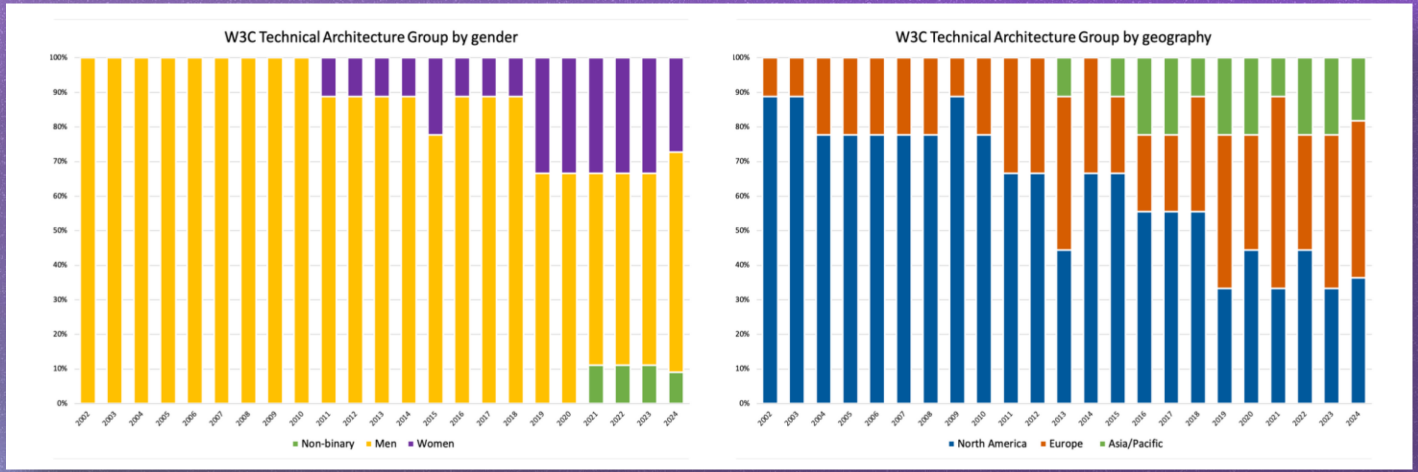
The World Wide Web Consortium is the organization behind multiple accessibility guidelines and standards recognized worldwide. With over 30 years of existence, it's mostly known for its work on the WCAG which are the worldwide standards in term of web accessibility.

They also created and updated a very extensive documentation on how to meet those standards with their Quick Reference on How to Meet WCAG. They are definitely one of the most important organizations providing resources on accessibility.



Making the web work

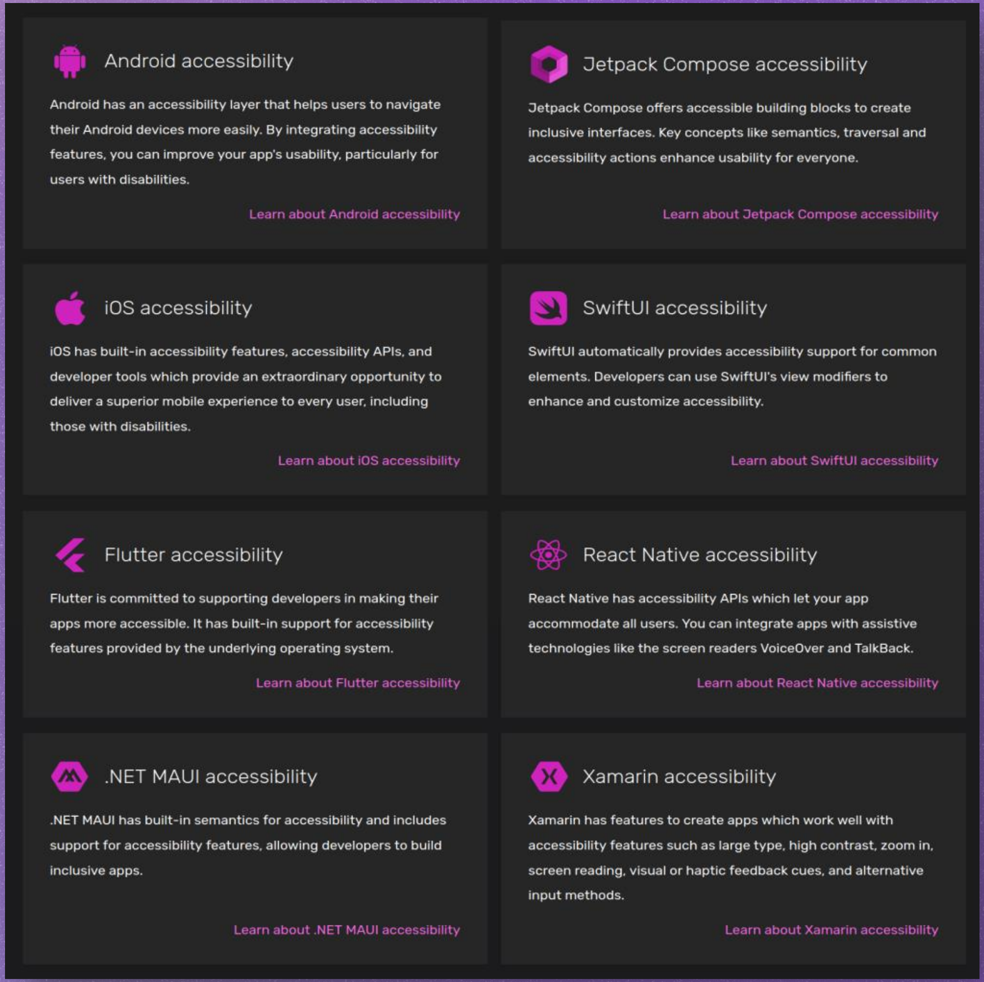
The World Wide Web Consortium (W3C) develops standards and guidelines to help everyone build a web based on the principles of accessibility, internationalization, privacy and security.



Non-profit collectives

The Appt Foundation

- Non-profit from the Netherlands
- Extensive technical documentation on mobile accessibility
- Great code samples



Android accessibility

Android has an accessibility layer that helps users to navigate their Android devices more easily. By integrating accessibility features, you can improve your app's usability, particularly for users with disabilities.

Learn about Android accessibility

Jetpack Compose accessibility

Jetpack Compose offers accessible building blocks to create inclusive interfaces. Key concepts like semantics, traversal and accessibility actions enhance usability for everyone.

Learn about Jetpack Compose accessibility

iOS accessibility

iOS has built-in accessibility features, accessibility APIs, and developer tools which provide an extraordinary opportunity to deliver a superior mobile experience to every user, including those with disabilities.

Learn about iOS accessibility

SwiftUI accessibility

SwiftUI automatically provides accessibility support for common elements. Developers can use SwiftUI's view modifiers to enhance and customize accessibility.

Learn about SwiftUI accessibility

Flutter accessibility

Flutter is committed to supporting developers in making their apps more accessible. It has built-in support for accessibility features provided by the underlying operating system.

Learn about Flutter accessibility

React Native accessibility

React Native has accessibility APIs which let your app accommodate all users. You can integrate apps with assistive technologies like the screen readers VoiceOver and TalkBack.

Learn about React Native accessibility

.NET MAUI accessibility

.NET MAUI has built-in semantics for accessibility and includes support for accessibility features, allowing developers to build inclusive apps.

Learn about .NET MAUI accessibility

Xamarin accessibility

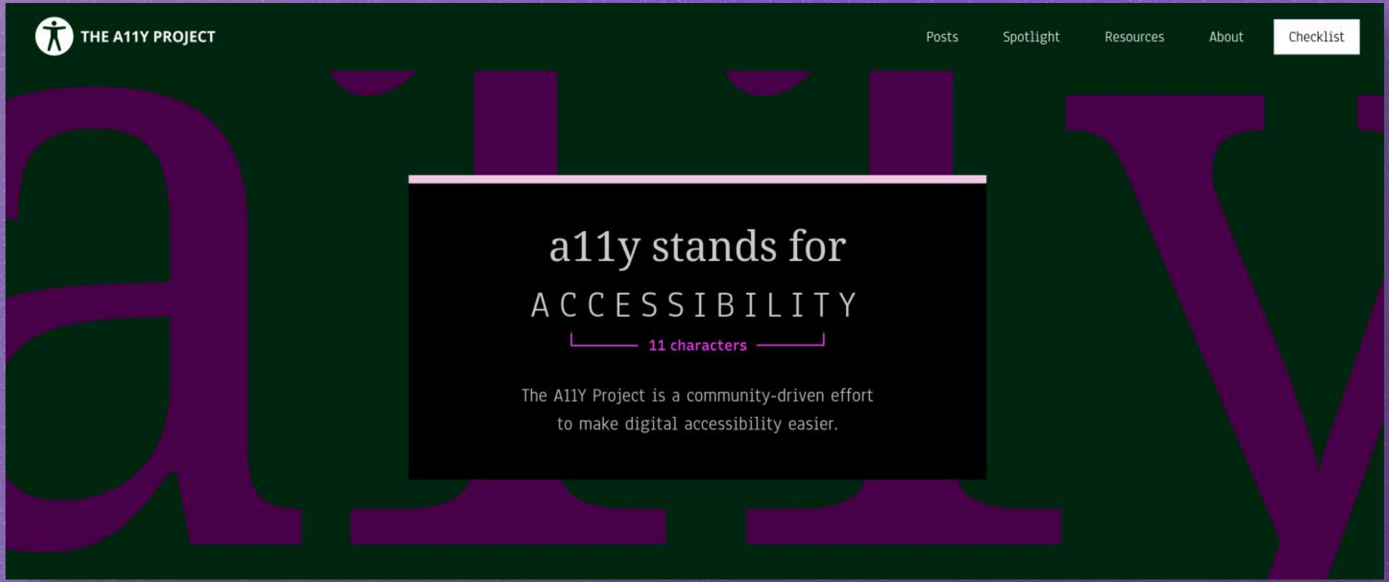
Xamarin has features to create apps which work well with accessibility features such as large type, high contrast, zoom in, screen reading, visual or haptic feedback cues, and alternative input methods.

Learn about Xamarin accessibility

Non-profit collectives

The A11Y Project

- **Great resource**
- Mainly (but not only) web accessibility
- Very extensive resources
- “Spotlight” section, dedicated to people working in the field
- List of medias with <u>dozen of carefully selected resources</u>.



a11y stands for ACCESSIBILITY

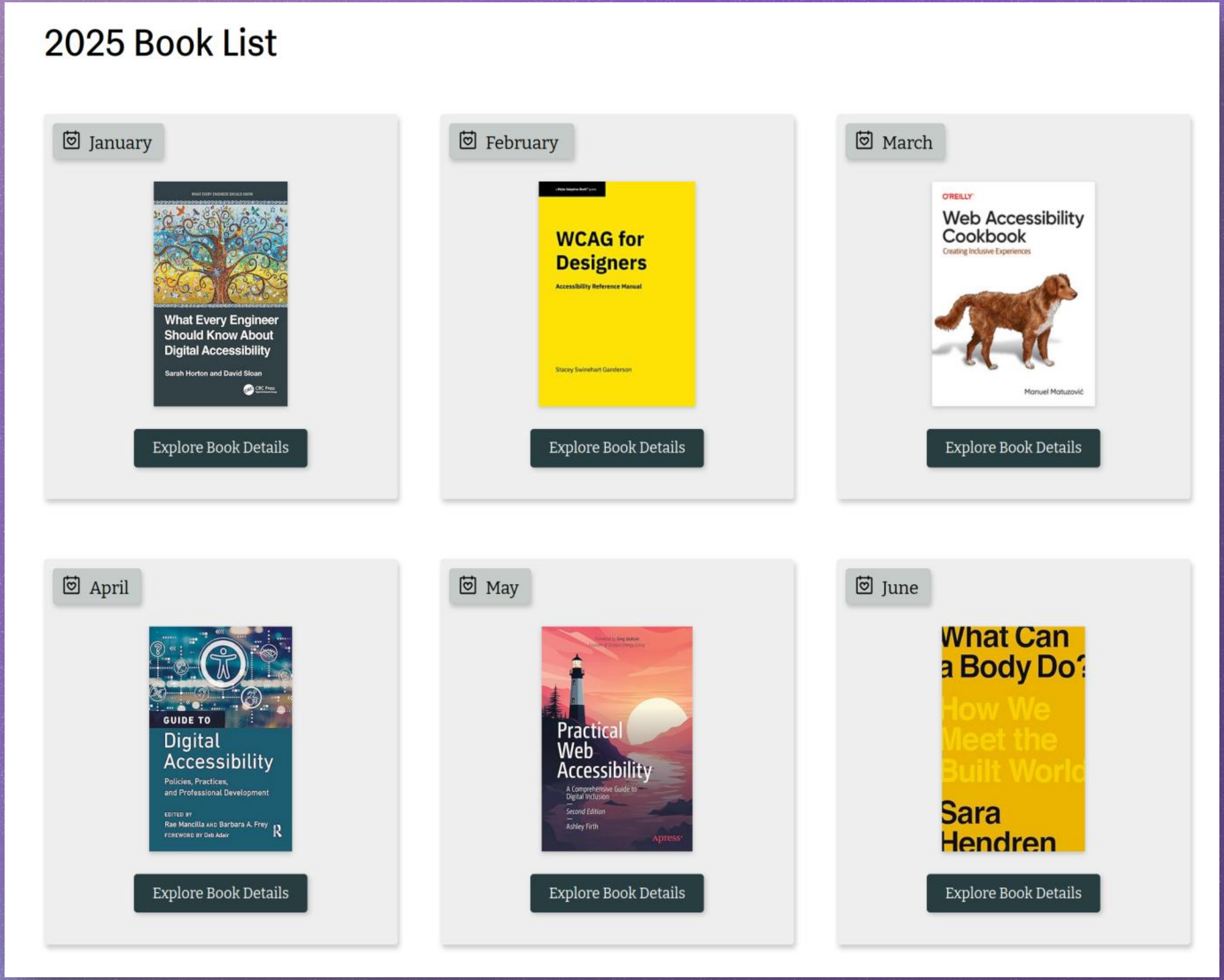
The A11Y Project is a community-driven effort to make digital accessibility easier.

Non-profit collectives

The Accessibility Book Club

The Accessibility Book club is a small organization releasing a yearly book list about accessibility.

Great for staying up to date!



2025 Book List

January
What Every Engineer Should Know About Digital Accessibility
Sarah Horton and David Sloan
Explore Book Details

February
WCAG for Designers
Accessibility Reference Manual
Stacey Swinehart Ganderson
Explore Book Details

March
O'REILLY
Web Accessibility Cookbook
Creating Inclusive Experiences
Manuel Matuzović
Explore Book Details

April
Guide to Digital Accessibility
Policies, Practices, and Professional Development
Edited by Rae Mancilla and Barbara A. Frey
Foreword by Deb Adair
Explore Book Details

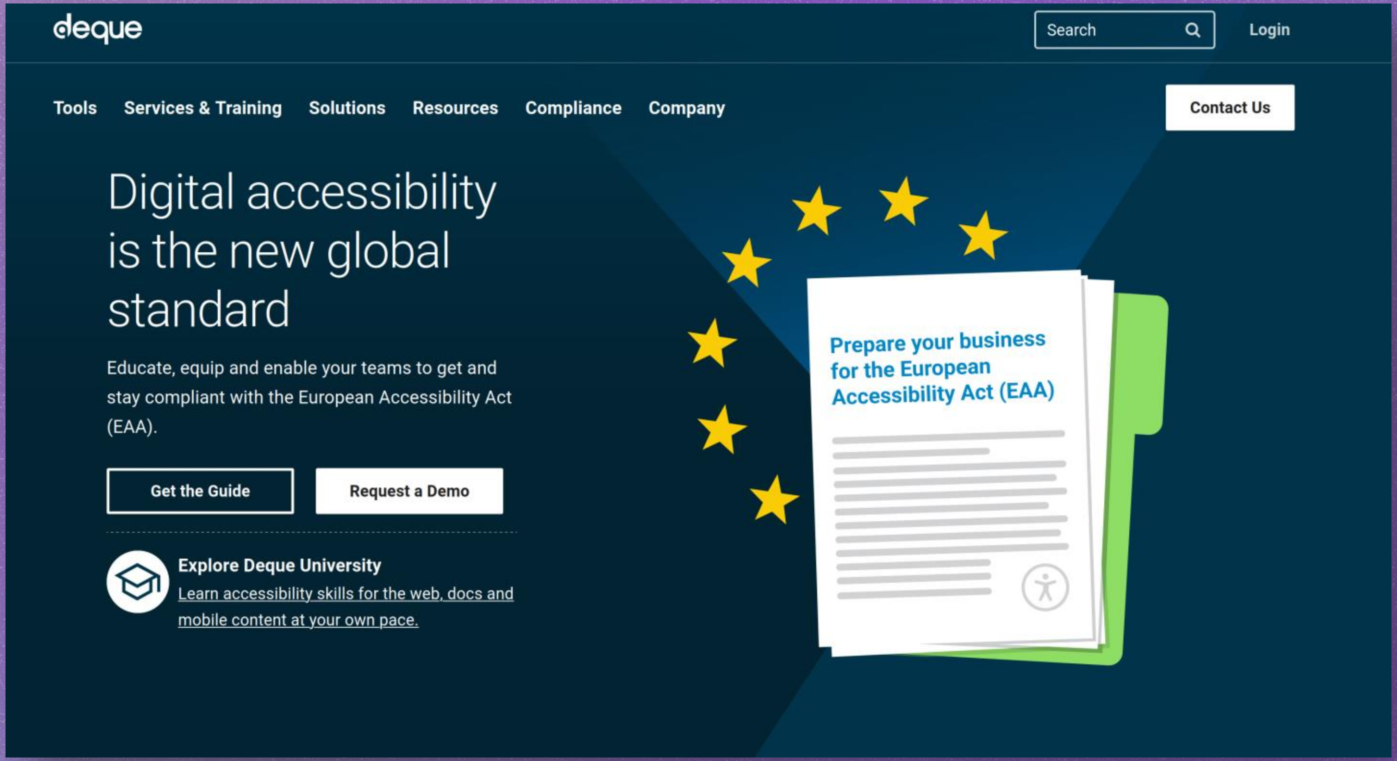
May
Practical Web Accessibility
A Comprehensive Guide to Digital Inclusion
Second Edition
Ashley Firth
Apress
Explore Book Details

June
What Can a Body Do? How We Meet the Built World
Sara Hendren
Explore Book Details

Consulting firms

Deque

- Reference in the field
- Axe, one of the most trusted web accessibility tool with over two billion downloads.
- Audits, trainings, books…
- Behind the accessibility of some of the biggest companies worldwide.



Digital accessibility is the new global standard

Educate, equip and enable your teams to get and stay compliant with the European Accessibility Act (EAA).

Explore Deque University

Learn accessibility skills for the web, docs and mobile content at your own pace.

Corporate actors

Apple

For years, Apple has been the reference for accessibility. Their tools and resources are among the best out there. Unfortunately, these tools and resources are directly targeted for their platforms.

Microsoft

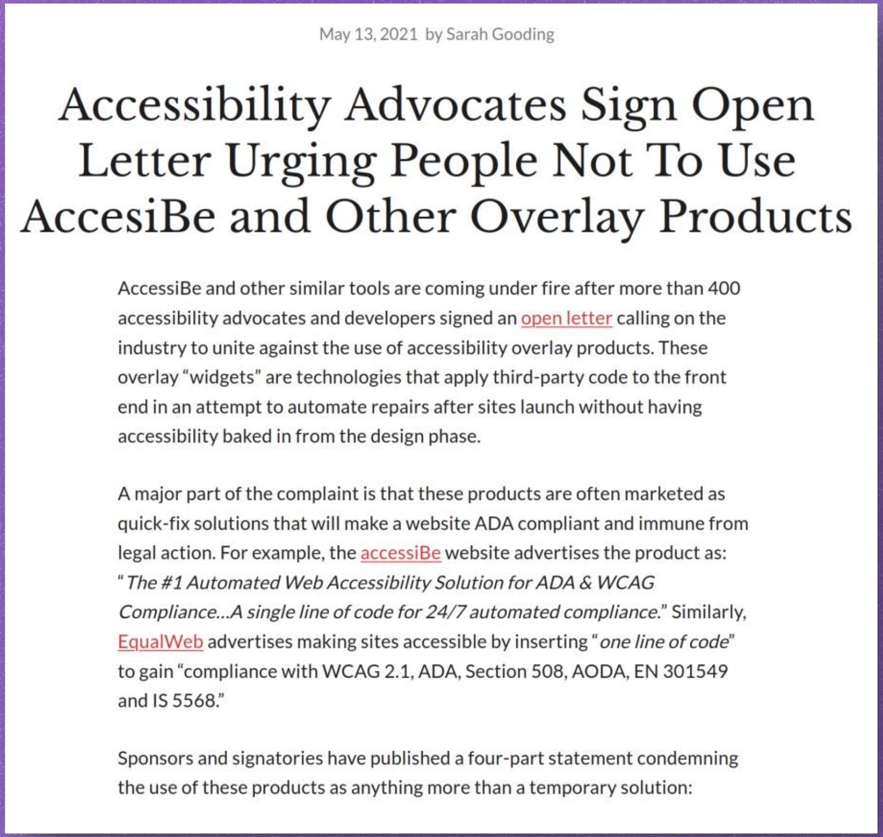
While we do not endorse all of their content, Microsoft has been publishing great extensive design resources about inclusiveness on their dedicated website.

What to look for and what to avoid?

Accessibility and AI-powered tools

400 accessibility advocates state:

- We will **never advocate, recommend**, or integrate an overlay which deceptively markets itself as **providing automated compliance with laws or standards.**
- We will always advocate for **the remediation of accessibility issues at the source of the original error.**
- We will refuse to stay silent when overlay vendors use deception to market their products.
- More specifically, we hereby advocate for the removal of accessiBe, AudioEye, UserWay, User1st, MK-Sense, MaxAccess, FACIL'iti, and all similar products and **encourage the site owners who've implemented these products to use more robust, independent, and permanent strategies to making their sites more accessible.**


May 13, 2021 by Sarah Gooding

Accessibility Advocates Sign Open Letter Urging People Not To Use AccesiBe and Other Overlay Products

AccessiBe and other similar tools are coming under fire after more than 400 accessibility advocates and developers signed an open letter calling on the industry to unite against the use of accessibility overlay products. These overlay "widgets" are technologies that apply third-party code to the front end in an attempt to automate repairs after sites launch without having accessibility baked in from the design phase.

A major part of the complaint is that these products are often marketed as quick-fix solutions that will make a website ADA compliant and immune from legal action. For example, the accessiBe website advertises the product as: "*The #1 Automated Web Accessibility Solution for ADA & WCAG Compliance…A single line of code for 24/7 automated compliance.*" Similarly, EqualWeb advertises making sites accessible by inserting "*one line of code*" to gain "compliance with WCAG 2.1, ADA, Section 508, AODA, EN 301549 and IS 5568."

Sponsors and signatories have published a four-part statement condemning the use of these products as anything more than a temporary solution:

Sarah Gooding, *Accessibility Advocates Sign Open Letter Urging People Not To Use AccesiBe and Other Overlay Products*. 2021

What to look for and what to avoid?

Green and red flags for choosing accessibility tools and experts

Look for...

- Structures that have disabled experts or value directly working with disabled people
- Structures that talk about inclusiveness rather than accessibility alone
- Structures that provide transparency about their own diversity distribution

Avoid...

- Automated accessibility overlay products
- Products that claims to have automated AI accessibility fixes
- Promises of quick fixes
- Structures that delivers a one-time accessibility enhancement without maintenance or knowledge transmission to your team

Assistive technologies: Desktop

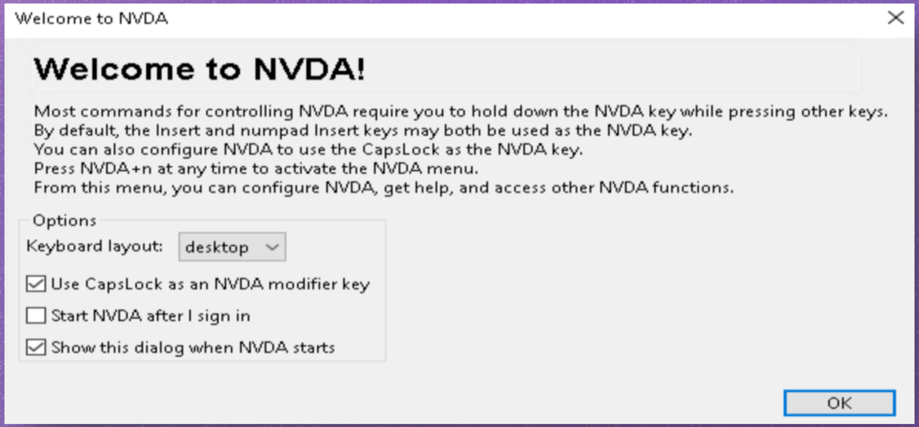
What is accessibility?

Using a screen reader and a keyboard

- **Screen readers** are the best way to quickly find and address accessibility issues affecting all assistive technologies.
- **Always check** if your modifications on the UI are **fully accessible only with your keyboard** — not a challenge for our beloved vim users.
- Use system default and, if possible, third-party screen readers. Consider implementing plug-ins.

Screen readers

- NVDA is open source and available on Windows.
- It's free and therefore is a very good tool to start testing accessibility.
- Very lightweight and stable.
- Less feature-full in complex apps like Excel.
- Many community plug-ins: consider building your own!



Welcome to NVDA

Welcome to NVDA!

Most commands for controlling NVDA require you to hold down the NVDA key while pressing other keys.
By default, the Insert and numpad Insert keys may both be used as the NVDA key.
You can also configure NVDA to use the CapsLock as the NVDA key.
Press NVDA+n at any time to activate the NVDA menu.
From this menu, you can configure NVDA, get help, and access other NVDA functions.

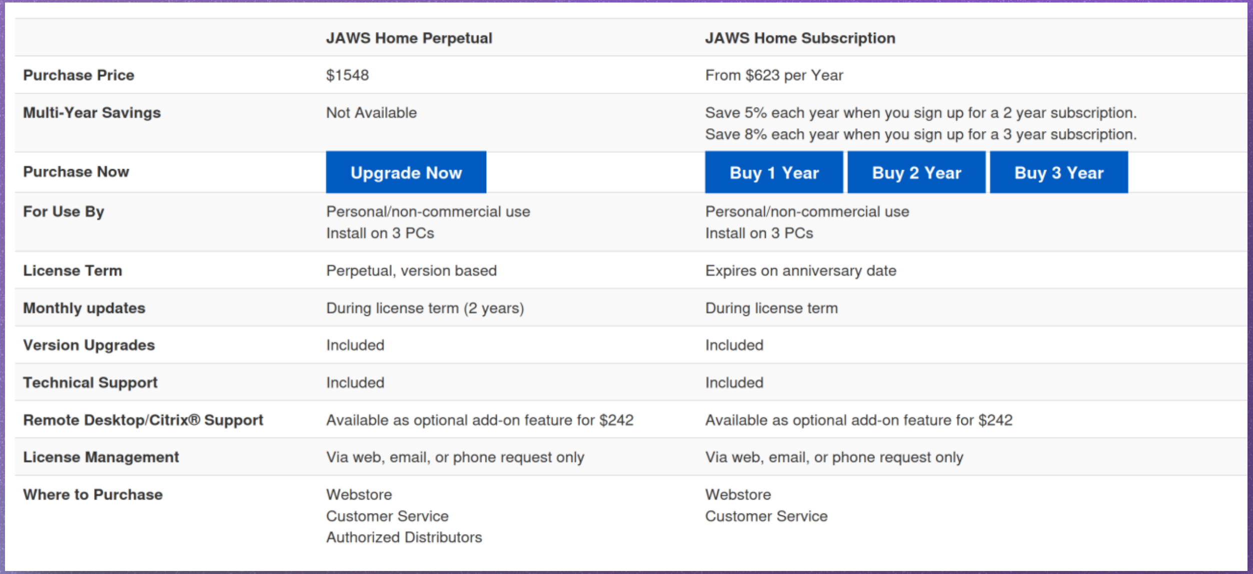
Options
Keyboard layout: desktop
[✓] Use CapsLock as an NVDA modifier key
[ ] Start NVDA after I sign in
[✓] Show this dialog when NVDA starts

Platform	Pros	Cons
Windows Theoretically Linux (via Wine)	• Free and open source • Very lightweight • Doesn't crash much	• Plugins for major productivity apps are often not as robust as JAWS

Screen readers

- Jaws is is the high-end professional screen reader for Windows.
- Allows for a lot of customization, has a lot of plugins especially for higher education.
- Can be overwhelming to learn
- Less stable than NVDA
- Proprietary and very expensive!

Platform	Pros	Cons
Windows	• Industry standard • Has a lot of plugins	• Crashes a lot • Requires a lot of RAM • Very expensive • Hard to learn • Can be confusing



	JAWS Home Perpetual	JAWS Home Subscription
Purchase Price	$1548	From $623 per Year
Multi-Year Savings	Not Available	Save 5% each year when you sign up for a 2 year subscription. Save 8% each year when you sign up for a 3 year subscription.
Purchase Now	Upgrade Now	Buy 1 Year Buy 2 Year Buy 3 Year
For Use By	Personal/non-commercial use Install on 3 PCs	Personal/non-commercial use Install on 3 PCs
License Term	Perpetual, version based	Expires on anniversary date
Monthly updates	During license term (2 years)	During license term
Version Upgrades	Included	Included
Technical Support	Included	Included
Remote Desktop/Citrix® Support	Available as optional add-on feature for $242	Available as optional add-on feature for $242
License Management	Via web, email, or phone request only	Via web, email, or phone request only
Where to Purchase	Webstore Customer Service Authorized Distributors	Webstore Customer Service

Screen readers

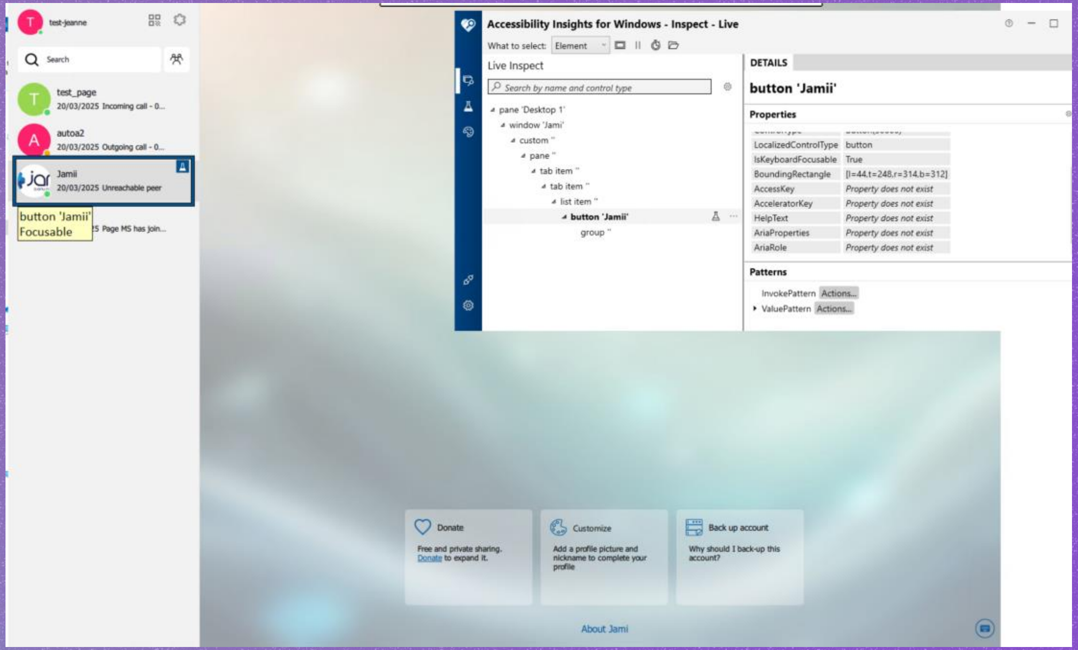
- VoiceOver is the native screen reader for Apple products.
- Known for being very easy to learn making for a fluid and intuitive experience.
- Only available on Apple machines, and thus to macOS/iOS users.

Platform	Pros	Cons
macOS	• Very easy to learn • Very fluid and intuitive	• Only available on macOS

Accessibility testing tools

- Accessibility Insights for Windows is a tool developed by Microsoft.
- Aims at visualizing the object hierarchy of an application.
- Only available on Apple machines, and thus to macOS/iOS users.

Platform	Pros	Cons
Windows	• One of the only tools available to check object tree and labels on desktop apps	• Only works on Windows • Doesn't provide much information about how to fix the issue encountered



- pane 'Desktop 1'
  - window 'Jami'
    - custom ''
      - pane ''
        - tab item ''
          - tab item ''
            - list item ''
              - button 'Jamii'
                - group ''

button 'Jamii'

LocalizedControlType	button
IsKeyboardFocusable	True
BoundingRectangle	[l=44,t=248,r=314,b=312]
AccessKey	Property does not exist
AcceleratorKey	Property does not exist
HelpText	Property does not exist
AriaProperties	Property does not exist
AriaRole	Property does not exist

InvokePattern Actions...
ValuePattern Actions...

Free and private sharing. Donate to expand it.
Add a profile picture and nickname to complete your profile
Why should I back-up this account?

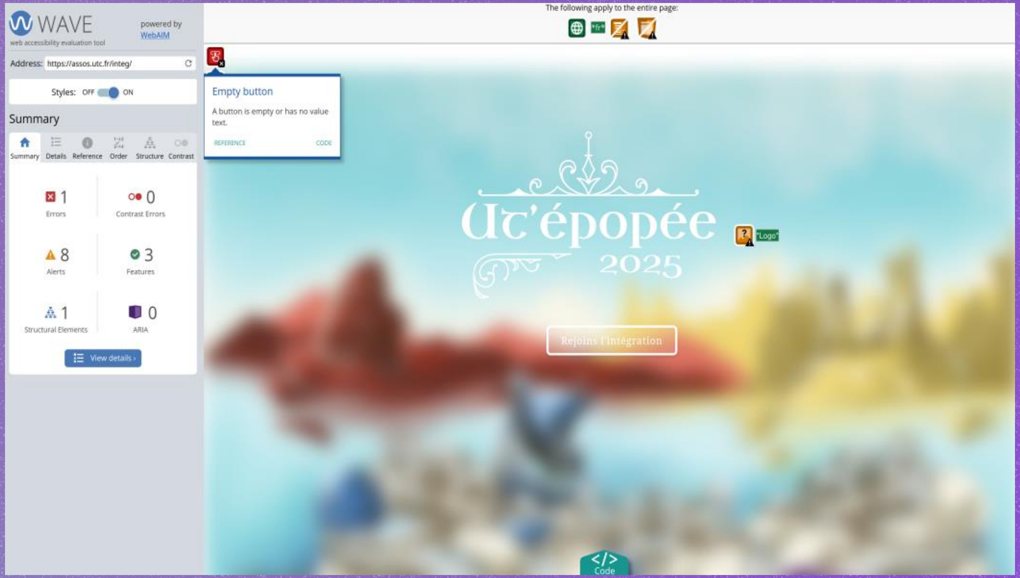
Accessibility testing tools

WAVE

web accessibility evaluation tool

- Wave is a testing software to automatically detect warnings and errors related to the WCAG Accessibility guidelines.
- As simple as entering your domain to see prevalent issues with your website
- Be careful not to believe blindly the feedback form automated tools lke Wave.

Platform	Pros	Cons
Web	• Very easy to use • Completely free	• Can be limited • Often misleading


WAVE web accessibility evaluation tool powered by WebAIM

Address: https://assos.utc.fr/integ/

Styles: OFF ON

Summary

1 Errors	0 Contrast Errors
8 Alerts	3 Features
1 Structural Elements	0 ARIA

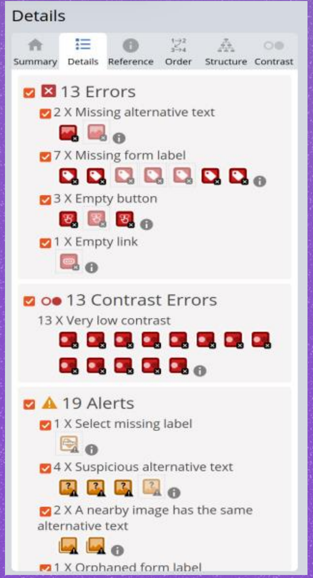
View details ›

The following apply to the entire page:

Empty button

A button is empty or has no value text.

REFERENCE CODE


Details

13 Errors
- 2 X Missing alternative text
- 7 X Missing form label
- 3 X Empty button
- 1 X Empty link

13 Contrast Errors
- 13 X Very low contrast

19 Alerts
- 1 X Select missing label
- 4 X Suspicious alternative text
- 2 X A nearby image has the same alternative text
- 1 X Orphaned form label

Accessibility testing tools

- **User groups are one of the most reliable ways to build an accessible and inclusive application.**
- You should try and make those user groups as diverse as possible.
- There should be a way of keeping track of the different problems raised by the users and the progress made on fixing them.
- The product and design team should have occasional sessions and calls with some users to discuss precisely how they would want to implement something

Assistive technologies: Mobile

What is accessibility?

Mobile screen readers

- Typically directly built and deeply integrated into the OS, whereas desktop screen readers are typically third-party apps.

- Navigate using specific gestures, whereas desktop screen readers commonly rely on keyboard navigation.

How to use?

- Quickly scan the interface by running your finger through the screen as if you were reading braille.
- Swipe left of right to navigate to the previous or next element in the interface.
- Double-tap to activate an element (e.g. a push button)

Alternative inputs

Switch control

- allows users to interact with app without needing the ability to precisely aim for the touchscreen elements.

Eye and face control

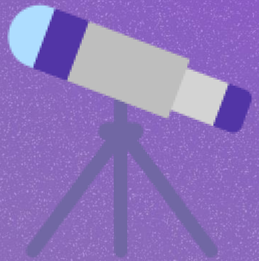
- Users can look at an element for a certain amount of time or perform a sound, move their head or even tap a switch while looking at an element to take action on it.

Voice control

- Users can also use their voice to choose which elements to select.
- Labeling your app properly allows them to call elements by their name rather than by numbers.

Accessibility settings

- There are many ways users can fine-tune their experience through accessibility settings. Your role as a software builder is to test your interface with many combinations of settings to make sure it works well.
- Check out this great resource for general information on platform-specific mobile assistive technologies.



Inclusiveness needs overview

From accessibility to inclusiveness

Main disabilities to account for in software

Disability is **not a binary state** but rather refers to a **wide variety of living experiences**, and that a lot of users experience **multiple disabilities at once.**

Consequently, **this list should not be seen as exhaustive.**

Main disabilities to account for in software

Motor disabilities

- affects up to 19% of users
- alternative input devices like switches or audio controlling

Hearing impairments

- affects 18% of users
- Provide redundant access to all auditive information

Motion sickness

Known triggers include:

- Zooms
- Flashing or blinking
- Animations playing automatically without user interaction
- Parallax effects

Speech or language

- dyslexia, illiteracy or elocution difficulties
- some of the solutions will include speech-to-text features, support for custom typefaces like Open Dyslexic and spell-checkers

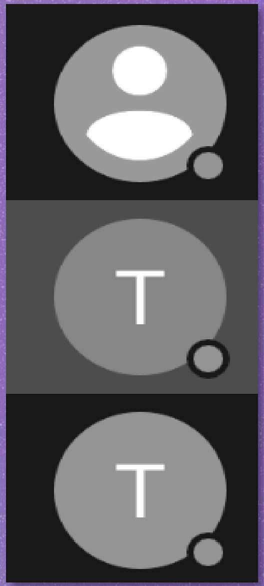
Low vision and blindness

- affects up to 17% of users
- Adaptive font sizes
- Support for assistive technologies like screen-readers or braille displays

Microsoft, *Accessibility (Design basics)*

Main disabilities to account for in software

Can you guess what is the accessibility issue in this screenshot?



