

**Blaise**



# Building an accessible questionnaire



Blaise



# Topics

- Accessibility in a nutshell
- WCAG
- Blaise and accessibility
- Best practices
- Demo



# Accessibility in a nutshell

- People deal with a great range of issues
  - Permanent
    - Physical disabilities
    - Neurological disabilities
  - Temporary
    - Situational events
  - Socio-economic
    - Environment
    - Culture
    - Financial



# Accessibility in a nutshell

- Difficulties in accessing content
  - Inadequate text size or color contrast
  - Missing alt text for images
  - Lack of captioning on videos
  - Illogical tab order
- Difficulties in responding to the content
  - Unclear error messages
  - Inadequate size of input boxes



# Accessibility in a nutshell

- Assistive technology
  - Screen readers
  - Magnifiers
  - Ergonomic keyboards
- Accessible design
  - Which allows for interaction with assistive technology



# Accessibility in a nutshell

- End user expectations
  - Multi-device support
    - End-users expect your questionnaire to sync/interact with different gadgets they own
  - A clear and logical path through your questionnaire
- If designed well, assistive technologies enhance, rather than impede, the efficiency of daily tasks
  - Industry standards and best practices play a crucial role in achieving this



# Accessibility in a nutshell

- (Web) accessibility
  - Ensures there are no barriers that prevent interaction with or access to questionnaires on the World Wide Web
  - Ensures a complete and accurate representation of the population by making sure everyone can participate in the survey
- Laws that require accessible websites
  - Proactive approach
    - Offer a customizable experience
      - Users can adjust font size, contrast, and navigation based on their needs



# Web Content Accessibility Guidelines (WCAG)

- Part of a series of global web accessibility guidelines
  - <https://www.w3.org/WAI/standards-guidelines/wcag/>
- Published by the [Web Accessibility Initiative](#) (WAI) of the [World Wide Web Consortium](#) (W3C)
  - This is the main international standards organization for the Internet
- A set of recommendations for making Web content more accessible
  - primarily for people with disabilities
  - but also for all user agents
    - including highly limited devices, such as smartphones or smartwatches



# Web Content Accessibility Guidelines (WCAG)

- The guidelines cover the following topics:
  - Perceivable
    - Can everyone perceive the content in some way or another?
  - Operable
    - Can everyone interact with it?
  - Understandable
    - Can everyone understand what is meant?
  - Robust
    - Is the behavior the same every time?
  - Conformance
    - Does it meet the success criteria?



# Web Content Accessibility Guidelines (WCAG)

- Success criteria
  - This only includes issues which cause people with disabilities proportionally greater problems than they would a person without a disability
    - It does not include 'normal' bad design choices
  - Testable success criteria are provided for each guideline
    - To see if your questionnaire conforms to the requirements
  - Three levels of conformance
    - A (lowest)
    - AA
      - This is the level most countries require
    - AAA (highest)



# What can Blaise do?

- It allows for accessible interviewing modes
  - Telephone interviewing (CATI)
  - Personal interviewing (CAPI)
- These interview modes help with accessibility in case of
  - Illiteracy
  - Bad motor skills
  - Visual disabilities
  - ...



# Blaise and CAWI

- Web interviewing (CAWI)
  - This is not accessible by design
  - But it is possible to adjust settings to comply with the WCAG guidelines



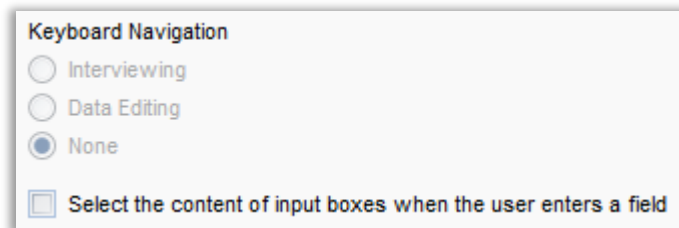
# Out-of-the-box accessible layout

- Why does Blaise not offer this?
  - Because people differ all over the world
    - What may be accessible to someone in America might be unreadable to someone in Africa
      - Based on their cultural and native environments
    - The best Blaise can do is offer functionality to make it accessible for your respondents



