

# **Content and Digital Design** **Accessibility Playbook**

|  |  |
| --- | --- |
| Date | Updates Made |
| 12/01/23 | * Images and Alternative Text section moved to <https://a11y.pearson.com/alt-text/> * Multimedia section moved to <https://a11y.pearson.com/multimedia/> |
| 12/02/22 | * Clarified character limits under alternative text * Clarified HTML coding for long descriptions * Clarified videos need cc, transcript, and AD |

Contents

[**Content and Digital Design**  **Accessibility Playbook** 1](#_Toc152573142)

[Intended Audience 4](#_Toc152573143)

[**INTRODUCTION TO ACCESSIBILITY** 4](#_Toc152573144)

[What is Web Accessibility? 5](#_Toc152573145)

[Need for Accessibility 5](#_Toc152573146)

[How people with disabilities use the web? 5](#_Toc152573147)

[Pearson Accessibility Guidelines 7](#_Toc152573148)

[WCAG 2.1 7](#_Toc152573149)

[**COLOR** 9](#_Toc152573150)

[Use color Wisely 10](#_Toc152573151)

[Color Contrast 15](#_Toc152573152)

[Contrast for Text Content 17](#_Toc152573153)

[Contrast for Non-Text Content 20](#_Toc152573154)

[Color contrast for Images 23](#_Toc152573155)

[**CONTENT AND LANGUAGE** 28](#_Toc152573156)

[Reading Order 29](#_Toc152573157)

[Abbreviations and Acronyms 31](#_Toc152573158)

[Sensory Characteristics 34](#_Toc152573159)

[Primary Language 36](#_Toc152573160)

[Secondary Language 38](#_Toc152573161)

[**FRAME AND PAGE TITLES** 40](#_Toc152573162)

[Page Titles 40](#_Toc152573163)

[Frame and iFrame Titles 45](#_Toc152573164)

[**HEADINGS** 48](#_Toc152573165)

[Heading Structure 48](#_Toc152573166)

[Heading Text 53](#_Toc152573167)

[**LISTS** 55](#_Toc152573168)

[Ordered Lists 55](#_Toc152573169)

[Unordered Lists 56](#_Toc152573170)

[Description Links 57](#_Toc152573171)

[**TABLES** 59](#_Toc152573172)

[Layout Tables 59](#_Toc152573173)

[Simple Data Tables 62](#_Toc152573174)

[Complex Data Tables 65](#_Toc152573175)

[Accessible Tables and ARIA 68](#_Toc152573176)

[**LINKS AND NAVIGATION** 70](#_Toc152573177)

[Descriptive Link Text 70](#_Toc152573178)

[Skip Links 74](#_Toc152573179)

[Landmarks 77](#_Toc152573180)

[**STEM** 80](#_Toc152573181)

[Math ML 80](#_Toc152573182)

[**ANIMATION** 83](#_Toc152573183)

[Moving and Blinking content 83](#_Toc152573184)

[Flickering Content 87](#_Toc152573185)

[**CSS AND ACCESSIBILITY** 89](#_Toc152573186)

[Text Resize 90](#_Toc152573187)

[Reflow 92](#_Toc152573188)

[Text Spacing 96](#_Toc152573189)

[Orientation 99](#_Toc152573190)

[Keyboard Focus 102](#_Toc152573191)

[**APPENDIX** 107](#_Toc152573192)

INTRODUCTION

Pearson is committed to accessibility and is taking all possible measures to make the online learning experience accessible for all learners. The Content and Digital Design Accessibility Playbook was put together to provide content producers with the guidance required for creating accessible learning experiences and offer buyers courses that are accessible out of the box.

The playbook includes techniques for implementing accessibility across different web page elements, such as headings, lists, images, multimedia, dynamic content, forms, etc. It identifies both techniques that must be adopted, and techniques that should be avoided, from an accessibility perspective. Along with the techniques, the playbook lists tips and tricks for enhancing accessibility.

The playbook comprises of examples of different Pearson online learning courses and lists possible solutions to help the team implement accessibility. Not only will the playbook serve as a valuable reference, it can also equip the team to implement accessibility while designing and developing all future Pearson courses.