Main disabilities to account for in software

Motor disabilities

- affects up to 19% of users
- alternative input devices like switches or audio controlling

Hearing impairments

- affects 18% of users
- Provide redundant access to all auditive information

Motion sickness

Known triggers include:

- Zooms
- Flashing or blinking
- Animations playing automatically without user interaction
- Parallax effects

Speech or language

- dyslexia, illiteracy or elocution difficulties
- some of the solutions will include speech-to-text features, support for custom typefaces like Open Dyslexic and spell-checkers

Low vision and blindness

- affects up to 17% of users
- Adaptive font sizes
- Support for assistive technologies like screen-readers or braille displays

Color blindness

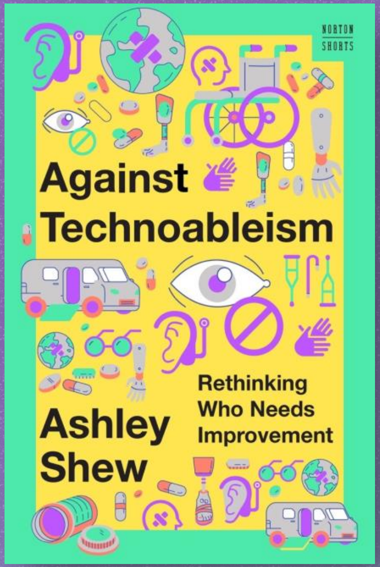
- Make sure your UI doesn't rely on color elements alone to convey information

Microsoft, *Accessibility (Design basics)*

Main disabilities to account for in software

Cognitive impairments

- Affects 16% of users
- Allow for customizability (UI simplification, product flexibility)



"Awni mentioned Discord as a social platform “easily adapted by autistic users to facilitate autistic-styled communication due to its flexibility both with custom emotes and for purpose-centric server organization.” Gardiner pointed to the importance of letting autistic people lead tech conversations with what they want and how they want to do things. This doesn’t mean jumping in to teach autistic people to play games in the way allistic people do but rather letting the “nothing about us without us” lesson of the disability rights movement carry into this space: to learn from autistic people how to use spaces autistically."

Additional needs and factors to take into consideration

***Non-exhaustive* list of human characteristics to keep in mind to try and include as many users as possible:**

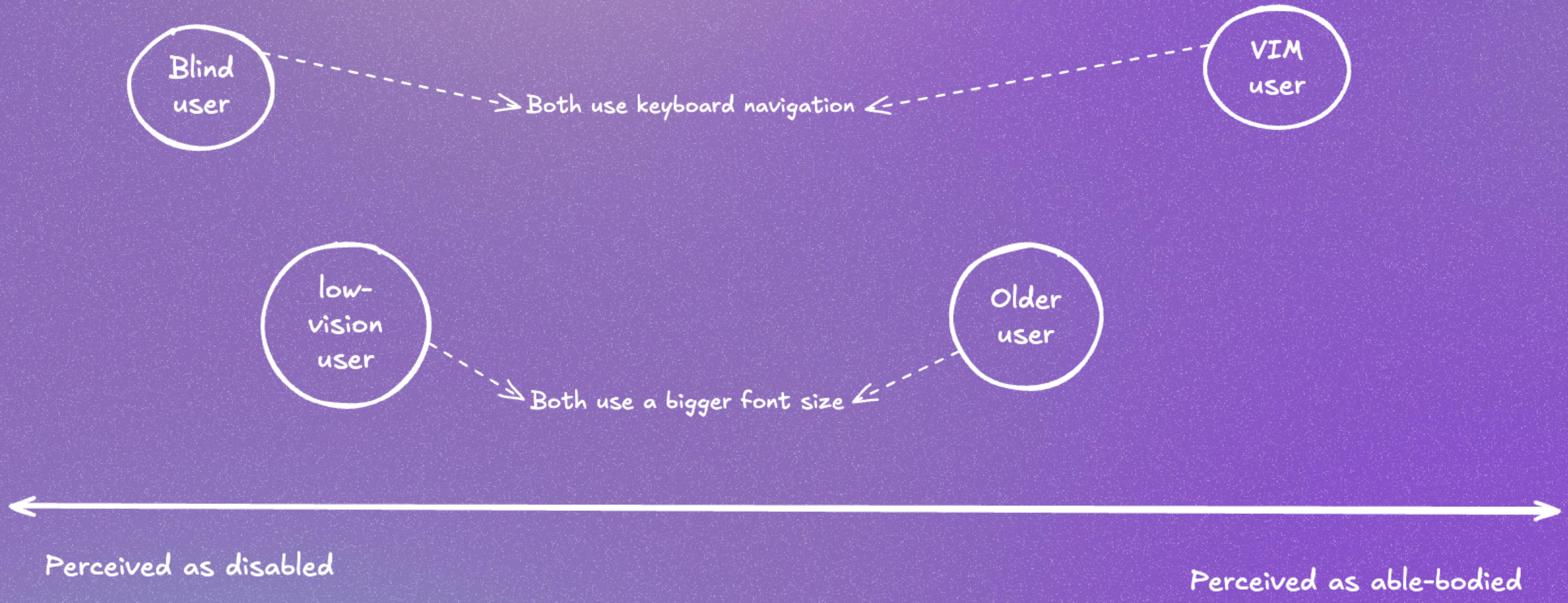
Age
Culture
Digital literacy
Education
Ethnicity
Gender, gender identity
Historical context
Language
Mental health
Nationality
Physical *and* cognitive (dis)abilities
Sexuality
Socio-economic context
…

You need to consider the intersections of all of those categories as well as the disabilities listed previously in the first section of this module.

We are all disabled and unique: accessibility, plurality and intersectionality

From accessibility to inclusiveness

Disability is a not a binary state



Therefore, it's crucial to not think disabled peoples as a minority but rather as the living diversity in which we all evolve.

What is inclusiveness

"the quality of including many different types of people and treating them all fairly and equally"

In simpler words, not being ableist is not enough. We must make sure that we are also not perpetuating other kinds of oppressions.

There is no inclusiveness without intersectionality.

Cambridge's dictionary's definition of inclusiveness

What is intersectionality?

Crenshaw defined intersectionality as a framework for understanding **specific living experiences of people situated at the intersection of discriminations and privileges.**

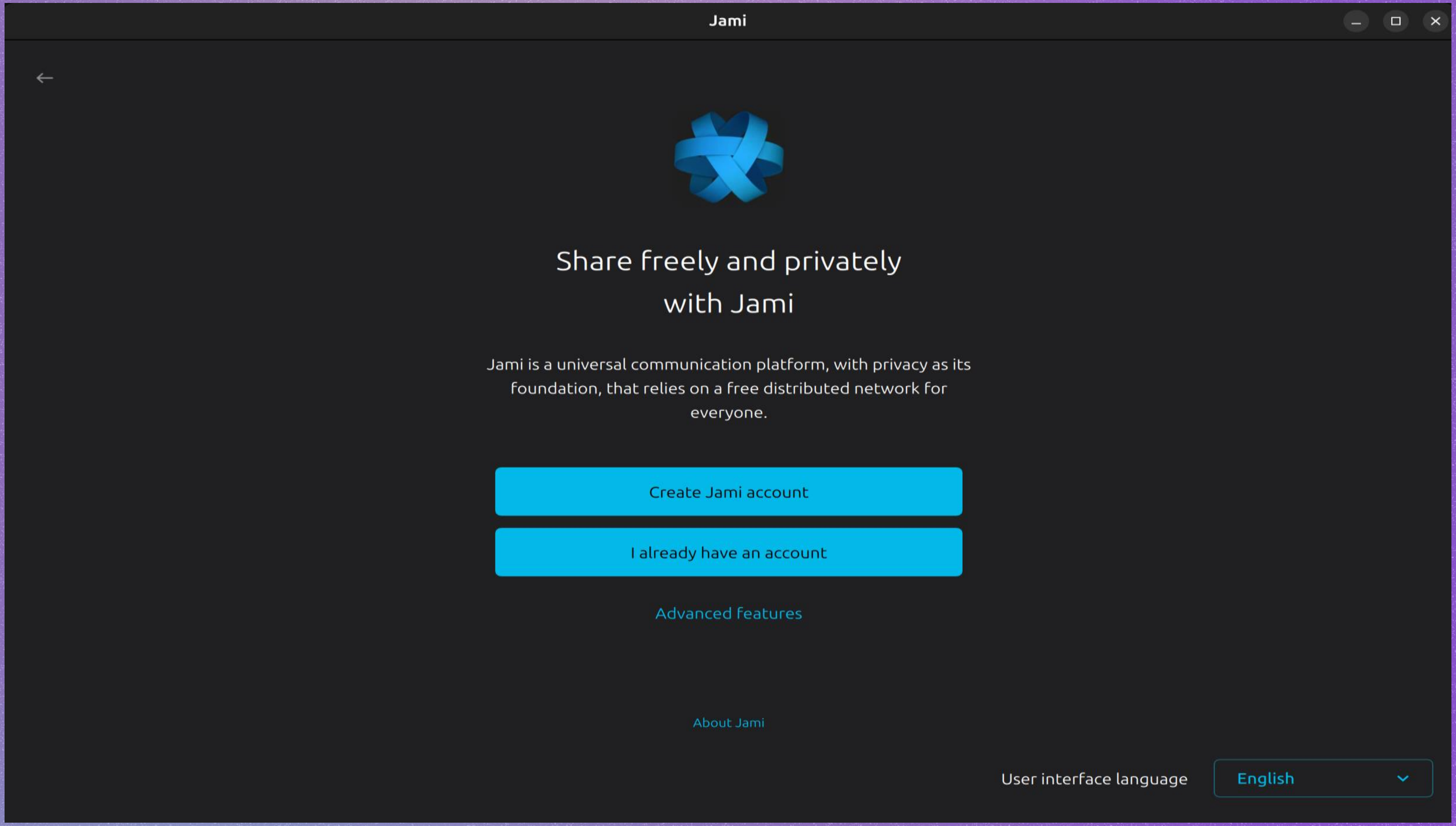
Crenshaw, Kimberle. "Mapping the Margins: Intersectionality, Identity Politics, and Violence against Women of Color." Stanford Law Review, vol. 43, no. 6, 1991, pp. 1241–99.

Intersectionality in software

- 17.7% of users with disabilities live with more than one condition (as of 2016).
- Designing for one disability leaves out a blank spot for users having that disability but also other conditions.
- It’s a common bias to forget that disabled people are not just disabled, but first and foremost people with their own unique life experiences and specificities.

We need to think about women who rely on wheelchairs, LGBTQIA+ people who don't speak the language of they country they live in, blind users who can't afford an iPhone - and those who are all of that at once!

Intersectionality in software


Jami

Share freely and privately with Jami

Jami is a universal communication platform, with privacy as its foundation, that relies on a free distributed network for everyone.

User interface language: English

Apple's automatic picture subtitling is another example !

Developer stance & user collaboration

From accessibility to inclusiveness

Understanding bias

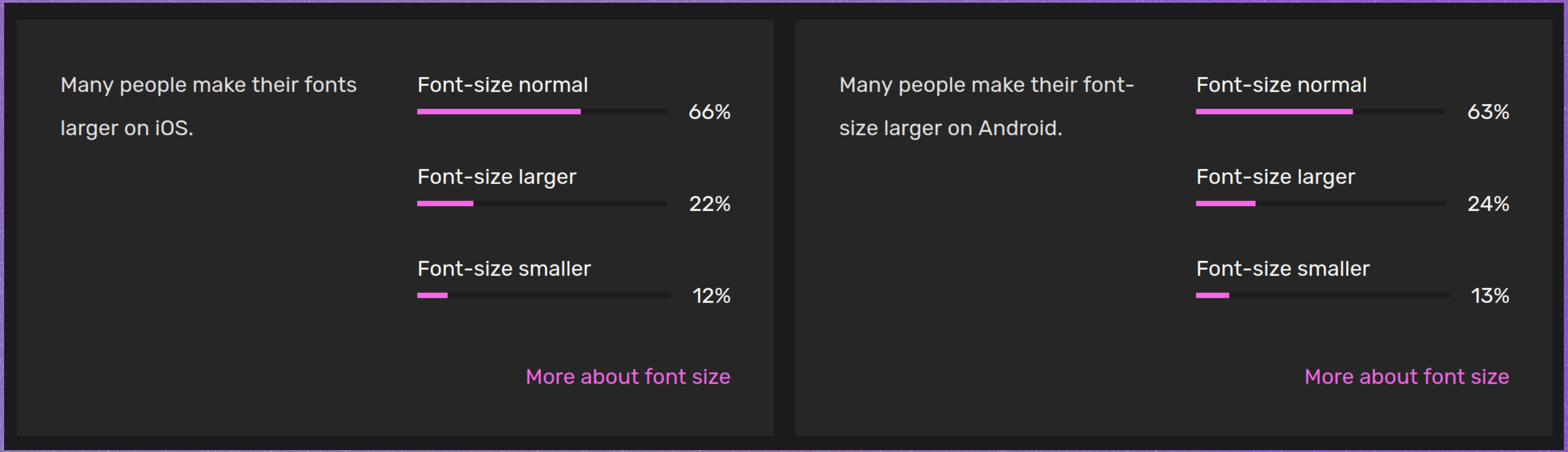
Count one point every time you answer with "yes".

Among the five people you spend the most time with:

1. Are all of them of the same ethnicity as you?
2. Do all of them have the same sexual orientation?
3. Do all of them have the same nationality as you?
4. Are all of them considered able-bodied?
5. Do all of them have a diploma (or, are pursuing one)?
6. Are all of them cisgender?
7. Are all of them working/studying or being retired?
8. Are all of them from the same or higher economic background as you?
9. Are most of them men?
10. Do they all speak the language of the country they live in?

Understanding disability

61% of Dutch Android users and 45% of Dutch iOS users have one or more accessibility settings activated on their phone.



Many people make their fonts larger on iOS.

Font-size normal 66%

Font-size larger 22%

Font-size smaller 12%

More about font size

Many people make their font-size larger on Android.

Font-size normal 63%

Font-size larger 24%

Font-size smaller 13%

More about font size

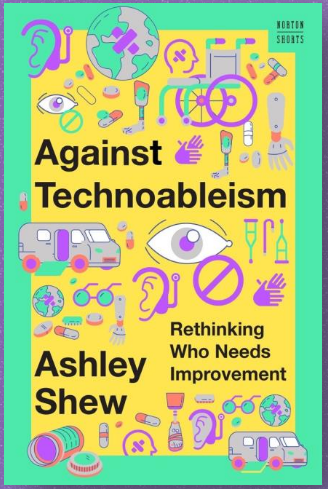
Understanding disability

"**Disability is not a binary state.** We all have abilities and limits to those abilities**. Disability happens when we have built something that doesn't work for someone with particular skills**."*

"Today, many of our ideas about able-bodiedness and disability come from classifications based on who is suitable for plantation or factory work: **we call people "disabled" when they can't perform "normal" amounts of physical labor.**"**

* Whitaker Rob.*Developing Inclusive Mobile Apps, Building Accessible Apps for iOS and Android*, 2020.
** Shew Ashley,*Against Technoableism: rethinking who needs improvement*. 2023

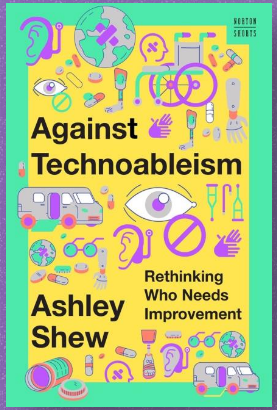
Disability experts and disabled experts



"Disabled people are "the real experts" (the title of Michelle Sutton's 2015 edited volume about and authored by autistic people) when it comes to technology and disability. We use technologies. We also reject them, grapple with them, or repurpose them. The views on technology we get from listening to disabled people often look very different from those of people educated in the medical and "helping" professions."*

** Shew Ashley, *Against Technoableism: rethinking who needs improvement*, 2023

Users amalgam



“Employing the word “users” can result in falling into the trap of thinking of users as one group – an amalgamation of people who are out there somewhere in the world using your app, a group of people that you’ll never meet and never know. This form of “group think” leads to creating a “one-size-fits-all” solution that, like everything claiming to be “one size fits all,” in reality fits no one.”*

❗ **There is no one solution, and if there were, you would probably not be the one to find it.** Your role as a software builder is to reach out to the real accessibility experts: those who rely on accessibility and experience its flaws every day, i.e. people with disabilities.

** Shew Ashley,*Against Technoableism: rethinking who needs improvement*. 2023

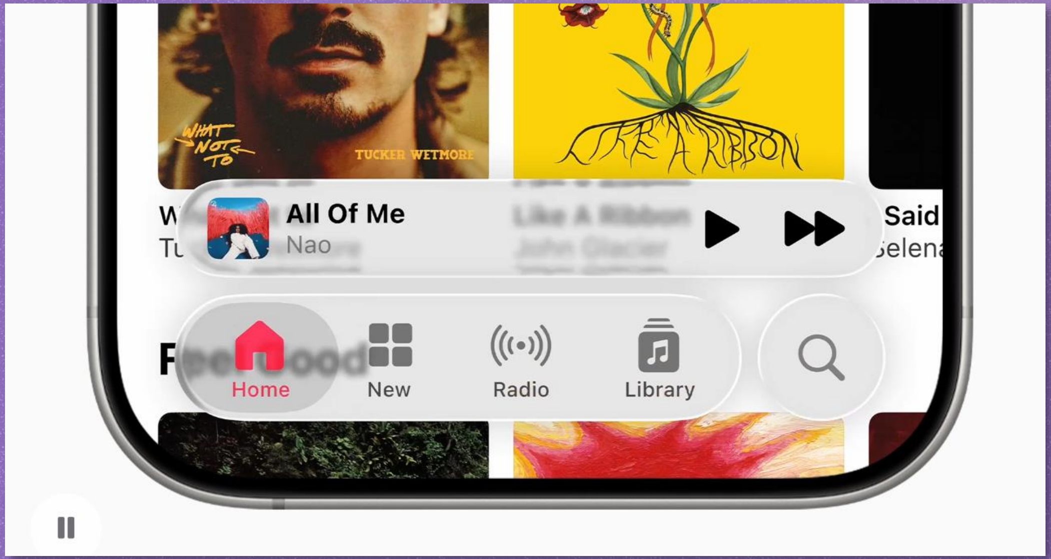
Why does open-source software struggle with accessibility and inclusiveness?

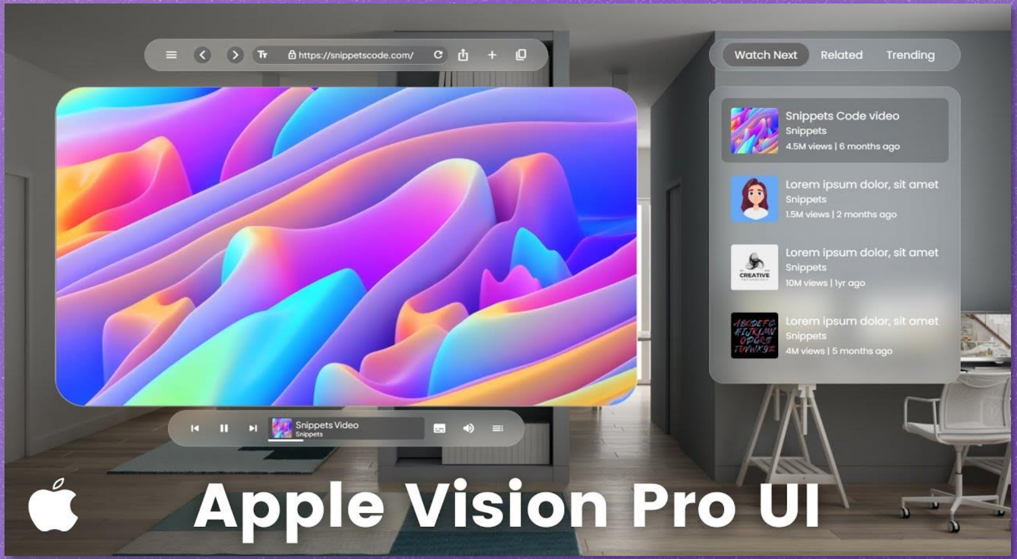
From accessibility to inclusiveness

Exclusion vs Accessibility

Exclusion is *automatic*, Accessibility requires *intentionality*.

Apple's liquid glass





https://www.apple.com/newsroom/2025/06/apple-introduces-a-delightful-and-elegant-new-software-design/
https://www.youtube.com/watch?v=l1sbZpwcTk0

The vision of FOSS

FOSS is about “**refus[ing] to break solidarity with other users.** [...] Users **will no longer be at the mercy of one programmer or company** which owns the sources and is in sole position to make changes.”

https://www.gnu.org/gnu/manifesto.en.html

FOSS according to the Free software foundation

“[F]ree software” is **a matter of liberty, not price**. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer.”

https://www.gnu.org/philosophy/free-sw.en.html#four-freedoms

The vision of FOSS

"GNU is not in the public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free."

https://www.gnu.org/gnu/manifesto.html

The four freedoms

A program is free software if the program's users have the **four essential freedoms:**

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help others (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

https://www.gnu.org/philosophy/free-sw.en.html#four-freedoms

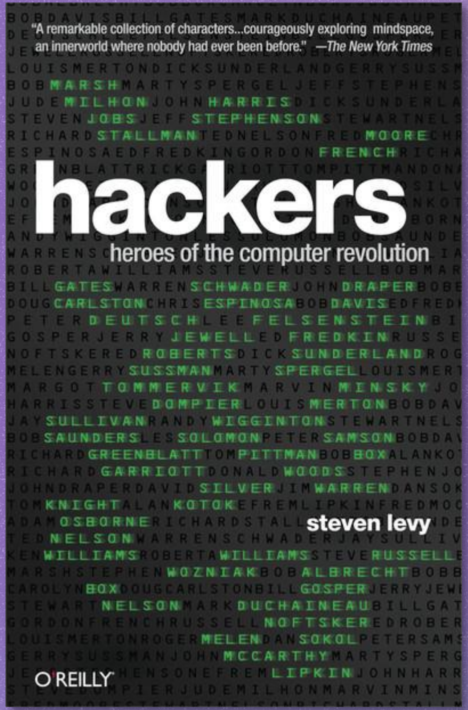
FOSS is everywhere

FOSS is not a niche ideology of idealist programmers.

- 96.3% of the top one million web servers are running Linux (the biggest and most advanced FOSS project to this day, with over 27.8 million lines of code)
- 85% of smartphones are Android and therefore Linux-based. When internet and our smartphones run Linux, FOSS runs the world.

* https://www.zdnet.com/home-and-office/networking/can-the-internet-exist-without-linux/
** https://linuxblog.io/85-of-all-smartphones-are-powered-by-linux/

The hackers ethics

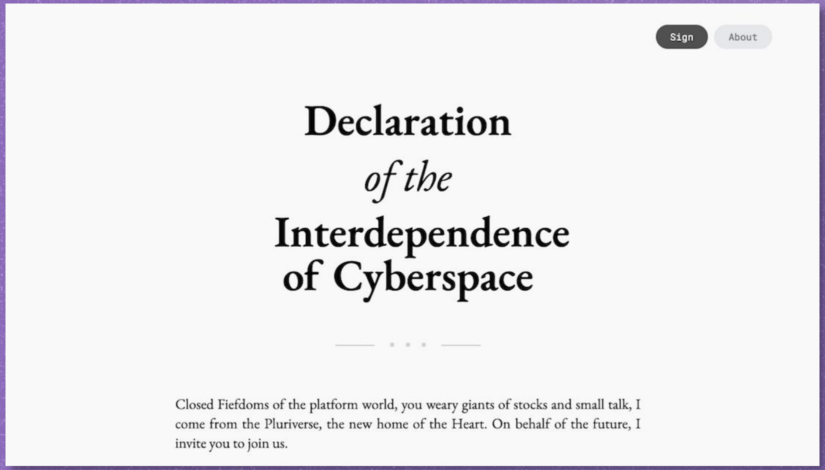

"A remarkable collection of characters...courageously exploring mindspace, an innerworld where nobody had ever been before." —The New York Times
hackers
heroes of the computer revolution
steven levy
O'REILLY

"- **Access to computers**—and anything that might teach you something about the way the world works—**should be unlimited and total**. Always yield to the Hands-On Imperative!

- **All information should be free**.
- **Mistrust Authority**—Promote Decentralization. Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
- You can create **art and beauty on a computer**.
- **Computers can change your life for the better.**"

https://www.oreilly.com/library/view/hackers/9781449390259/

Ideology in the early cyberspace



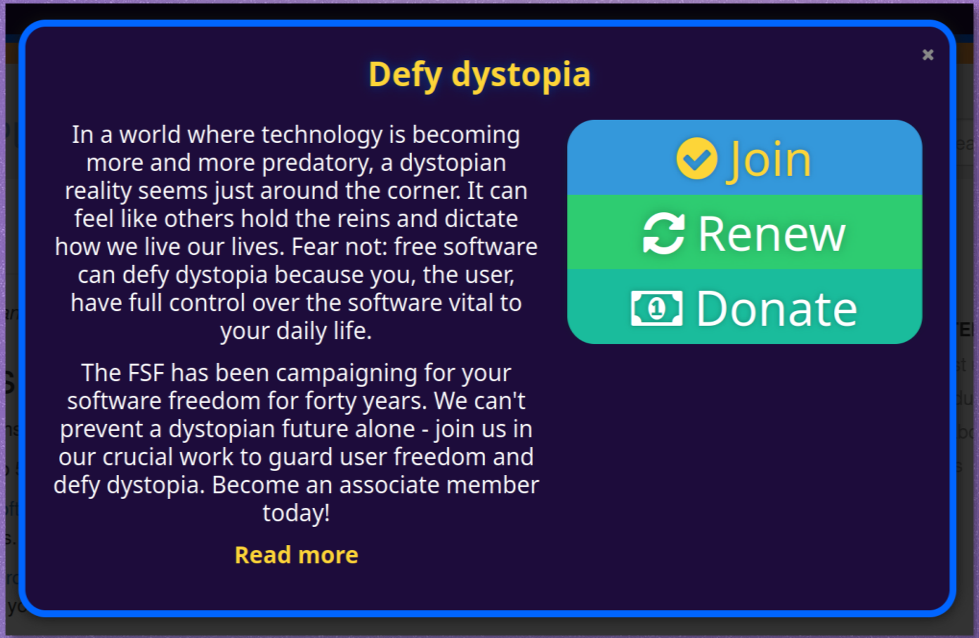
Declaration *of the* Interdependence of Cyberspace

Closed Fiefdoms of the platform world, you weary giants of stocks and small talk, I come from the Pluriverse, the new home of the Heart. On behalf of the future, I invite you to join us.

"Governments of the Industrial World, you weary giants of flesh and steel, I come from Cyberspace, the new home of Mind. On behalf of the future, I ask you of the past to leave us alone. You are not welcome among us. You have no sovereignty where we gather. We have no elected government, nor are we likely to have one, so I address you with no greater authority than that with which liberty itself always speaks. I declare the global social space we are building to be naturally independent of the tyrannies you seek to impose on us. You have no moral right to rule us nor do you possess any methods of enforcement we have true reason to fear."

A Declaration of Independence of the Cyberspace, J. P. Barlow, 1996

Today FOSS ideology



Defy dystopia

In a world where technology is becoming more and more predatory, a dystopian reality seems just around the corner. It can feel like others hold the reins and dictate how we live our lives. Fear not: free software can defy dystopia because you, the user, have full control over the software vital to your daily life.

The FSF has been campaigning for your software freedom for forty years. We can't prevent a dystopian future alone - join us in our crucial work to guard user freedom and defy dystopia. Become an associate member today!

Read more

"Dystopian worlds don't just pop into existence: they are built by the choices of those with substantial power, taking freedom from others. Free software has unlimited power for making our world actually better for people (not Big Tech), but we need to join together and ensure that freedom can thrive."

https://www.fsf.org/appeal/2025-spring/free-software-can-defy-dystopia

Motivation for joining the FSF

“The world is heading down a dangerous, greedy, closed-source for-profit path. Here's to free software, now and for years to come!” -- Ellie Hayden

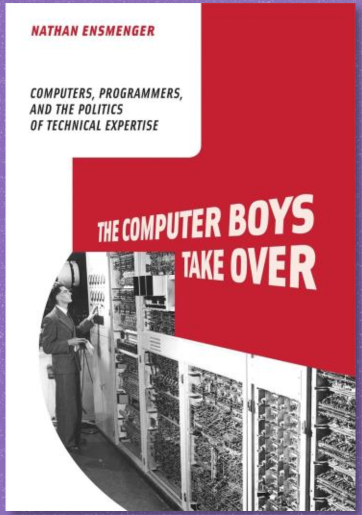
“Just to join the people for a free world. Compartir es amor.” -- Francisco Javier Delgado Grande

“I have gotten tired of big tech and unfair intellectual property laws everywhere I turn. I want to participate in and promote a world that I actually want to live in.” -- Isaac Martinolich

“Let's make the world a better place!” -- Olivia Cimno

https://www.fsf.org/appeal/2025-spring/free-software-can-defy-dystopia

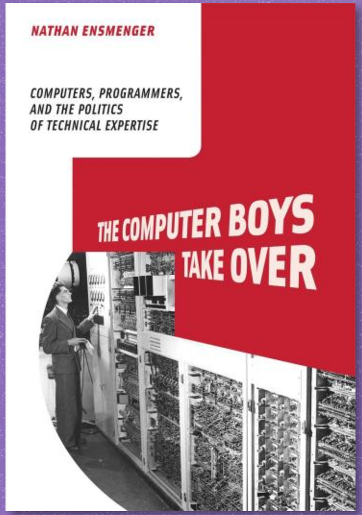
Motivations for early software


NATHAN ENSMENGER
COMPUTERS, PROGRAMMERS, AND THE POLITICS OF TECHNICAL EXPERTISE
THE COMPUTER BOYS TAKE OVER

"It was not until the late 1960s that software became a product that could be purchased separately from a computer, and even then software as code represented only a small component of a larger software system of services and support."

Ensmenger, Nathan. *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. The MIT Press, 2010.

Early software contributors


NATHAN ENSMENGER
COMPUTERS, PROGRAMMERS, AND THE POLITICS OF TECHNICAL EXPERTISE
THE COMPUTER BOYS TAKE OVER

"It is no coincidence that the first software workers were women. The use of the word software in this context is, of course, anachronistic— the word itself would not be introduced until 1958— but the hierarchical distinctions and gender connotations it embodies— between " hard " technical mastery, and the " softer, " more social (and implicitly, of secondary importance) aspects of computer work— are applicable even in the earliest of electronic computing development projects (...) it was clearly the male computer engineers who were significant."

Ensmenger, Nathan. *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. The MIT Press, 2010.

Contributors to early software

F/OSS

FOSS is needed for Defense

FOSS is deeply needed by states and other centralized systems of power. Indeed the U.S. Department of Defense states that their security "depends on (OSS) applications and strategies" with at the time of an 2003 survey, 44 examples of FOSS software use in the Departement of Defense".

"The FOSS communities contribute to DoD security in two ways. Firstly, it has produced infrastructure software such as OpenBSD with low rates of software failure combined with early and rapid closure of security holes, which makes such systems useful as the security linchpins in broader security strategies. Secondly, the FOSS communities have had a long-term fascination with developing more and more sophisticated applications for identifying and analyzing security holes in networks and computers, resulting in FOSS products such as SARA and Snort that are invaluable to in-depth analyses of security risks."

Use of Free and Open Source Software (FOSS) in the U.S. Department of Defense. 2003

What is innovation?

There is no natural way technical innovation is going.

Only technologies receiving enough time, effort and therefore money eventually yield results that are perceived as “progress”.

Corporate contributions to FOSS

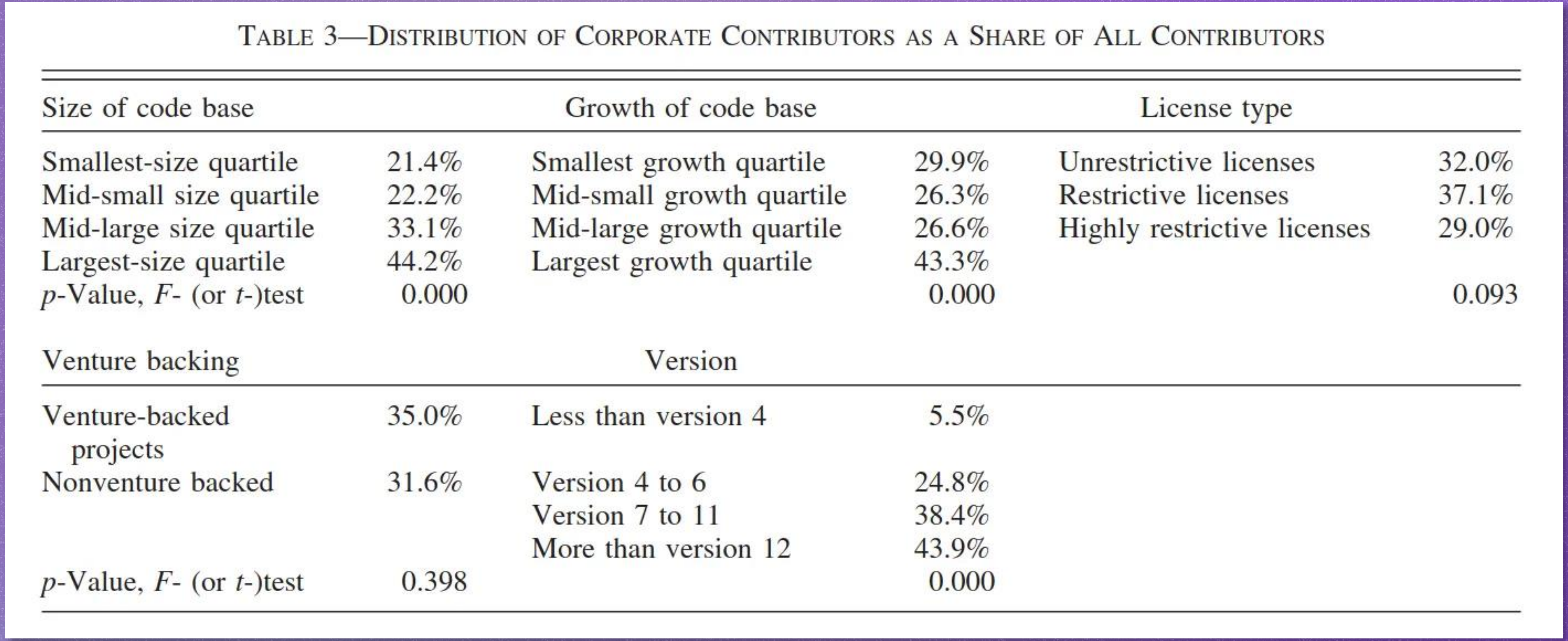
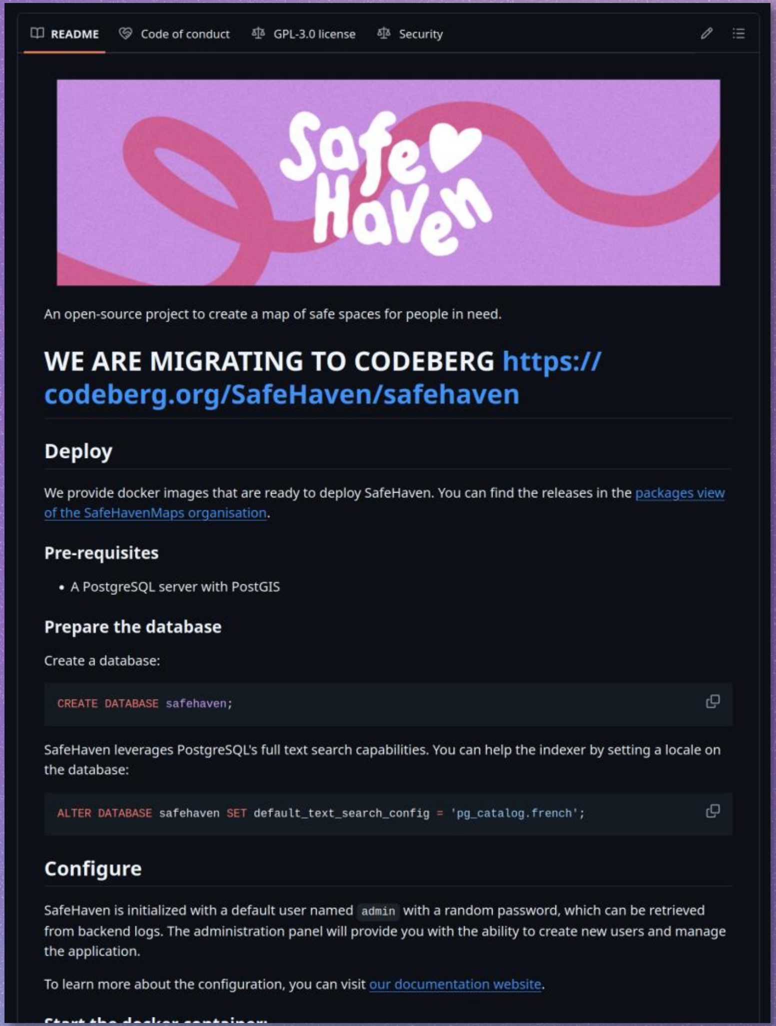


TABLE 3—DISTRIBUTION OF CORPORATE CONTRIBUTORS AS A SHARE OF ALL CONTRIBUTORS

Size of code base		Growth of code base		License type	
Smallest-size quartile	21.4%	Smallest growth quartile	29.9%	Unrestrictive licenses	32.0%
Mid-small size quartile	22.2%	Mid-small growth quartile	26.3%	Restrictive licenses	37.1%
Mid-large size quartile	33.1%	Mid-large growth quartile	26.6%	Highly restrictive licenses	29.0%
Largest-size quartile	44.2%	Largest growth quartile	43.3%		
p-Value, *F*- (or *t*-)test	0.000		0.000		0.093
Venture backing		**Version**			
Venture-backed projects	35.0%	Less than version 4	5.5%		
Nonventure backed	31.6%	Version 4 to 6	24.8%		
		Version 7 to 11	38.4%		
		More than version 12	43.9%		
p-Value, *F*- (or *t*-)test	0.398		0.000		

The Dynamics of Open-Source Contributors By JOSH LERNER, PARAG A. PATHAK, AND JEAN TIROLE*

FOSS projects that are useful to minorities



An open-source project to create a map of safe spaces for people in need.

WE ARE MIGRATING TO CODEBERG https://codeberg.org/SafeHaven/safehaven

Deploy

We provide docker images that are ready to deploy SafeHaven. You can find the releases in the packages view of the SafeHavenMaps organisation.

Pre-requisites

- A PostgreSQL server with PostGIS

Prepare the database

Create a database:

```
CREATE DATABASE safehaven;
```

SafeHaven leverages PostgreSQL's full text search capabilities. You can help the indexer by setting a locale on the database:

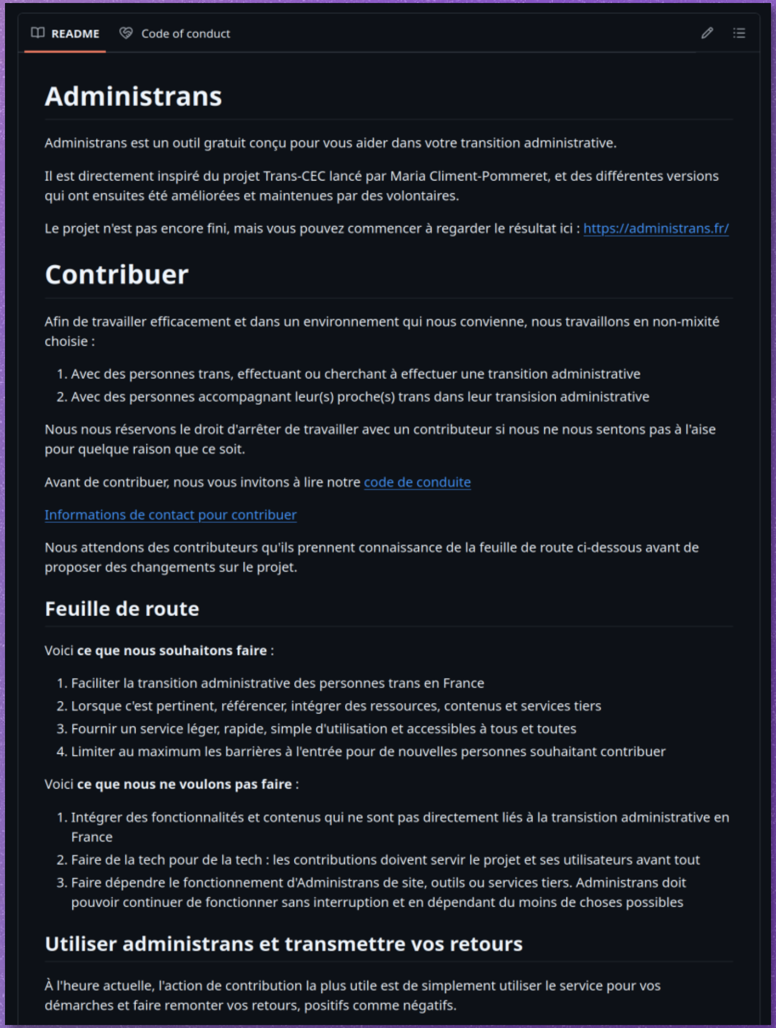
```
ALTER DATABASE safehaven SET default_text_search_config = 'pg_catalog.french';
```

Configure

SafeHaven is initialized with a default user named `admin` with a random password, which can be retrieved from backend logs. The administration panel will provide you with the ability to create new users and manage the application.

To learn more about the configuration, you can visit our documentation website.