# Example

Navigation examples	Blaise	Web standard (keyboard navigation)
Between questions	Mouse, (Shift +) tab, (Shift +) Enter	(Shift +) Tab
Within questions (enumeration/set)	Mouse, (Shift +) tab, (Shift +) Enter	Arrows
Selecting/deselecting a checkbox	Mouse, Space bar	Space bar
Selecting/deselecting a radio button	Mouse, Space bar	Space bar
Links	Mouse, Space bar, Enter	Enter
Buttons	Mouse, Space bar, Enter	Space bar, Enter
Next Page	Enter or through a button	Only through a button



Keyboard Navigation

Interviewing

Data Editing

None

Select the content of input boxes when the user enters a field

- These are web standards
  - People who use accessibility technology are used to this other way of navigating



# Blaise and Accessibility

- Basic functionalities
  - Colors
  - Fonts
  - Sizes
    - elements
    - adaptive design
  - Navigation options
- ...but different requirements



# Specific accessibility functionalities

- Disabling Auto Focus (Data Entry Setting)
  - There will be no automatic focus on the first (required) question
- Heading Levels (Source code and/or Resource Database)
  - <H1> to <H6>
- Timeout Warning (Resource Database and Data Entry Setting)
  - Session time out
- Navigation (Data Entry Setting)
  - Keyboard navigation options
- Skip Links (Resource Database and Data Entry Setting)
  - For jumping directly to the requested part of the page
- Validation (Layout Designer)
  - Blaise will tell you if certain things are not accessible



# Data entry settings – Accessibility options

The screenshot shows the 'AccessibilityDemo Solution - Blaise Control Centre' interface. The 'Data Entry Settings' panel is active, with the 'Accessibility options' section highlighted by a red box. A callout box titled 'Accessibility options' provides a detailed view of the settings:

- Use accessibility optimization
- Disable Auto Focus
- Use Skip Links
- Optimize for readability without stylesheets



# Data entry settings – Accessibility options

- **Disable AutoFocus**
  - When a page is loaded, no focus is set
- **Use Skip Links**
  - Check this box to tell the system you want to use Skip Links
- **Optimize for readability without stylesheets**
  - Replaces some Blaise elements with native browser versions
    - It removes the custom HTML that Blaise uses to make controls have more functionalities or a nicer appearance
  - Note that this is mostly meant for fully blind respondents
    - It's best to offer a separate layout for this, as things may differ visually from your design
- **Session Timeout (Session tab)**
  - Server timeout = 20 minutes by default
  - Use in combination with a timer and a warning text



# Accessibility properties in the Resource Database

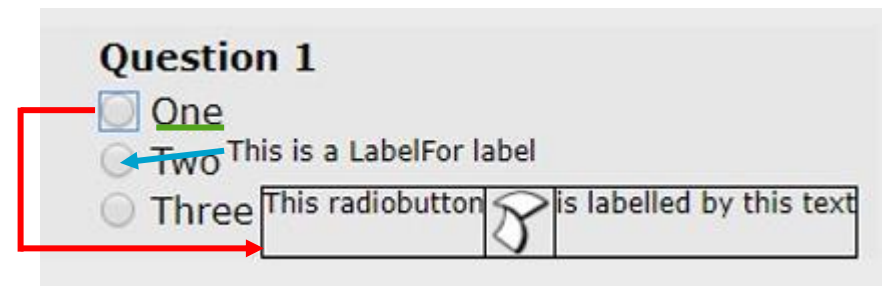
- Check info properties
  - AnswerRequired: RoleName, Text, TextKey, TextSource
  - RangeError: RoleName, Text, TextKey, TextSource
- Appearance properties
  - Textbox
  - Category buttons / radiobuttons / checkboxes
  - Date picker / date wheel / time picker / dropdown list
- BlankChar
  - For Edit Masks to help with understanding what needs to be filled in



# Accessibility properties in the Resource Database

## ■ LabelFor

- This is always a text element
- It has a functional purpose. Its accessible purpose is a bonus
- Can only point to a control in the same template
- The screen reader will use the text that is defined for this element when the mentioned element is focused, see e.g. the Enumeration template in the Visually Impaired sample
- Html: for