### **Intended Audience**

The Content and Digital Design Accessibility Playbook is intended for:

* Authors and content writers
* Designers
* Content managers and producers
* Courseware Quality Assurance (QA) teams

## **INTRODUCTION TO ACCESSIBILITY**

Accessibility refers to making a product or service usable for as many people as possible. In the digital space, accessibility ensures that websites, applications, documents, online learning solutions etc. are usable for all users including those with disabilities. Often accessibility is viewed at something that is meant for people with disabilities. However, accessibility benefits each one of us in different situations.

Today digital accessibility has been adopted in most countries across the globe. To make accessibility happen, there are standards, guidelines and specifications that outline ways to make it a reality. Technologies include support for developing accessible solutions and the demand for accessible products is increasing by the day.

In simple terms, accessibility is ensuring that everyone can use a product or service irrespective of their abilities or situation of use. In physical space, we may refer to accessibility as providing ramps to allow people on a wheelchair come in easily similarly in the case of digital accessibility it refers to making the online solutions available for all users including those using assistive technologies.

### **What is Web Accessibility?**

Web accessibility is a practice where websites, online learning courses, web applications, documents etc. are designed, developed, and maintained in a manner that people with disabilities can use them with accessibility features turned on as well as with assistive technologies. Web accessibility makes it possible for people with disabilities to access the content, interact with the functionality, understand the content, and use the content with their assistive technology.

Accessible web content ensures that learners with mobility disabilities can enroll in online learning course without having to go through the hassle of waiting in long queues for their admissions. Accessible web content makes it possible for learners with hearing disabilities to listen to online lectures by following synchronized captions. Similarly, learners with visual impairments can perceive complex images by reading the alternate descriptions with their assistive technologies and those with learning disabilities can access the content by applying custom stylesheets.

### **Need for Accessibility**

Web plays a significant role in our day to day life. Most of our daily activities from learning, shopping, banking, recreation, employment, healthcare, and government procedures happen online. Access to web content is therefore essential for all users including for those with disabilities.

Some of the key benefits of accessibility include:

* Ensures access to essential online activities for all users
* Increase in customer base
* Helps to conform with accessibility standards and guidelines.
* Builds brand loyalty
* Drives innovation by making solutions that are accessible for wide range of users
* Adhere to legal requirements
* Contribute towards building an inclusive society.

### **How people with disabilities use the web?**

Ever thought how someone who cannot see will access online learning simulations? Or how someone who cannot listen will access a video explaining the solar system? How someone with upper limb difficulties will perform a science experiment? Technology acts as an enabler for people with disabilities and they can enroll to online learning courses, appear in an online exam, read books, participate in discussion forums, shop online, book their travel tickets, apply for credit cards and many more day to day activities on the web.

Evolution of the web has made life very easy for all of us and more so for people with disabilities as they can accomplish their tasks independently. People with disabilities use different assistive technologies to access the web either using a computer, tablet, or mobile device.

Let us learn about few of the assistive technologies used by people with disabilities to browse the web:

**Visual Disabilities**

* **Screen readers:** reads out the information displayed on a web page for their users. Users can access the information by listening to the screen reader and using a keyboard to input the data or give commands. Screen readers rely on semantic markup to identify the information for their users. Screen readers also provide their users with option to access web page elements using different keystrokes.
* **Screen magnifiers:** enlarges the information displayed on a web page for their users. Users can access the information by either using a mouse or a keyboard. Since users access the information in an enlarged form, not all the information is visible on the screen. This requires users to scroll a lot to access the information.
* **Refreshable Braille Displayers:** is a piece of hardware which can be attached to a computer, tablet, or mobile phone. It works in combination with a screen reader. The information read out by a screen reader is refreshed on the braille displayer and users can touch the braille pins and read the information. Users can read the information using the braille displayer whereas they can input the data using a keyboard. Refreshable braille displayers are used by blind and deaf-blind users.

**Hearing Disabilities:**

* Synchronized captions help people with hearing disabilities access videos and text transcripts help them to access audio-only content with ease. They do not require any assistive technology as such to access the web content.

**Learning Disabilities:**

* **Text to speech synthesizers (TTS):** reads out the text displayed on the screen as well as highlights the text. This helps users in understanding the information when it is read out to them. TTS software also provide users with options to view the information using different contrast themes.
* **Word prediction tools:** predicts words while users are typing data. These tools are found very helpful by users with learning disabilities especially while filling up online forms as they do not have to memorize spellings.