Start the docker container



Administrans

Administrans est un outil gratuit conçu pour vous aider dans votre transition administrative.

Il est directement inspiré du projet Trans-CEC lancé par Maria Climent-Pommeret, et des différentes versions qui ont ensuites été améliorées et maintenues par des volontaires.

Le projet n'est pas encore fini, mais vous pouvez commencer à regarder le résultat ici : https://administrans.fr/

Contribuer

Afin de travailler efficacement et dans un environnement qui nous convienne, nous travaillons en non-mixité choisie :

1. Avec des personnes trans, effectuant ou cherchant à effectuer une transition administrative
2. Avec des personnes accompagnant leur(s) proche(s) trans dans leur transision administrative

Nous nous réservons le droit d'arrêter de travailler avec un contributeur si nous ne nous sentons pas à l'aise pour quelque raison que ce soit.

Avant de contribuer, nous vous invitons à lire notre code de conduite

Informations de contact pour contribuer

Nous attendons des contributeurs qu'ils prennent connaissance de la feuille de route ci-dessous avant de proposer des changements sur le projet.

Feuille de route

Voici **ce que nous souhaitons faire** :

1. Faciliter la transition administrative des personnes trans en France
2. Lorsque c'est pertinent, référencer, intégrer des ressources, contenus et services tiers
3. Fournir un service léger, rapide, simple d'utilisation et accessibles à tous et toutes
4. Limiter au maximum les barrières à l'entrée pour de nouvelles personnes souhaitant contribuer

Voici **ce que nous ne voulons pas faire** :

1. Intégrer des fonctionnalités et contenus qui ne sont pas directement liés à la transistion administrative en France
2. Faire de la tech pour de la tech : les contributions doivent servir le projet et ses utilisateurs avant tout
3. Faire dépendre le fonctionnement d'Administrans de site, outils ou services tiers. Administrans doit pouvoir continuer de fonctionner sans interruption et en dépendant du moins de choses possibles

Utiliser administrans et transmettre vos retours

À l'heure actuelle, l'action de contribution la plus utile est de simplement utiliser le service pour vos démarches et faire remonter vos retours, positifs comme négatifs.

The disregard for low-profile tasks

"It is argued that the high-profile, high-prestige tasks get disproportionate attention compared to the low-profile, low-prestige tasks such as application documentation."

Open source and accessibility: advantages and limitations. Michael Heron 1, Vicki L Hanson and Ian Ricketts

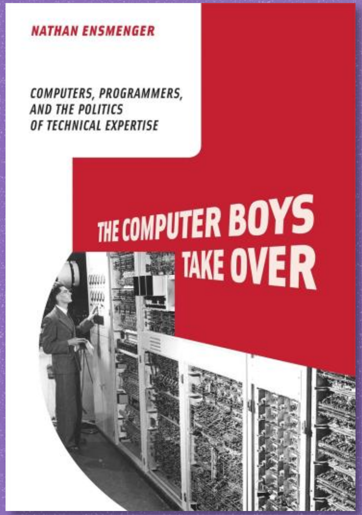
But is it only the fault of the economic system?

OSS and designing "for"

"This is a pattern in OSS even outside of the context of crisis but often exacerbated by crisis. Designers, researchers, product managers, community managers and documentarians are excluded, often not explicitly, but implicitly by the 'coding focussed' culture of OSS. (...)The dangerous gap being that the structures in place currently lack an inclusive, contextually (crisis) relevant and sustainable way of building and maintaining the OSS along with the community itself being inclusive, contextually relevant and sustainable. This can in the most harmful situations, add to a colonisation of crisis, aid relief and technology where communities are designed 'for' not 'by' and 'with'."

Eriol Fox *Talking-with-humanitarian-tech-group-about-OSS* (draft)

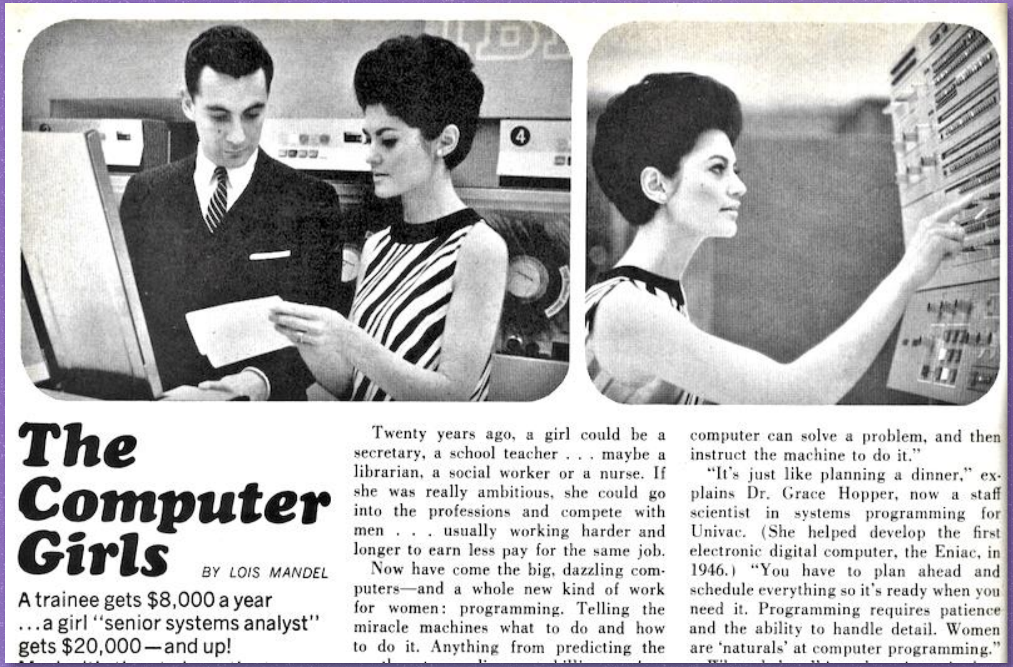
Programmers and operators


NATHAN ENSMENGER
COMPUTERS, PROGRAMMERS, AND THE POLITICS OF TECHNICAL EXPERTISE
THE COMPUTER BOYS TAKE OVER

"the telephone switchboardlike appearance of the ENIAC programming cable-and-plug panels reinforced the notion that programmers were mere machine operators, that programming was more handicraft than science, more feminine than masculine, more mechanical than intellectual"

Ensmenger, Nathan. *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. The MIT Press, 2010.

Women and software development



The Computer Girls BY LOIS MANDEL

A trainee gets $8,000 a year ...a girl ''senior systems analyst'' gets $20,000—and up!

Twenty years ago, a girl could be a secretary, a school teacher . . . maybe a librarian, a social worker or a nurse. If she was really ambitious, she could go into the professions and compete with men . . . usually working harder and longer to earn less pay for the same job.

Now have come the big, dazzling computers—and a whole new kind of work for women: programming. Telling the miracle machines what to do and how to do it. Anything from predicting the

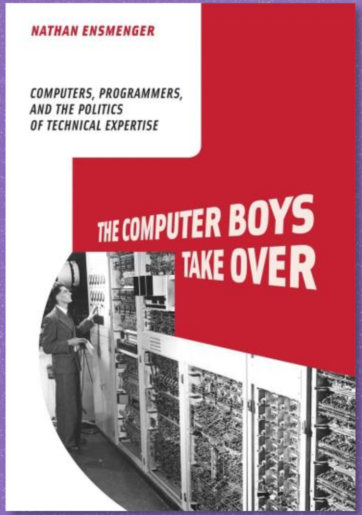
computer can solve a problem, and then instruct the machine to do it."

"It's just like planning a dinner," explains Dr. Grace Hopper, now a staff scientist in systems programming for Univac. (She helped develop the first electronic digital computer, the Eniac, in 1946.) "You have to plan ahead and schedule everything so it's ready when you need it. Programming requires patience and the ability to handle detail. Women are 'naturals' at computer programming."

"(Programming) is just like planning a dinner. (...) This is the age of the Computer Girls. There are twenty thousand of them in the United States already. (...) Women are ' naturals' at computer programming."

Ensmenger, Nathan. *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. The MIT Press, 2010.

Programmers recruitment bias


NATHAN ENSMENGER
COMPUTERS, PROGRAMMERS, AND THE POLITICS OF TECHNICAL EXPERTISE
THE COMPUTER BOYS TAKE OVER

"This bias toward male programmers was not so much deliberate as it was convenient— a combination of laziness, ambiguity, and traditional male privilege. The fact that the use of lazy screening practices inadvertently excluded large numbers of potential female trainees was simply never considered. But the increasing assumption that the average programmer was also male did play a key role in the establishment of a highly masculine programming subculture."

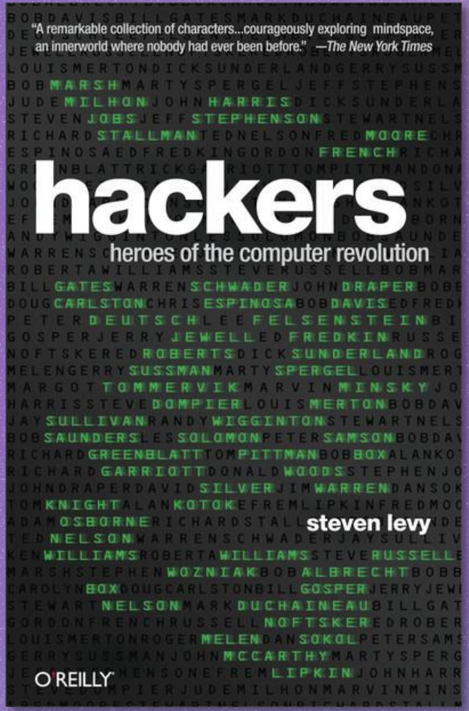
Ensmenger, Nathan. *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. The MIT Press, 2010.

FOSS lack of intersectionnality

When one of his colleague was accused of assaulting a 17 years old, he stated in a leaked email that "the most plausible scenario is that she presented herself to him as entirely willing." and that the definition of 'rape' shouldn't "depends on minor details such as which country it was in or whether the victim was 18 years old or 17". He also stated on his blog in 2011 about child pornography that "Having a photo or drawing does not hurt anyone, so and if you or I think it is disgusting, that is no excuse for censorship."

https://stallman.org/notes/2011-mar-jun.html
https://www.vice.com/en/article/famed-computer-scientist-richard-stallman-described-epstein-victims-as-entirely-willing/
https://arstechnica.com/tech-policy/2019/09/richard-stallman-leaves-mit-after-controversial-remarks-on-rape/

The hackers ethics


"A remarkable collection of characters...courageously exploring mindspace, an innerworld where nobody had ever been before." —The New York Times
hackers
heroes of the computer revolution
steven levy
O'REILLY

"- **Access to computers**—and anything that might teach you something about the way the world works—**should be unlimited and total**. Always yield to the Hands-On Imperative!

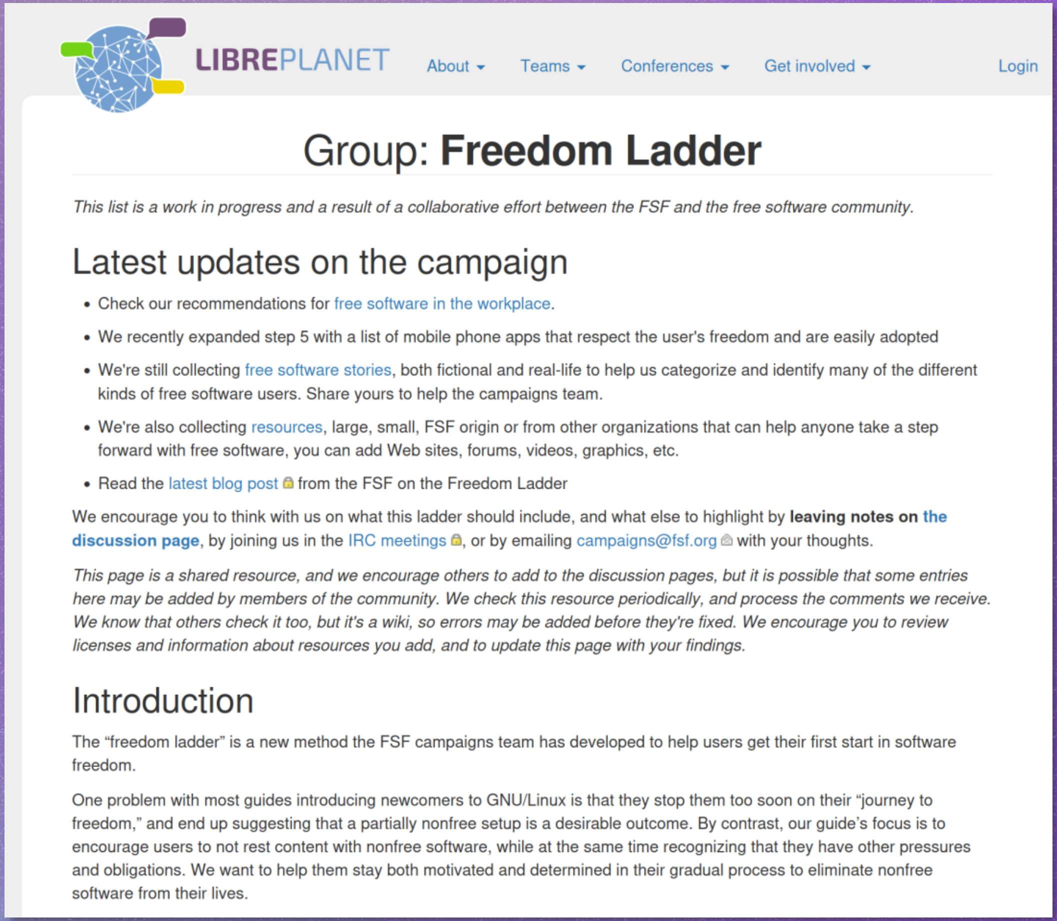
- **All information should be free**.
- **Mistrust Authority**—Promote Decentralization. Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
- You can create **art and beauty on a computer**.
- **Computers can change your life for the better.**"

https://www.oreilly.com/library/view/hackers/9781449390259/

Inclusiveness is rethinking what is a good programmer

Inclusiveness cannot happen in this exclusive culture that only swears in technical proficiency.

The freedom ladder



Group: **Freedom Ladder**

This list is a work in progress and a result of a collaborative effort between the FSF and the free software community.

Latest updates on the campaign

- Check our recommendations for free software in the workplace.
- We recently expanded step 5 with a list of mobile phone apps that respect the user's freedom and are easily adopted
- We're still collecting free software stories, both fictional and real-life to help us categorize and identify many of the different kinds of free software users. Share yours to help the campaigns team.
- We're also collecting resources, large, small, FSF origin or from other organizations that can help anyone take a step forward with free software, you can add Web sites, forums, videos, graphics, etc.
- Read the latest blog post from the FSF on the Freedom Ladder

We encourage you to think with us on what this ladder should include, and what else to highlight by **leaving notes on the discussion page**, by joining us in the IRC meetings, or by emailing campaigns@fsf.org with your thoughts.

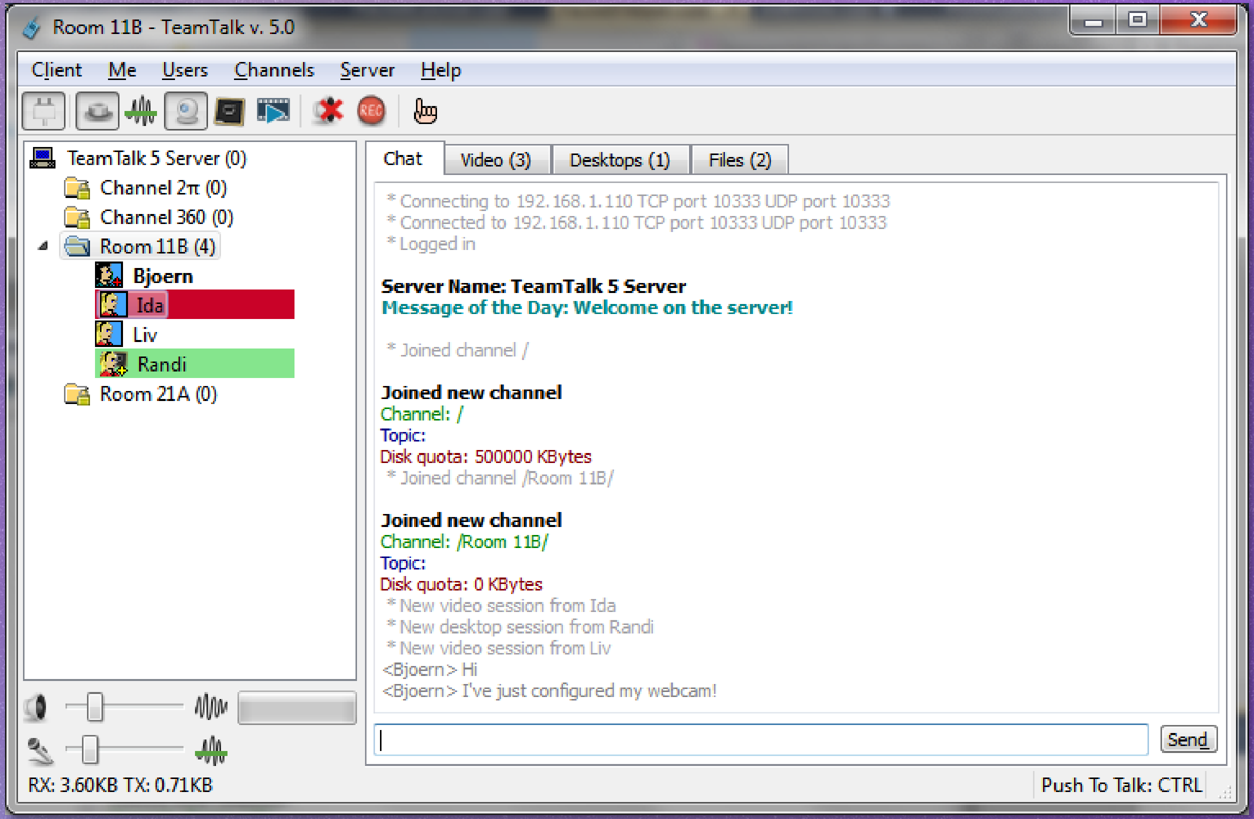
This page is a shared resource, and we encourage others to add to the discussion pages, but it is possible that some entries here may be added by members of the community. We check this resource periodically, and process the comments we receive. We know that others check it too, but it's a wiki, so errors may be added before they're fixed. We encourage you to review licenses and information about resources you add, and to update this page with your findings.

Introduction

The “freedom ladder” is a new method the FSF campaigns team has developed to help users get their first start in software freedom.

One problem with most guides introducing newcomers to GNU/Linux is that they stop them too soon on their “journey to freedom,” and end up suggesting that a partially nonfree setup is a desirable outcome. By contrast, our guide’s focus is to encourage users to not rest content with nonfree software, while at the same time recognizing that they have other pressures and obligations. We want to help them stay both motivated and determined in their gradual process to eliminate nonfree software from their lives.

TeamTalk5



```
* Connecting to 192.168.1.110 TCP port 10333 UDP port 10333
* Connected to 192.168.1.110 TCP port 10333 UDP port 10333
* Logged in

Server Name: TeamTalk 5 Server
Message of the Day: Welcome on the server!

* Joined channel /

Joined new channel
Channel: /
Topic:
Disk quota: 500000 KBytes
* Joined channel /Room 11B/

Joined new channel
Channel: /Room 11B/
Topic:
Disk quota: 0 KBytes
* New video session from Ida
* New desktop session from Randi
* New video session from Liv
<Bjoern> Hi
<Bjoern> I've just configured my webcam!
```

RX: 3.60KB TX: 0.71KB

Push To Talk: CTRL

Advices for open source accessibility projects

"1. Associate the software with a trusted brand if possible, such as a university, charity, or software publisher.

2. Ensure a single, certifiable location from which the ‘official’ installer can be downloaded.

3. Incorporate digital signing of plug-ins so as to ensure they are reviewed before being made available to end users.

4. Make available the necessary technical architecture to support genuinely collaborative community support.

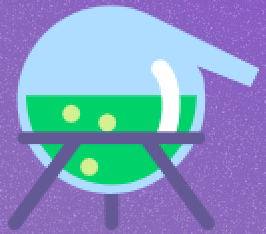
5. Inculcate a good ‘civic‘mind-set for support by recruiting the most helpful of contributors as moderators.

6.Provide public recognition for contributions that are considered ‘high value’ for the project in the form of visible achievements"

Open source and accessibility: advantages and limitations. Michael Heron 1, Vicki L Hanson and Ian Ricketts

For and by diversity

FOSS should be made by a diversity of people for the diversity of people called humanity.



Bring inclusiveness and accessibility to your pipeline

From accessibility to inclusiveness

Incorporating inclusiveness into your workflows

The structure of your workflows shape the application you are delivering.

By explicitly incorporating accessibility and inclusiveness throughout your design and development pipeline, you are making sure that you and your team are actively working towards an application that will be usable and welcoming to everyone.

Accessibility and inclusiveness essentials

1. Have inclusiveness in mind since the ideation
2. Make it a goal for your designs
3. Iterate on implementation
4. Test
5. Get user feedback

Having an inclusiveness/accessibility coach

The coaches can be the ones **who monitor with other developers the progress of the current accessibility roadmap,** without doing all of the accessibility-related work themselves.

They are their team's reference on accessibility issues.

Barrell, Dylan. *Agile Accessibility Handbook*. Amplify Publishing, 2020.

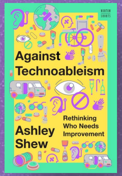
Building empathy

We all have a **singular, partial perspective of the world**. These comes with **biases** but also **insights** and **experiences**.

This can lead to an "**empathy gap**" (as called in Barrell's book) where accessibility and inclusiveness feel like remote and niche preoccupations.

- **Share stories of users who took advantage of the inclusiveness of your app**
- **Host events where team members can try and use your app with assistive technologies or simulated disabilities, and reflect on what could be improved.**

Barrell, Dylan. *Agile Accessibility Handbook*. Amplify Publishing, 2020.



Jill's story

“When I was in grad school, [...] I made great friends with my fellow student Jill. [...] She passed away over a decade ago now, but back then, we hung out with a larger group from our cohort, and Jill and I got along especially well. We had classes together, occasionally shopped together, and she’d invite us all to her apartment for group hangs. She was originally from New Jersey but had moved to New Mexico, where the climate was better for her. She loved New Mexico. She was just away to earn a master’s degree: two years in our program in Virginia, then she planned to return home to New Mexico to teach. She was full of jokes, and it was easy to be with her.

On our campus, disability accessibility was . . . shitty, to say the least. [T]he apartment she ended up in wasn’t even truly accessible. It had a bathroom with an appropriate sink and toilet, but the shower wasn’t great, and the kitchen was just the same standard kitchen that was in all the apartments in the complex, so she had to be creative. [...] The parking lot was always overfilled on football game days. On one of those days, we were going to hang out, and then she was going to go over to the library for an event. I drove over, found a parking spot way out in some grass, and trekked over to her apartment.

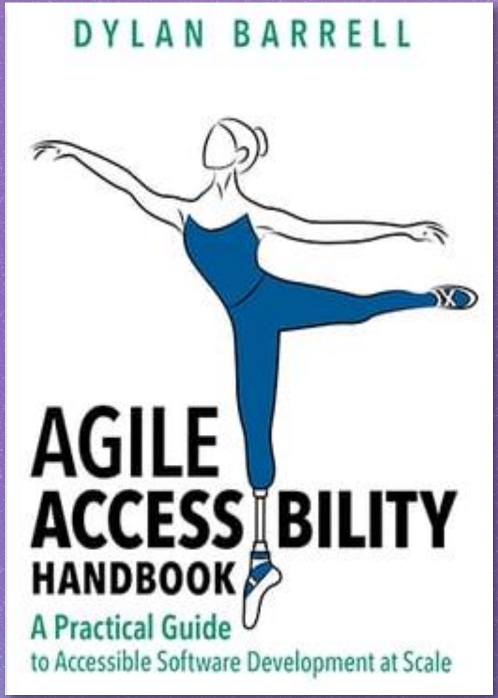
Jill was on the phone when I got there. Someone had parked next to her giant white van in the yellow-slashed area that is the open zone next to her disability parking spot. [....] She could neither get to the side of her van to get in nor open the door to allow the van’s wheelchair lift to fold out. She had already called the apartment complex once that day. Although the complex had signs all around the lot saying that it would tow away non-resident cars (which this one was!), it was game day, so they didn’t want to call a tow company. They said they couldn’t do anything and told her to call the police. She called the non-emergency police line, and they said it was the domain of her apartment complex. And it was game day after all, so the police were busy too. She was shuttled back and forth like a tennis ball. No one was going to do anything.

[...] She encouraged me to let the air out of the car’s tires. I regret that I was too much of a wuss at age twenty-three to actually do it. We already knew the police weren’t going to come about the car, so why didn’t I? Often, past me disappoints my currently disabled self.

I think of all the infrastructure that failed here—that was really built to fail, because it was never built to consider disabled people. It did the bare minimum: the physical complex (and administrators, and the police) met a few legally required standards, but they did not actually enforce the civil-rights law that is the Americans with Disabilities Act (ADA). Few complexes met even the lowest bar, so her housing options were limited—she was glad to even get that dumpy, inconvenient apartment. The systems in place supposedly to give her equal access—including her leasing office and the police—were indifferent to actually ensuring that she had it. Jill missed the event she said she’d help with at our library. Turned out she was accustomed to these system failures. At least that day wasn’t a failure with life-or-death stakes, like the ones that would delay her access to gynecological care and ultimately lead to her death less than two years later.”*

* Shew Ashley,*Against Technoableism: rethinking who needs improvement*. 2023

The agile accessibility Handbook



If you want to learn further on this topic, **the Agile Accessibility Handbook** mentioned previously is a **very solid starting point, tackling big picture transformation processes as well as small-scale team practices.**

Inclusive Design 101

Designing inclusive and accessible software

What is an inclusive app?

Inclusiveness is

“The quality of including many different types of people and treating them fairly and equally” (Cambridge Dictionary)

An inclusive app is

one that can not only be used completely, comfortably and safely by all users, but also in which they can all thrive, feel welcome and express themselves.

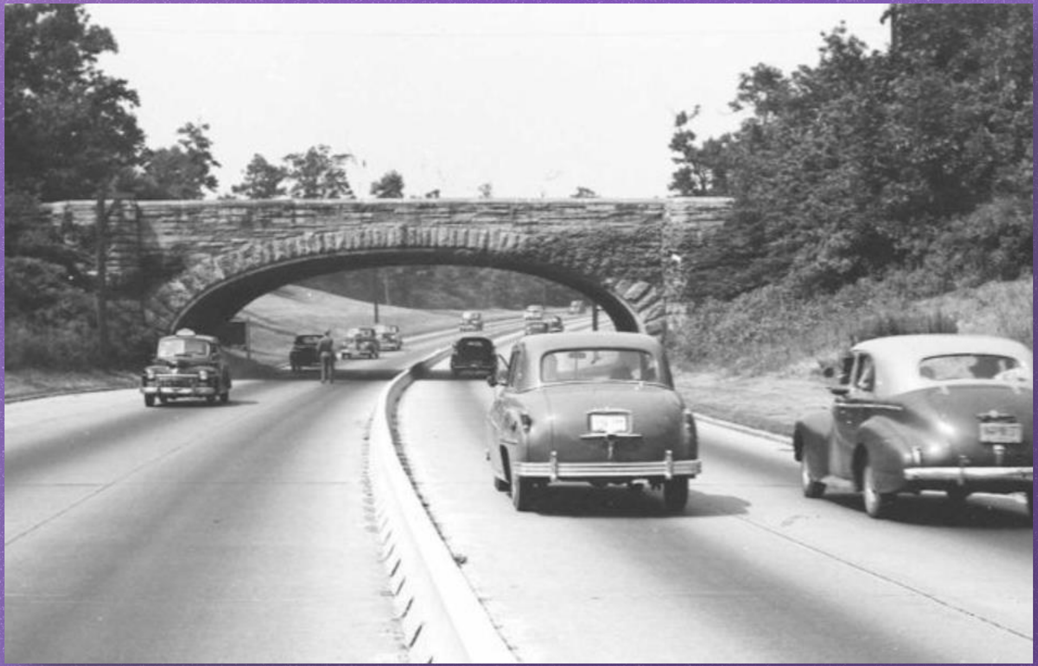
What is an inclusive app?

Are there human characteristics that could prevent a user from understanding, navigating, interacting with or feeling well with the provided content?

Age
Culture
Digital literacy
Education
Ethnicity
Gender, gender identity
Historical context
Language
Mental health
Nationality
Physical *and* cognitive (dis)abilities
Sexuality
Socio-economic context
…

Designing for inclusiveness

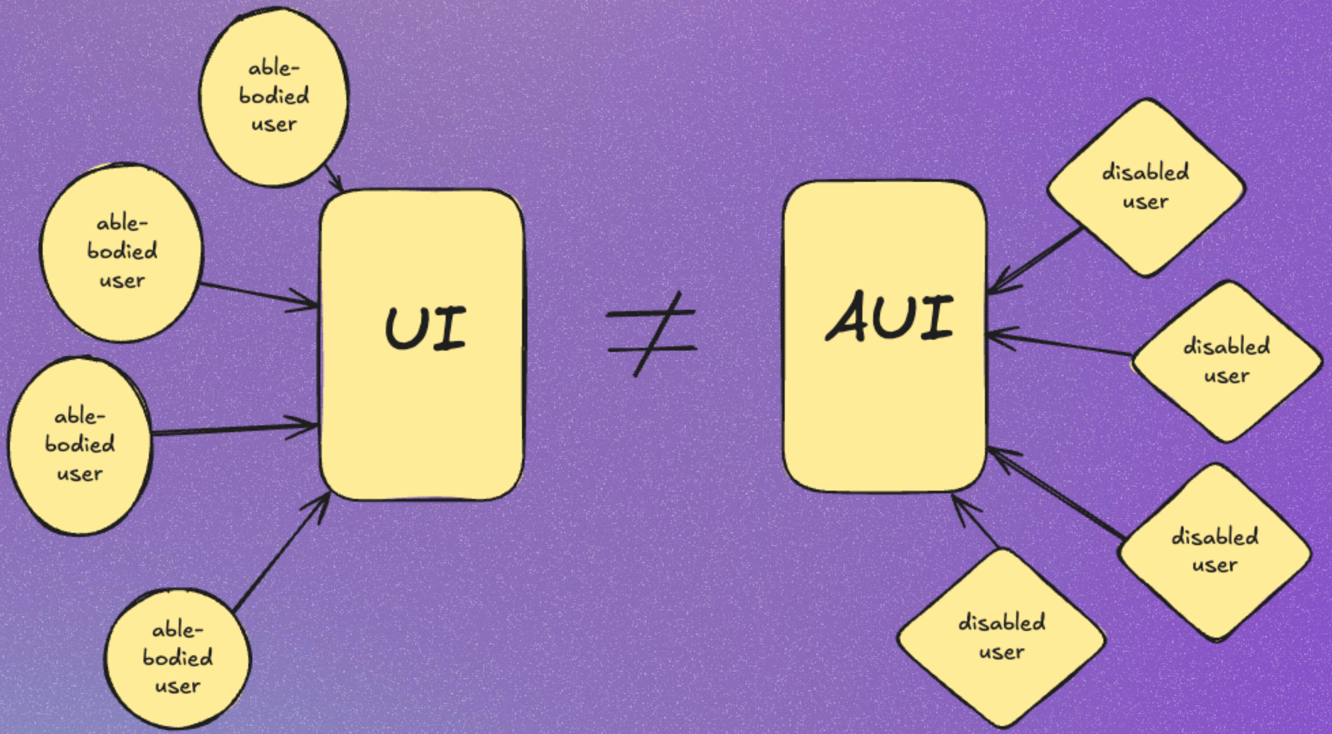
- While implementation is important, **design is crucial** to build an inclusive app.
- You and your work are the team's compass towards a great – and inclusive – product!
- Don't ever "make a feature accessible" again. Instead, build it with inclusiveness in mind.
- Is this bridge inclusive?



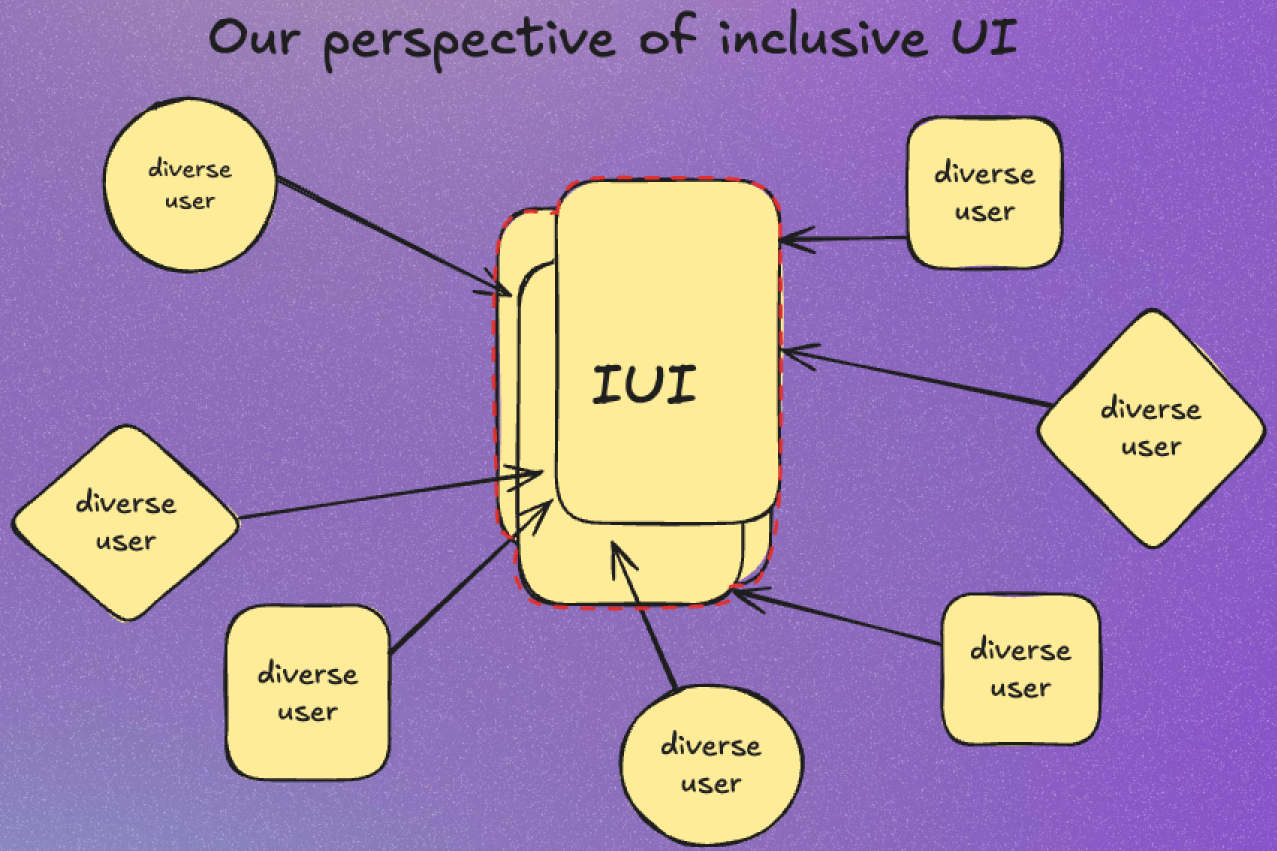
Winner, Langdon (1980). *Do artifacts have politics?* Daedalus 109 (1): pp. 121--136.

What is an IUI?

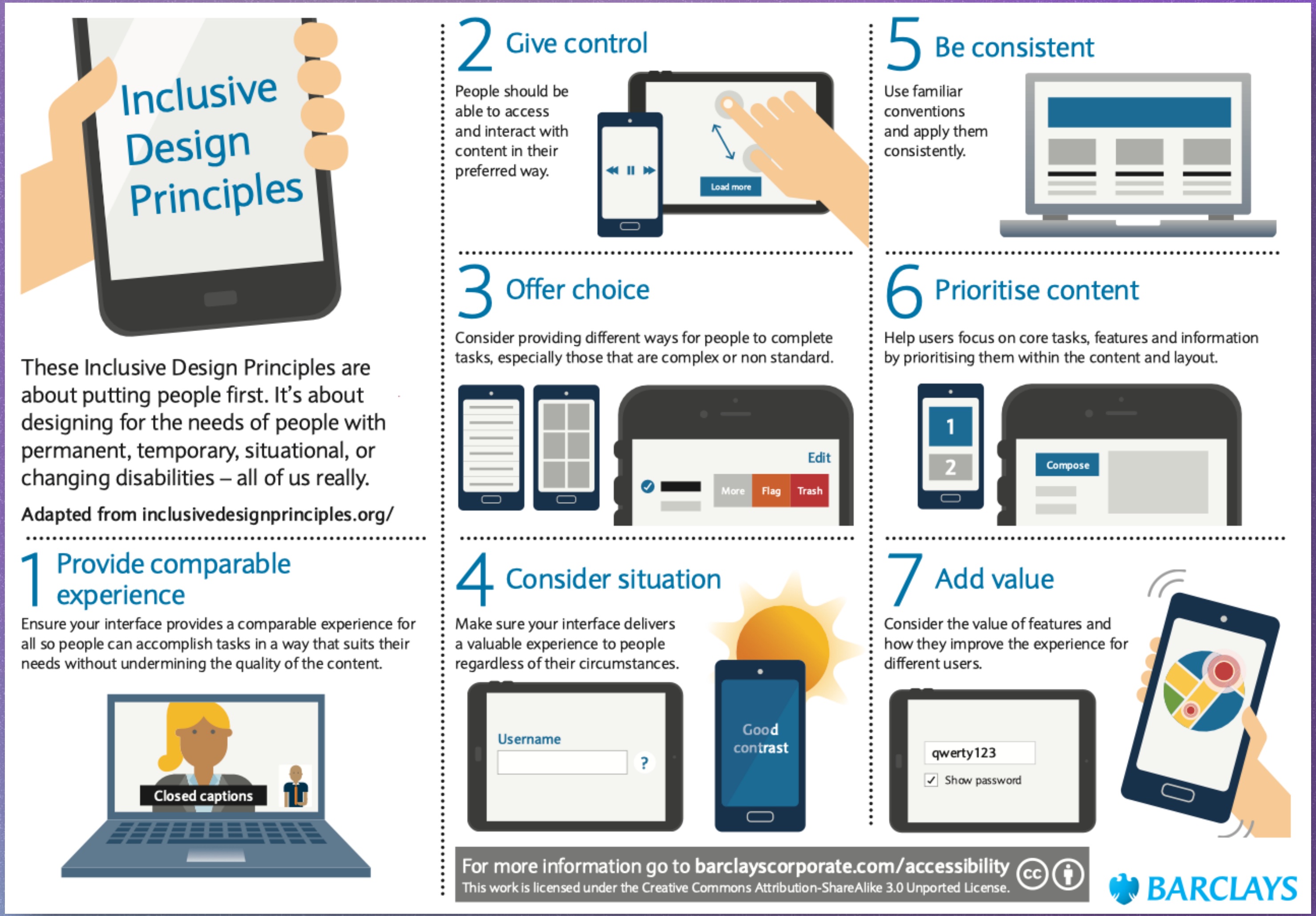
Common perspective of accessible UI



What is an IUI?



Inclusive Design Principles


Inclusive Design Principles
Load more
Edit
More
Flag
Trash
1
2
Compose
Closed captions
Username
?
Good contrast
qwerty123
✓ Show password

These Inclusive Design Principles are about putting people first. It's about designing for the needs of people with permanent, temporary, situational, or changing disabilities – all of us really.

Adapted from inclusivedesignprinciples.org/

1 Provide comparable experience

Ensure your interface provides a comparable experience for all so people can accomplish tasks in a way that suits their needs without undermining the quality of the content.

2 Give control

People should be able to access and interact with content in their preferred way.

3 Offer choice

Consider providing different ways for people to complete tasks, especially those that are complex or non standard.

4 Consider situation

Make sure your interface delivers a valuable experience to people regardless of their circumstances.

5 Be consistent

Use familiar conventions and apply them consistently.

6 Prioritise content

Help users focus on core tasks, features and information by prioritising them within the content and layout.

7 Add value

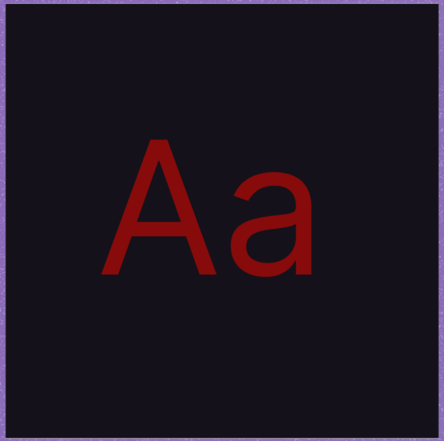
Consider the value of features and how they improve the experience for different users.