## ■ LabelledBy (preferred)

- Can only point to a control (container) in the same template
- The screen reader will use the text from the control this points at when this element is focused
- Html: aria-labelledby



# Accessibility properties in the Resource Database

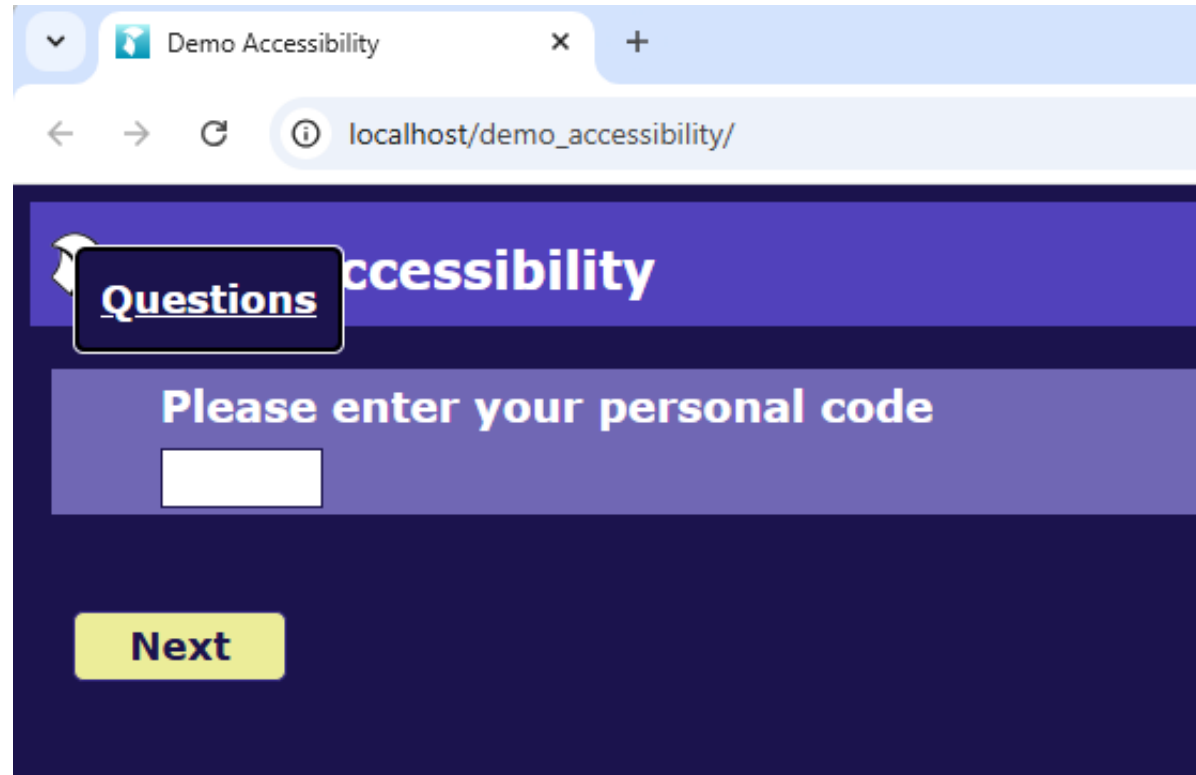
- ScreenReaderText properties
  - Alternative to LabelFor and LabelledBy
  - RoleName, Text, TextKey, TextSource
  - Html: aria-label
- NOTE: LabelFor, LabelledBy and ScreenreaderText cannot be used in combination with each other.
  - LabelledBy wins if you specify a ScreenreaderText as well



# Accessibility properties in the Resource Database

- SkipLink properties

- AccessibilityRole
- IsSkipLinkTarget
- SkipLinkDescription
- SkipLinkIndex
- SkipLinkName



- Don't forget to turn on UseSkipLinks in Data Entry Settings



- TabIndex

- To change the tab order (not recommended: only change this if you have a very good use case)

- Timer control

- To set alerts for when a session is about to expire
- Don't forget to set Session Timeout in Data Entry Settings if applicable
  - Server time-out is 20 minutes by default

## Demo Accessibility

For security reasons, your session will be ended within a few minutes. Please click Continue to keep your session alive.