**Mobility Disabilities:**

* **Adaptive keyboards:** come in different shapes and sizes to meet user’s needs. Split keyboards (where the standard keyboard is split in two parts), one handed keyboard (which are adapted for users who have one hand), keyboards used in the bank that have multiple keys with combination of the digit 0 etc. are some of the examples of adaptive keyboards. These keyboards are used by people with mobility disabilities to meet their needs.
* **Sip-n-puff devices:** are used by people with severe mobility disabilities, such as those who have issues with upper and lower limbs. With sip-n-puff devices, people can use their breath to click on a button or a link. While typing, people using sip-n-puff devices use the onscreen keyboards (readily available in different operating systems) with their device to fill up the details.
* **Onscreen keyboards:** are virtual keyboards that are used by people who find it difficult to press the keys on a hardware keyboard. Onscreen keyboards are often used in combination with other hardware devices, such as eye tracking tools, head mouse etc.
* **Voice recognition systems:** is a software using which people can control their computer by giving voice commands. Users can click on links or buttons, dictate their details while filling up forms etc. Adding descriptive label for links and buttons allow users to give accurate voice commands while accessing information on the web.

Some of the assistive technologies are today available with different operating systems, such as Windows, Mac OS, iOS, Android, Linux, Chrome OS etc. whereas all the modern browsers also provide users with a range of accessibility features to choose from. All in all, people with disabilities have many choices when it comes to assistive technologies and accomplish their online tasks.

### **Pearson Accessibility Guidelines**

Accessibility guidelines are available for different technologies, such as web, software, PDF, Videos etc. When it comes to web content the universal guidelines all countries and companies follow is the Web Content Accessibility Guidelines (WCAG). This is true to Pearson as well. Pearson Accessibility Guidelines are aligned with the latest version of WCAG, i.e. WCAG 2.1 at Level AA.

### **WCAG 2.1**

WCAG 2.1 was published on 5th June 2018. Web Content Accessibility Guidelines (WCAG) 2.1 are developed by the World Wide Web Consortium (W3C) Web Accessibility Initiative (WAI) group to help individuals, companies, and governments across the globe to implement accessibility in their web content.

The Web Content Accessibility Guidelines provide directions for making the web content more accessible for users with different types of disabilities.

WCAG 2.1 comprises of 13 guidelines that are organized under 4 principles:

1. **Perceivable - Information and user interface components must be presentable to users in ways they can perceive.** 
   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.
   2. Time-based Media: Provide alternatives for time-based media.
   3. Adaptable: Create content that can be presented in different ways (For example, simpler layout) without losing information or structure.
   4. Distinguishable: Make it easier for users to see and hear content including separating foreground from background.
2. **Operable - User interface components and navigation must be operable.** 
   1. Keyboard Accessible: Make all functionality available from a keyboard.
   2. Enough Time: Provide users enough time to read and use content.
   3. Seizures: Do not design content in a way that is known to cause seizures.
   4. Navigable: Provide ways to help users navigate, find content, and determine where they are.
   5. **Input Modalities**
3. **Understandable - Information and the operation of user interface must be understandable.** 
   1. Readable: Make text content readable and understandable.
   2. Predictable: Make Web pages appear and operate in predictable ways.
   3. Input Assistance: Help users avoid and correct mistakes.
4. **Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.**

**4.1.** Compatible: Maximize compatibility with current and future user agents, including assistive technologies.

Each guideline has testable success criteria for conformance at Level A, Level AA or Level AAA. In all, there are 78 success criteria that are grouped under the above-mentioned 13 guidelines. Often the questions that come up include: Do we have to comply with all the 78 success criteria? What is acceptable? How much is enough?

Below is a brief explanation of WCAG conformance levels:

Level A - Must be done

Level AA - Should be done

Level AAA - Can be done

Across the world, Level AA conformance is what organizations and governments aim to achieve. WCAG 2.0 is approved as an ISO standard: ISO/IEC 40500:2012. WCAG 2.1 is an extension of WCAG 2.0 and it includes all the success criteria’s that were part of WCAG 2.0.