For more information go to **barclayscorporate.com/accessibility**

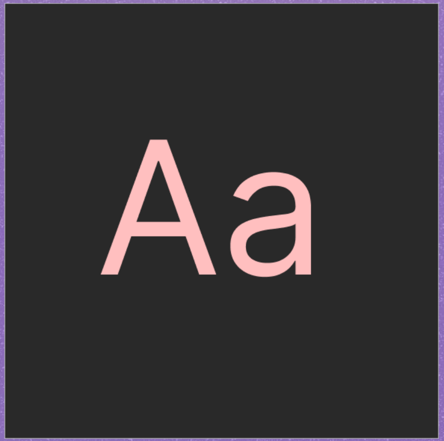
BARCLAYS

Now, how do you concretely make an IUI?

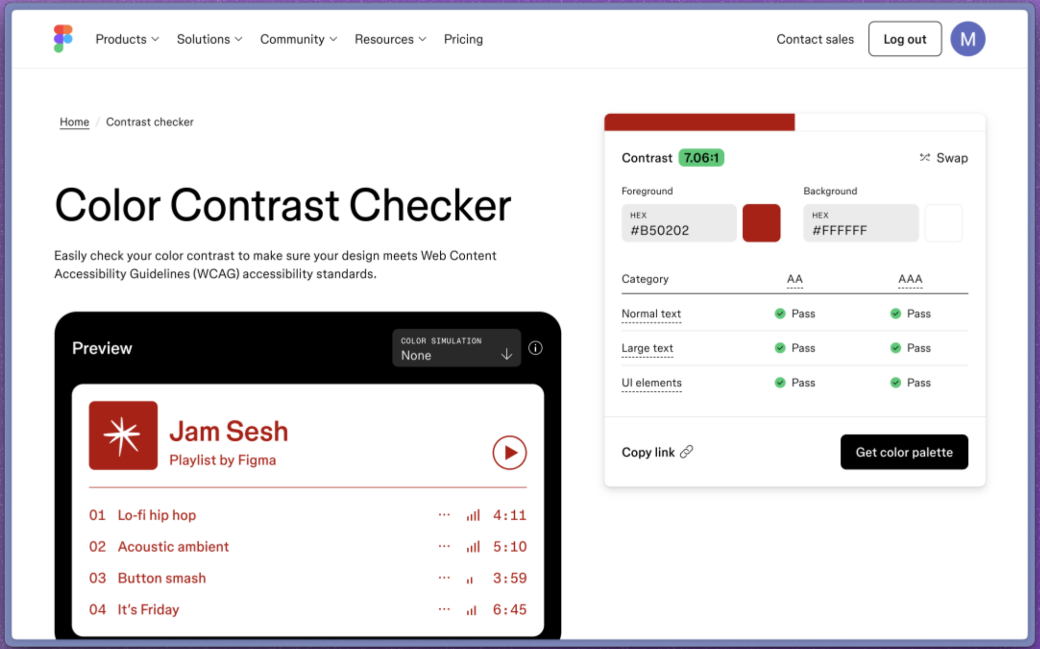
The colors are contrasted



✗



✓



Home / Contrast checker

Color Contrast Checker

Easily check your color contrast to make sure your design meets Web Content Accessibility Guidelines (WCAG) accessibility standards.

Preview

COLOR SIMULATION: None

Jam Sesh
Playlist by Figma

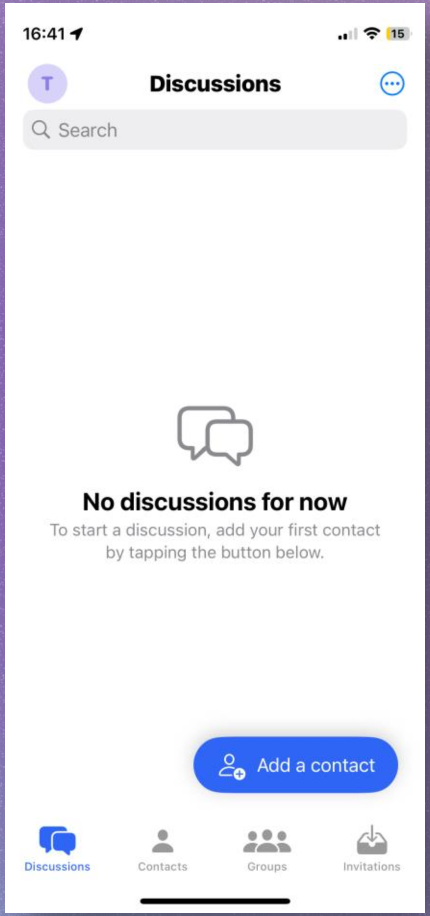
```
01  Lo-fi hip hop        4:11
02  Acoustic ambient     5:10
03  Button smash         3:59
04  It's Friday          6:45
```

Contrast 7.06:1 — Swap

Foreground HEX: #B50202
Background HEX: #FFFFFF

Category	AA	AAA
Normal text	✓ Pass	✓ Pass
Large text	✓ Pass	✓ Pass
UI elements	✓ Pass	✓ Pass

The colors are not necessary to understand the UI


16:41

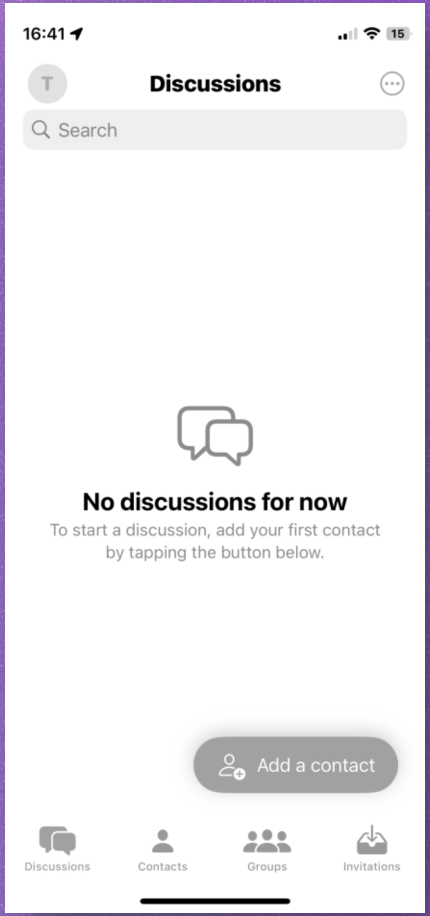
Discussions

Search

No discussions for now

To start a discussion, add your first contact by tapping the button below.

vs


16:41

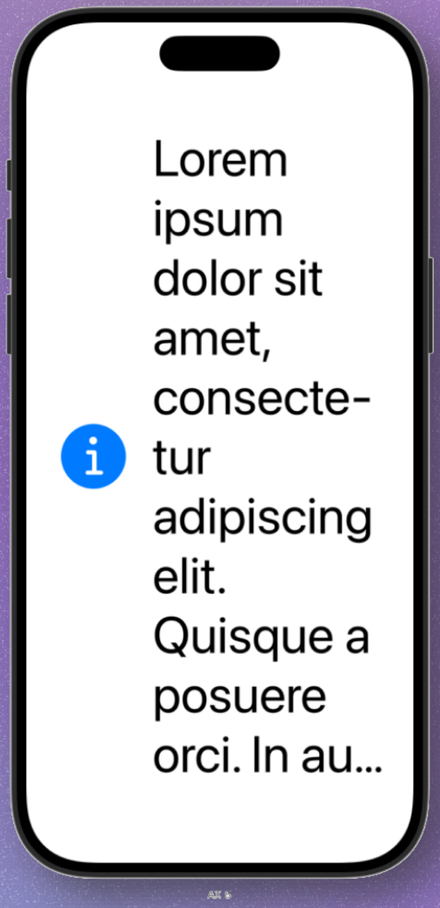
Discussions

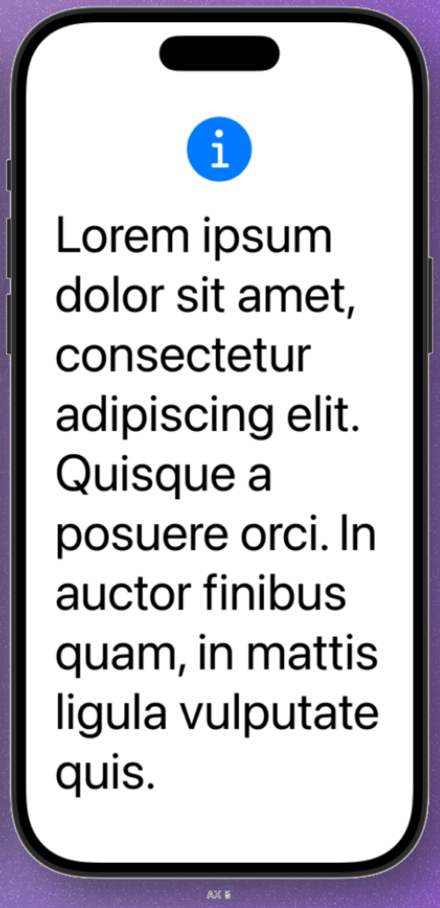
Search

No discussions for now

To start a discussion, add your first contact by tapping the button below.

The layout is responsive to custom font sizes

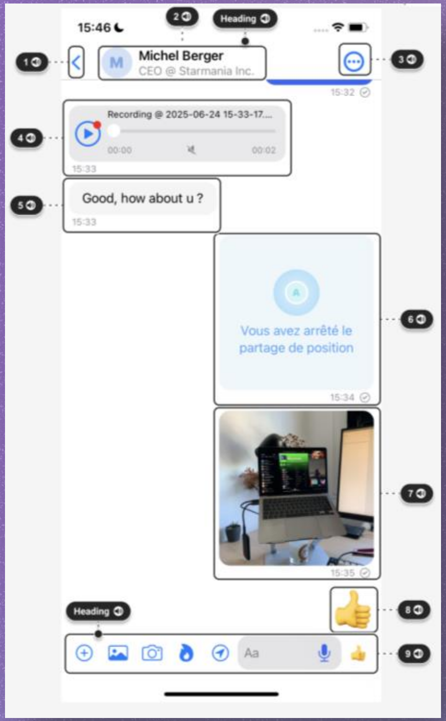

Lorem ipsum dolor sit amet, consecte-tur adipiscing elit. Quisque a posuere orci. In au...


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque a posuere orci. In auctor finibus quam, in mattis ligula vulputate quis.

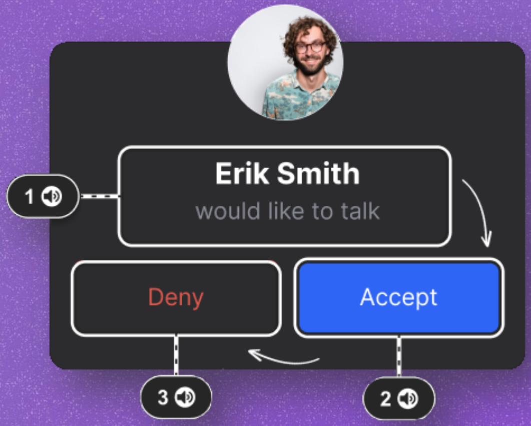
The interactions do not require physical and/or motor abilities

- Avoid **timeouts**
- Provide **shortcuts**
- Tolerate **errors**
- Protect **privacy**
- Support **assistive technologies**

The content can be accessed using assistive technologies



"Michel Berger. 3 unread messages. You said: Thanks ! at 9:07. Message status: sent. Pinned. Actions available.”



1. Grouping ---------------> 2. Labelling ---------> 3. Ordering

The UI is simple and intuitive

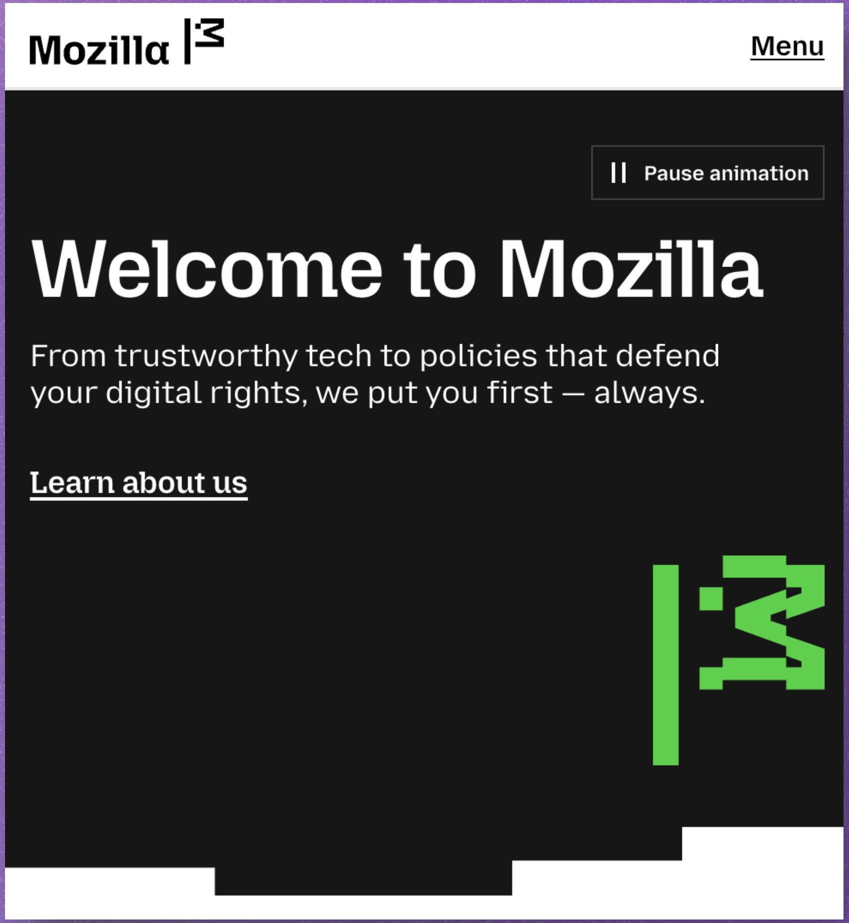
- Keep the interface as less cluttered and as consistent as possible.
- Moving or animated content should be pausable.
- Limit the number of text font and sizes used.
- Avoid pure black-on-white text.
- Body text should be naturally (not fully) justified.
- Keep written content clear, concise and free of acronyms or idioms.
- Provide alternatives to text (with images, graphs, screen reader support, etc.)
- Limit vivid colors, especially around content.
- "Don't insist on accuracy for written input. Instead, provide options." Depending on the platform, implementing auto-completion, spell-checkers, dictation, etc. can be helpful.
- Avoid dark patterns, and moreover, be very clear on the consequences of the action taken by the users on your interface.

Whitaker, Rob (2020). *Developing Inclusive Mobile Apps: Building Accessible Apps for iOS and Android*, Apress Berkley.

The UI has options to prevent motion sickness

Known triggers include:

- Zooms
- Flashing or blinking
- Animations playing automatically without user interaction
- Parallax effects



Welcome to Mozilla

From trustworthy tech to policies that defend your digital rights, we put you first — always.

Learn about us

The UI adapts well to other cultures

Colors

Language

RTL support

User's characteristics are not assumed

- **Content representing people & activities**
  - Show various ethnicities, body types, *etc.*
  - Represent a good variety of relatable activities and cultural references

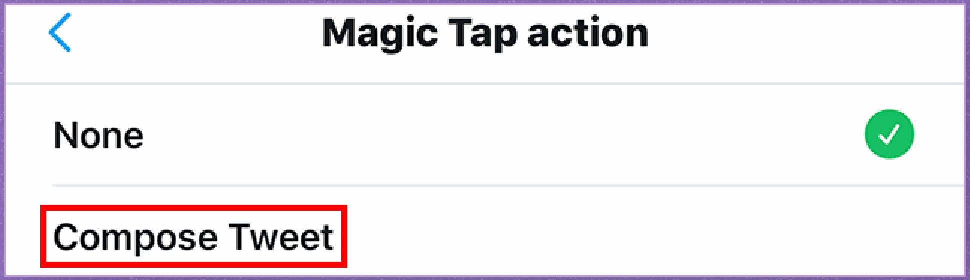
- **Gender identity & expression**
  - Provide options & custom entries
  - Avoid using gendered pronouns
  - Use gender-neutral icons
  - Offer the possibility to use a preferred name

- **Access to technology**
  - 20% of users only access the internet via their phone

- **Digital literacy**

Getting creative

Building great inclusive experiences is the occasion to thinking about new ways of interacting with your app.

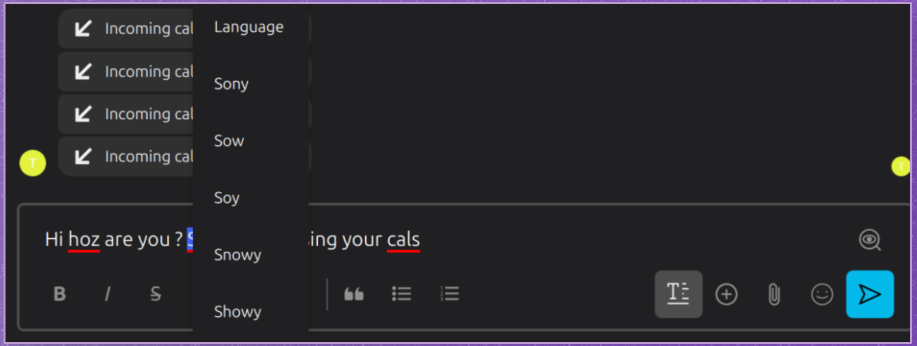


Magic Tap action

None ✓

Compose Tweet

Twitter's "Magic Tap" action settings

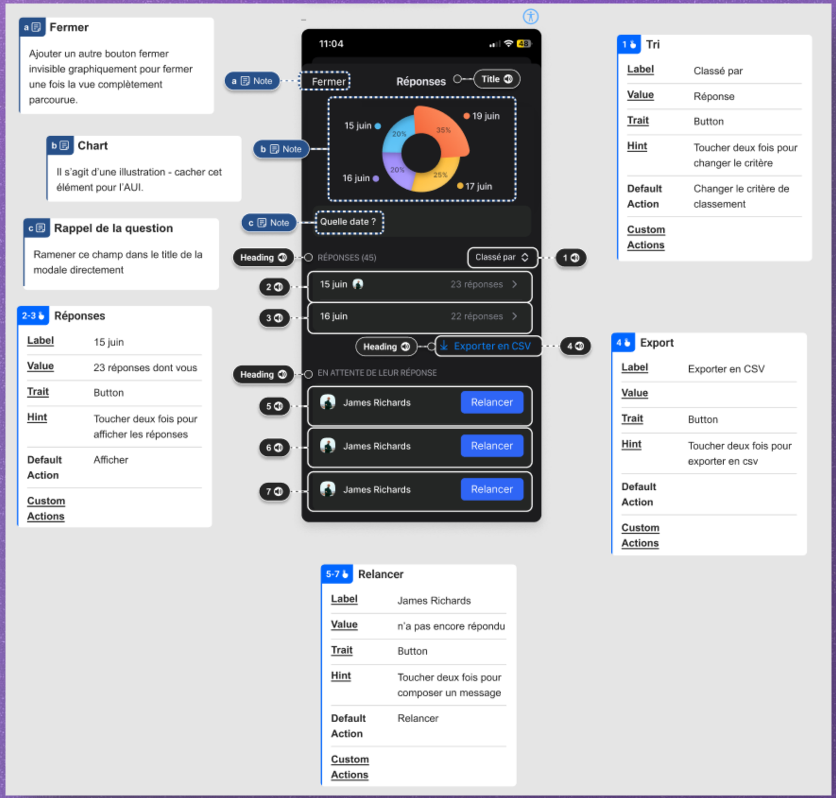


Jami's spellchecker

Communicating inclusive design

To achieve the intended IUX, communication is key:

- Communicating the accessibility of IUIs can be challenging, as accessibility features are often graphically invisible
- Use design annotations
- Test using assistive technologies and provide feedback



a Fermer

Ajouter un autre bouton fermer invisible graphiquement pour fermer une fois la vue complètement parcourue.

b Chart

Il s'agit d'une illustration - cacher cet élément pour l'AUI.

c Rappel de la question

Ramener ce champ dans le title de la modale directement

2-3 Réponses

Label	15 juin
Value	23 réponses dont vous
Trait	Button
Hint	Toucher deux fois pour afficher les réponses
Default Action	Afficher
Custom Actions	

1 Tri

Label	Classé par
Value	Réponse
Trait	Button
Hint	Toucher deux fois pour changer le critère
Default Action	Changer le critère de classement
Custom Actions	

4 Export

Label	Exporter en CSV
Value	
Trait	Button
Hint	Toucher deux fois pour exporter en csv
Default Action	
Custom Actions	

5-7 Relancer

Label	James Richards
Value	n'a pas encore répondu
Trait	Button
Hint	Toucher deux fois pour composer un message
Default Action	Relancer
Custom Actions	

Designing an IUI on a whiteboard

Design an account creation screen for a social media app.

•The interface should include:

- Fields to fill in social media information (like name, gender, hobbies, *etc.*)
- A button to send the form
- A way of going back to the home page
- A checkbox to accept the service's TOS, with a link to access them

Wrapping up

- An inclusive app is one using which users can thrive with and feel welcome in.
- Designers have a big responsibility regarding inclusiveness as they are their team’s compass towards delivering an outstanding experience for everyone.
- The IUI should not be seen as a standard UI that was made accessible, nor as a separate UI, but as an ideal to strive for, for the entire UI to be inclusive and welcoming for everyone.
- Making you interface inclusive is an occasion to get creative and to think about new innovative ways of interacting with your app.
- Building an IUI is a long process, that can start with baby steps today.

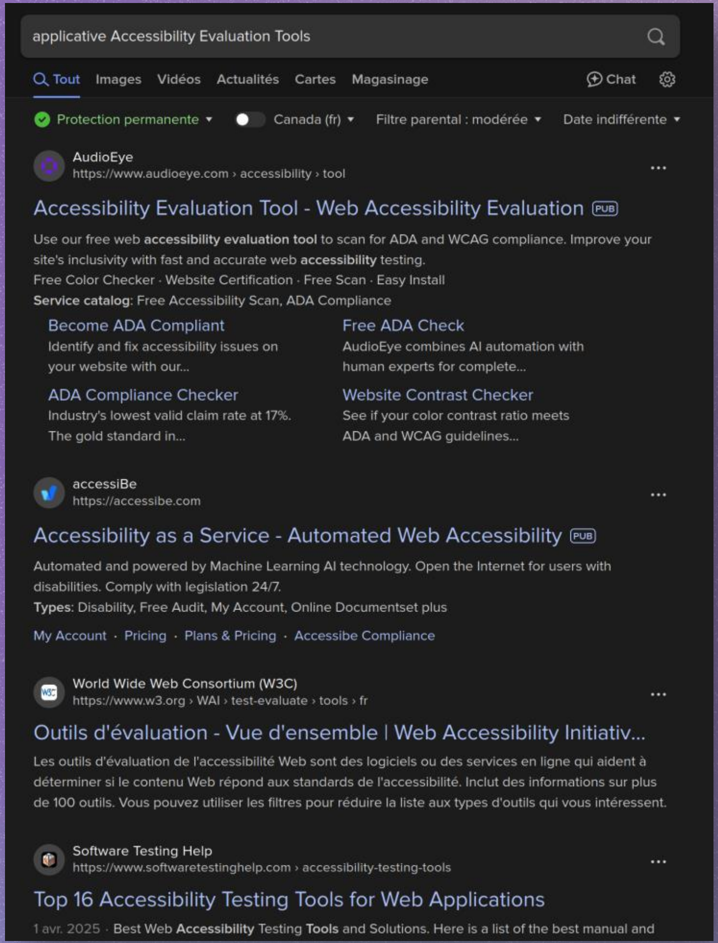
Remember: **good design is inclusive, and good design benefits everyone!**

Inclusive design for Desktop

Designing inclusive and accessible software

What are the resources ?

…and why are there so few of them ?



applicative Accessibility Evaluation Tools

Tout Images Vidéos Actualités Cartes Magasinage Chat

Protection permanente ▾ Canada (fr) ▾ Filtre parental : modérée ▾ Date indifférente ▾

AudioEye
https://www.audioeye.com › accessibility › tool

Accessibility Evaluation Tool - Web Accessibility Evaluation PUB

Use our free web **accessibility evaluation tool** to scan for ADA and WCAG compliance. Improve your site's inclusivity with fast and accurate web **accessibility** testing.
Free Color Checker · Website Certification · Free Scan · Easy Install
Service catalog: Free Accessibility Scan, ADA Compliance

Become ADA Compliant
Identify and fix accessibility issues on your website with our...

Free ADA Check
AudioEye combines AI automation with human experts for complete...

ADA Compliance Checker
Industry's lowest valid claim rate at 17%. The gold standard in...

Website Contrast Checker
See if your color contrast ratio meets ADA and WCAG guidelines...

accessiBe
https://accessibe.com

Accessibility as a Service - Automated Web Accessibility PUB

Automated and powered by Machine Learning AI technology. Open the Internet for users with disabilities. Comply with legislation 24/7.
Types: Disability, Free Audit, My Account, Online Documentset plus

My Account · Pricing · Plans & Pricing · Accessibe Compliance

World Wide Web Consortium (W3C)
https://www.w3.org › WAI › test-evaluate › tools › fr

Outils d'évaluation - Vue d'ensemble | Web Accessibility Initiativ...

Les outils d'évaluation de l'accessibilité Web sont des logiciels ou des services en ligne qui aident à déterminer si le contenu Web répond aux standards de l'accessibilité. Inclut des informations sur plus de 100 outils. Vous pouvez utiliser les filtres pour réduire la liste aux types d'outils qui vous intéressent.

Software Testing Help
https://www.softwaretestinghelp.com › accessibility-testing-tools

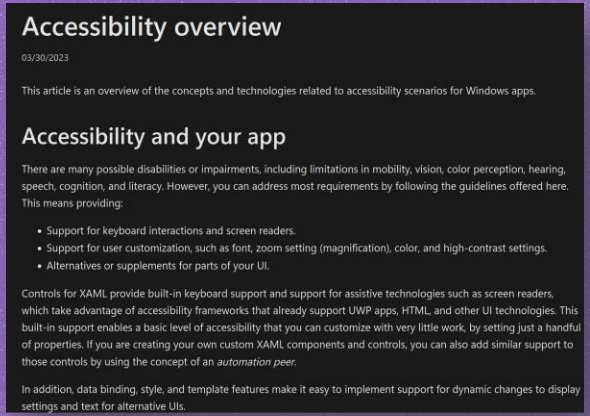
Top 16 Accessibility Testing Tools for Web Applications

1 avr. 2025 · Best Web **Accessibility** Testing **Tools** and Solutions. Here is a list of the best manual and

The number of people who need to create and deploy a website is on a completely different scale than the number of persons that need to develop a native desktop applications.

What are the resources ?

...and why are there so few of them ?



Accessibility overview

03/30/2023

This article is an overview of the concepts and technologies related to accessibility scenarios for Windows apps.

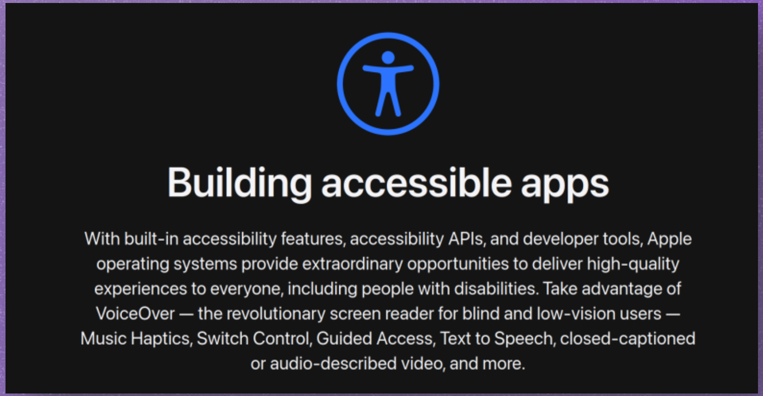
Accessibility and your app

There are many possible disabilities or impairments, including limitations in mobility, vision, color perception, hearing, speech, cognition, and literacy. However, you can address most requirements by following the guidelines offered here. This means providing:

- Support for keyboard interactions and screen readers.
- Support for user customization, such as font, zoom setting (magnification), color, and high-contrast settings.
- Alternatives or supplements for parts of your UI.

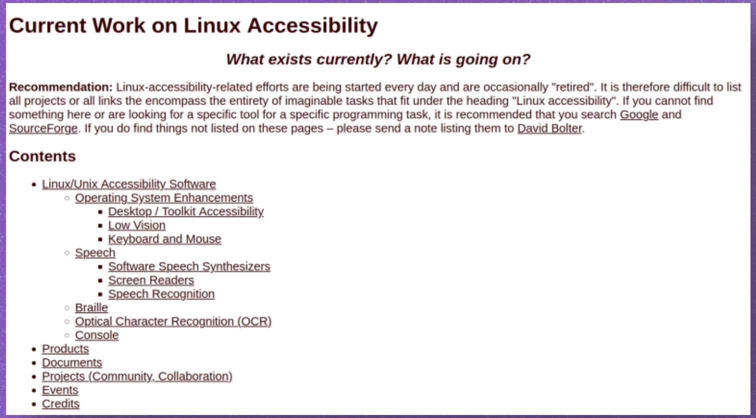
Controls for XAML provide built-in keyboard support and support for assistive technologies such as screen readers, which take advantage of accessibility frameworks that already support UWP apps, HTML, and other UI technologies. This built-in support enables a basic level of accessibility that you can customize with very little work, by setting just a handful of properties. If you are creating your own custom XAML components and controls, you can also add similar support to those controls by using the concept of an *automation peer*.

In addition, data binding, style, and template features make it easy to implement support for dynamic changes to display settings and text for alternative UIs.



Building accessible apps

With built-in accessibility features, accessibility APIs, and developer tools, Apple operating systems provide extraordinary opportunities to deliver high-quality experiences to everyone, including people with disabilities. Take advantage of VoiceOver — the revolutionary screen reader for blind and low-vision users — Music Haptics, Switch Control, Guided Access, Text to Speech, closed-captioned or audio-described video, and more.



Current Work on Linux Accessibility

What exists currently? What is going on?

Recommendation: Linux-accessibility-related efforts are being started every day and are occasionally "retired". It is therefore difficult to list all projects or all links the encompass the entirety of imaginable tasks that fit under the heading "Linux accessibility". If you cannot find something here or are looking for a specific tool for a specific programming task, it is recommended that you search Google and SourceForge. If you do find things not listed on these pages – please send a note listing them to David Bolter.

Contents

Microsoft. *Accessibility overview for Windows*
Apple. *Developing accessible apps*
JP Schnapper-Casteras & Janet Hopkins. *Current Work on Linux Accessibility*, 2004

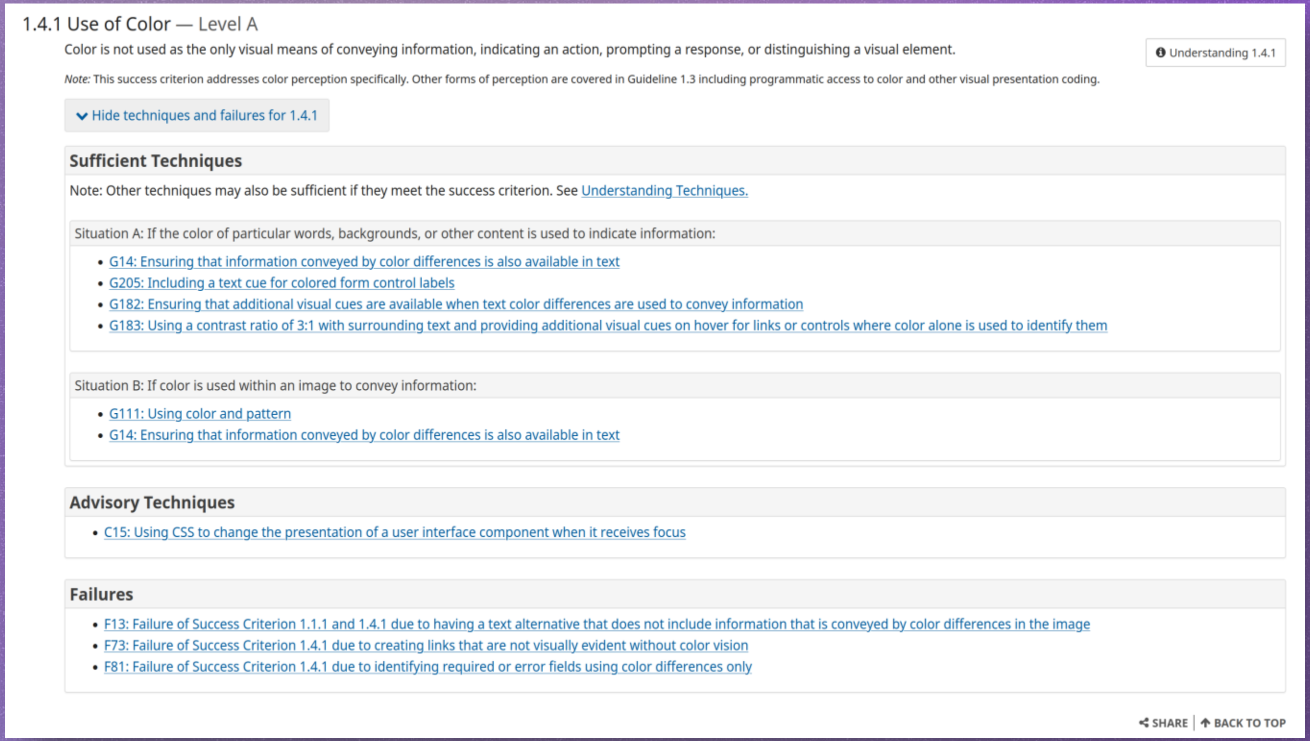
What are the resources ?

...and why are there so few of them ?

How to deal with contextual elements, different panes, complex UI elements that you typically don't find on a web page ?

❗ **Working on applicative accessibility is not memorizing a fix set of rules to apply but rather learning how to find relevant solutions to problems you encounter.**

The WCAG standard



1.4.1 Use of Color — Level A

Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

Note: This success criterion addresses color perception specifically. Other forms of perception are covered in Guideline 1.3 including programmatic access to color and other visual presentation coding.

Sufficient Techniques

Note: Other techniques may also be sufficient if they meet the success criterion. See Understanding Techniques.

Situation A: If the color of particular words, backgrounds, or other content is used to indicate information:

- G14: Ensuring that information conveyed by color differences is also available in text
- G205: Including a text cue for colored form control labels
- G182: Ensuring that additional visual cues are available when text color differences are used to convey information
- G183: Using a contrast ratio of 3:1 with surrounding text and providing additional visual cues on hover for links or controls where color alone is used to identify them

Situation B: If color is used within an image to convey information:

- G111: Using color and pattern
- G14: Ensuring that information conveyed by color differences is also available in text

Advisory Techniques

- C15: Using CSS to change the presentation of a user interface component when it receives focus

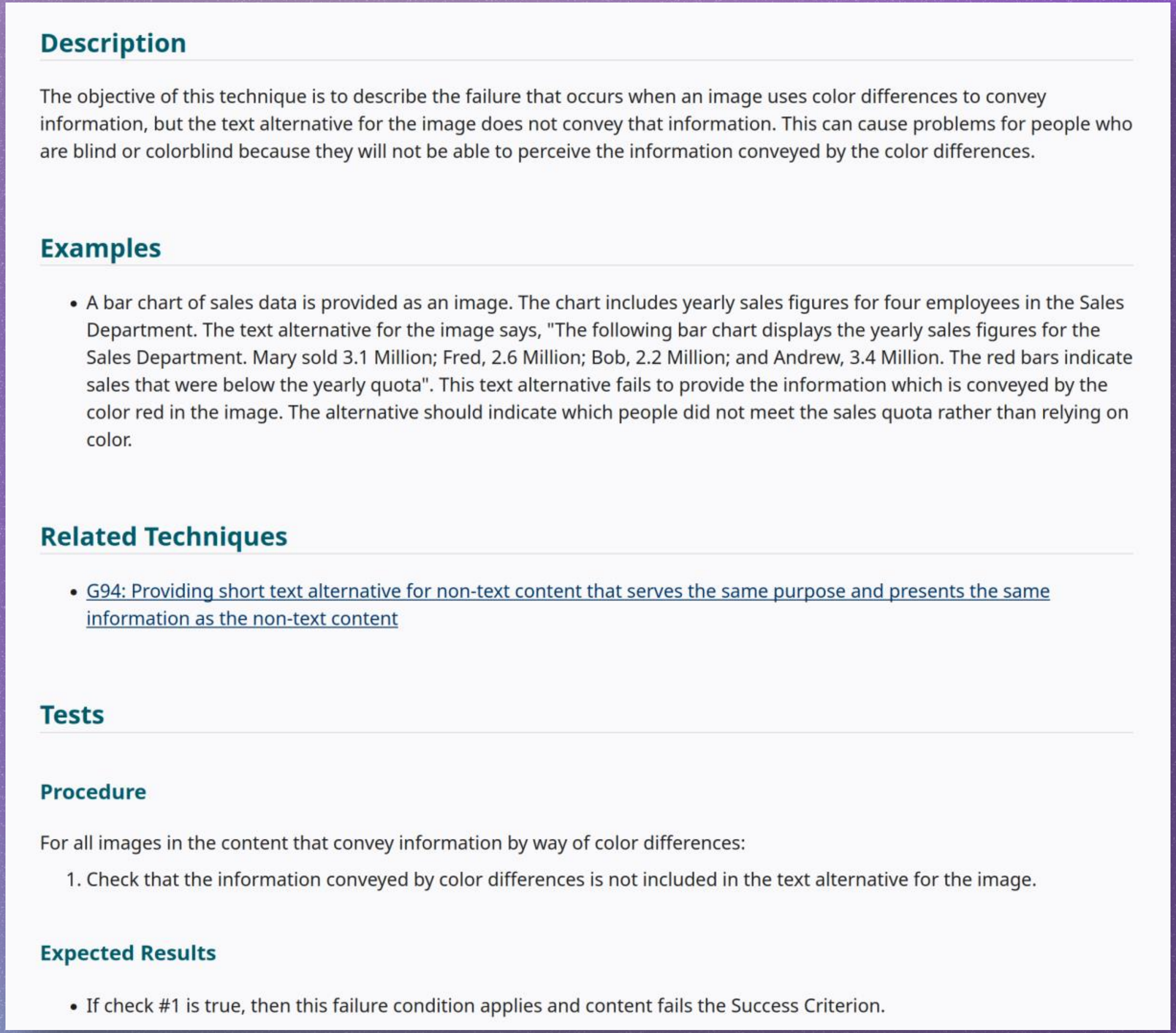
Failures

- F13: Failure of Success Criterion 1.1.1 and 1.4.1 due to having a text alternative that does not include information that is conveyed by color differences in the image
- F73: Failure of Success Criterion 1.4.1 due to creating links that are not visually evident without color vision
- F81: Failure of Success Criterion 1.4.1 due to identifying required or error fields using color differences only

The WCAG aims at providing shared worldwide accessibility guidelines for web content.

We especially recommend using the *How to Meet WCAG (Quick Reference)* from the Web Accessibility Initiative.

The WCAG standard



Description

The objective of this technique is to describe the failure that occurs when an image uses color differences to convey information, but the text alternative for the image does not convey that information. This can cause problems for people who are blind or colorblind because they will not be able to perceive the information conveyed by the color differences.

Examples

- A bar chart of sales data is provided as an image. The chart includes yearly sales figures for four employees in the Sales Department. The text alternative for the image says, "The following bar chart displays the yearly sales figures for the Sales Department. Mary sold 3.1 Million; Fred, 2.6 Million; Bob, 2.2 Million; and Andrew, 3.4 Million. The red bars indicate sales that were below the yearly quota". This text alternative fails to provide the information which is conveyed by the color red in the image. The alternative should indicate which people did not meet the sales quota rather than relying on color.

Related Techniques

- G94: Providing short text alternative for non-text content that serves the same purpose and presents the same information as the non-text content

Tests

Procedure

For all images in the content that convey information by way of color differences:

1. Check that the information conveyed by color differences is not included in the text alternative for the image.

Expected Results

- If check #1 is true, then this failure condition applies and content fails the Success Criterion.

Expected navigation with screen readers

Tab & Backtab

- The main ways of navigating trough the UI
- Pressing it should respectively send the focus to the next and previous focusable elements of the UI.
- should not be used to navigate through the elements of a list or focused static text.

The arrow keys

- The up and down arrow keys are used to navigate inside an element.
- While pressing tab would focus and then exit a list, the arrows should be used to navigate inside of the said list.

