

Valget er ditt

Universell utforming

Dette dokumentet beskriver hvordan Valget er ditt tilfredsstiller kravene til universell utforming i W3C WAI WCAG 2.1 AA.

Fra krav til leveransen:

Spillet skal utformes i samsvar med standard Web Content Accessibility Guidelines 2.0 (WCAG 2.0)/NS/ISO/IEC 40500:2012, på nivå A og AA så langt det er mulig.

Contents

| | |
|-----------------------------|---|
| 1. Perceivable | 2 |
| 2. Operable | 4 |
| 3. Understandable | 5 |
| 4. Robust..... | 5 |
| 5. Testing methods | 6 |
| Screen Readers | 6 |
| W3C Markup Validation | 6 |

1. Perceivable

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

| Guideline | Level | Solution | Tested |
|---|-------|--|---|
| 1.1.1 Non-text Content | A | All still images have alt-texts. All graphic buttons without text have titles or aria labels that describe their function. | Yes, with screen readers |
| 1.2.1 Audio-only and Video-only | A | Audio-only and Video-only content are not present in the program. | |
| 1.2.2 Captions (pre-recorded) | A | All videos in the program have captions. | Yes |
| 1.2.3 Audio Description or Media Alternative (pre-recorded) | A | The video sound tracks contain all essential information, so an extra audio description of the visual content is not necessary. | |
| 1.2.4 Captions (live) | AA | Live content is not present in the program. | |
| 1.2.5 Audio Description (Pre-recorded) | AA | The video sound tracks contain all essential information, so an extra audio description of the visual content is not necessary. | |
| 1.3.1 Info and Relationships | A | Standard HTML5 semantic elements mark up the structure. Aria labels identify any non-standard use of elements (e.g. option buttons in multiple-choice questions and visual feedback). | Yes, with screen readers |
| 1.3.2 Meaningful Sequence | A | The DOM order matches the visual order. | Yes, with screen readers |
| 1.3.3 Sensory Characteristics | A | Instructions do not rely solely on sensory characteristics. | |
| 1.3.4 Orientation | AA | The content does not restrict its view and operation to a single display orientation, such as portrait or landscape | |
| 1.3.5 Identify Input Purpose | AA | There are no input fields collecting information about the user. | |
| 1.4.1 Use of Colour | A | Colour is not the only visual means of conveying information. | |
| 1.4.2 Audio Control | A | The teacher can pause videos that start playing automatically if they need to adjust the audio volume. | Yes |
| 1.4.3 Contrast (Minimum) | AA | The visual presentation of text and images of text has a contrast ratio of at least 4.5:1. | Yes, with Web AIM WAVE and NCSU Colour Contrast Analyser plugins for Chrome |

| Guideline | Level | Solution | Tested |
|----------------------------------|-------|---|--------|
| 1.4.4 Resize text | AA | Text can be resized without assistive technology up to 200 percent without loss of content or functionality | Yes |
| 1.4.5 Images of Text | AA | The program uses text to convey information rather than images of text. | |
| 1.4.10 Reflow | AA | The content requires two-dimensional layout for usage and so does not support this criterion. | |
| 1.4.11 Non-text Contrast | AA | The visual presentation has a contrast ratio of at least 3:1 against adjacent colour(s): <ul style="list-style-type: none"> • User Interface Components • Graphical Objects | |
| 1.4.12 Text Spacing | AA | No loss of content or functionality occurs if the user increases line, paragraph, letter, or word spacing. | |
| 1.4.13 Content on Hover or Focus | AA | Hover/focus content does not obscure other content, is hoverable and persistent. | |

2. Operable

User interface components and navigation must be operable.