**As of June 2020, all new Pearson online learning content should aim to satisfy WCAG 2.1 Level A and Level AA success criteria. For more information about Pearson’s accessibility guidelines, see** [Accessibility Guidelines for Higher Education](https://www.pearson.com/accessibility-guidelines.html)**.­**

## **COLOR**

Color is an integral part of any web page. Color adds life to a web page. Color needs to be used wisely as far as accessibility is concerned. Use colors that contrast well to ensure that they are legible for all users in different situations of use. Color should be used in combination of text as well as other visual clues to convey information, purpose and meaning.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


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.

1.4.3 Contrast (Minimum) Level AA

The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following:

**Large Text:** Large scale text and images of large-scale text has a contrast ratio of at least 3:1.

**Incidental:** Text or images of text that are a part of inactive user interface component, that are pure decoration, that are not visible to anyone, or that are a part of a picture that contains significant other visual content, have no contrast requirement;

**Logotypes:** Text that is a part of a logo or brand name has no contrast requirement.

1.4.11 Non-text Contrast Level AA

The visual [presentation](https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html#dfn-presentation) of the following have a [contrast ratio](https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html#dfn-contrast-ratio) of at least 3:1 against adjacent color(s):

**User Interface Components**

Visual information required to identify [user interface components](https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html#dfn-user-interface-component) and [states](https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html#dfn-state), except for inactive components or where the appearance of the component is determined by the user agent and not modified by the author;

**Graphical Objects**

Parts of graphics required to understand the content, except when a particular presentation of graphics is [essential](https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html#dfn-essential) to the information being conveyed.

### **Use color Wisely**

Colors should be used wisely on the web, as information conveyed with color alone might not be available for people with visual disabilities.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1

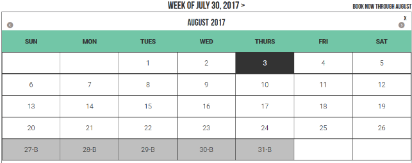

|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.4.1 Use of color | A |

#### How to implement

This section lists the techniques that need to be implemented to ensure that color is not used as the sole means to convey important information in different scenarios.

* Use additional visual cues along with color.

In the example below, booked dates are indicated using color along with the letter ‘B’. This will make the information about the booked dates available for color-blind users as well as blind users.



* Indicate currently selected item:

The information can be made available to color-blind users by formatting the text as bold. Similarly, the information can be made available for blind users by adding hidden text “selected” or via the aria-current attribute depending on the context.

* Contextual Links

For contextual links, use underline or bold font style along with color. The underline for contextual links should appear:

* Ideally, always, or on mouse hover and keyboard focus.
* Ensure that the contrast between contextual link texts with the surrounding text is 3:1.

Use color along with textual alerts:

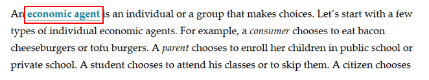
When a form is submitted with invalid data, use icons and textual alert along with color to indicate an error. For example, use icons and text messages along with color to indicate an error.

Color is helpful to easily identify invalid form fields but when color alone is used, users with visual disabilities will not be able to understand that an error has occurred.

#### Practices to Apply & Avoid

* Supplement color with additional visual cues such as:
* Patterns: …., ----
* Symbols: \*
* Distinguish contextual links from the surrounding text using color as well as by underlining them or changing the font style to bold.

For example, on the Pearson website, certain links within the paragraph are distinguished using color and are underlined.



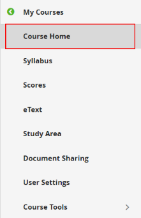
* Do not use color alone to distinguish contextual links (e.g. links within paragraphs) from the surrounding text.

For example, on the Pearson website, certain links within the paragraph are distinguished using color alone.



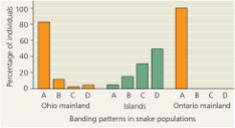
* Do not indicate the currently selected item using color alone.

For example, in the Pearson biology course, the currently selected tab “Syllabus” is indicated using color alone.



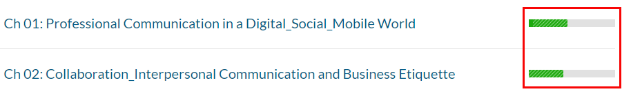
* Supplement color with icons and text to convey response for a user's action.
* Do not use color alone to indicate data in graphs.

For example, in the Pearson Biology course, color alone is used to indicate two different sets of data.