Enter

- Should act as if you were clicking on an element.
- In some contexts it can be used to close informative popups as an alternative to the escape key

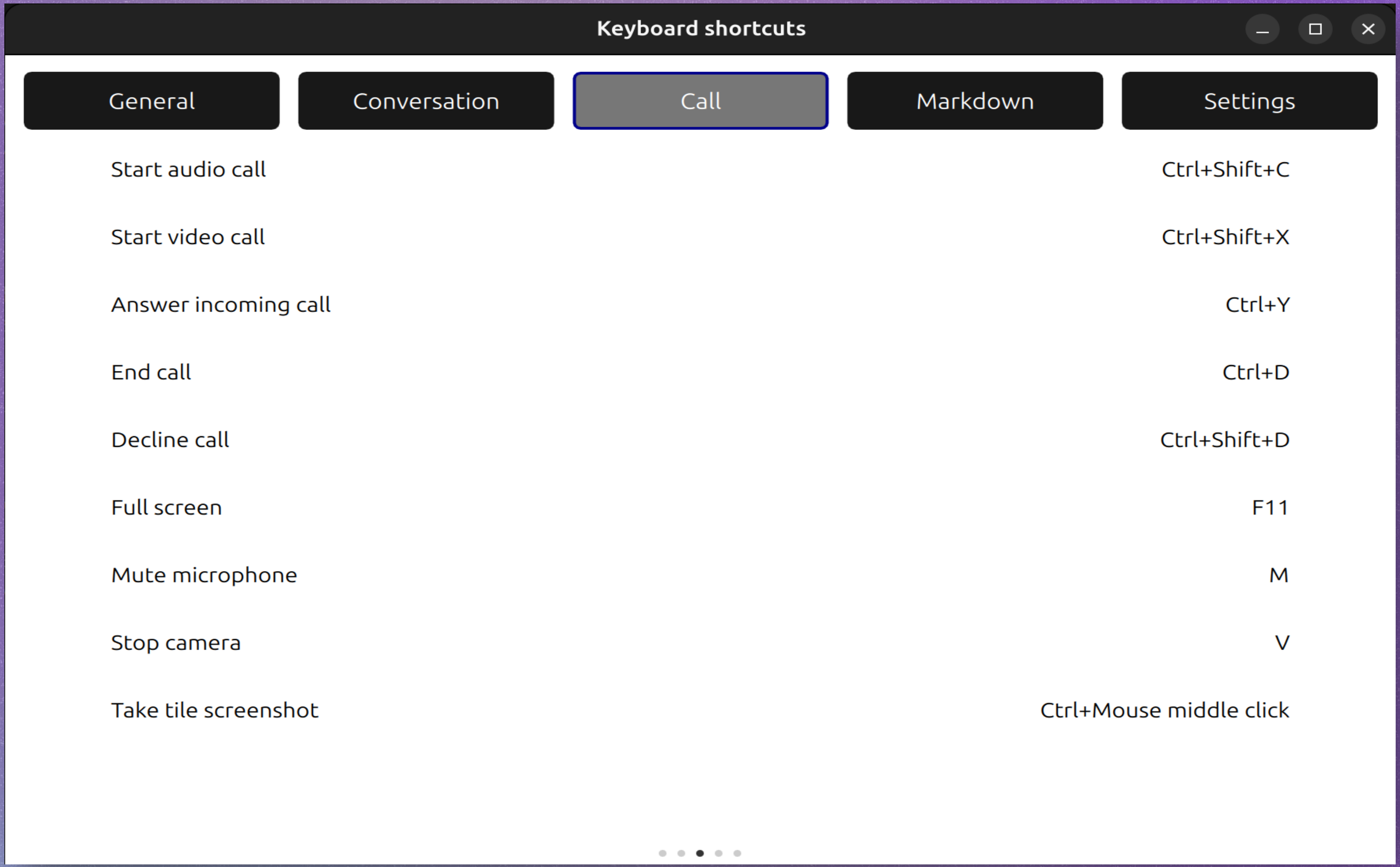
Escape

- Supposed to close popups and dropdowns.
- Having it broken is a common problem in many applications called a "focus trap"

Alt

- Firstly it should close and escape fields where tab inputs would be absorbed (some text inputs for example).
- Secondly it should focus the menu attached to the top of an application. Here is an example in VSCode.

Shortcuts



Start audio call	Ctrl+Shift+C
Start video call	Ctrl+Shift+X
Answer incoming call	Ctrl+Y
End call	Ctrl+D
Decline call	Ctrl+Shift+D
Full screen	F11
Mute microphone	M
Stop camera	V
Take tile screenshot	Ctrl+Mouse middle click

Inclusive design for Mobile

Designing inclusive and accessible software

Mobile accessibility and inclusiveness are critical

1. **Mobile devices are protheses**

2. **Mobile devices are used in a great variety of contexts**

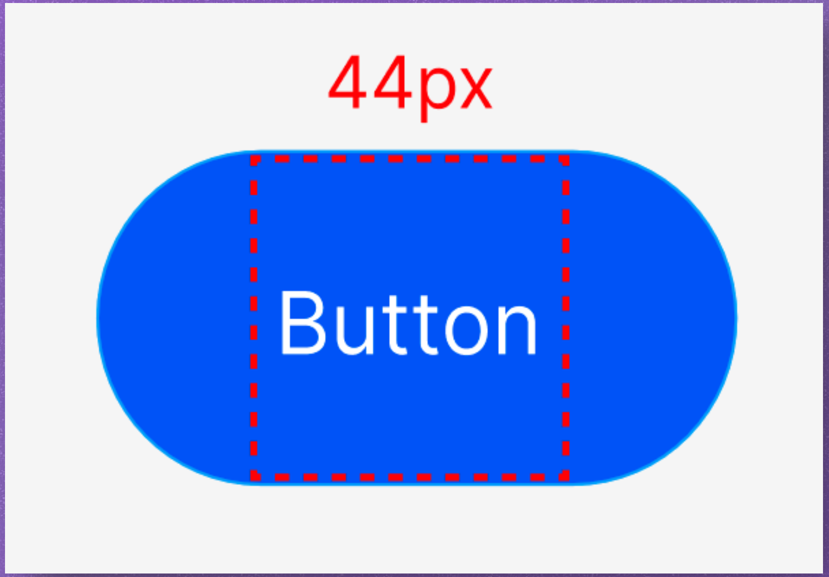
3. **Mobile devices are some people's only access to your platform** (20% of Internet users only had a smartphone to access the Internet)

Whitaker, Rob (2020). *Developing Inclusive Mobile Apps: Building Accessible Apps for iOS and Android*, Apress Berkley.

Specificities of mobile accessibility and inclusiveness

Gestures

- Provide sufficient and consistent padding for **touch targets.**
- Support alternative gestures so that your users have different ways of executing an action.
- Integrate with (or even implement) tools allowing for shortcuts, automation and voice assistants.
- Allow users to use screen readers, switches, keyboards, etc.



A button with a text of size 12.

Vocabulary of mobile accessible design

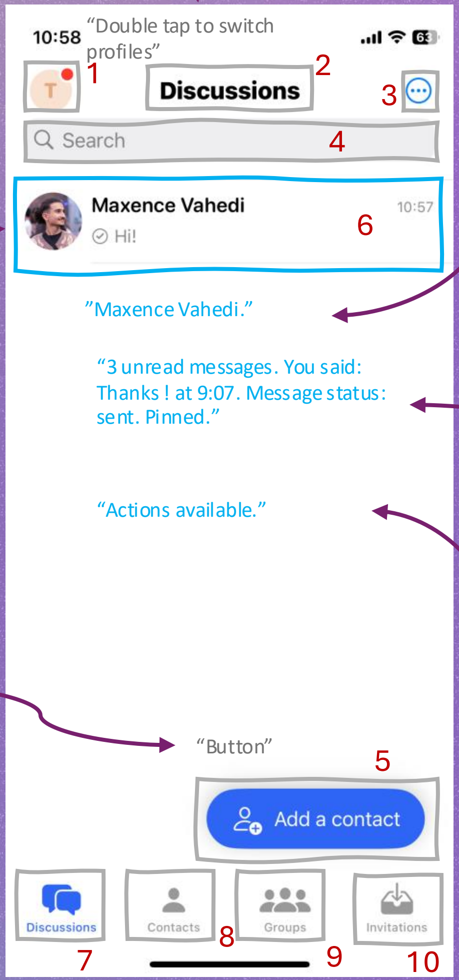
Hint

Accessibility group

- An element which will be accessed atomically
- For example, a button and its title should grouped as they share the same semantics.

Trait

- Qualifies an element on its types or roles
- Examples: "Button", "Static text", "Toggle", "Allows direct interaction"



Label

- Qualifies the name of an element
- Should be as short as possible and reflect visual UI's text

Value

- Information about the element, like its state or any other information that is not in the label

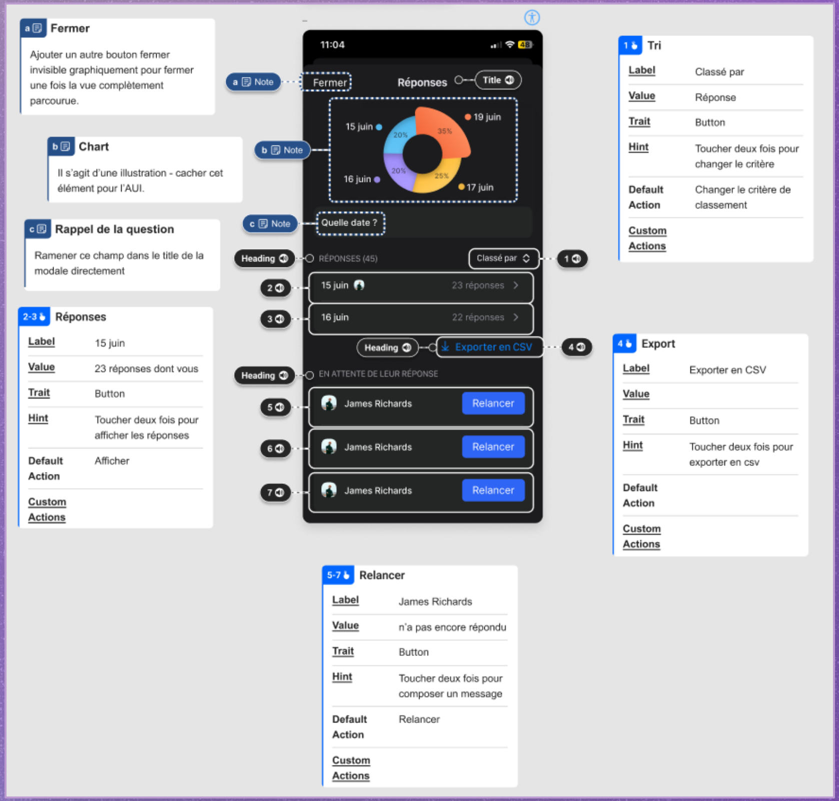
Action

- An action the user can take on an element
- Either be primary or secondary/custom

Focus order

- Allows to specify in which order the elements of your UI should be accessed
- Always indicate it in your designs

Incorporating accessibility to your design files



a Fermer

Ajouter un autre bouton fermer invisible graphiquement pour fermer une fois la vue complètement parcourue.

b Chart

Il s'agit d'une illustration - cacher cet élément pour l'AUI.

c Rappel de la question

Ramener ce champ dans le title de la modale directement

2-3 Réponses

Label	15 juin
Value	23 réponses dont vous
Trait	Button
Hint	Toucher deux fois pour afficher les réponses
Default Action	Afficher
Custom Actions	

1 Tri

Label	Classé par
Value	Réponse
Trait	Button
Hint	Toucher deux fois pour changer le critère
Default Action	Changer le critère de classement
Custom Actions	

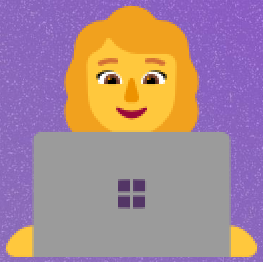
4 Export

Label	Exporter en CSV
Value	
Trait	Button
Hint	Toucher deux fois pour exporter en csv
Default Action	
Custom Actions	

5-7 Relancer

Label	James Richards
Value	n'a pas encore répondu
Trait	Button
Hint	Toucher deux fois pour composer un message
Default Action	Relancer
Custom Actions	

You can find online many free accessibility annotation kits made by the Figma community.



Inclusive code 101

Programming inclusive and accessible software

Accessible code is good code

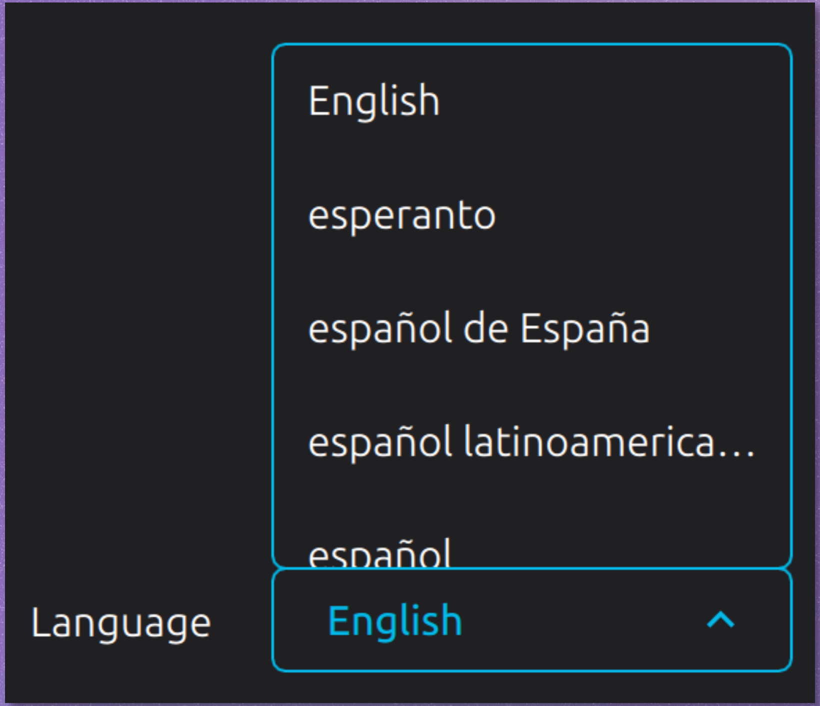
Custom components and accessibility

- **Bad accessibility can often be a symptom of flawed code.**
- You code can be good but will likely never be as robust as one iterated upon by a community of developers or a big company.

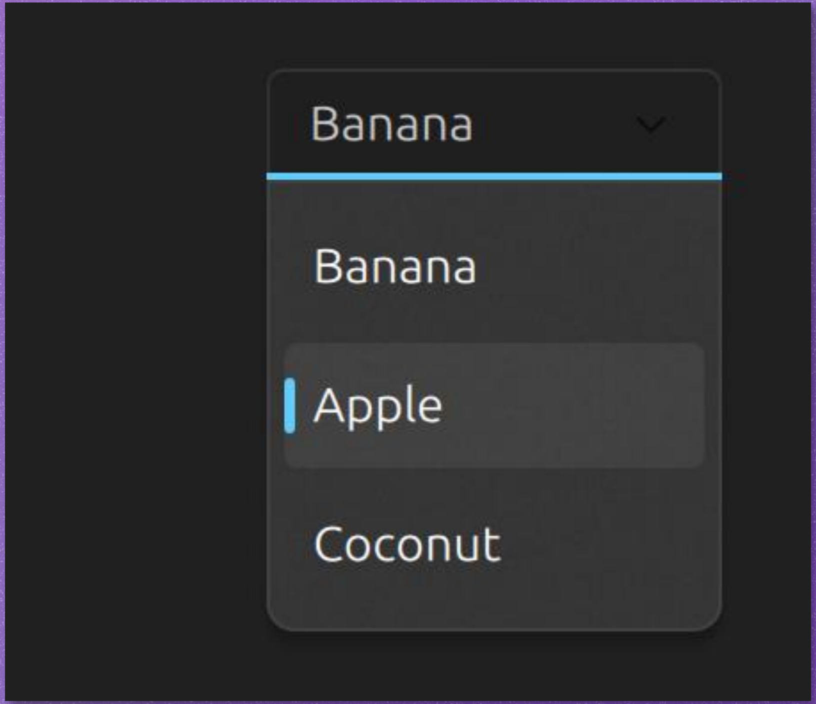
Don't reinvent (or program in our case) the wheel!

Accessible code is good code

Custom components and accessibility



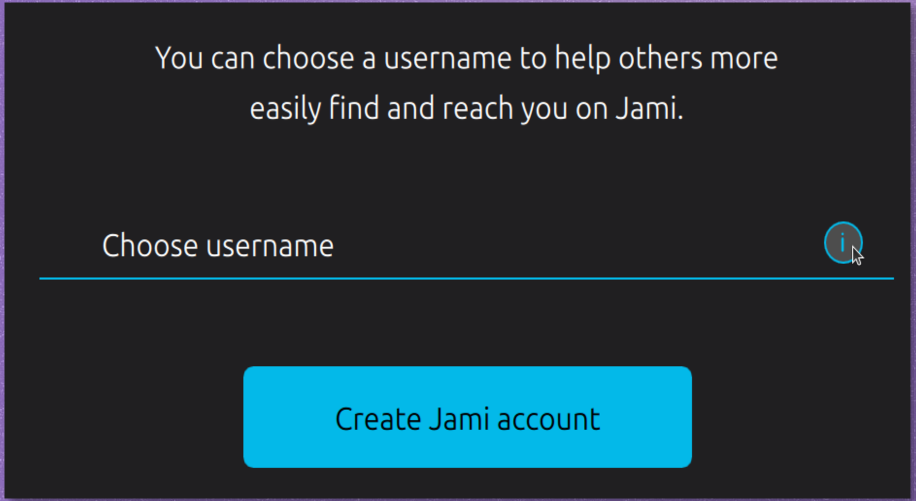
Custom component

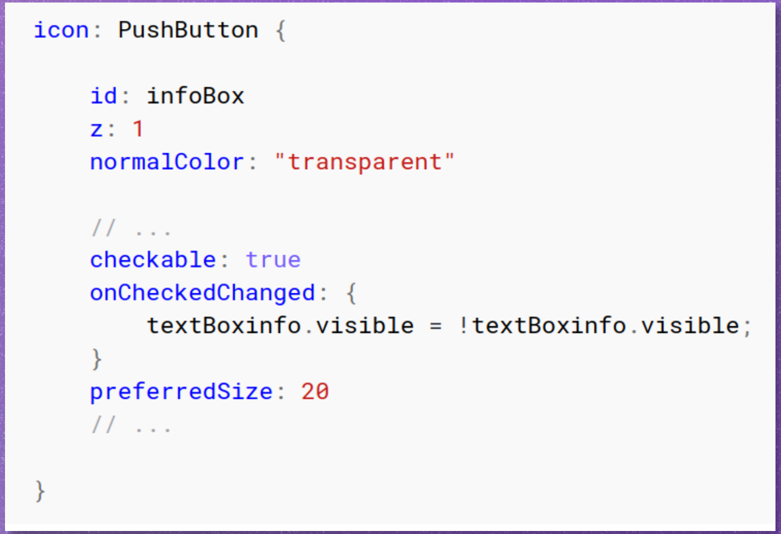


Native component

Accessible code is good code

What bad accessibility says about our code: a few concrete examples

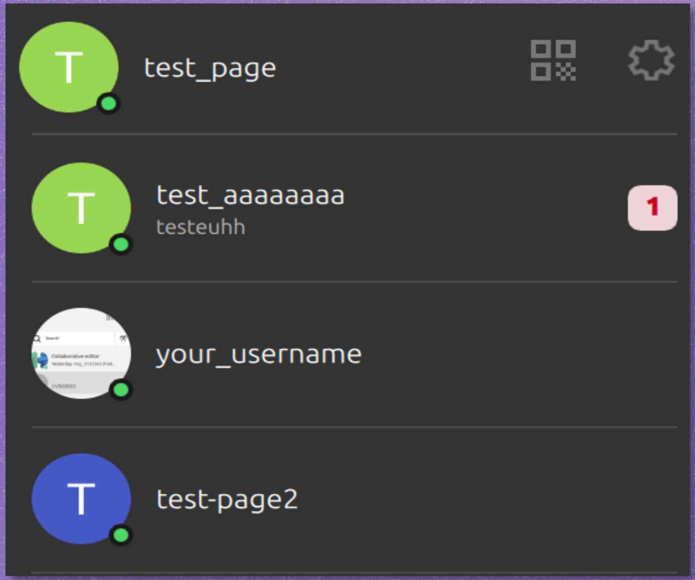

You can choose a username to help others more easily find and reach you on Jami.


```
icon: PushButton {

    id: infoBox
    z: 1
    normalColor: "transparent"

    // ...
    checkable: true
    onCheckedChanged: {
        textBoxinfo.visible = !textBoxinfo.visible;
    }
    preferredSize: 20
    // ...

}
```

Accessibility and inclusiveness in programming have one key benefit: **making you think twice about your code**. As said previously, stumbling into an accessibility issue might mean that you could be doing something better in a different way.

Accessible code is good code

What bad accessibility says about our code: a few concrete examples



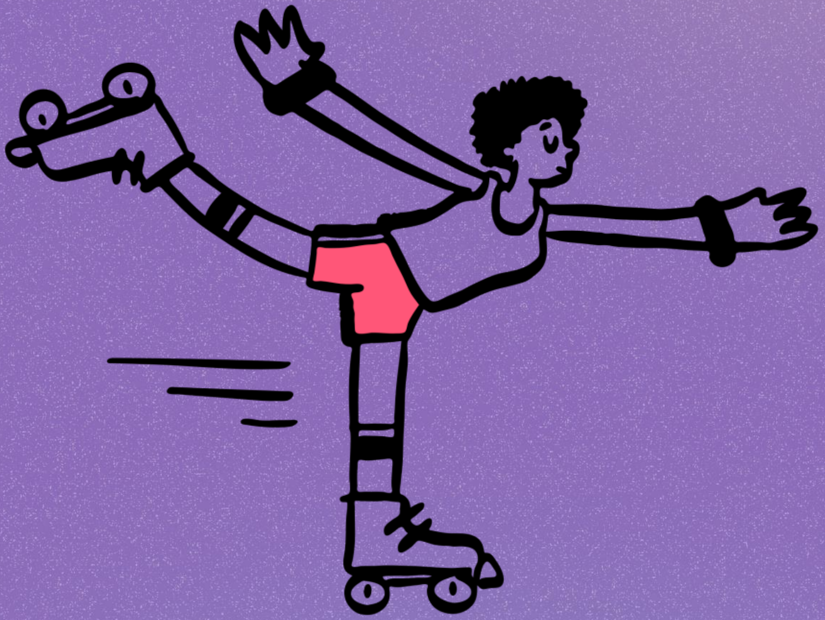
```
Keys.onUpPressed: verticalScrollBar.decrease()
Keys.onDownPressed: verticalScrollBar.increase()
```

Accessibility and inclusiveness in programming have one key benefit:
making you think twice about your code.

Accessible code is good code

Focus and explicit key navigation (Desktop)

Keyboard navigation breaks very often.

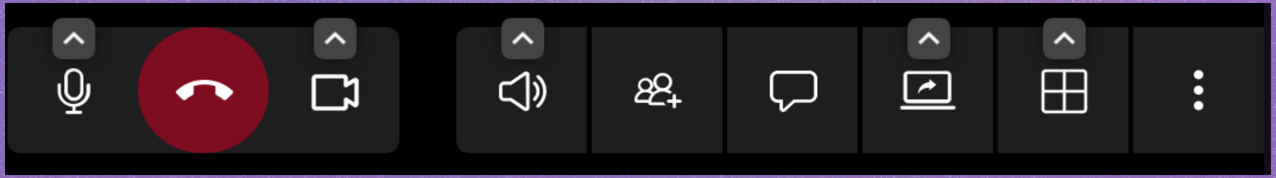


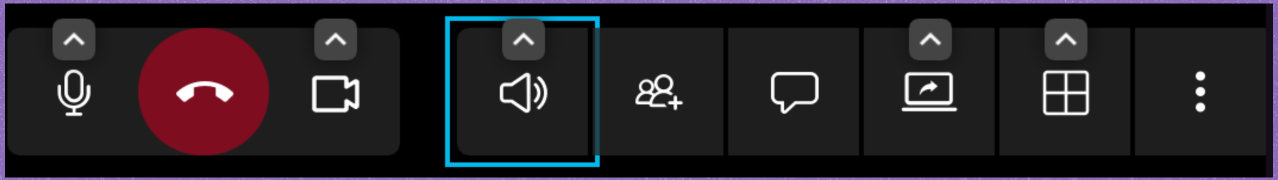
- Try and make it focusable so that your framework can automatically incorporate it in its navigation.
- You could want to use explicit keyboard navigation, to force a component to send you to another component that you has chosen.

```
KeyNavigation.tab: isTheExtendedViewLoaded()? settingsButton : moreInfosButton
KeyNavigation.backtab: newAccountButton
KeyNavigation.up: isTheExtendedViewLoaded()? fromBackupButton : alreadyHaveAccount
KeyNavigation.down: KeyNavigation.tab
```

Accessible code is good code

Fading components





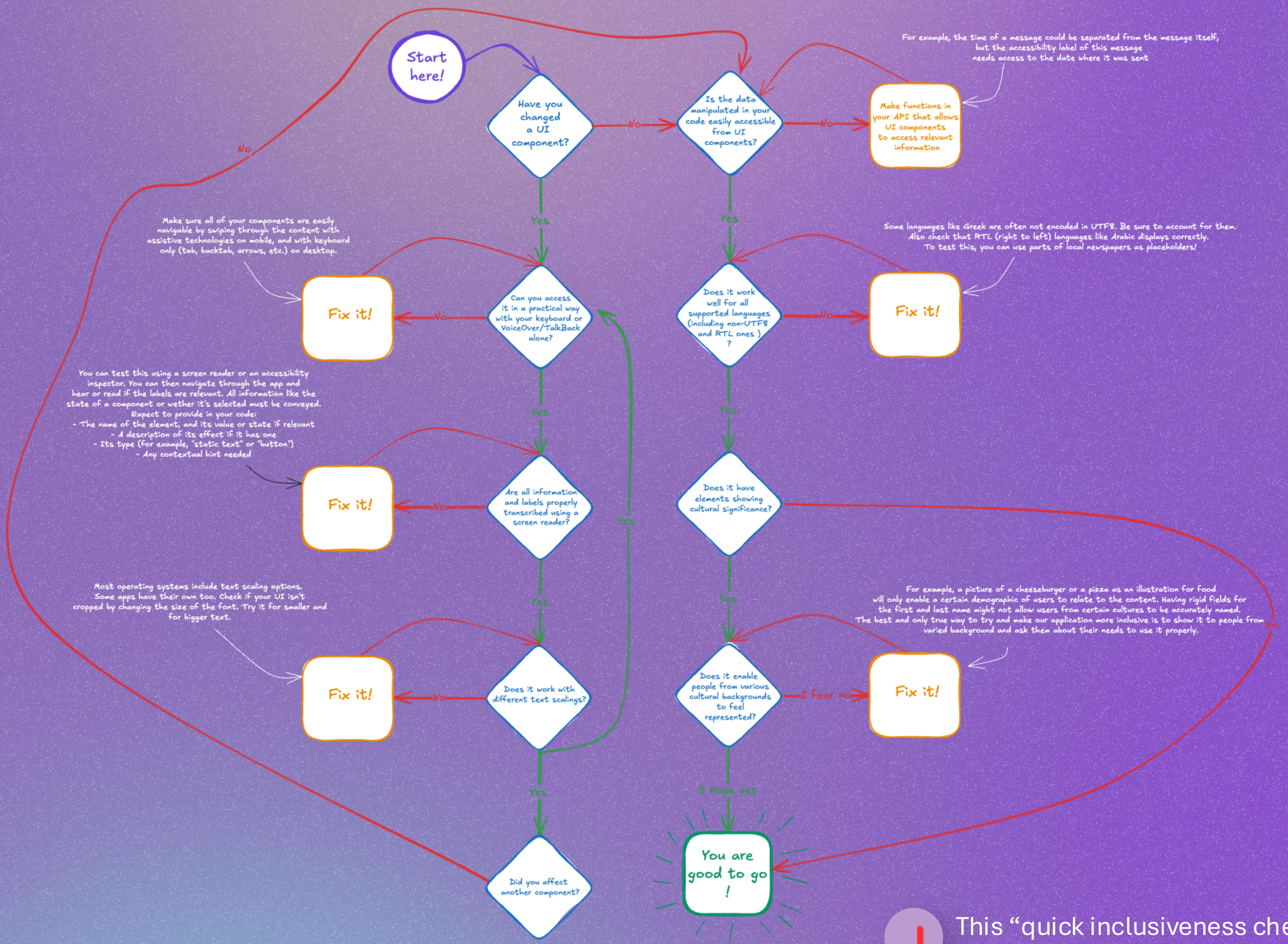
1. Any navigation key press should make the fading component visible again
2. The component shouldn't fade if it is focused, or if one of its children items is.
3. The selected element should be clearly identifiable by the user

Development process

What can be automated?

- CI can detect missing accessibility labels and unreachable elements.
- Some frameworks (including SwiftUI and Jetpack Compose) provide ways of automating UI tests that check accessibility.

No amount of checklists and tests in CI can replace feedback and testing from diverse user groups.


Start here!
Have you changed a UI component?
No
Yes
Is the data manipulated in your code easily accessible from UI components?
No
Make functions in your API that allows UI components to access relevant information
Yes
Can you access it in a practical way with your keyboard or VoiceOver/TalkBack alone?
No
Fix it!
Yes
Are all information and labels properly transcribed using a screen reader?
No
Fix it!
Yes
Does it work with different text scalings?
No
Fix it!
Yes
Did you affect another component?
No
Yes
Does it work well for all supported languages (including non-UTF8 and RTL ones) ?
No
Fix it!
Yes
Does it have elements showing cultural significance?
No
Yes
Does it enable people from various cultural backgrounds to feel represented?
I fear no
Fix it!
I hope yes
You are good to go !

For example, the time of a message could be separated from the message itself, but the accessibility label of this message needs access to the date where it was sent

Make sure all of your components are easily navigable by swiping through the content with assistive technologies on mobile, and with keyboard only (tab, backtab, arrows, etc.) on desktop.

Some languages like Greek are often not encoded in UTF8. Be sure to account for them. Also check that RTL (right to left) languages like Arabic displays correctly. To test this, you can use parts of local newspapers as placeholders!

You can test this using a screen reader or an accessibility inspector. You can then navigate through the app and hear or read if the labels are relevant. All information like the state of a component or wether it's selected must be conveyed.
Expect to provide in your code:
- The name of the element, and its value or state if relevant
- A description of its effect if it has one
- Its type (for example, "static text" or "button")
- Any contextual hint needed

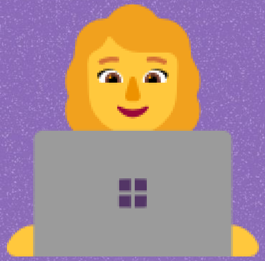
Most operating systems include text scaling options. Some apps have their own too. Check if your UI isn't cropped by changing the size of the font. Try it for smaller and for bigger text.

For example, a picture of a cheeseburger or a pizza as an illustration for food will only enable a certain demographic of users to relate to the content. Having rigid fields for the first and last name might not allow users from certain cultures to be accurately named. The best and only true way to try and make our application more inclusive is to show it to people from varied background and ask them about their needs to use it properly.

This “quick inclusiveness check” only aims at checking you are not breaking any pre-existing accessibility.

Development process

Accessible API's & back-end



- **Naming conventions:**
  - Avoid abbreviations, which can be difficult to understand to some people due to language, screen reader use, or cognitive disabilities.
  - Make sure the name is as clear and complete as possible.
- **Documenting and commenting:**
  - Write clear sentences with simple language that will enable everyone to understand the purpose of your code.
  - If your documentation is built to HTML, check with the WAVE tool that it is WCAG-compliant.

Inclusive features and programming for Apple platforms

Programming inclusive and accessible software

Apple's accessibility model

Hint

Accessibility Element

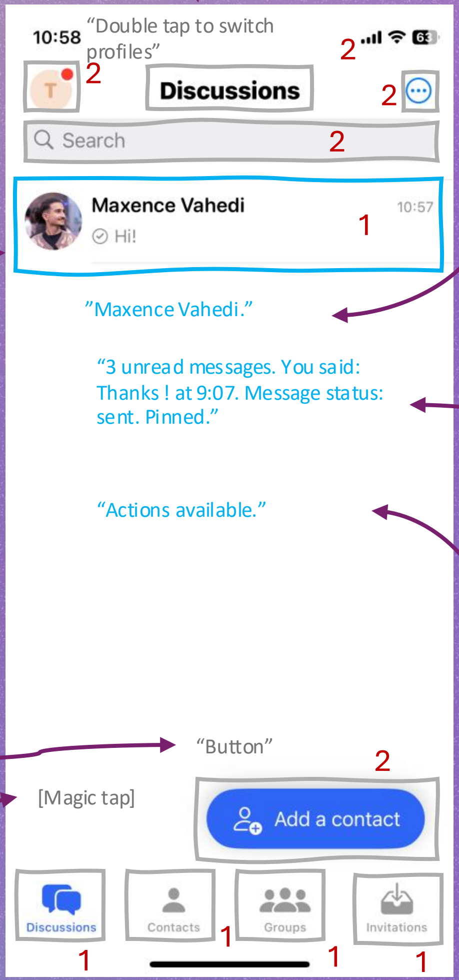
- Every UI Component provided by Apple is an accessibility element
- Will be accessed atomically (i.e. without dividing it into multiple elements) by cursor interfaces (like VoiceOver or Switch Control

Trait

- Qualifies an element on its types or roles
- Examples: "Button", "Static text", "Toggle", "Allows direct interaction"

Custom content/gestures

- Additional information and customized gestures can be implemented



Label

- Qualifies the name of an element
- Should be as short as possible and reflect visual UI's text

Value

- Information about the element, like its state or any other information that is not in the label

Action

- An action the user can take on an element
- **Primary:** function `() -> Void` that will be triggered by a double-tap in VoiceOver, or a tap with Switch Control.
- **Secondary:** `[String: () -> Void]`

Sort priority

- Value given to prioritize sorting

VoiceOver

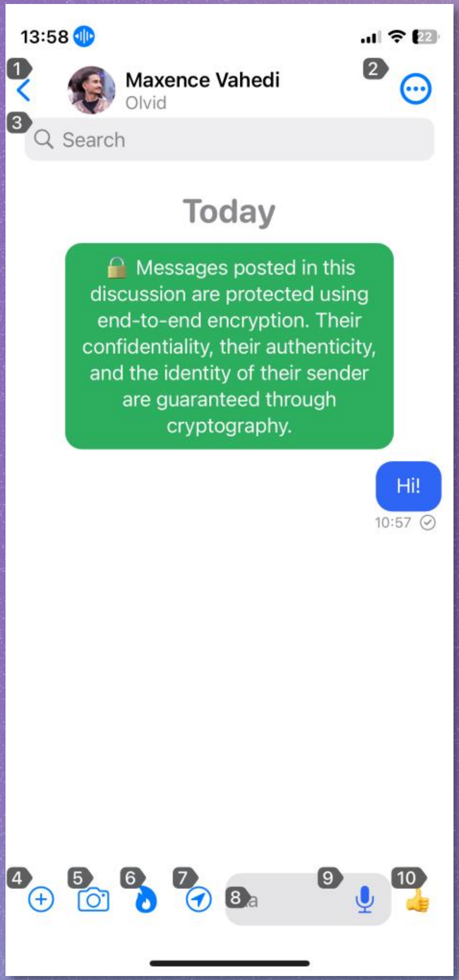
Apple's screen reader

- **Gesture-based** screen reader and interface for blind and low-vision users.
- Best way to test your app's accessibility for assistive technologies.
- Supporting custom rotors actions is a great way to provide quick app-specific actions that go beyond the bare necessity.

Learning how to use VoiceOver is crucial to making your app accessible to everyone!

Voice control

Interacting through voice



🔒 Messages posted in this discussion are protected using end-to-end encryption. Their confidentiality, their authenticity, and the identity of their sender are guaranteed through cryptography.

Hi!

10:57

- Allows people with physical or motor disabilities to interact with their iPhones without touching the screen.

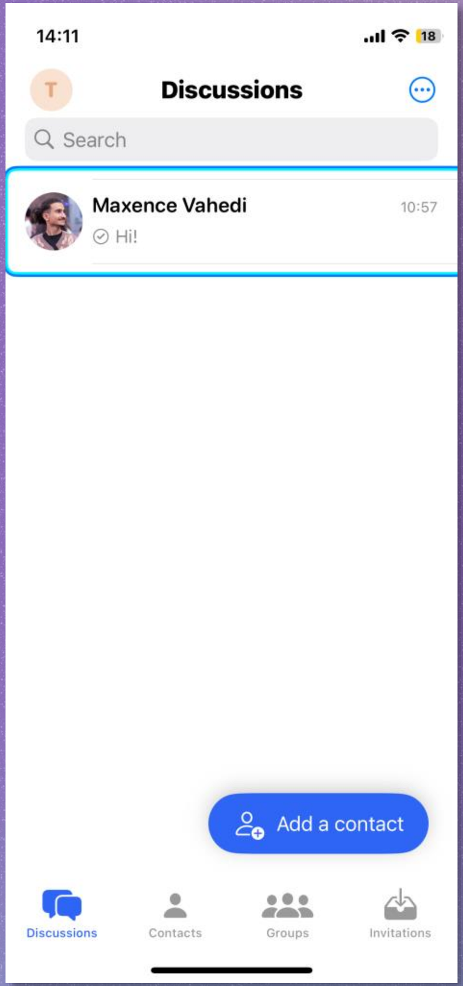
- Users can perform any action as long as it is exposed to assistive technologies as with VoiceOver.

- Selecting elements is done by either:
  - Saying the <u>label</u> of the elements (make it short and consistent with on-screen text)
  - Saying the number of the elements

- Make sure actionable the elements are marked as such

Switch Control

Interacting through external devices

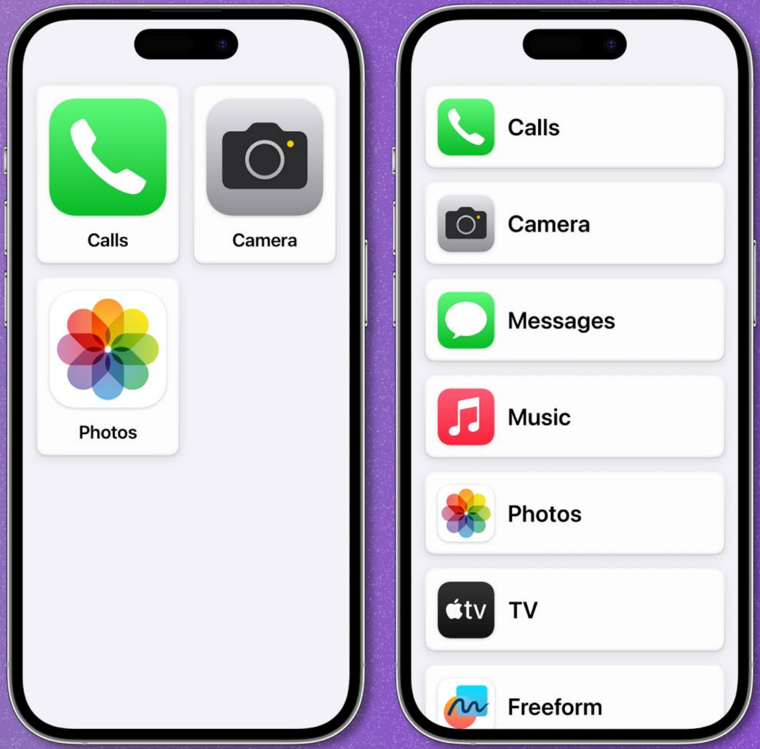


- Lets users navigate your app with the switch devices of their choice, and/or mouth gestures
- Clear and efficient navigation and grouping is crucial to making it usable
- In particular, the order of the elements is crucial

Assistive Access

Interacting through voice

- Provide a minimal experience for people with certain cognitive disabilities or who are less familiar with new technologies
- OS-wide feature, app specific implementations
- **Guided access** allows these users to get a tailored experience with custom restrictions, allowing to disable keyboard inputs, volume buttons, *etc*



Accessibility in SwiftUI

The ViewModifier approach

- Accessibility in SwiftUI relies heavily on accessibility modifiers.
- They are ViewModifiers altering the accessibility tree's structure and nodes of your app.
- Their list can be very inspiring!

Accessibility in SwiftUI

Grouping elements

```
var body: some View {
    Group {
        NiceView()
        VeryNiceView()
    }
    .accessibilityElement(children: <#T##AccessibilityChildBehavior#>)
}
```

As said earlier, there are different ways of grouping elements, each corresponding to a case of the `AccessibilityChildBehavior` enum.