Continue



# ARIA Attributes

- Assistive technology uses so-called Aria Attributes
  - Listed in the html as **aria-`<attribute>`=“`<value>`”**, e.g. `aria-hidden=“true”`

Aria Attributes	Blaise Property	Where can you define this?	Purpose
label	ScreenReaderText	Resource Database (control) Source code (RoleText)	Screen readers will read these texts if there is no aria-labelledby attribute present
labelledby	LabelledBy	Resource Database (control)	A screen reader will read the text associated with the element mentioned in this property when this control is focused. The text needs to be visible on screen.
disabled	Enabled	Resource Database (control)	This control can be interacted with or not, depending on the value of this property
hidden	Visibility	Resource Database (control)	This control is visible or not, depending on the value of this property



# HTML Attributes

- Html has additional attributes which can also be used for accessibility
  - Listed in the html as **<attribute>=“<value>”**, e.g. role=“presentation”

Html attributes	Blaise Property	Where can you define this?	Purpose
labelledby	LabelledBy	Resource Database (control)	Tells the browser this element belongs to the element mentioned in its value
for	LabelFor	Resource Database (control)	Tells the browser this element belongs to the element mentioned in its value. In Blaise it holds the text for the mentioned control.
role	AccessibilityRole	Resource Database (control in template)	Tells the web page the function of the section/element
tabindex	TabIndex	Resource Database (control in template)	Defines the order in which you can tab through the site



# Best practices during development

- Always take accessibility into account from the start
- Offer simplified texts for easier understanding
  - Or better: use them for everyone
- Limit the number of questions
  - Per page
  - In total
- Add clear images plus descriptions that help get the message across
  - If the message is otherwise difficult to grasp
- Use the same style (language, layout) everywhere
  - Especially within new waves of the same project
- Check that your questions and layout are methodologically correct
  - Do a pilot with a panel e.g. to test this if you are not sure



# Best practices – accessibility settings

- Give a warning if the session is about to run out
  - with an option to continue!
- It is best to use the defaults for AccessibilityRole, TabIndex and SkipLinkIndex specifically
  - If you change a setting, you will be responsible for making sure everything is still accessible
  - This sometimes means using combinations of settings to get it right
- Good to know: Assistive technology will follow the DOM of the HTML



# Best practices – Accessibility settings

- It is recommended to make sure your page follows a logical order
  - use `tabindex = -1` for elements that should not be tabbable
  - leave the rest at 0
- Add screen reader texts
  - this includes alt texts for images
    - If they are decorative only, these should not be tabbable and they don't need an alt text
  - For tables with a header text, row texts and category texts, make sure to concatenate all texts in the `ScreenReaderText` of a radiobutton
- Add Skip Links
  - Make sure you have 2 or 3 sections on your page to jump to
    - E.g. Questions (main grid), Navigation (navigation grid), Links (footer with links to relevant sites)



# Some practical examples

- Your company logo should be tabbable (tabindex = 0 or 1), with a clear description so that respondents know this is a legit survey from an official company
  - To do this:
    - Set role to **banner**
    - Set Visibility to **Visible**
    - Add a good, clear **ScreenReaderText**
      - e.g. "Official logo for the Dutch Central Bureau of Statistics"



# Some practical examples

- Make sure that headers use the correct headings
  - Make sure to start at <H1> on each page, then work your way down to <H6> in a logical order, e.g.:

Datamodel Title: <H1>

DATAMODEL Life Survey “<H1>Life Survey</H1>”

Section: <H2>

“<H2>Personal Questions</H2>”

Subsection: <H3>

“<H3>Medical History</H3>

This section will ask some questions about your medical history.”

Subsection: <H3>

“<H3>Family</H3>

This section will ask some questions about your relation to your family.”