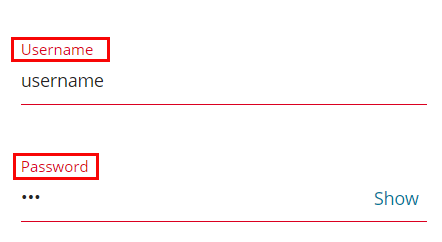
* Ensure to supplement color with other visual elements such as patterns, varying line styles or symbols to indicate data in graphs and charts.

For example, in the Pearson Business Communication course, color is supplemented with pattern to show the progress on a progress bar.



* Do not use color alone to indicate response for a user's action, such as indicating an error for a form field when the user submits a form with invalid data.

For example, on the Pearson login page, color alone is used to indicate an error.



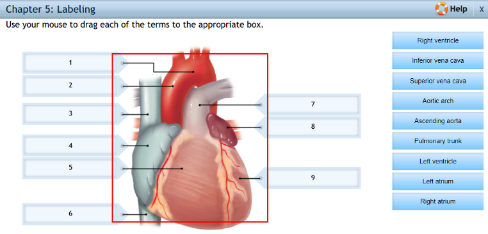
* Ensure to provide textual clues along with using color.

For example, in the Pearson Business Law course, the progress bar for both Homework and Quizzes is indicated by color as well as through supporting text.



* Ensure to provide lead lines to associate labels with selected sections of the image.

For example, in the Pearson Medical Terminology course, lead lines are provided for this interactive activity of labelling.



 Tip:

Color contrast for contextual link text with the surrounding text should be 3:1.

#### How to test for Accessibility?

To test a web page for use of color alone, perform the following steps:

1. Open the web page that needs to be tested.
2. Check if color alone is used on the page to:
   1. Identify the state of an interactive element, such as currently selected tab, button being pressed etc.
   2. Indicate a response for user action, such as form labels displayed using red color to indicate an error.
   3. Important information conveyed using color alone, such as values in a chart or graph are conveyed using color alone.
3. If any of the conditions mentioned in step 2 above are true, then it is an accessibility violation as per WCAG 2.1 success criteria 1.4.1 at Level A.

To check the color contrast for contextual links, perform the following steps:

1. Open Color Contrast Analyser (CCA).
2. Open the page that includes a contextual link in any browser.
3. Right-click on the link and select “Inspect” option from the context menu to open the browser’s Inspect panel.
4. Take the hex code of link’s foreground text as well as the surrounding text’s color.
5. Add the link and surrounding text color hex codes in the respective text fields available in Color Contrast Analyser (CCA).
6. Color contrast ratio along with its compliance result will be displayed.
7. If the color contrast ratio is less than 3:1 then it is an accessibility violation of WCAG 2.1 success criteria 1.4.1 at Level A.

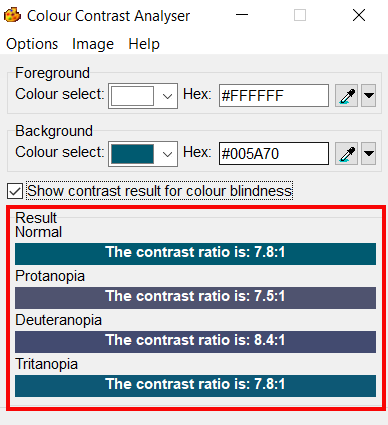
### **Color Contrast**

Colors used for displaying text and images of text (text created as an image) should have sufficient contrast with their background to make the information legible for users.

* Color blindness is the inability to perceive differences between some colors. Users who are color blind have difficulty in differentiating color or between red and green, or blue and yellow.   
  Ensuring a website follows the WCAG guidelines for color contrast ratio, color should not affect a person’s ability to perceive important content.

Types of color blindness:

* + **Tritanopia:** makes it hard to tell the difference between blue and green, and between yellow and red.
  + **Protanopia:** makes red look greener and less bright.
  + **Deuteranopia:** most common type of red-green color blindness. It makes green look more red.
  + Monochromacy: cannot see color at all (gray scale).



* For text:  
  For standard text (font lesser than 14pt) to be readable, it is mandatory for it to have a color contrast of 4.5:1 with its background.
  + For large text (14pt bold or 18pt) to be readable, it is mandatory for it to have a color contrast of 3:1 with its background.
  + There are exceptions. Incidental texts which are not required to meet the contrast requirements are: an inactive user interface element, logo, text for pure decoration, text with is not visible to anyone and text which is part of a picture which does not contain significant information.