Accessibility in SwiftUI

The ViewModifier approach

.COMBINE

creates a new parent accessibility element, drops its children elements from the accessibility tree, and computes new accessibility attributes.

Useful for very simple combinations

.IGNORE

creates new accessibility element, also drop its children, but don't compute any accessibility attribute for free.

Good practice, nudges intentionality

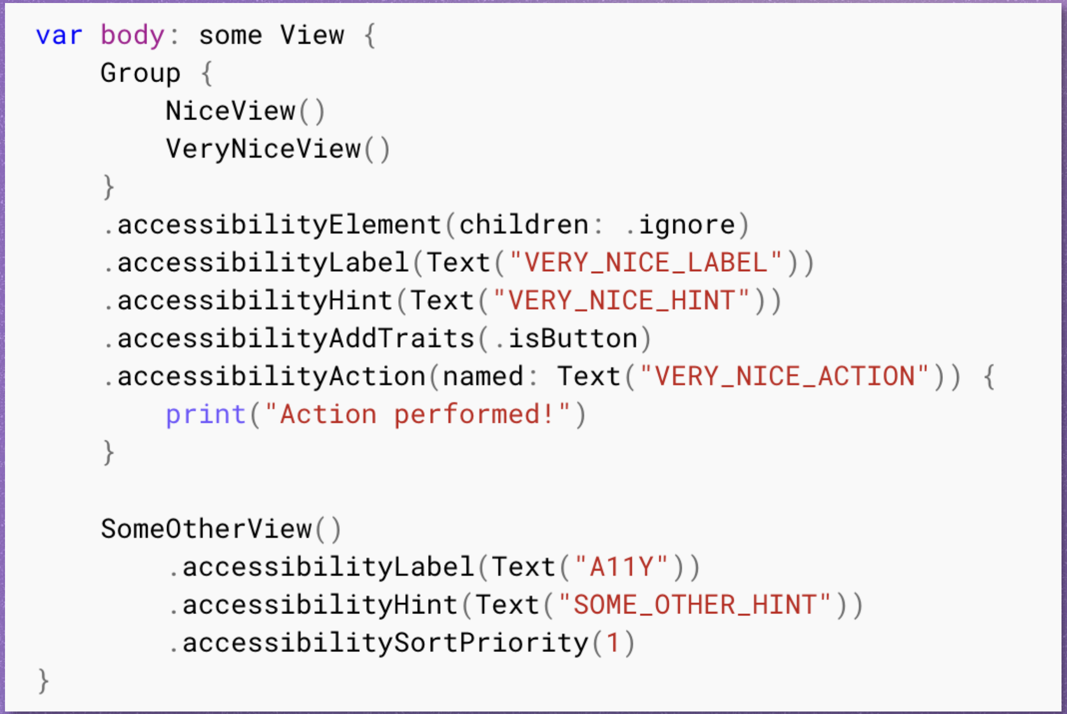
.CONTAIN

creates an accessibility element that isn't a leaf on the tree, i.e. that won't be directly accessed.

Useful for collections of items

Accessibility in SwiftUI

Limits to the ViewModifier approach



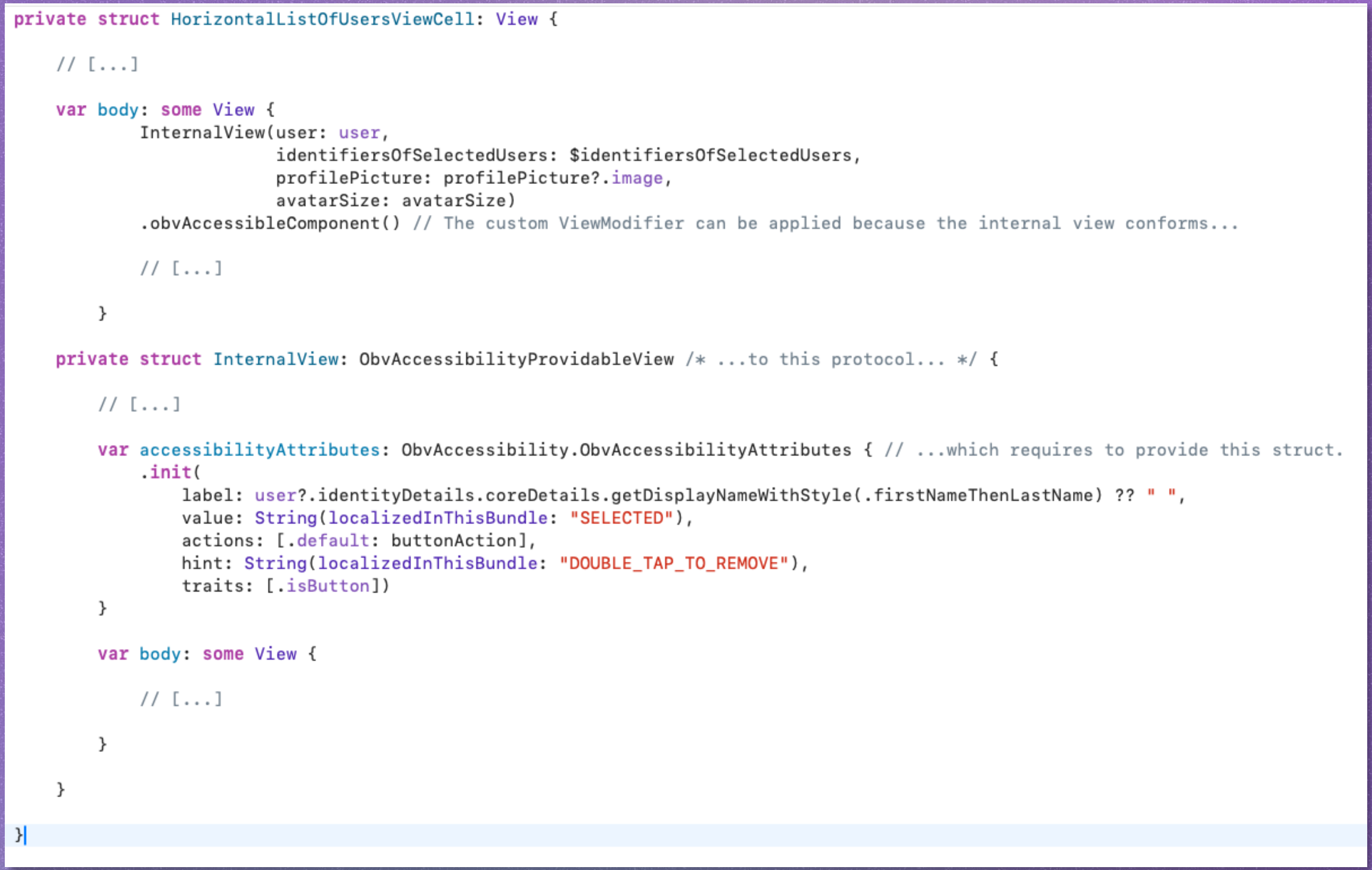
```swift
var body: some View {
    Group {
        NiceView()
        VeryNiceView()
    }
    .accessibilityElement(children: .ignore)
    .accessibilityLabel(Text("VERY_NICE_LABEL"))
    .accessibilityHint(Text("VERY_NICE_HINT"))
    .accessibilityAddTraits(.isButton)
    .accessibilityAction(named: Text("VERY_NICE_ACTION")) {
        print("Action performed!")
    }

    SomeOtherView()
        .accessibilityLabel(Text("A11Y"))
        .accessibilityHint(Text("SOME_OTHER_HINT"))
        .accessibilitySortPriority(1)
}
```

- The accessibility-related code is hard to find. In bigger and more complex views, it gets scattered all over the view's code.
- If same snippets are likely to be repeated everywhere.
- Yet, consistency across the codebase is not enforced, nor nudged.

We need to find a way of facilitating, structuring and harmonizing accessibility in the codebase.

Accessibility in SwiftUI

Building a custom accessibility architecture

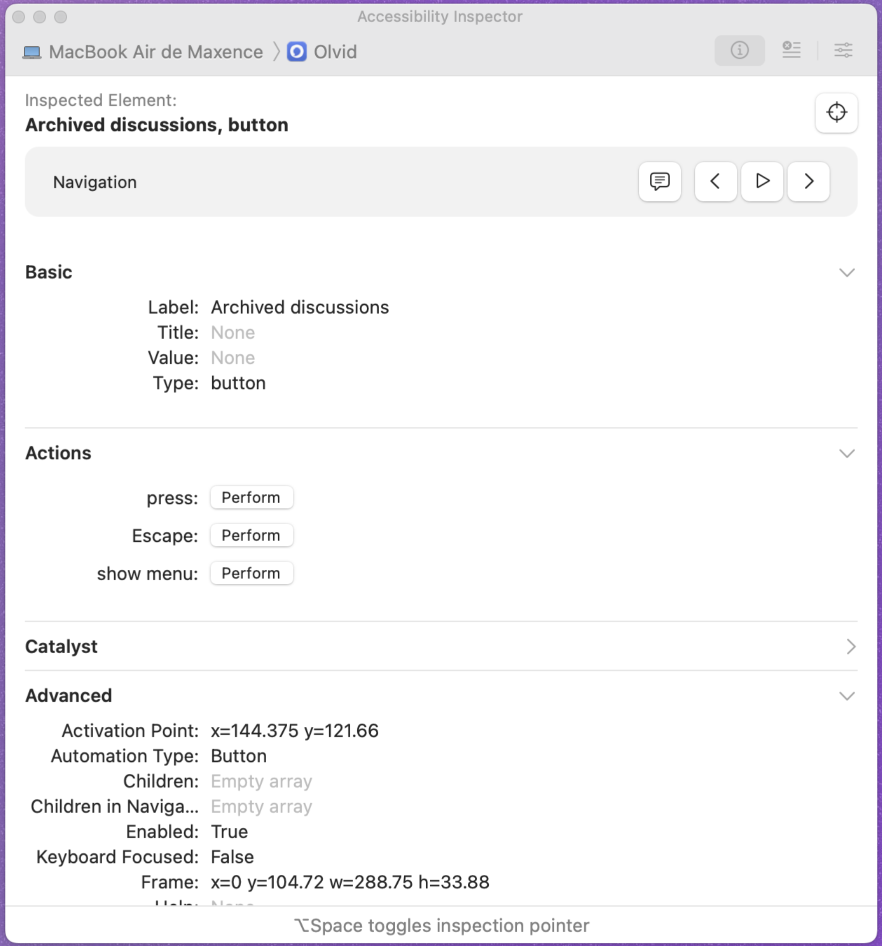


```
private struct HorizontalListOfUsersViewCell: View {

    // [...]

    var body: some View {
        InternalView(user: user,
                     identifiersOfSelectedUsers: $identifiersOfSelectedUsers,
                     profilePicture: profilePicture?.image,
                     avatarSize: avatarSize)
        .obvAccessibleComponent() // The custom ViewModifier can be applied because the internal view conforms...

        // [...]

    }

    private struct InternalView: ObvAccessibilityProvidableView /* ...to this protocol... */ {

        // [...]

        var accessibilityAttributes: ObvAccessibility.ObvAccessibilityAttributes { // ...which requires to provide this struct.
            .init(
                label: user?.identityDetails.coreDetails.getDisplayNameWithStyle(.firstNameThenLastName) ?? " ",
                value: String(localizedInThisBundle: "SELECTED"),
                actions: [.default: buttonAction],
                hint: String(localizedInThisBundle: "DOUBLE_TAP_TO_REMOVE"),
                traits: [.isButton])
        }

        var body: some View {

            // [...]

        }

    }

}
```

- **Example with Olvid's approach:**
  - A protocol requiring to provide accessibility attributes.
  - A struct giving those accessibility attributes.
  - A view modifier that can be applied on any view conforming to the protocol, applying those accessibility attributes.

Accessibility Inspector

- Inspect any accessibility element on your device, whether physical or emulated
- Very useful when developing to get a textual representation of your components' accessibility attributes
- Can also perform accessibility audits that are fast but not very insightful.


Accessibility Inspector

MacBook Air de Maxence 〉 Olvid

Inspected Element:
Archived discussions, button

Navigation

Basic

Label:	Archived discussions
Title:	None
Value:	None
Type:	button

Actions

press:	Perform
Escape:	Perform
show menu:	Perform

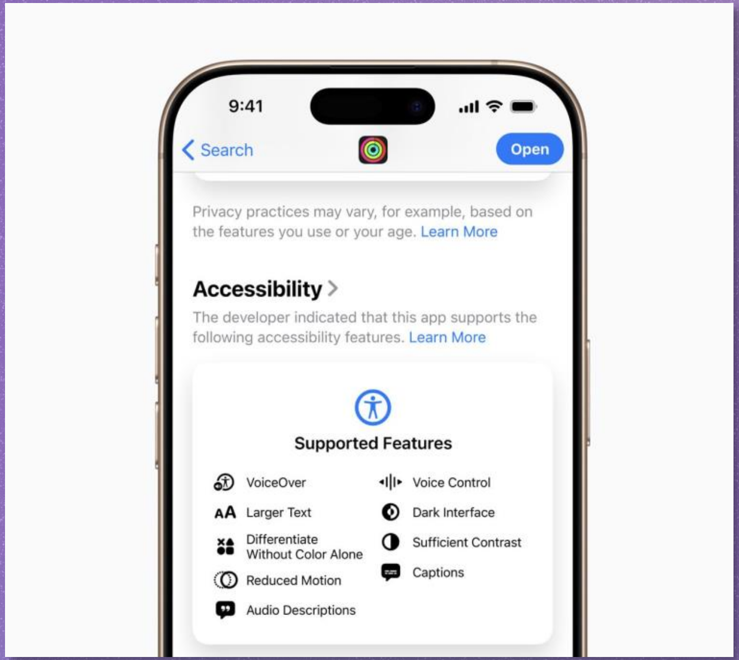
Catalyst

Advanced

Activation Point:	x=144.375 y=121.66
Automation Type:	Button
Children:	Empty array
Children in Naviga...	Empty array
Enabled:	True
Keyboard Focused:	False
Frame:	x=0 y=104.72 w=288.75 h=33.88

⌥Space toggles inspection pointer

Accessibility Nutrition Labels



Privacy practices may vary, for example, based on the features you use or your age. Learn More

Accessibility >

The developer indicated that this app supports the following accessibility features. Learn More

Supported Features

VoiceOver	Voice Control
Larger Text	Dark Interface
Differentiate Without Color Alone	Sufficient Contrast
Reduced Motion	Captions
Audio Descriptions	

- New feature **coming with iOS 26**
- **Help users get an idea of what assistive technologies you support** in your app
- Opportunity for developers **to be valued for their accessibility work**

Wrapping up

- Apple provides **powerful tools** for you to achieve great accessibility in your apps.
- **Apple's accessibility model is based on elements and attributes**. Providing as many as relevant will help assistive technologies work well with your solution.
- **Knowing about the features used by your users is crucial** as it is the first step towards understanding their needs. Some features listed in the module can also be a source of inspiration for you to provide innovative and intuitive way of interacting with your app.
- SwiftUI accessibility leverages the flexibility of view modifiers, which is sometimes paid at the price of code readability and structure. Once you've identified patterns in the way you and your team use Apple accessibility's framework, **you can build helper functions that will facilitate, harmonize and structure accessibility across your codebase.**
- **Accessibility Nutrition Labels** allow developers to let the users know which accessibility features they can expect when downloading your app.

Inclusive programming with Qt

Programming inclusive and accessible software

Accessibility labels with Qt

The role

- Will indicate to the accessibility technology users (like a screen-reader or a braille display) how they can interact with this specific object
- Can take values such as Button, StaticText, Heading, dialog

The name

- Is the main title of your component, like for example "Create account Button" or "Language selection ComboBox".

The description

- Is the main title of your component, like for example "Create account Button" or "Language selection ComboBox".

```
Accessible.role: Accessible.Button
Accessible.name: toolTipText
Accessible.description: JamiStrings.qrCodeExplanation
```

Accessibility Labels with Qt

```
"Michel Berger. 3 unread messages. You said: Thanks!
at 9:07.
Message status: sent. Pinned. Actions available.”
```

```
Accessible.role: Accessible.StaticText
Accessible.name: {
    let name = isOutgoing ? JamiStrings.inReplyToYou: UtilsAdapter.getBestNameForUri(CurrentAccount.id, Author)
    return name + ": " + Body + " " + formattedTime + " " + formattedDay
}
Accessible.description: {
    let status = ""
    if (bubble.isEdited) status += JamiStrings.edited + " "
    if (IsLastSent) status += JamiStrings.sent + " "
    return status + (readers.length > 0 ? JamiStrings.readBy + " " + readers.join(", "): "")
}
```

Accessibility Labels with Qt

You need to make sure that all focusable elements are reached.

Before manually setting keyboard navigation, you should try and use the base focus property of Qt. For example, setting the focus of a component to true might be enough for Qt to recognize it as a reachable interactive element.

```
    keyNavigationEnabled: true
    keyNavigationWraps: false
    focus: true
    activeFocusOnTab: true
```

```
KeyNavigation.backtab: addAccountItem
KeyNavigation.tab: shareButton
KeyNavigation.up: addAccountItem
KeyNavigation.down: shareButton
```

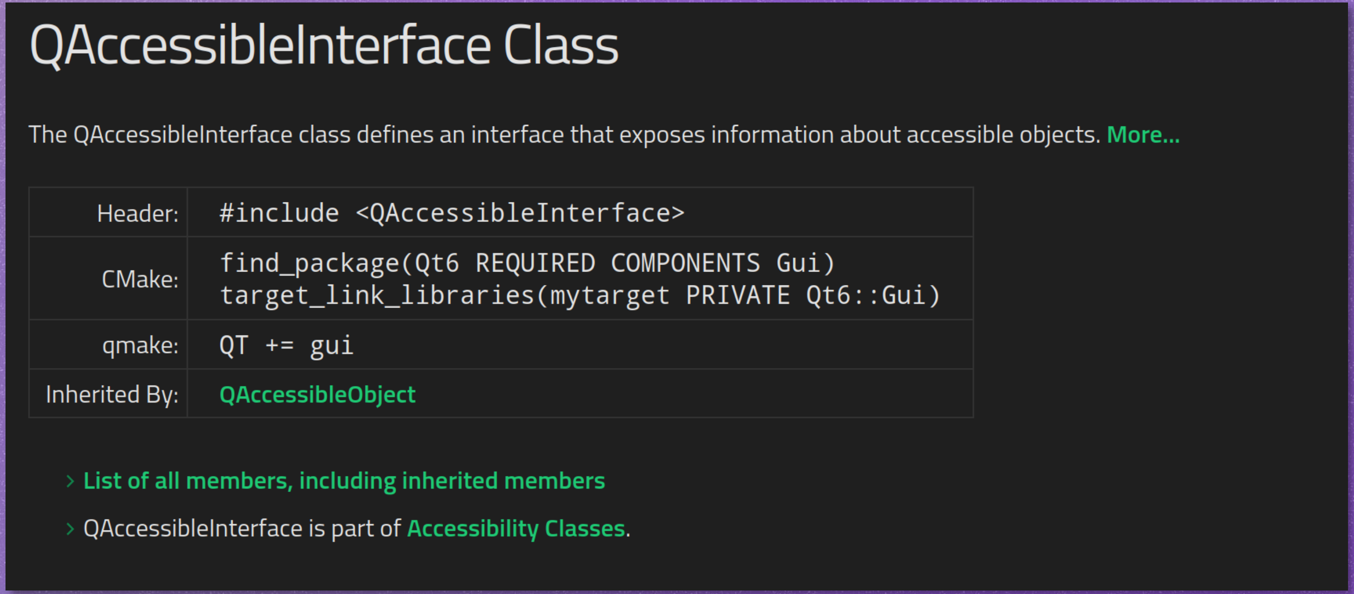
Keyboard navigation for our example

```
// Rather than doing this inside the message itself, we do it at the ListView level
ListView {
    id: root
    // We try to minimize custom code for navigation and therefore use
    // the Qt base property of a List view rather than setting
    // KeyNavigation.tab each time
    keyNavigationEnabled: true
    keyNavigationWraps: false
}
```

The AccessibleInterface Class

The QAccessibleInterface Class is useful to create inclusive user interfaces.

It implements a pure virtual API that allows assistive technologies like braille displays or screen readers to directly access information about accessible objects.



QAccessibleInterface Class

The QAccessibleInterface class defines an interface that exposes information about accessible objects. More...

Header:	`#include <QAccessibleInterface>`
CMake:	`find_package(Qt6 REQUIRED COMPONENTS Gui)` `target_link_libraries(mytarget PRIVATE Qt6::Gui)`
qmake:	`QT += gui`
Inherited By:	QAccessibleObject

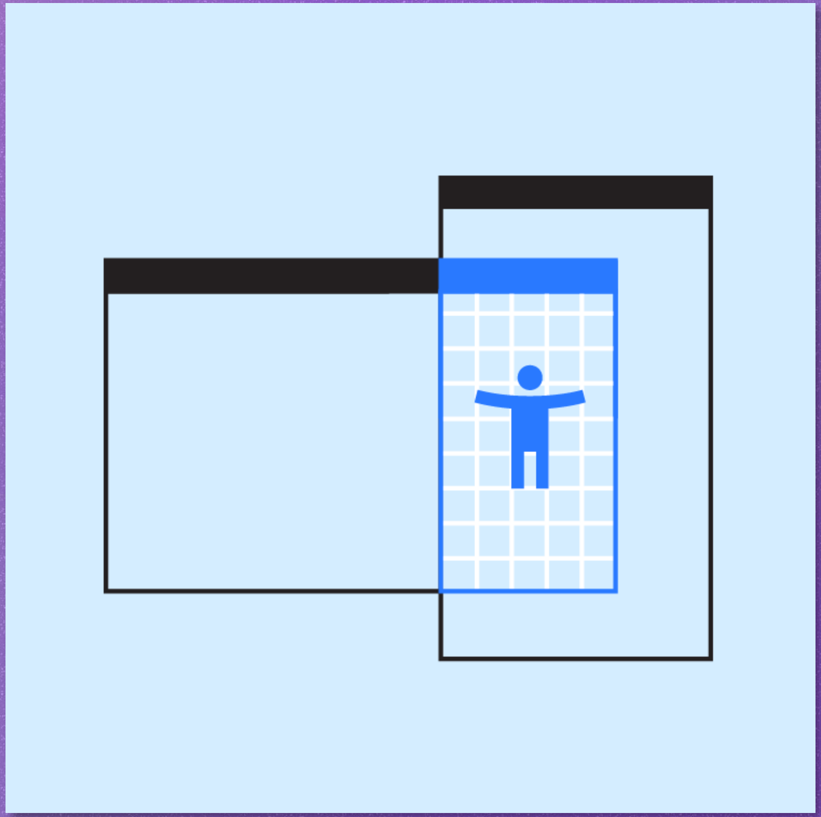
- List of all members, including inherited members
- QAccessibleInterface is part of Accessibility Classes.

Inclusive programming for Android

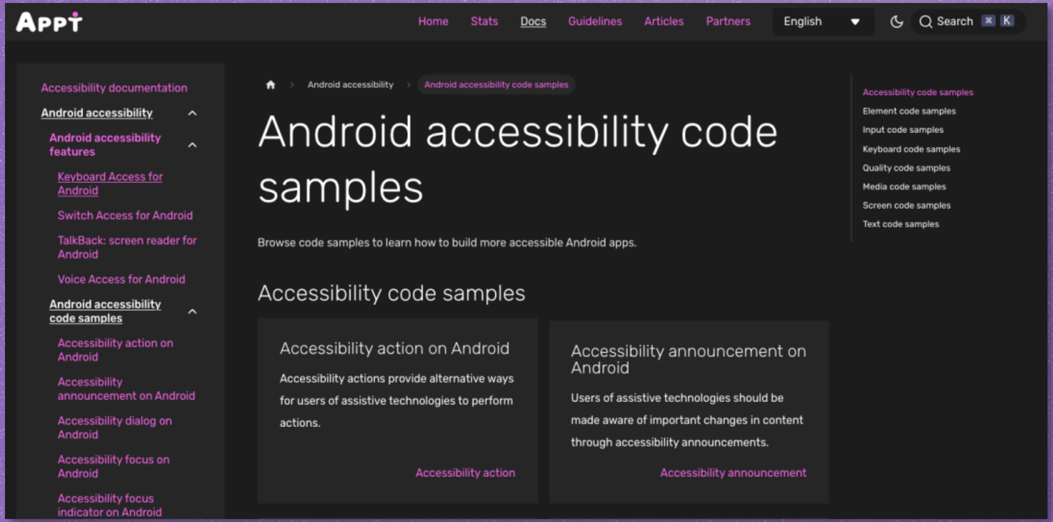
Programming inclusive and accessible software

Android's accessibility model

- Accessibility is at Material's core
- Native android components come with built-in accessibility
- **If you use native components, most of your work as a developer is to contextualize those elements** by customizing the labels, actions and by building a consistent navigation through the accessibility tree.
- Android's **accessibility tree** is a hierarchical organization of your app's on-screen components that is made available to assistive technologies. It is automatically derived from your code.



We recommend



Accessibility documentation
Android accessibility
Android accessibility features
Keyboard Access for Android
Switch Access for Android
TalkBack: screen reader for Android
Voice Access for Android
Android accessibility code samples
Accessibility action on Android
Accessibility announcement on Android
Accessibility dialog on Android
Accessibility focus on Android
Accessibility focus indicator on Android

Android accessibility › Android accessibility code samples

Android accessibility code samples

Browse code samples to learn how to build more accessible Android apps.

Accessibility code samples

Accessibility action on Android

Accessibility actions provide alternative ways for users of assistive technologies to perform actions.

Accessibility action

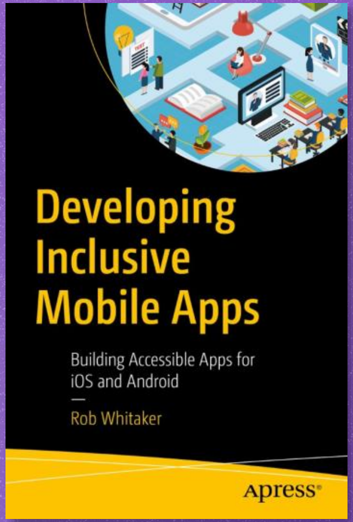
Accessibility announcement on Android

Users of assistive technologies should be made aware of important changes in content through accessibility announcements.

Accessibility announcement

Accessibility code samples
Element code samples
Input code samples
Keyboard code samples
Quality code samples
Media code samples
Screen code samples
Text code samples

APPT's code samples



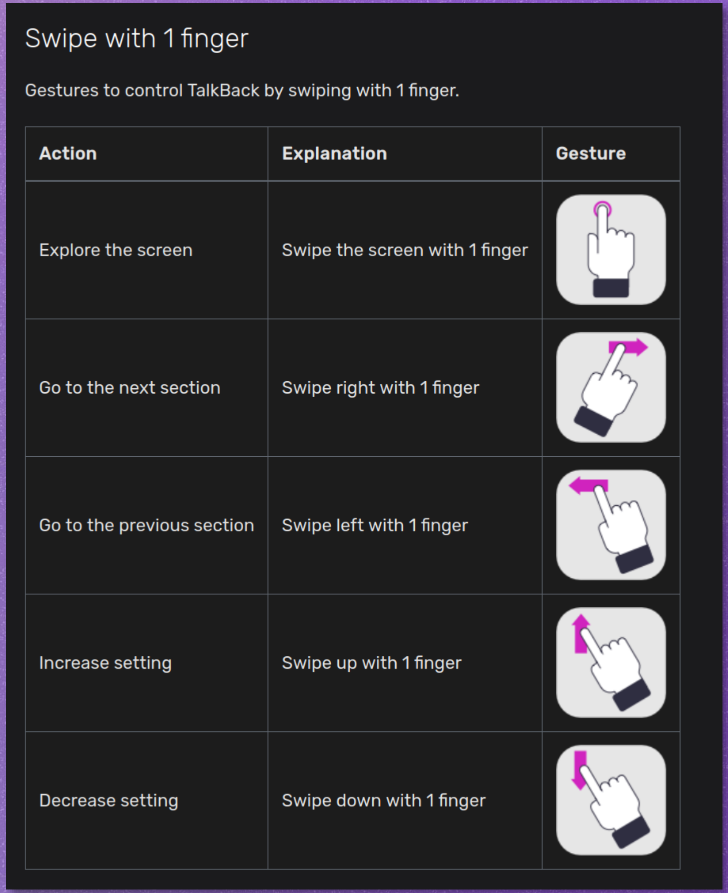
Developing Inclusive Mobile Apps
Building Accessible Apps for iOS and Android
Rob Whitaker
Apress

Rob Whitaker's book

Talkback

TalkBack is the most prominent assistive technology available on Android.

- It works very much like VoiceOver, letting users navigate through your app (via the accessibility tree) without needing to see the screen.
- By making your app work with TalkBack, you are doing most of (if not all) the job required for it to be usable with any assistive technology.



Swipe with 1 finger

Gestures to control TalkBack by swiping with 1 finger.

Action	Explanation	Gesture
Explore the screen	Swipe the screen with 1 finger	
Go to the next section	Swipe right with 1 finger	
Go to the previous section	Swipe left with 1 finger	
Increase setting	Swipe up with 1 finger	
Decrease setting	Swipe down with 1 finger	

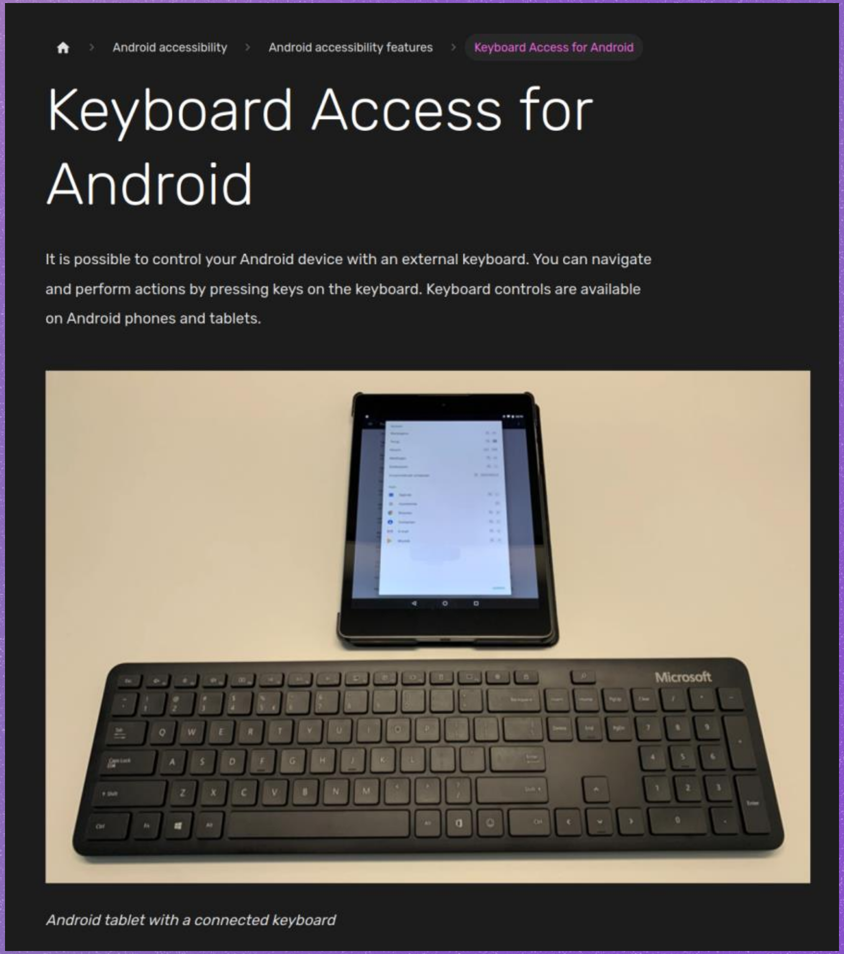
Switch access

- **Switch Access is Android's accessibility feature allowing to control the device using a switch,** that can either be:
  - External (game controller, accessibility switch, keyboard, etc.)
  - Internal (phone buttons, intended for developers)
- Make sure to check that actions requiring complex gestures, like drag-and-dropping, are supported by adding accessibility actions.
- If you've implemented a tap listener on an element that is not marked as actionable, Switch Access won't pick it up.

Keyboard access

Android allows users to control their device using a keyboard.

- Also based on the accessibility tree
- Providing keyboard shortcuts (Jetpack Compose, XML) and custom keyboard order (Jetpack Compose, XML) will contribute to delivering a great UX to keyboard users.



Android accessibility > Android accessibility features > Keyboard Access for Android

Keyboard Access for Android

It is possible to control your Android device with an external keyboard. You can navigate and perform actions by pressing keys on the keyboard. Keyboard controls are available on Android phones and tablets.

Android tablet with a connected keyboard

Voice access

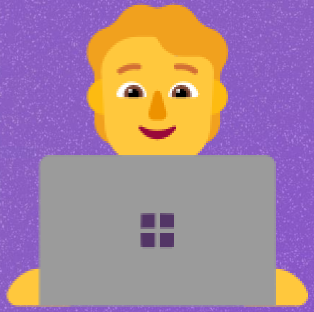
Voice Access is an accessibility service developed by Google and available in the Play Store.

- It aims at providing vocal access to the whole device.
- Unlike Apple's Voice Control, this software uses numbers rather than labels to interact with elements.
- Users can also use text values (and not content description) to select an item of the UI.

Third-party assistive technologies

Unlike Apple, Android has opened their accessibility API, allowing anyone to build their own accessibility service.

While you probably don't want to implement one specifically for your app (unless you *really* have specific accessibility needs), it's good to know that there are plenty of tools to choose from!



Display size, text size

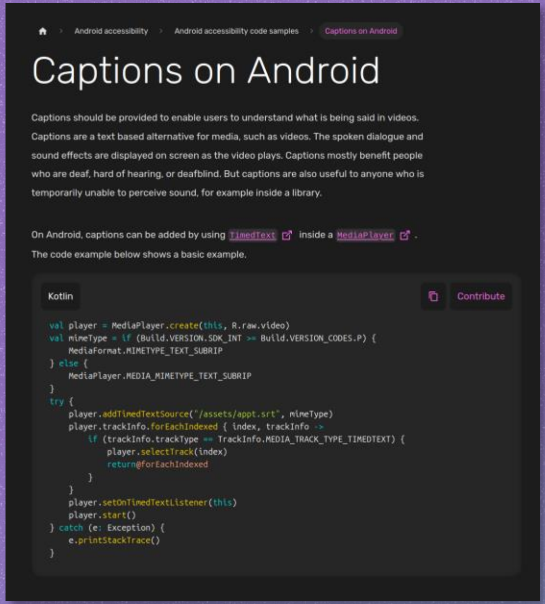
Display and text sizes might vary dramatically based on user needs. Making your app adaptable to those variables is quite straightforward:

- Make sure your text font sizes are indeed affected by system settings. If not, consider using *sp* as unit for your sizes where you are required to use fixed values.

- Avoid using `maxLines` to prevent text truncation.

Captions

Captions can easily be implemented in your Android app for media content.

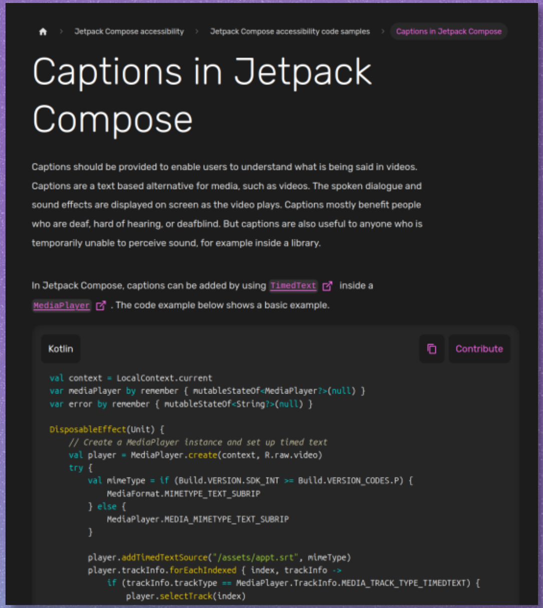


Captions on Android

Captions should be provided to enable users to understand what is being said in videos. Captions are a text based alternative for media, such as videos. The spoken dialogue and sound effects are displayed on screen as the video plays. Captions mostly benefit people who are deaf, hard of hearing, or deafblind. But captions are also useful to anyone who is temporarily unable to perceive sound, for example inside a library.

On Android, captions can be added by using `TimedText` inside a `MediaPlayer` . The code example below shows a basic example.

```kotlin
val player = MediaPlayer.create(this, R.raw.video)
val mimeType = if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.P) {
    MediaFormat.MIMETYPE_TEXT_SUBRIP
} else {
    MediaPlayer.MEDIA_MIMETYPE_TEXT_SUBRIP
}
try {
    player.addTimedTextSource("/assets/appt.srt", mimeType)
    player.trackInfo.forEachIndexed { index, trackInfo ->
        if (trackInfo.trackType == TrackInfo.MEDIA_TRACK_TYPE_TIMEDTEXT) {
            player.selectTrack(index)
            return@forEachIndexed
        }
    }
    player.setOnTimedTextListener(this)
    player.start()
} catch (e: Exception) {
    e.printStackTrace()
}
```

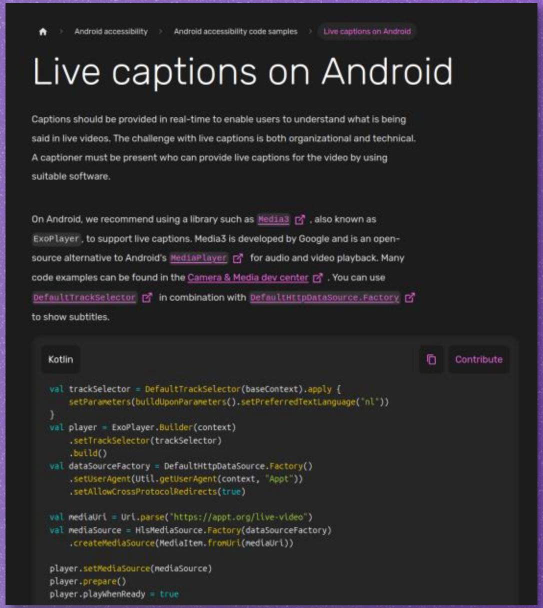

Jetpack Compose accessibility
Jetpack Compose accessibility code samples
Captions in Jetpack Compose
Kotlin
Contribute

Captions in Jetpack Compose

Captions should be provided to enable users to understand what is being said in videos. Captions are a text based alternative for media, such as videos. The spoken dialogue and sound effects are displayed on screen as the video plays. Captions mostly benefit people who are deaf, hard of hearing, or deafblind. But captions are also useful to anyone who is temporarily unable to perceive sound, for example inside a library.

In Jetpack Compose, captions can be added by using `TimedText` inside a `MediaPlayer` . The code example below shows a basic example.

```kotlin
val context = LocalContext.current
var mediaPlayer by remember { mutableStateOf<MediaPlayer?>(null) }
var error by remember { mutableStateOf<String?>(null) }

DisposableEffect(Unit) {
    // Create a MediaPlayer instance and set up timed text
    val player = MediaPlayer.create(context, R.raw.video)
    try {
        val mimeType = if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.P) {
            MediaFormat.MIMETYPE_TEXT_SUBRIP
        } else {
            MediaPlayer.MEDIA_MIMETYPE_TEXT_SUBRIP
        }

        player.addTimedTextSource("/assets/appt.srt", mimeType)
        player.trackInfo.forEachIndexed { index, trackInfo ->
            if (trackInfo.trackType == MediaPlayer.TrackInfo.MEDIA_TRACK_TYPE_TIMEDTEXT) {
                player.selectTrack(index)
```

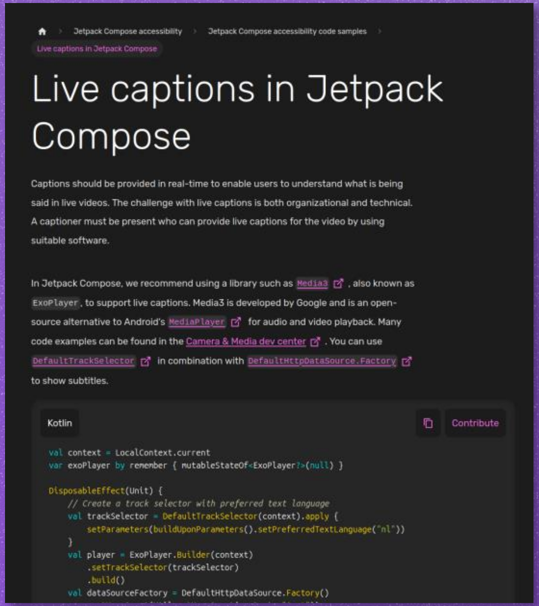


Live captions on Android

Captions should be provided in real-time to enable users to understand what is being said in live videos. The challenge with live captions is both organizational and technical. A captioner must be present who can provide live captions for the video by using suitable software.