Section: <H2>

“<H2>Work Questions</H2>”



# Some practical examples

- For a table with a list of questions with the same answer options, your screen reader needs to read out all relevant texts for each radio button
- Choose Literal for ScreenReaderTextSource
- Add an expression to the ScreenReaderText of the radioButton:

```
Field.Parent.GetRoleText('Question') +  
Field.GetRoleText('Question') +  
Category.GetRoleText('Category')
```

The screenshot shows the Visual Studio WPF Designer interface for a 'Category RadioButton'. The 'Properties' window is open, and the 'ScreenReaderText' property is highlighted in green. The 'ScreenReaderTextSource' property is set to 'Literal'. Red arrows point to the 'ScreenReaderText' and 'ScreenReaderTextSource' properties. The 'ScreenReaderText' property is currently empty, and the 'ScreenReaderTextSource' property is set to 'Literal'. The 'ScreenReaderRole' property is set to 'Category'. The 'ScreenReaderTextK' property is set to 'None'. The 'ScreenReaderTextS' property is set to 'Literal'. The 'ShowFocus' property is set to 'True'. The 'SpeedKey' property is set to 'None'. The 'OutlineWidth' property is set to '2'. The 'RowIndex' property is set to '0'. The 'RowSpan' property is set to '1'. The 'Name' property is set to 'RadioButton'. The 'Description' property is set to 'Category Radio button'. The 'Image' property is set to 'None'. The 'References' property is set to 'None'.

# Some practical examples

- Links should be clearly different from the rest of the text
  - When not clicked
  - When clicked
- This can be done by giving links a different font color and underlining
  - This is the default way
    - Most people will recognize underlined, colored text as a link, as it is a web standard
    - But make sure the colors you use in your house style have enough contrast
  - If you adjust links, make sure they still differ in at least 2 ways from other text
    - When clicked, they should change color
  - Make sure to check color contrast for both the clicked and the unclicked link!



# Testing for Accessibility

- WCAG-EM Report Tool (<https://www.w3.org/WAI/eval/report-tool/>) (paid)
  - For experienced accessibility evaluators
  - Structured report, but no (automated) checks for accessibility
- ANDI (free)
  - <https://www.ssa.gov/accessibility/andi/help/install.html>
  - Add a Content Security Policy to Blaise using a .bcsp file
- SilkTide (browser extension for Chrome and Edge) (free)
  - <https://silktide.com/toolbar/>
- WAVE (browser extension for Chrome, Firefox and Edge) (free)
  - <https://wave.webaim.org/extension/>
- DevTools (F12 in browser | Console | Issues) (free)



# Testing for Accessibility

- Keyboard
  - Test navigation
- Screenreaders:
  - NVDA – Non-Visual Desktop Access (free)
    - <https://www.nvaccess.org/>
  - JAWS - Job Access With Speech (paid)
    - <https://www.freedomscientific.com/products/software/jaws/>
  - Screen readers for specific operating systems:
    - Windows: Narrator (free)
    - macOS/iOS: VoiceOver (free)
    - Android: TalkBack (free)
    - Linux: Orca (free)

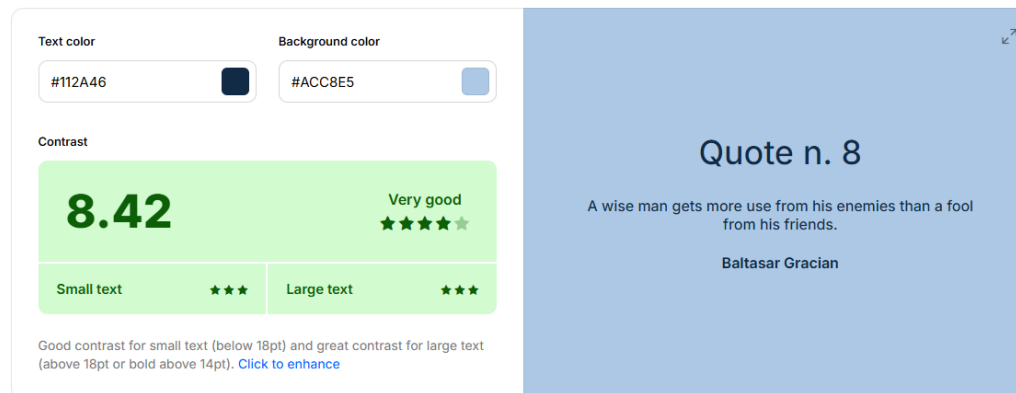


# Testing for Accessibility

- Color Contrast testers
  - E.g. <https://colors.co/> → Contrast Checker (scroll down a bit to find this)
  - Contrast should be 4,5:1 or 3:1 for larger fonts (18pt or 14 pt bold)

## Color Contrast Checker

Calculate the contrast ratio of text and background colors.