| Guideline | Level | Solution | Tested |
|--|-------|---|--------|
| 2.1.1 Keyboard | A | All functionality of the content is operable through a keyboard interface. | Yes |
| 2.1.2 No Keyboard Trap | A | The user can always use the tab key to move focus out of a component. | Yes |
| 2.1.4 Character Key Shortcuts | A | The program does not use keyboard shortcuts. | |
| 2.2.1 Timing Adjustable | A | There are no time limits in the program. | |
| 2.2.2 Pause, Stop, Hide | A | There is no moving, blinking, scrolling, or auto-updating information, apart from videos which can be paused. | Yes |
| 2.3.1 Three Flashes or Below Threshold | A | The program does not contain anything that flashes more than three times in any one-second period. | |
| 2.4.1 Bypass Blocks | A | The program sets the focus to the main content area each time it loads a new item, so keyboard users can access the content directly. | Yes |
| 2.4.2 Page Titled | A | Web pages have titles that describe topic or purpose. | |
| 2.4.3 Focus Order | A | The DOM is structured so focusable components receive focus in an order that preserves meaning and operability. | Yes |
| 2.4.4 Link Purpose (In Context) | A | The purpose of any links can be determined from the link text alone. | |
| 2.4.5 Multiple Ways | AA | All web pages in the program are part of a process and so do not require multiple ways to locate them. | |
| 2.4.6 Headings and Labels | AA | Headings and labels describe topic or purpose where relevant. | |
| 2.4.7 Focus Visible | AA | The keyboard focus indicator is visible. | Yes |
| 2.5.1 Pointer Gestures | A | The program does not use pointer gestures. | |
| 2.5.2 Pointer Cancellation | A | Pointer functionality does not use the down event and the program provides abort and undo mechanisms. | |
| 2.5.3 Label in Name | A | The "for" attribute is used to link labels and user interface components. | |
| 2.5.4 Motion Actuation | A | The program has no functions that are operated by device or user motion. | |

3. Understandable

Information and the operation of user interface must be understandable.

| Guideline | Level | Solution | Tested |
|---|-------|--|--------|
| 3.1.1 Language of Page | A | The program sets the HTML element's language attribute. | Yes |
| 3.1.2 Language of Parts | AA | The program does not include any parts that are not in the main language. | |
| 3.2.1 On Focus | A | No components automatically initiate a change of context when they receive focus. | Yes |
| 3.2.2 On Input | A | No components automatically change context when their settings are changed. | Yes |
| 3.2.3 Consistent Navigation | AA | Repeated navigational elements occur in the same relative order each time they appear. | Yes |
| 3.2.4 Consistent Identification | AA | The program identifies components that have the same functionality consistently. | Yes |
| 3.3.1 Error Identification | A | Users are informed if they have entered an invalid code. | Yes |
| 3.3.2 Labels or Instructions | A | All input fields have initial text describing what should be entered. | Yes |
| 3.3.3 Error Suggestion | AA | It is not possible to provide error suggestions for invalid codes. | |
| 3.3.4 Error Prevention (Legal, Financial, Data) | AA | Submissions are either reversable or confirmed. | Yes |

4. Robust

Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

| Guideline | Level | Solution | Tested |
|-------------------------|-------|---|---|
| 4.1.1 Parsing | A | All elements have complete start and end tags, are nested according to specification, do not contain duplicate attributes and use unique IDs. | Yes - using the W3C Markup Validation Service |
| 4.1.2 Name, Role, Value | A | The program uses standard HTML controls, which meet this success criterion. | |

5. Testing methods

This section details the methods used to test the solution.

W3C Markup Validation Service

We used the Markup Validation Service at <https://validator.w3.org/>. We had to use the "Validate by Direct Input" option, as this allowed us to paste in after dynamic markup generation that occurs on page load. The "Validate by URI" or "Validate by File Upload" options would test the incomplete source elements and thus report errors.

Method:

1. Choose "Validate by Direct Input"
2. Paste a document element into the test text field: <!DOCTYPE html>
3. Open "More Options" and select "Show Source" to make locating errors easier
4. Navigate to the course's start page
5. Choose "Inspect element" to open the web browser's DOM Explorer in Developer Tools
6. Copy the entire html element
7. Paste the html element after the document element
8. Click the **Check** button
9. Repeat from step 4 for each different content type in the course, i.e.:
 - Free-text questions
 - Multiple-choice questions
 - Simulation
 - Text with links
 - Text with image
 - Video

Screen Readers

We tested using the following screen readers, operating systems and browsers:

- NVDA - Windows - Internet Explorer, Firefox and Chrome
- ChromeVox - Windows - Chrome
- VoiceOver - OS X - Safari

We validated functionality for all content types in the course. We observed some differences in how different screen reader / browser combinations presented information, especially after dynamic updates to areas marked as "aria-live".

WAVE - web accessibility evaluation tool

We used the WAVE extension for Chrome to validate accessibility for all content types in the course.