On Android, we recommend using a library such as `Media3` , also known as `ExoPlayer`, to support live captions. Media3 is developed by Google and is an open-source alternative to Android's `MediaPlayer` for audio and video playback. Many code examples can be found in the Camera & Media dev center . You can use `DefaultTrackSelector` in combination with `DefaultHttpDataSource.Factory` to show subtitles.

```kotlin
val trackSelector = DefaultTrackSelector(baseContext).apply {
    setParameters(buildUponParameters().setPreferredTextLanguage("nl"))
}
val player = ExoPlayer.Builder(context)
    .setTrackSelector(trackSelector)
    .build()
val dataSourceFactory = DefaultHttpDataSource.Factory()
    .setUserAgent(Util.getUserAgent(context, "Appt"))
    .setAllowCrossProtocolRedirects(true)

val mediaUri = Uri.parse("https://appt.org/live-video")
val mediaSource = HlsMediaSource.Factory(dataSourceFactory)
    .createMediaSource(MediaItem.fromUri(mediaUri))

player.setMediaSource(mediaSource)
player.prepare()
player.playWhenReady = true
```


Jetpack Compose accessibility
Jetpack Compose accessibility code samples
Live captions in Jetpack Compose
Kotlin
Contribute

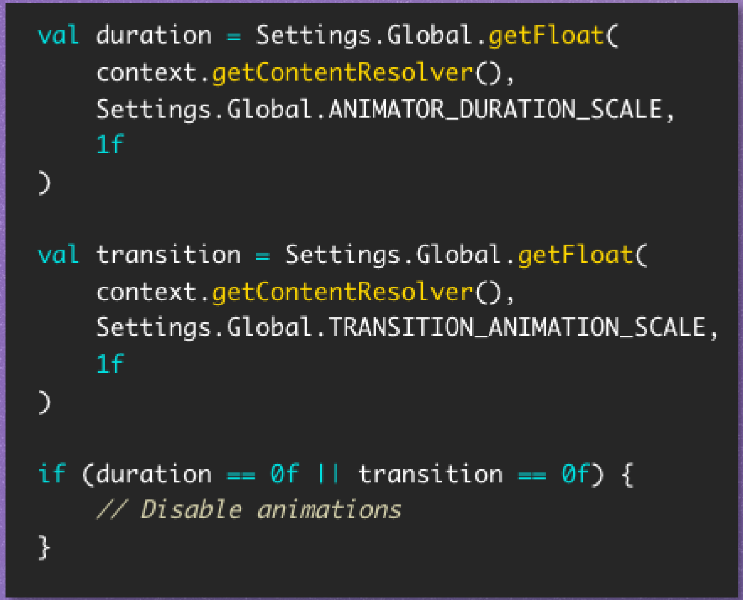
Live captions in Jetpack Compose

Captions should be provided in real-time to enable users to understand what is being said in live videos. The challenge with live captions is both organizational and technical. A captioner must be present who can provide live captions for the video by using suitable software.

In Jetpack Compose, we recommend using a library such as `Media3` , also known as `ExoPlayer`, to support live captions. Media3 is developed by Google and is an open-source alternative to Android's `MediaPlayer` for audio and video playback. Many code examples can be found in the Camera & Media dev center . You can use `DefaultTrackSelector` in combination with `DefaultHttpDataSource.Factory` to show subtitles.

```kotlin
val context = LocalContext.current
var exoPlayer by remember { mutableStateOf<ExoPlayer?>(null) }

DisposableEffect(Unit) {
    // Create a track selector with preferred text language
    val trackSelector = DefaultTrackSelector(context).apply {
        setParameters(buildUponParameters().setPreferredTextLanguage("nl"))
    }
    val player = ExoPlayer.Builder(context)
        .setTrackSelector(trackSelector)
        .build()
    val dataSourceFactory = DefaultHttpDataSource.Factory()
```

Reduced animations



```kotlin
val duration = Settings.Global.getFloat(
    context.getContentResolver(),
    Settings.Global.ANIMATOR_DURATION_SCALE,
    1f
)

val transition = Settings.Global.getFloat(
    context.getContentResolver(),
    Settings.Global.TRANSITION_ANIMATION_SCALE,
    1f
)

if (duration == 0f || transition == 0f) {
    // Disable animations
}
```

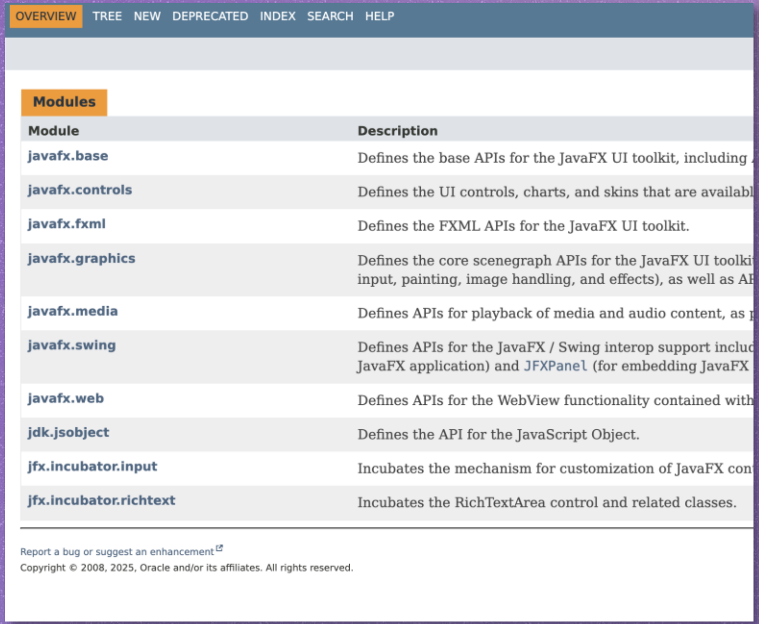
Some users prefer or need to reduce animations.

- While Android doesn't directly indicate through the API whether the user has the option activated or not, you can determine it with this little workaround.
- Providing the possibility to stop automatic and repeating animations, as well as flashes or blinkings, zooms and parallax effects is crucial to make sure users can use your app safely and comfortably.

Inclusive programming with Java

Programming inclusive and accessible software

JavaFX Accessibility



Modules

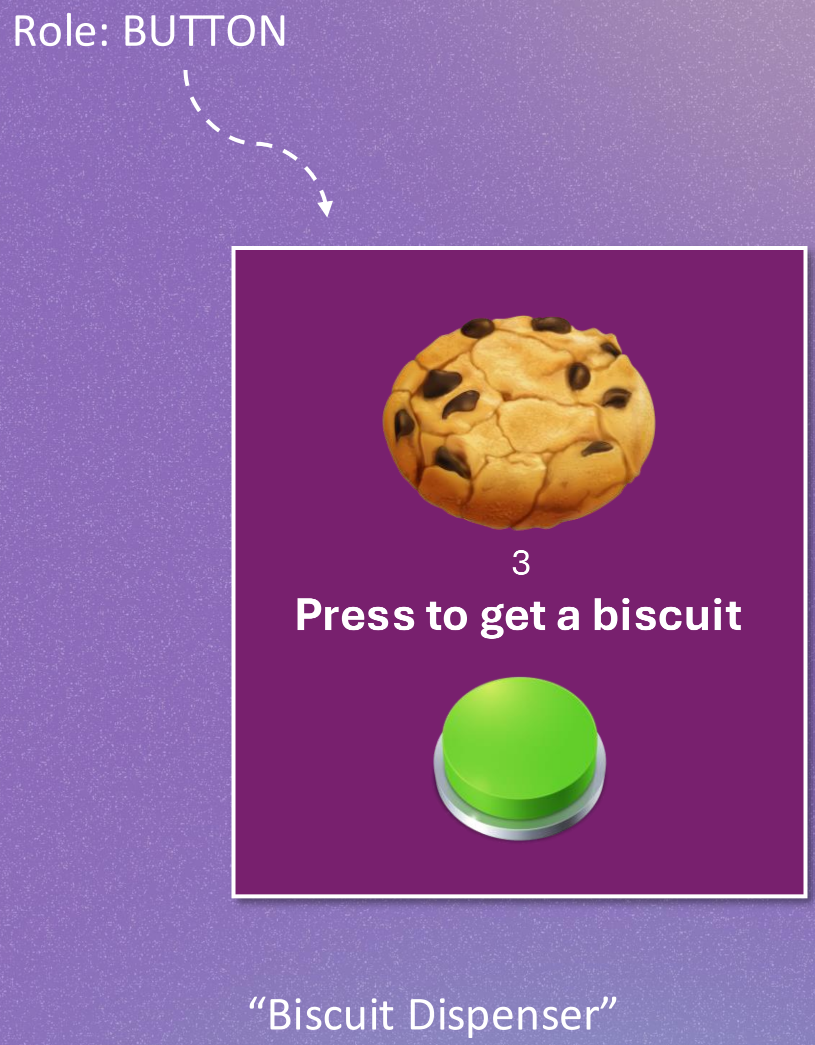
Module	Description
javafx.base	Defines the base APIs for the JavaFX UI toolkit, including
javafx.controls	Defines the UI controls, charts, and skins that are availabl
javafx.fxml	Defines the FXML APIs for the JavaFX UI toolkit.
javafx.graphics	Defines the core scenegraph APIs for the JavaFX UI toolki input, painting, image handling, and effects), as well as AI
javafx.media	Defines APIs for playback of media and audio content, as
javafx.swing	Defines APIs for the JavaFX / Swing interop support incluc JavaFX application) and JFXPanel (for embedding JavaFX
javafx.web	Defines APIs for the WebView functionality contained with
jdk.jsobject	Defines the API for the JavaScript Object.
jfx.incubator.input	Incubates the mechanism for customization of JavaFX con
jfx.incubator.richtext	Incubates the RichTextArea control and related classes.

Report a bug or suggest an enhancement

Copyright © 2008, 2025, Oracle and/or its affiliates. All rights reserved.

- JavaFX has a **complete accessibility API**
- Unfortunately**, its documentation is very rudimentary** (only an API reference)
- In this module, we'll **provide the basics** along with some **common accessibility implementations** with JavaFX

Accessible Role

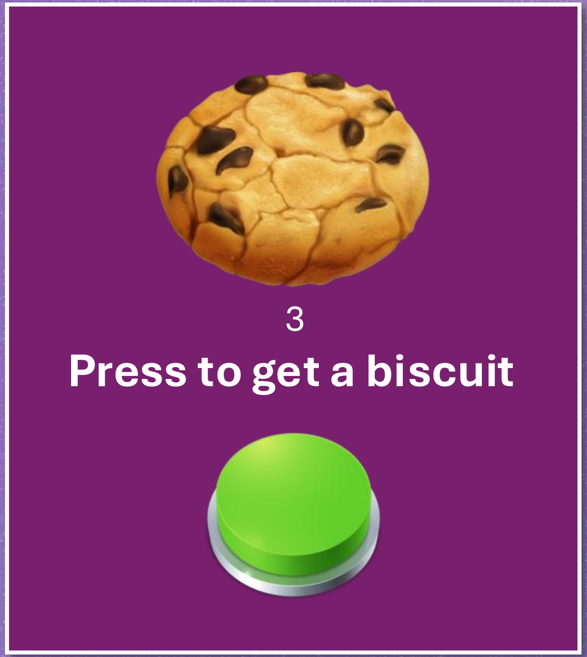


- The `accessibleRole` gives information to assistive technologies so they can behave accordingly. There are many roles possible from which you can choose for example BUTTON.

- The `accessibleRoleDescription` can be used to override `accessibleRole's` textual representation, which is read by the screen reader. This is useful, for example, if you have an element with the role Button that you would like the screen reader to call *"biscuit dispenser"*.

Describing and explaining

Role: BUTTON

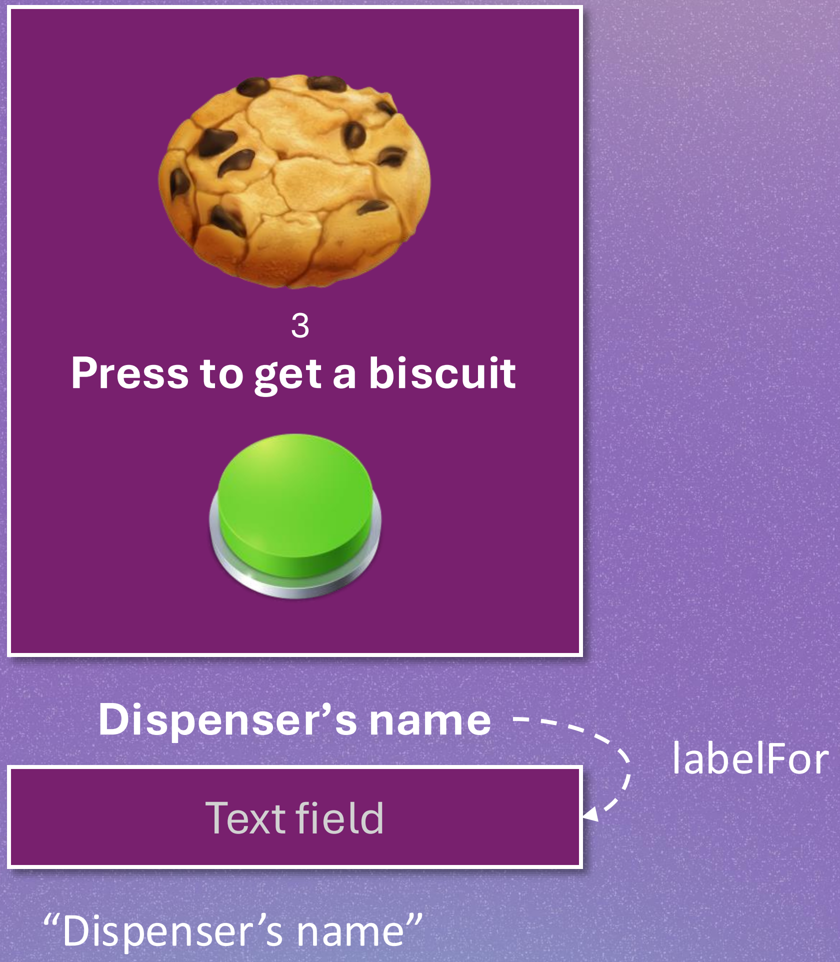


"Biscuit Dispenser,
Three biscuits left,
Dispenses a biscuit when activated"

- The `accessibleText` should describe the content of the Node. For our example, the accessible text could read "3 biscuits left". For some scenarios like for buttons, it might not be necessary to set it as labels will automatically be read out loud by screen readers if linked accordingly.

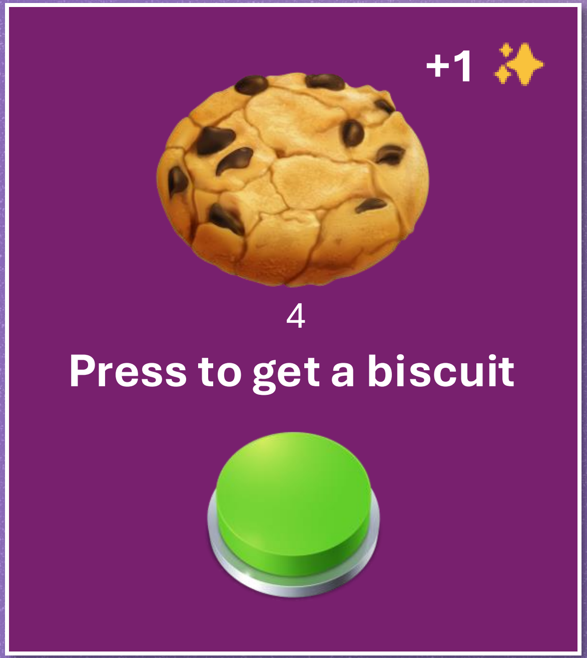
- The `accessibleHelp` should be used to give contextual hints to help users understand where they are or what are the possible interactions with an element. For our previous example, this could read "dispenses a biscuit when activated".

The labelFor property



- The `labelFor` property is used to link a label to an input (like a text field or a slider). This is not to be confused with an accessibility label as called in many other frameworks - the equivalent is `accessibleText` in JavaFX. It could for example be a (visual) label saying "edit the biscuit dispenser’s name" next to an associated text field.

Notifying changes



"Biscuit Dispenser,
Four biscuits left,
Dispenses a biscuit when activated"

- By default, changes on elements might not be notified. To make sure they are, you can use the `notifyAccessibleAttributeChanged(AccessibleAttribute attributes)`. As explained in Jonathan Giles' presentation of the JavaFX accessibility API for Oracle, the method can for example be called in an override of the invalidated() Node's method.

Useful features

- Check if an assistive technology is running with Platform's `accessibilityActiveProperty`.
- Make elements focusable `setFocusTraversable(boolean value)`
- Focus elements `with` `requestFocus()`
- Customize or trigger accessibility actions with `executeAccessibilityAction`.
- Provide dynamic accessibility properties with `queryAccessibleAttribute`.

building inclusive free and open-source software

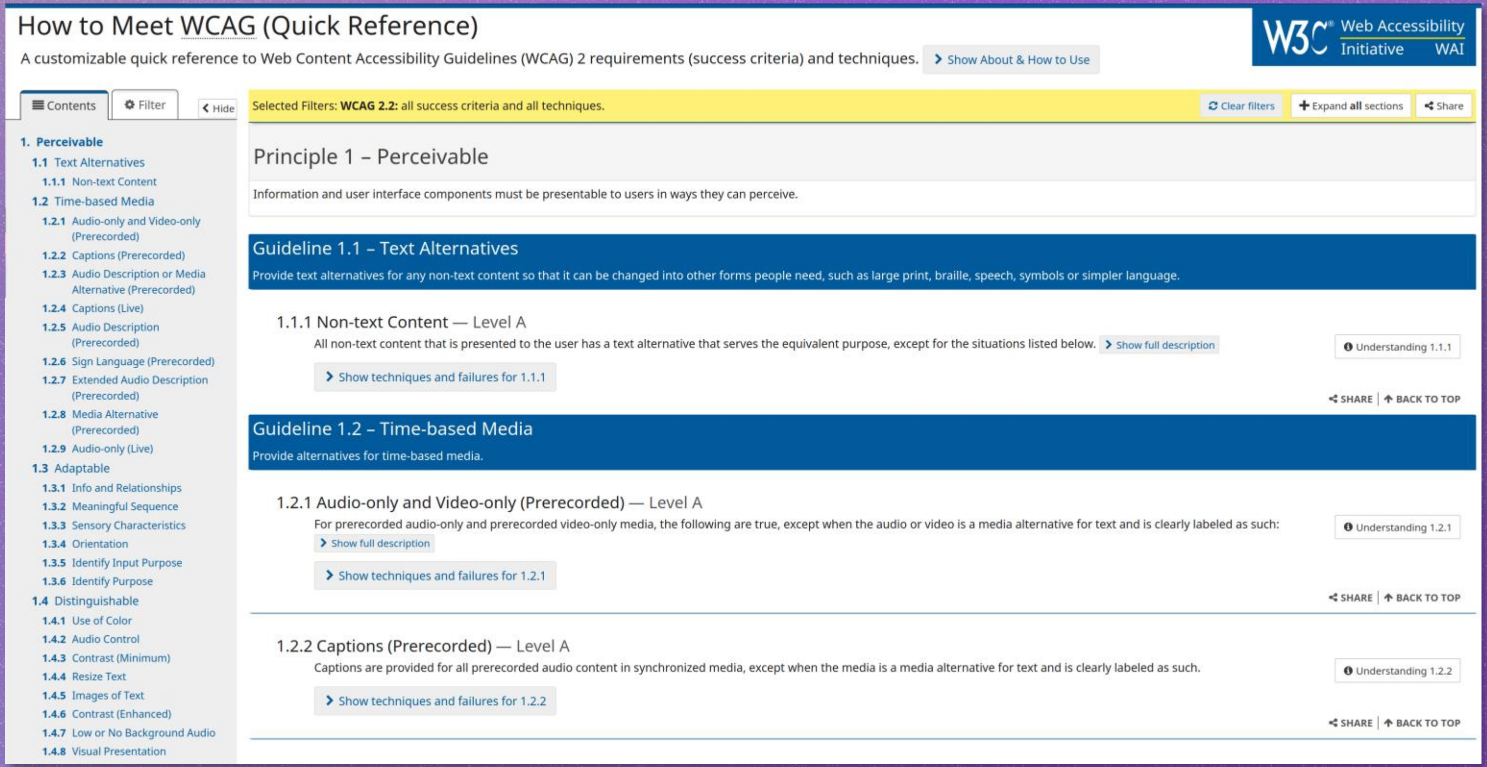
Inclusive Web programming

Programming inclusive and accessible software

The WCAG guidelines

These are the international guidelines for web accessibility, and it should also be yours.

We especially recommend using the *How to Meet WCAG (Quick Reference)* from the Web Accessibility Initiative.



How to Meet WCAG (Quick Reference)

A customizable quick reference to Web Content Accessibility Guidelines (WCAG) 2 requirements (success criteria) and techniques.

Selected Filters: **WCAG 2.2:** all success criteria and all techniques.

Principle 1 – Perceivable

Information and user interface components must be presentable to users in ways they can perceive.

Guideline 1.1 – Text Alternatives

Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

1.1.1 Non-text Content — Level A

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below.

Guideline 1.2 – Time-based Media

Provide alternatives for time-based media.

1.2.1 Audio-only and Video-only (Prerecorded) — Level A

For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such:

1.2.2 Captions (Prerecorded) — Level A

Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such.

The *inclusive Components* project

The Inclusive Components blog and pattern library should be one of your privileged resource going forward with web accessibility.

It provides articles, documentation and code snippets for making elements such as collapsible sections, tooltips, toggle buttons, etc. inclusive. It's truly an enormous and immensely helpful work.

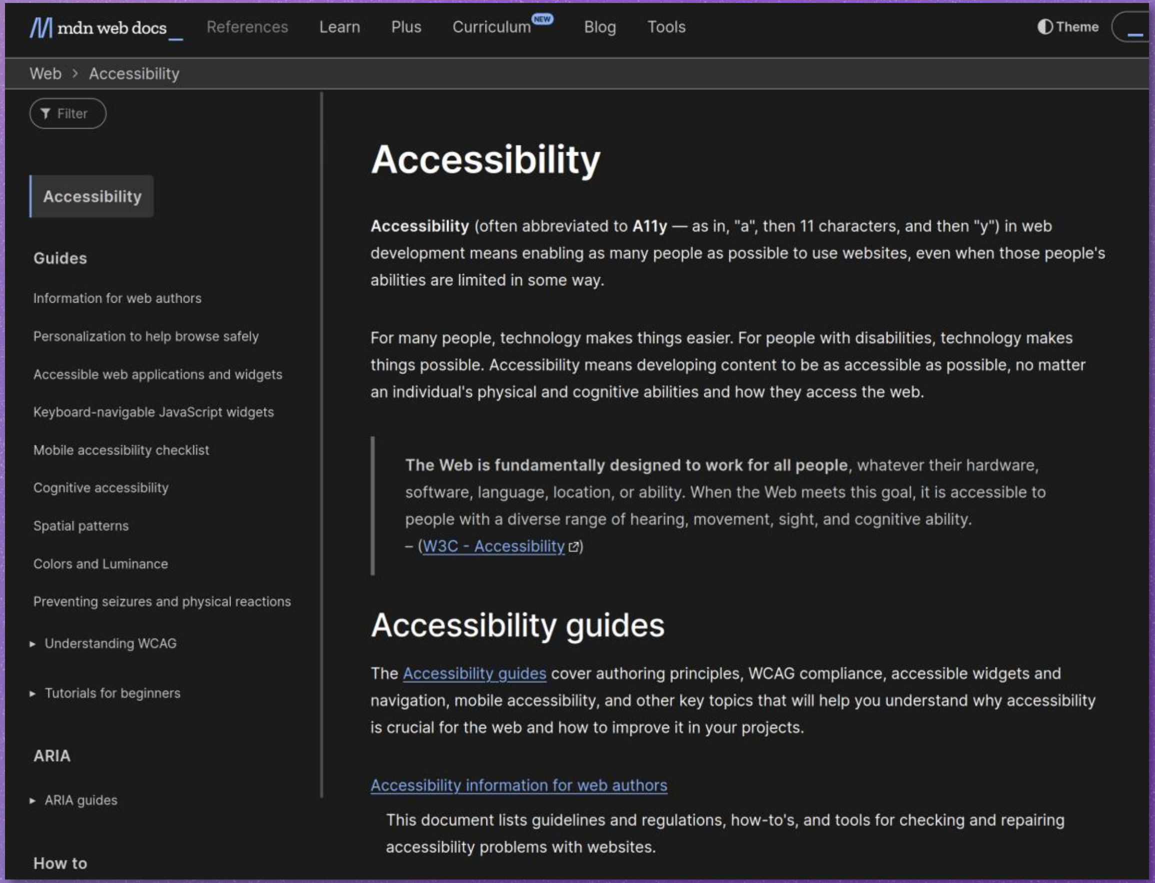
Inclusive Components

A blog trying to be a pattern library. All about designing inclusive web interfaces, piece by piece.

Mozilla's accessibility documentation

When it comes to essentials of web development, Mozilla always has a resource. Accessibility is not an exception!

Their very extensive and beginner-friendly reference is a great starting point.



Accessibility

Accessibility (often abbreviated to **A11y** — as in, "a", then 11 characters, and then "y") in web development means enabling as many people as possible to use websites, even when those people's abilities are limited in some way.

For many people, technology makes things easier. For people with disabilities, technology makes things possible. Accessibility means developing content to be as accessible as possible, no matter an individual's physical and cognitive abilities and how they access the web.

> **The Web is fundamentally designed to work for all people**, whatever their hardware, software, language, location, or ability. When the Web meets this goal, it is accessible to people with a diverse range of hearing, movement, sight, and cognitive ability.
> – (W3C - Accessibility)

Accessibility guides

The Accessibility guides cover authoring principles, WCAG compliance, accessible widgets and navigation, mobile accessibility, and other key topics that will help you understand why accessibility is crucial for the web and how to improve it in your projects.

Accessibility information for web authors

This document lists guidelines and regulations, how-to's, and tools for checking and repairing accessibility problems with websites.

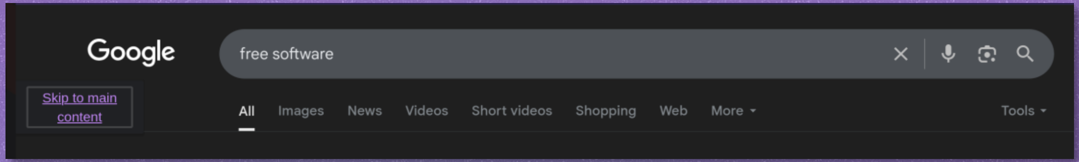
Keyboard usage

Navigation

Navigation inside a web page or a webapp is often not trivial to design.

Skip to content

Most websites should include a "Skip to main content". This allows users to navigate directly to the content of a page without going through all of the menus, situated at the top of almost every single page they navigate.



Heading hierarchy

This one is easy. When navigating a web page, screen readers and other assistive technologies will generate a table of content for the user using the headings.

Menus

Designing and implementing usable and accessible menus is a requirement to making great web interfaces. This very extensive documentation provides insightful guidelines and code snippets to implement your menus in an accessible way.

JavaScript

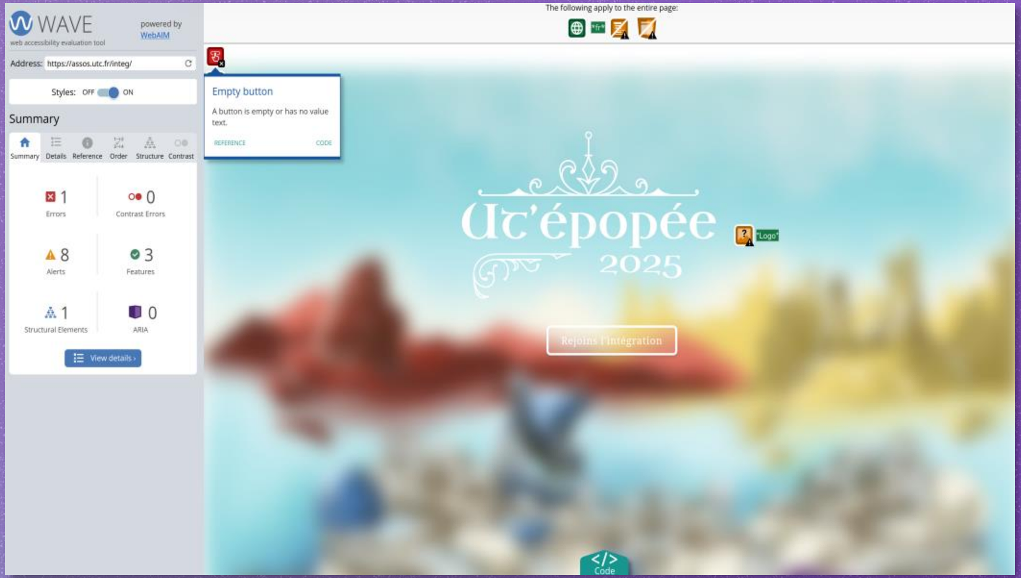
A Guide To Keyboard Accessibility: JavaScript is article that provides detailed insights and code snippets about accessibility programming using JavaScript.

Accessibility testing tools

WAVE
web accessibility evaluation tool

- Wave is a testing software to automatically detect warnings and errors related to the WCAG Accessibility guidelines.
- As simple as entering your domain to see prevalent issues with your website
- Be careful not to believe blindly the feedback form automated tools lke Wave.

Platform	Pros	Cons
Web	• Very easy to use • Completely free	• Can be limited • Often misleading


WAVE
web accessibility evaluation tool
powered by WebAIM
The following apply to the entire page:
Ut'épopée 2025
Rejoins l'intégration
Code

Address: https://assos.utc.fr/integ/

Styles: OFF ON

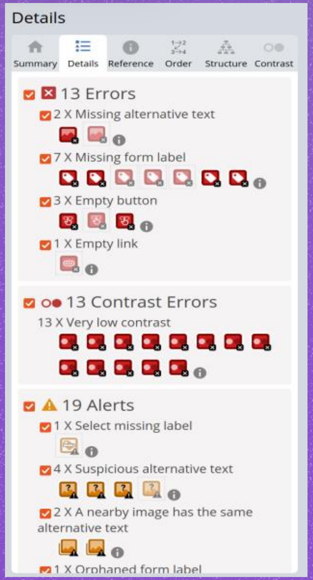
Summary

Summary | Details | Reference | Order | Structure | Contrast

1 Errors
0 Contrast Errors
8 Alerts
3 Features
1 Structural Elements
0 ARIA

View details ›

Empty button
A button is empty or has no value text.
REFERENCE CODE



Details

Summary | Details | Reference | Order | Structure | Contrast

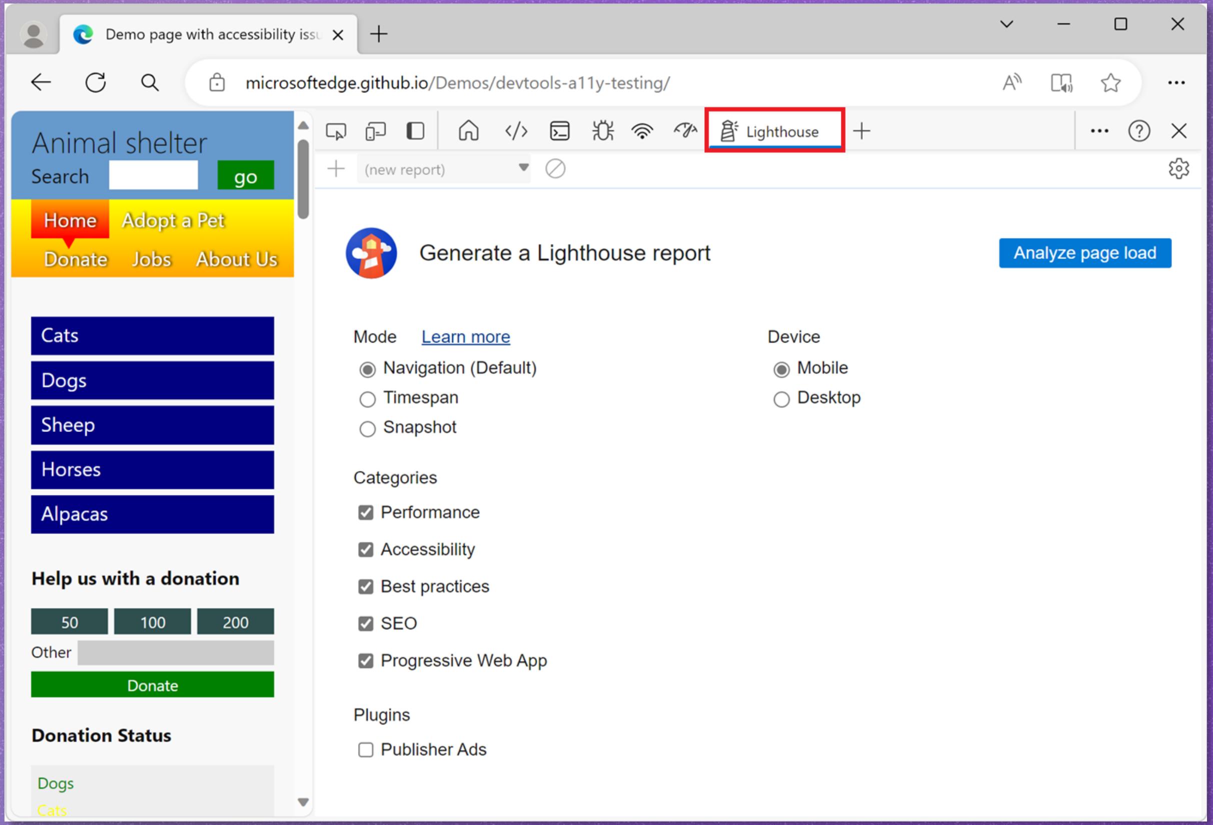
13 Errors
- 2 X Missing alternative text
- 7 X Missing form label
- 3 X Empty button
- 1 X Empty link

13 Contrast Errors
- 13 X Very low contrast

19 Alerts
- 1 X Select missing label
- 4 X Suspicious alternative text
- 2 X A nearby image has the same alternative text
- 1 X Orphaned form label

Accessibility testing tools

- Lighthouse is a web accessibility testing software developed by Microsoft and integrated in Microsoft Edge.
- It works quite similarly to wave. You can check Microsoft's documentation on how to use it.



Demo page with accessibility issu

microsoftedge.github.io/Demos/devtools-a11y-testing/

Generate a Lighthouse report

Mode Learn more
- [✓ Navigation (Default)]
- Timespan
- Snapshot

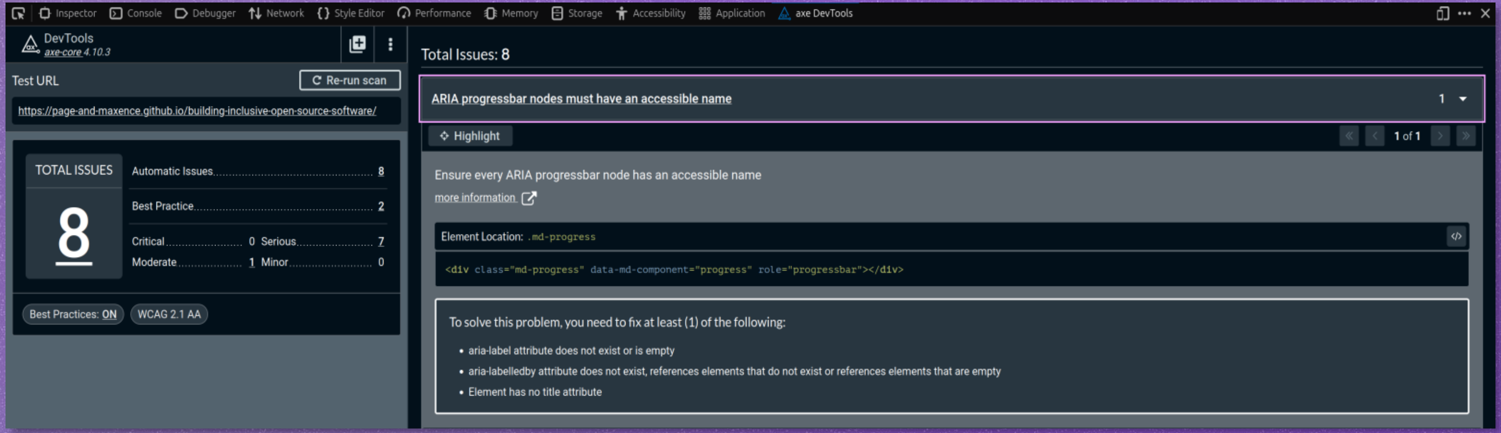
Device
- [✓ Mobile]
- Desktop

Categories
- [✓] Performance
- [✓] Accessibility
- [✓] Best practices
- [✓] SEO
- [✓] Progressive Web App

Plugins
- [ ] Publisher Ads

Accessibility testing tools

- Axe is one of the most used web accessibility tool with over two billion downloads.
- Developed and maintained by Deque
- Its feedback is very comprehensible, providing insights on how to fix specific issues and even providing information about "Best Practices".



Test URL

https://page-and-maxence.github.io/building-inclusive-open-source-software/

TOTAL ISSUES 8

Automatic Issues 8

Best Practice 2

Critical .... 0 Serious 7

Moderate .... 1 Minor 0

Total Issues: 8

ARIA progressbar nodes must have an accessible name 1

Ensure every ARIA progressbar node has an accessible name

more information

Element Location: .md-progress

```
<div class="md-progress" data-md-component="progress" role="progressbar"></div>
```

To solve this problem, you need to fix at least (1) of the following:

- aria-label attribute does not exist or is empty
- aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
- Element has no title attribute

Conclusion, bibliography and useful resources

To conclude and go further

What is accessibility

- From a product standpoint, it's about ensuring everything you build can be used by anyone, no matter their physical or cognitive abilities.
- From a legal standpoint, it is an obligation in many countries, guaranteeing an equal right to autonomy and participation of all.

- From a software standpoint, digital accessibility comes down to making apps optimally perceivable, operable and understandable to people who live with disabilities.

- From a political standpoint, accessibility is striving fight back against the structural tendency to exclusion of disabled people and attempting to provide them with a place (virtual or not) where they feel at ease and free to express themselves.

When products are not built with accessibility in mind, they end up excluding many people, denying them access to places, tools or even communication.

Additional needs and factors to take into consideration

Non-exhaustive **list of human characteristics to keep in mind to try and include as many users as possible:**

Age
Culture
Digital literacy
Education
Ethnicity
Gender, gender identity
Historical context
Language
Mental health
Nationality
Physical *and* cognitive (dis)abilities
Sexuality
Socio-economic context
…

You need to consider the intersections of all of those categories as well as the disabilities listed previously in the first section of this module.

Intersectionality in software

- 17.7% of users with disabilities live with more than one condition (as of 2016).
- Designing for one disability leaves out a blank spot for users having that disability but also other conditions.
- It’s a common bias to forget that disabled people are not just disabled, but first and foremost people with their own unique life experiences and specificities.

We need to think about women who rely on wheelchairs, LGBTQIA+ people who don't speak the language of they country they live in, blind users who can't afford an iPhone - and those who are all of that at once!

Intersectional thinking is key

- 17.7% of users with disabilities live with more than one condition (as of 2016).
- Designing for one disability leaves out a blank spot for users having that disability but also other conditions.
- It’s a common bias to forget that disabled people are not just disabled, but first and foremost people with their own unique life experiences and specificities.

❗ **It's crucial to not think disabled peoples as a minority but rather as the living diversity in which we all evolve.**

Accessibility and inclusiveness need to be intentional

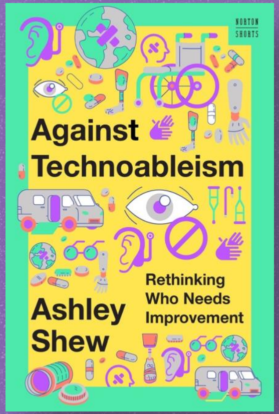
"Disability is not a binary state. We all have abilities and limits to those abilities. Disability happens when we have built something that doesn’t work for someone with particular skills."*

- By default, we naturally tend to design, program and test for people like us or the people we spend time with.
- Not thinking about accessibility is a privilege because most disabled people don't have the choice to think about it.
- Not developing with accessibility and inclusiveness in mind will reinforce biases against minorities in software.

Accessibility and inclusiveness are a shift in how you perceive and take into account otherness through your building processes, not a task to be done.

* Whitaker Rob.*Developing Inclusive Mobile Apps, Building Accessible Apps for iOS and Android*, 2020.