The screenshot shows the Color Contrast Checker interface. On the left, there are two color pickers: 'Text color' with hex code #112A46 and 'Background color' with hex code #ACC8E5. Below these, the 'Contrast' section displays a large green box with the value '8.42' and the text 'Very good' with five stars. Underneath, two smaller green boxes show 'Small text' with three stars and 'Large text' with three stars. At the bottom, a note states: 'Good contrast for small text (below 18pt) and great contrast for large text (above 18pt or bold above 14pt). [Click to enhance](#)'. On the right, a blue rectangular area simulates the background with the text: 'Quote n. 8', 'A wise man gets more use from his enemies than a fool from his friends.', and 'Baltasar Gracian'.



# Checklist

- Survey design decisions
  - Design in a logical order
  - Trade off between fully accessible and visually appealing
    - If known upfront consider offering a separate URL
- Check color contrasts
  - 4,5:1 or
  - 3:1 for large texts (18pt or 14 pt bold)
- Define AccessibilityRole for landmarks
  - Banner (header), Main (questions), Navigation (next button etc), Content Info (footer)
- Define SkipLink containers
  - Point to landmarks
    - Set SkipLink properties
  - Set UseSkiplink to true
- Use of LabelledBy or LabelFor
  - The label should be visible for a screen reader to read it
  - Point to the name of the control it is labeled by or for
    - in same template
- ScreenReaderTexts
  - Most important for enumerations
  - Concatenate textroles
- Define instructive error texts
  - Refer to the question and add a suggestion on how to resolve the error
- Define alt text for images
  - If meaningful
  - Use ScreenReaderText textrole (.blax) or property (.blrd)
- Timeout warning
  - Set session time out, set a timer at least 20 sec before end of session, add CurrentPage() action under button
- Test your instrument
  - Screen reader (NVDA, JAWS)
  - ANDI (or SilkTide or similar)
  - WCAG-EM Report Tool
  - <https://www.w3.org/WAI/WCAG22/quickref/?versions=2.2>



# Conclusion

- Blaise can handle designing an accessible questionnaire
- But your organization will have to make a choice on how to use it
  - Based on the local laws, and requirements from the organization
- The options are:
  - One fully accessible layout
    - abled respondents might become frustrated
  - One layout, where a compromise is chosen
    - neither the abled nor the disabled will have the best experience
  - Multiple optimized layouts, where each respondent can pick a layout that meets their needs best
    - compromises on inclusivity



# Final note

The goal isn't "all or nothing"

Every improvement you make will help make your questionnaires more accessible

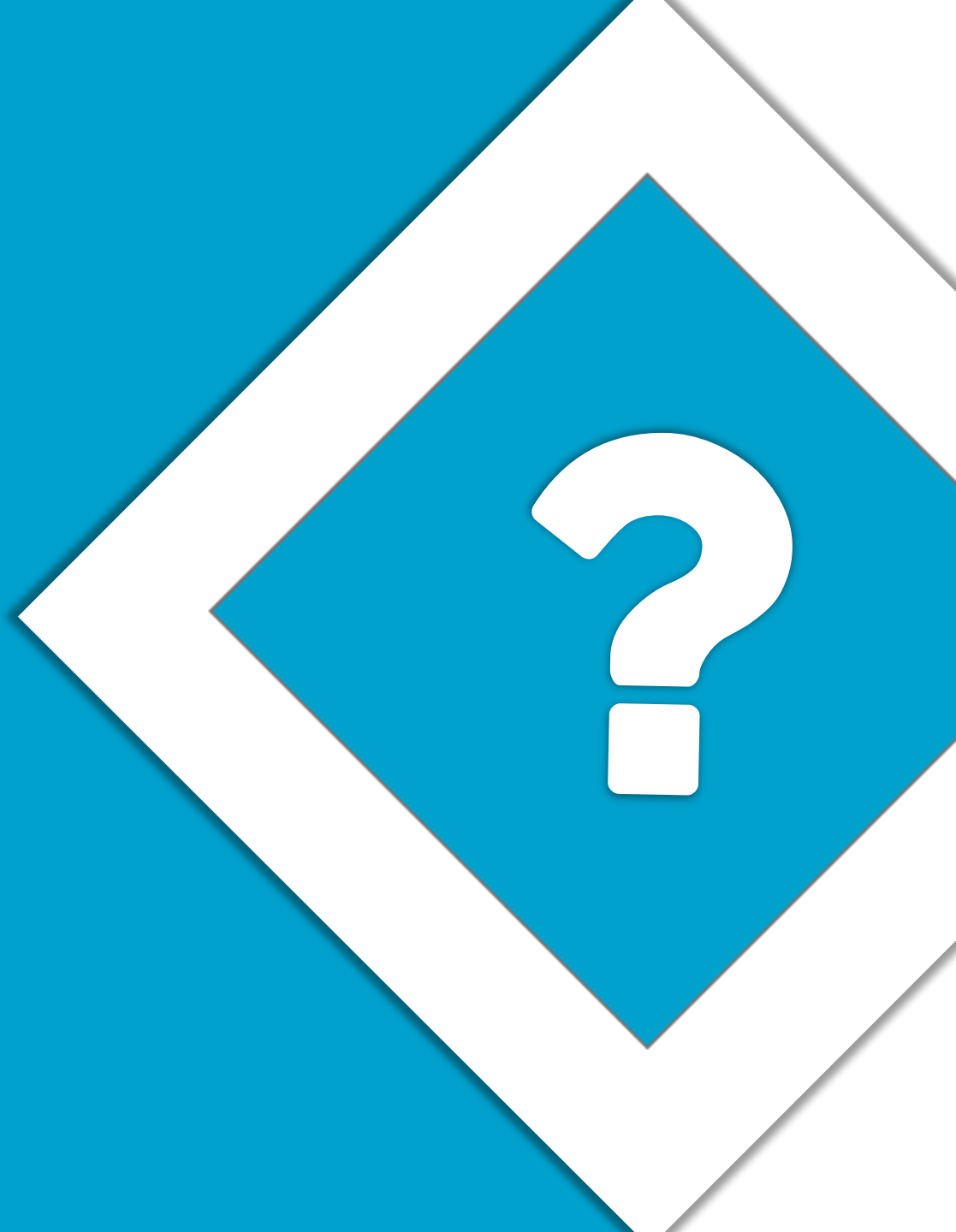
But keep in mind that you comply with your local accessibility laws.  
Most often this means complying with WCAG 2.2 AA at the least.