Disabled people are the real experts on accessibility



"Disabled people are “the real experts” (the title of Michelle Sutton’s 2015 edited volume about and authored by autistic people) when it comes to technology and disability. We use technologies. We also reject them, grapple with them, or repurpose them. The views on technology we get from listening to disabled people often look very different from those of people educated in the medical and “helping” professions."*

We encourage following this motto: "nothing about them without them”.

** Shew Ashley, *Against Technoableism: rethinking who needs improvement*, 2023

There isn't one "accessible and inclusive design"

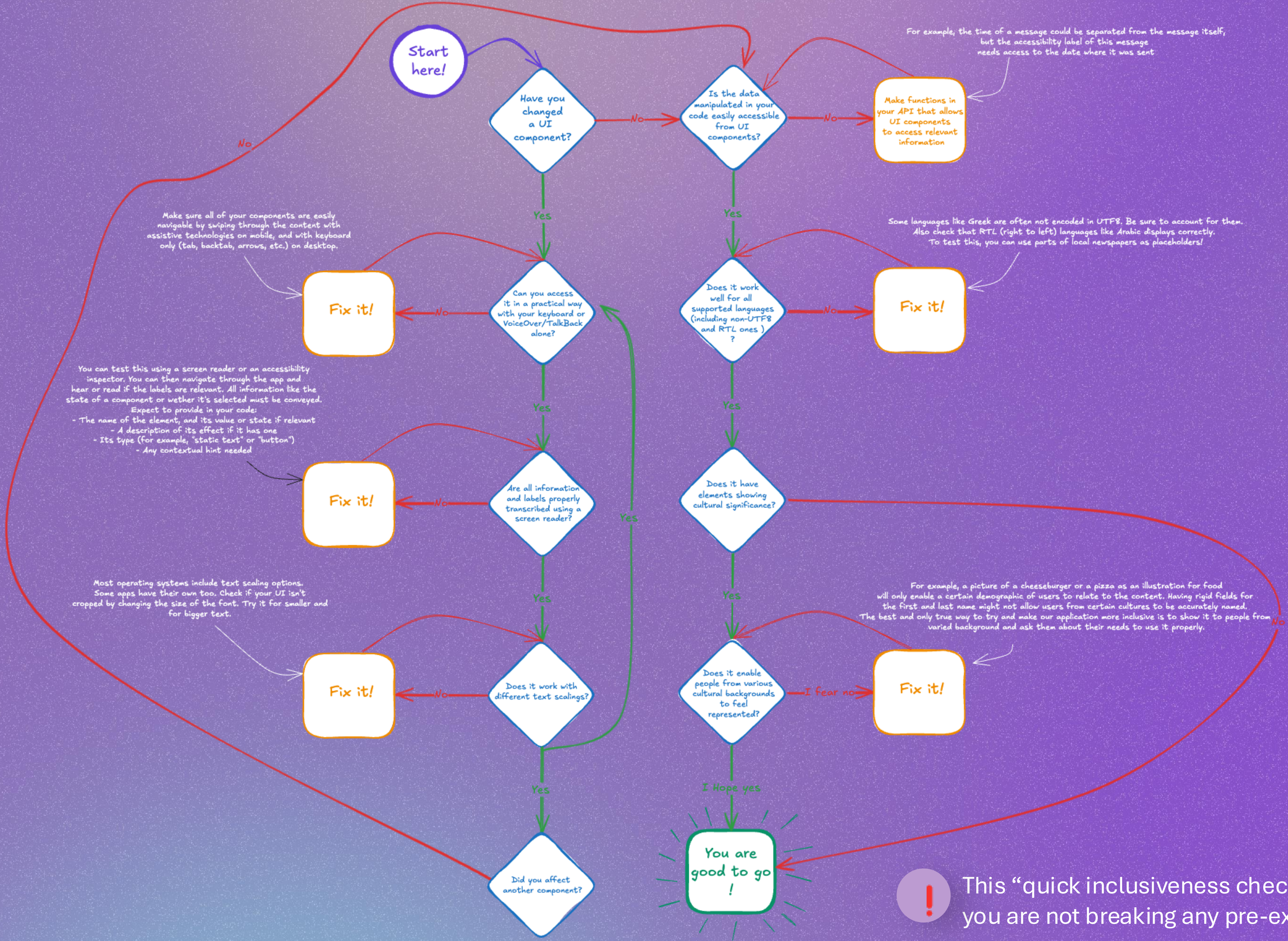
- While implementation is important, **design is crucial** to build an inclusive app.
- Don't ever "make a feature accessible" again. Instead, build it with inclusiveness in mind.

People are diverse, solutions to their needs too.
Experiment, fail, get feedback, improve and nurture empathy!

Accessible code is good code

- **Bad accessibility can often be a symptom of flawed code.** Accessibility and inclusiveness have one key benefit: making you think twice about your code.
- You code can be good but will likely never be as robust as one iterated upon by a community of developers or a big company.

Your code cannot be good without being accessible and if it's not accessible, it's probably hiding something more.


Start here!
Have you changed a UI component?
No
Yes
Is the data manipulated in your code easily accessible from UI components?
No
Make functions in your API that allows UI components to access relevant information
Yes
Can you access it in a practical way with your keyboard or VoiceOver/TalkBack alone?
No
Fix it!
Yes
Are all information and labels properly transcribed using a screen reader?
No
Fix it!
Yes
Does it work with different text scalings?
No
Fix it!
Yes
Yes
Did you affect another component?
No
Does it work well for all supported languages (including non-UTF8 and RTL ones) ?
No
Fix it!
Yes
Does it have elements showing cultural significance?
No
Yes
Does it enable people from various cultural backgrounds to feel represented?
I fear no
Fix it!
I Hope yes
You are good to go !

For example, the time of a message could be separated from the message itself, but the accessibility label of this message needs access to the date where it was sent

Make sure all of your components are easily navigable by swiping through the content with assistive technologies on mobile, and with keyboard only (tab, backtab, arrows, etc.) on desktop.

You can test this using a screen reader or an accessibility inspector. You can then navigate through the app and hear or read if the labels are relevant. All information like the state of a component or wether it's selected must be conveyed.
Expect to provide in your code:
- The name of the element, and its value or state if relevant
- A description of its effect if it has one
- Its type (for example, "static text" or "button")
- Any contextual hint needed

Most operating systems include text scaling options. Some apps have their own too. Check if your UI isn't cropped by changing the size of the font. Try it for smaller and for bigger text.

Some languages like Greek are often not encoded in UTF8. Be sure to account for them. Also check that RTL (right to left) languages like Arabic displays correctly. To test this, you can use parts of local newspapers as placeholders!

For example, a picture of a cheeseburger or a pizza as an illustration for food will only enable a certain demographic of users to relate to the content. Having rigid fields for the first and last name might not allow users from certain cultures to be accurately named. The best and only true way to try and make our application more inclusive is to show it to people from varied background and ask them about their needs to use it properly.

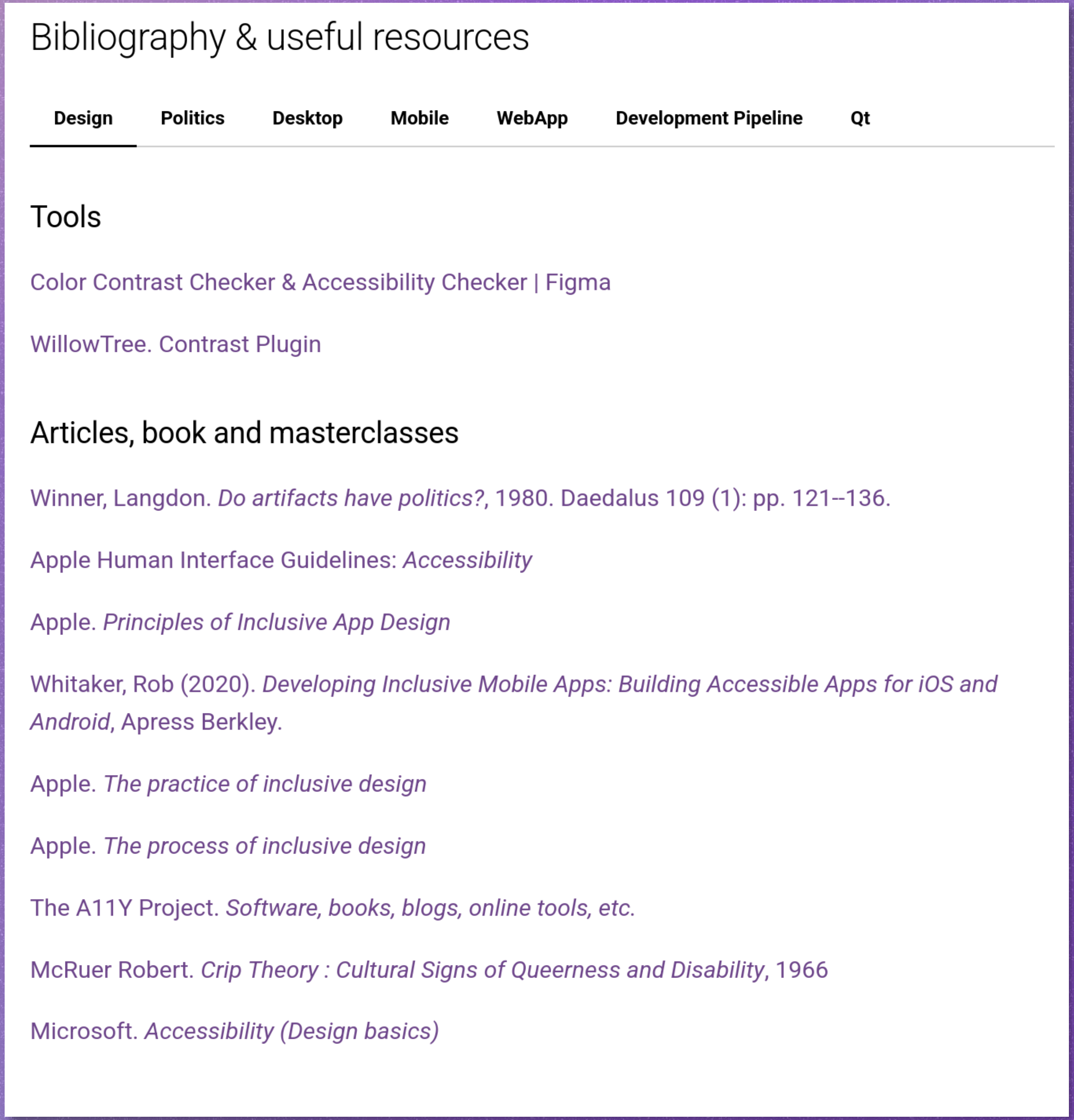
! This "quick inclusiveness check" only aims at checking you are not breaking any pre-existing accessibility.

Perspectives

Bibliography and useful resources

Bibliography and useful resources

Design



Bibliography & useful resources

Design | Politics | Desktop | Mobile | WebApp | Development Pipeline | Qt

Tools

Color Contrast Checker & Accessibility Checker | Figma

WillowTree. Contrast Plugin

Articles, book and masterclasses

Winner, Langdon. *Do artifacts have politics?*, 1980. Daedalus 109 (1): pp. 121--136.

Apple Human Interface Guidelines: *Accessibility*

Apple. *Principles of Inclusive App Design*

Whitaker, Rob (2020). *Developing Inclusive Mobile Apps: Building Accessible Apps for iOS and Android*, Apress Berkley.

Apple. *The practice of inclusive design*

Apple. *The process of inclusive design*

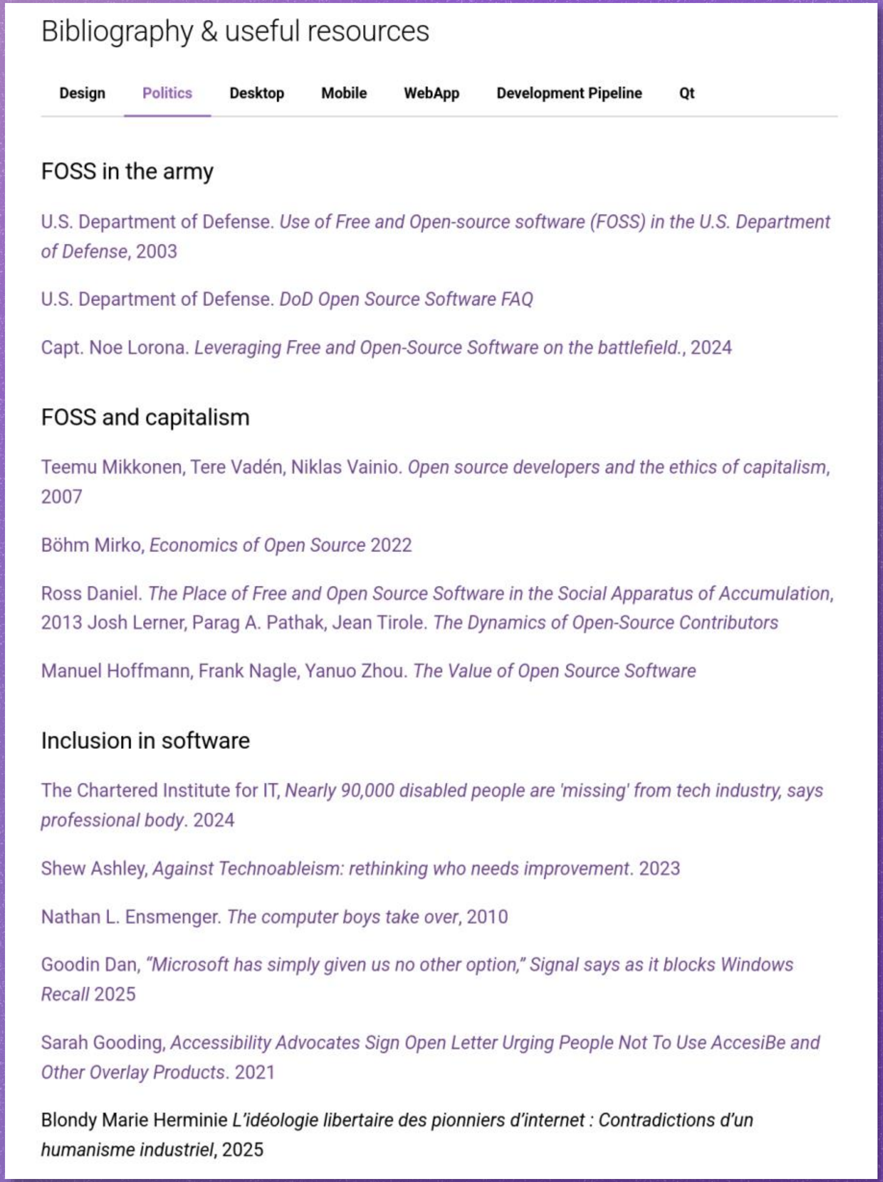
The A11Y Project. *Software, books, blogs, online tools, etc.*

McRuer Robert. *Crip Theory : Cultural Signs of Queerness and Disability*, 1966

Microsoft. *Accessibility (Design basics)*

Bibliography and useful resources

Politics



Bibliography & useful resources

FOSS in the army

U.S. Department of Defense. *Use of Free and Open-source software (FOSS) in the U.S. Department of Defense*, 2003

U.S. Department of Defense. *DoD Open Source Software FAQ*

Capt. Noe Lorona. *Leveraging Free and Open-Source Software on the battlefield.*, 2024

FOSS and capitalism

Teemu Mikkonen, Tere Vadén, Niklas Vainio. *Open source developers and the ethics of capitalism*, 2007

Böhm Mirko, *Economics of Open Source* 2022

Ross Daniel. *The Place of Free and Open Source Software in the Social Apparatus of Accumulation*, 2013 Josh Lerner, Parag A. Pathak, Jean Tirole. *The Dynamics of Open-Source Contributors*

Manuel Hoffmann, Frank Nagle, Yanuo Zhou. *The Value of Open Source Software*

Inclusion in software

The Chartered Institute for IT, *Nearly 90,000 disabled people are 'missing' from tech industry, says professional body*. 2024

Shew Ashley, *Against Technoableism: rethinking who needs improvement*. 2023

Nathan L. Ensmenger. *The computer boys take over*, 2010

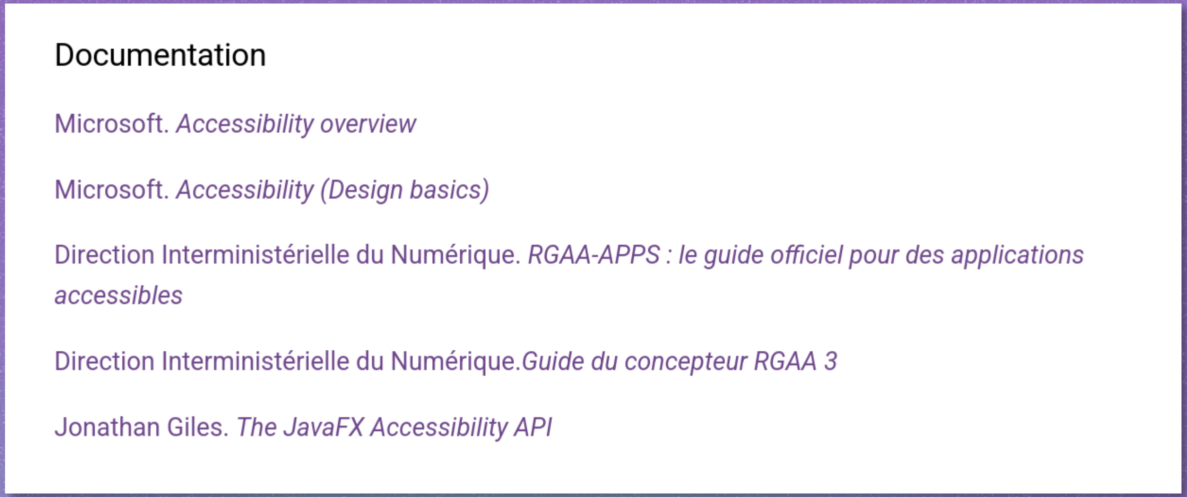
Goodin Dan, *"Microsoft has simply given us no other option," Signal says as it blocks Windows Recall* 2025

Sarah Gooding, *Accessibility Advocates Sign Open Letter Urging People Not To Use AccesiBe and Other Overlay Products*. 2021

Blondy Marie Herminie *L'idéologie libertaire des pionniers d'internet : Contradictions d'un humanisme industriel*, 2025

Bibliography and useful resources

Desktop



Documentation

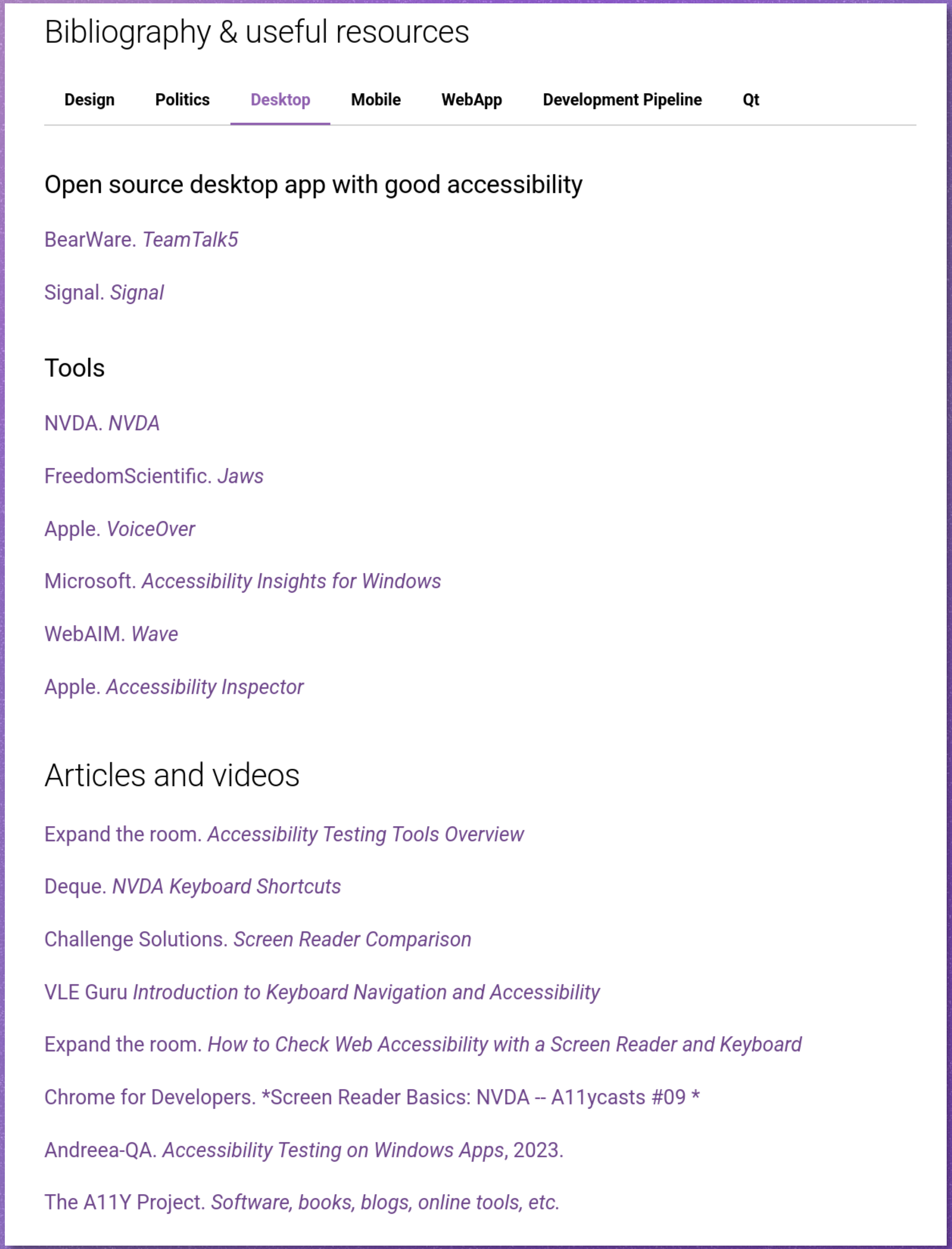
Microsoft. *Accessibility overview*

Microsoft. *Accessibility (Design basics)*

Direction Interministérielle du Numérique. *RGAA-APPS : le guide officiel pour des applications accessibles*

Direction Interministérielle du Numérique.*Guide du concepteur RGAA 3*

Jonathan Giles. *The JavaFX Accessibility API*



Bibliography & useful resources

Open source desktop app with good accessibility

BearWare. *TeamTalk5*

Signal. *Signal*

Tools

NVDA. *NVDA*

FreedomScientific. *Jaws*

Apple. *VoiceOver*

Microsoft. *Accessibility Insights for Windows*

WebAIM. *Wave*

Apple. *Accessibility Inspector*

Articles and videos

Expand the room. *Accessibility Testing Tools Overview*

Deque. *NVDA Keyboard Shortcuts*

Challenge Solutions. *Screen Reader Comparison*

VLE Guru *Introduction to Keyboard Navigation and Accessibility*

Expand the room. *How to Check Web Accessibility with a Screen Reader and Keyboard*

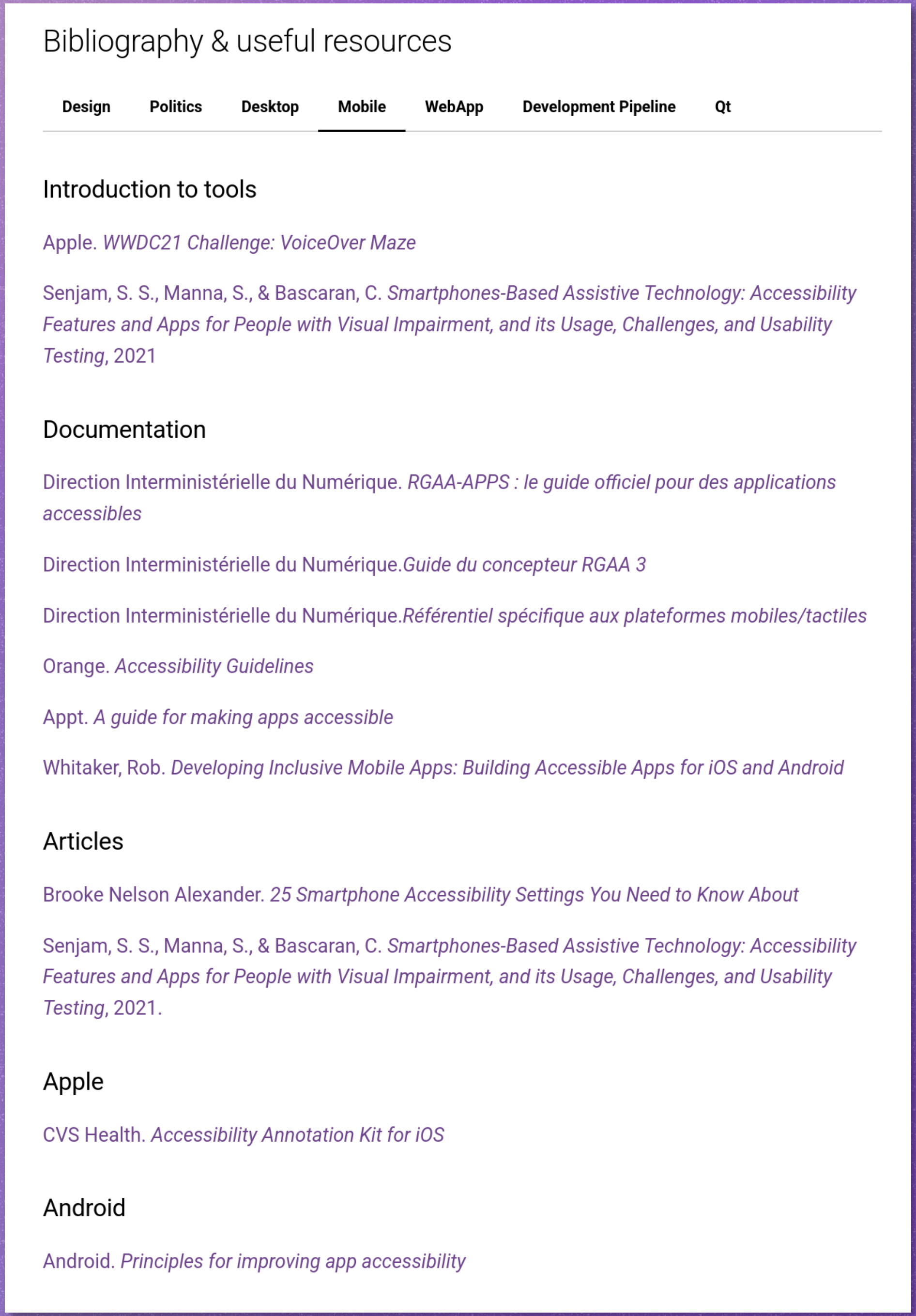
Chrome for Developers. *Screen Reader Basics: NVDA -- A11ycasts #09 *

Andreea-QA. *Accessibility Testing on Windows Apps*, 2023.

The A11Y Project. *Software, books, blogs, online tools, etc.*

Bibliography and useful resources

Mobile



Bibliography & useful resources

Introduction to tools

Apple. *WWDC21 Challenge: VoiceOver Maze*

Senjam, S. S., Manna, S., & Bascaran, C. *Smartphones-Based Assistive Technology: Accessibility Features and Apps for People with Visual Impairment, and its Usage, Challenges, and Usability Testing*, 2021

Documentation

Direction Interministérielle du Numérique. *RGAA-APPS : le guide officiel pour des applications accessibles*

Direction Interministérielle du Numérique.*Guide du concepteur RGAA 3*

Direction Interministérielle du Numérique.*Référentiel spécifique aux plateformes mobiles/tactiles*

Orange. *Accessibility Guidelines*

Appt. *A guide for making apps accessible*

Whitaker, Rob. *Developing Inclusive Mobile Apps: Building Accessible Apps for iOS and Android*

Articles

Brooke Nelson Alexander. *25 Smartphone Accessibility Settings You Need to Know About*

Senjam, S. S., Manna, S., & Bascaran, C. *Smartphones-Based Assistive Technology: Accessibility Features and Apps for People with Visual Impairment, and its Usage, Challenges, and Usability Testing*, 2021.

Apple

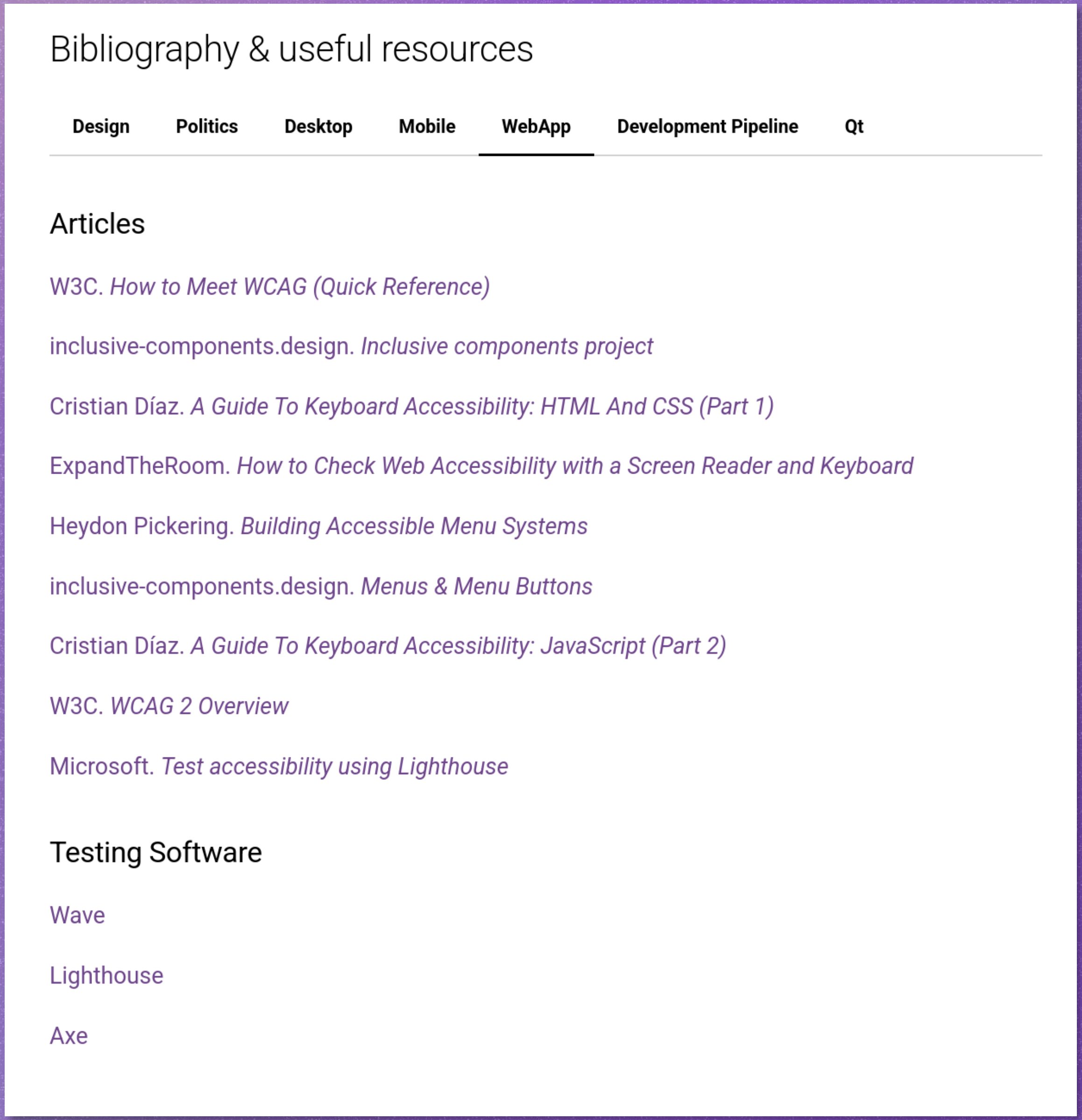
CVS Health. *Accessibility Annotation Kit for iOS*

Android

Android. *Principles for improving app accessibility*

Bibliography and useful resources

Webapp



Bibliography & useful resources

Articles

W3C. *How to Meet WCAG (Quick Reference)*

inclusive-components.design. *Inclusive components project*

Cristian Díaz. *A Guide To Keyboard Accessibility: HTML And CSS (Part 1)*

ExpandTheRoom. *How to Check Web Accessibility with a Screen Reader and Keyboard*

Heydon Pickering. *Building Accessible Menu Systems*

inclusive-components.design. *Menus & Menu Buttons*

Cristian Díaz. *A Guide To Keyboard Accessibility: JavaScript (Part 2)*

W3C. *WCAG 2 Overview*

Microsoft. *Test accessibility using Lighthouse*

Testing Software

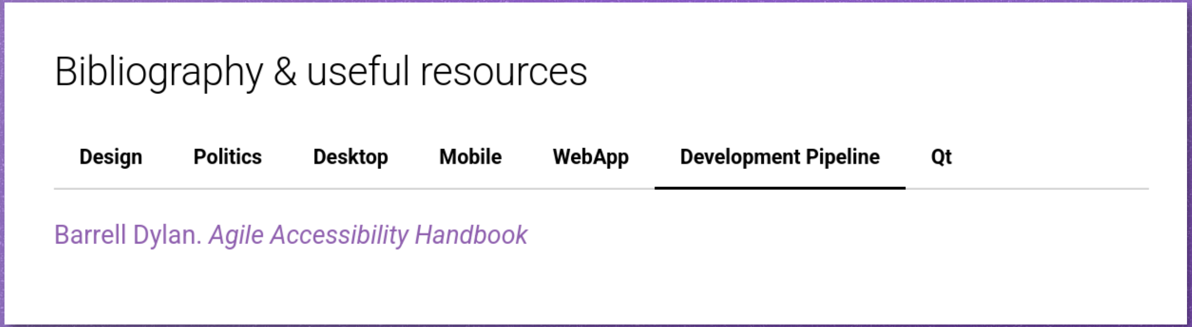
Wave

Lighthouse

Axe

Bibliography and useful resources

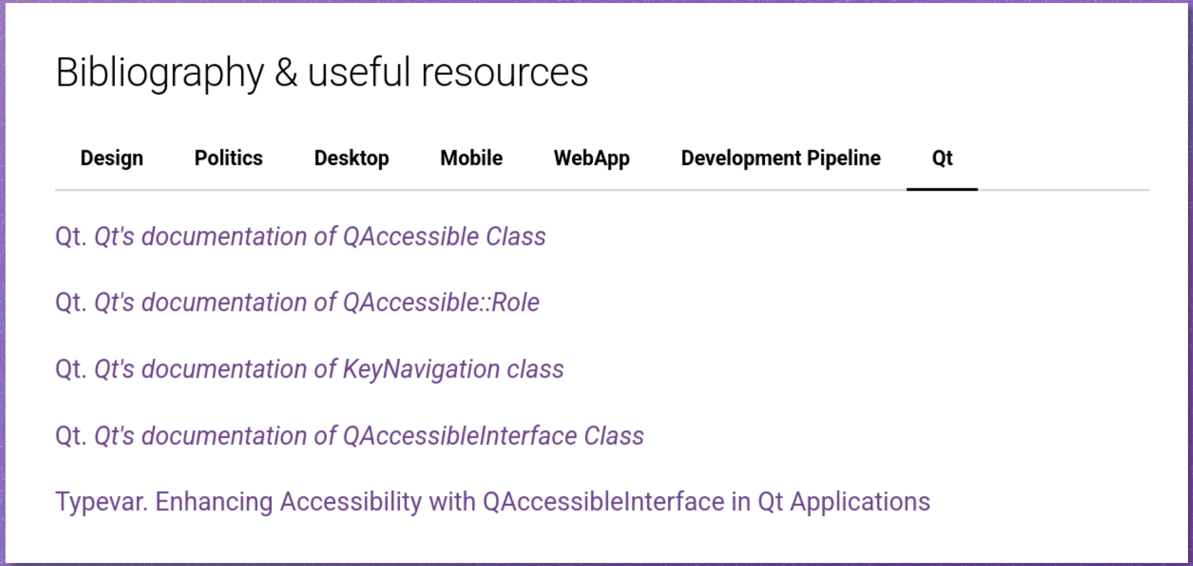
Development pipeline


Bibliography & useful resources

Barrell Dylan. *Agile Accessibility Handbook*

Bibliography and useful resources

Qt



Bibliography & useful resources

Qt. *Qt's documentation of QAccessible Class*

Qt. *Qt's documentation of QAccessible::Role*

Qt. *Qt's documentation of KeyNavigation class*

Qt. *Qt's documentation of QAccessibleInterface Class*

Typevar. Enhancing Accessibility with QAccessibleInterface in Qt Applications

Where you go next!

To conclude and go further

Workshop

For each team:

- Choose your inclusiveness coach
- Determine the key blockers for the inclusiveness of your product
- Define your epics and tasks

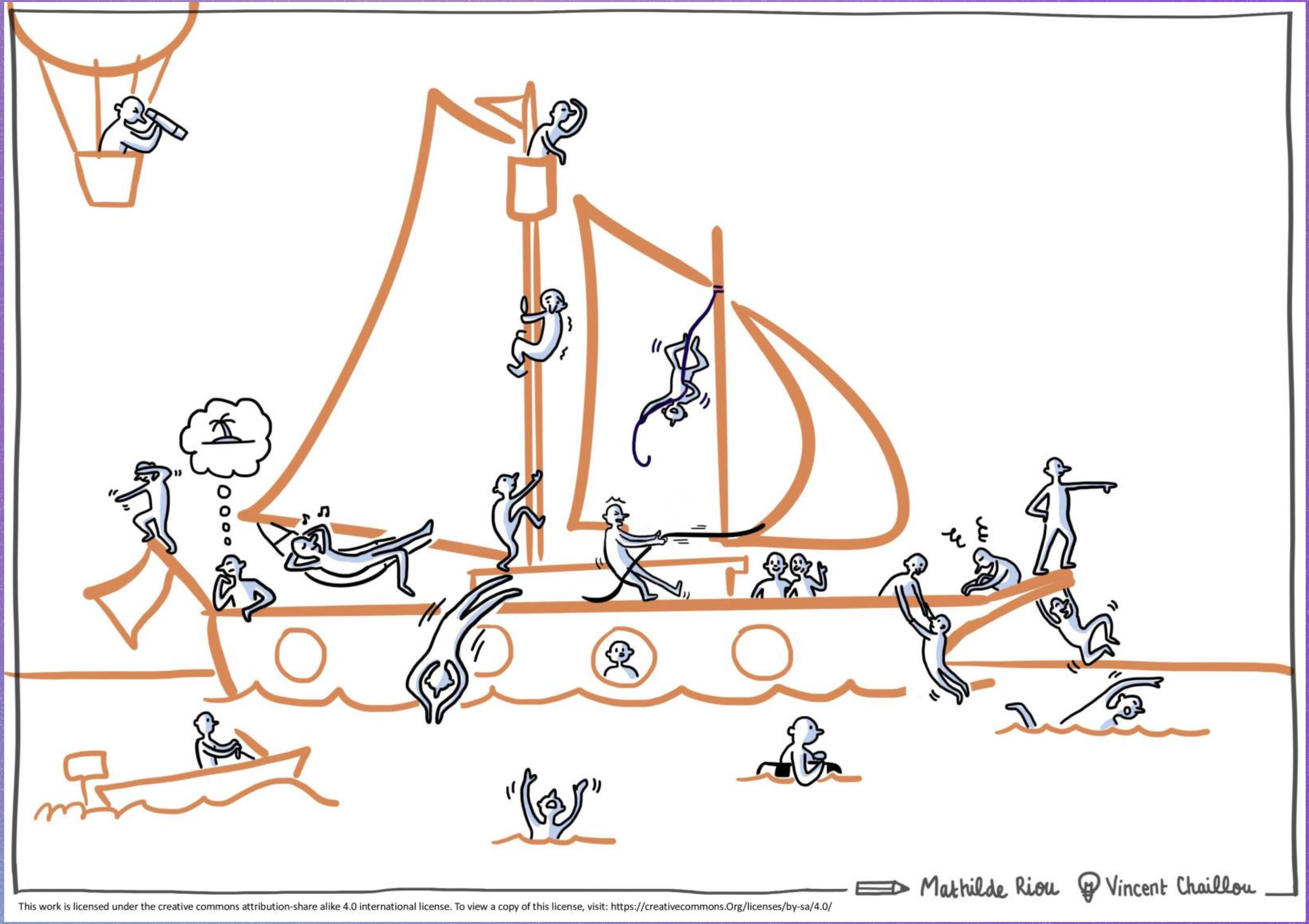
Collective sharing

To conclude and go further

What did other learn?

1. One thing that you learned
2. One thing that you felt
3. One thing that will change in your work

Together on a boat



Thank you for listening