DEMO



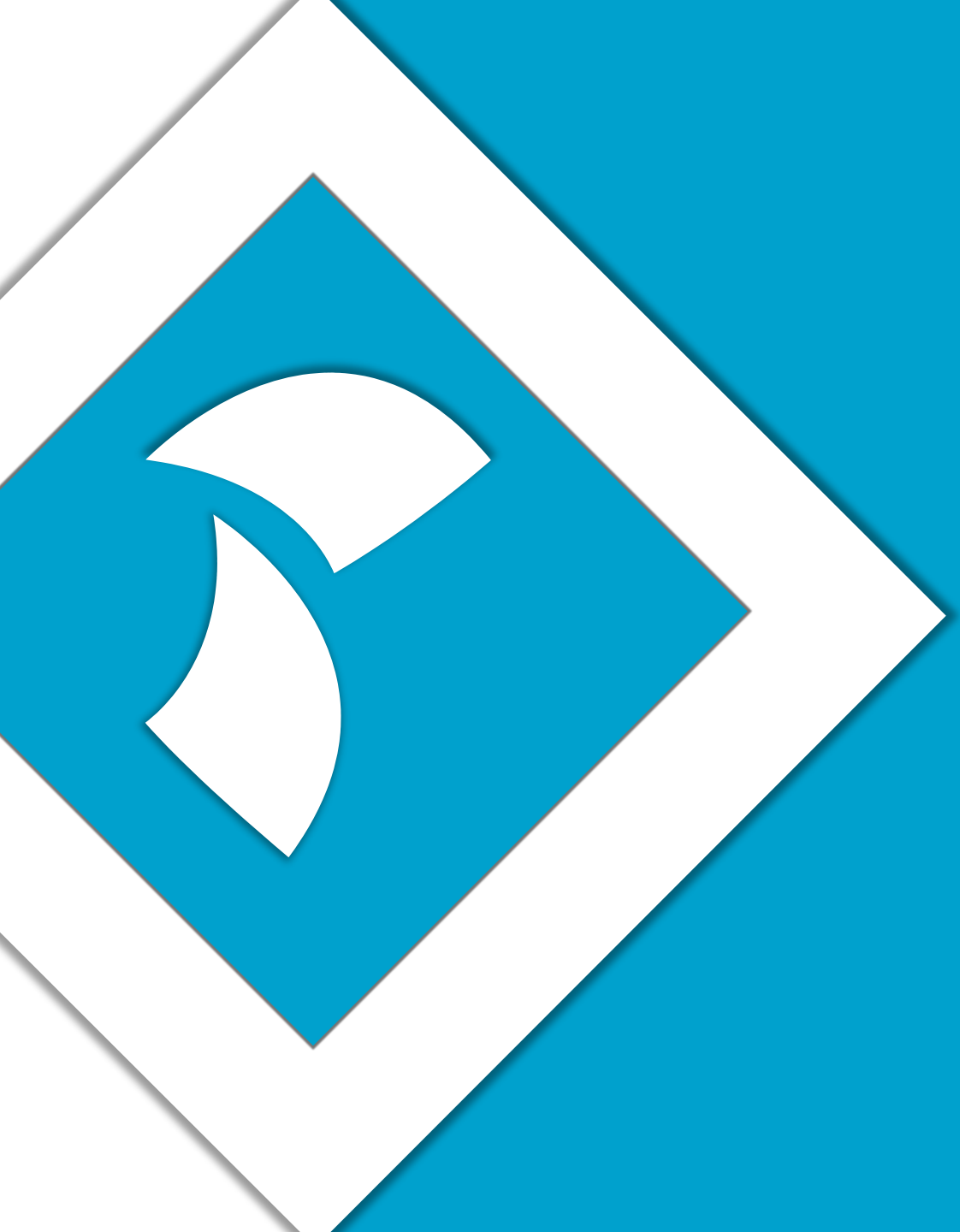
Questions



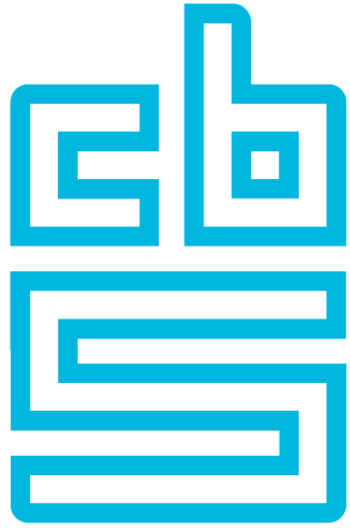
# For more detailed information, check out the Help Section: Accessibility

- Blaise site | Online Documentation 5 (<http://help.blaise.com>)
- Click the link in the upper right corner of the Control Centre
- Contact Blaise Support: [blaise@cbs.nl](mailto:blaise@cbs.nl)
- Or use the portal of our JIRA ticketing system <https://blaisesupport.cbs.nl>





**Thank you  
for your time**



**Blaise**

**Gaining deeper understanding**



[www.blaise.com](http://www.blaise.com)



[blaise@cbs.nl](mailto:blaise@cbs.nl)



[@Blaise5](https://www.youtube.com/@Blaise5)