Non-Text:  
For non-text elements such as user interface components and graphical objects, it is mandatory for the contrast ratio to be above 3:1 with its adjacent color.

* + User interface components are the controls for different functions. These controls in their different states such as active, hover etc. should have a contrast ratio higher than 3:1 with their background. Non-text contrast requirements are not applicable if the user interface components are in an inactive state. They are also not applicable to visual borders that are added by the browser when a link or any other interactive element receives focus.
  + Graphical objects are those elements that convey important information, such as graphs, diagrams, key icons etc.  Non-text contrast requirements are applicable to images that convey important information and different components of the graphical object need to have minimum contrast of 3:1. Elements can be excluded from the non-text contrast requirements if changing the contrast will result in incorrect interpretation of information.

### **Contrast for Text Content**

Colors used for displaying text and images of text (text created as an image) should have sufficient contrast with their background to make the information legible for users.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.4.3 Contrast (Minimum) | AA |

#### How to implement

This section lists the techniques that need to be implemented to ensure that contrast between the foreground text is sufficient with its background in different scenarios.

UI Designers Need to…

Two colors are considered to have good visibility if the contrast ratio between the foreground text and background is sufficient. Darken either the foreground or the background color to ensure that it passes the contrast requirement outlined by WCAG 2.0.

Check the contrast ratio between foreground text colors with the background using color contrast testing tools before using them on a web page. The minimum contrast requirements outlined by WCAG 2.1 are:

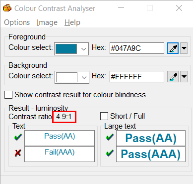
|  |  |
| --- | --- |
| **Text Size** | **Contrast Ratio** |
| Standard size text (anything less than 14pt) | 4.5:1 |
| Large-size text (14pt bold or 18pt) | 3:1 |

#### Practices to Apply & Avoid

* Use colors that provide enough contrast between the foreground text and background.

In the example below on the Pearson website, sufficient color contrast is used for displaying the link text such as “Assignment 1" and “Introduction to Mastering Biology for Non-Majors”. The color contrast ratio for the text is "4.9:1".

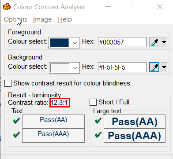
Link text with a good colour contrast ratio with it's background



* Ensure large-scale text (14pt bold or 18pt) has a color contrast ratio of 3:1 with its background.

For example, on the Pearson website, sufficient color contrast is used to display large text. Color contrast ratio "12.3:1" is used for displaying the text "Investing with us".





#### How to test for Accessibility?

To check the color contrast for text, perform the following steps:

1. Open Color Contrast Analyser (CCA).
2. Open the page for which color contrast of text needs to be checked in any browser.
3. Right-click on the text and select “Inspect” option from the context menu to open the browser’s Inspect panel.
4. Take the hex code of text’s foreground color as well as the hex code of the background color.
5. Add the foreground text and background color hex codes in the respective text fields available in CCA.
6. Color contrast ratio along with its compliance result will be displayed. Compliance will be displayed for:
7. Standard text size and large text size
8. Results are displayed for Level AA as well as Level AAA.
9. Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement.
10. Text that is part of a logo or brand name has no contrast requirement.
11. If the color contrast ratio is less than 4.5:1 for standard text or less than 3:1 for large text, then it is an accessibility violation of WCAG 2.1 success criteria 1.4.3 at Level AA.

### **Contrast for Non-Text Content**

Color contrast needs to be enough not only for textual content available on web pages but also for all the important non-textual content. The contrast between non-textual content with the adjacent color needs to be 3:1.

Non-text contrast requirements were introduced in WCAG 2.1 and is applicable to:

User interface controls in different states, such as focus, active, hover etc. This applies to different user interface components, such as links, form controls, buttons etc. For example, if the background color of a link changes on focus then the contrast between the link's background color with the text font as well as page's background color should be at least 3:1. Non-text contrast requirements are not applicable if the for user interface components in an inactive state. Similarly, non-text contrast requirements do not apply to visual borders that are added by the browser when a link or any other interactive element receives focus.

Graphical objects that convey important information, such as graphs, diagrams, key icons etc.  Non-text contrast requirements are applicable to images that convey important information and different components need to have minimum contrast of 3:1, such as lines of a line graph, labels of a bar graph, key icons, numerical and textual data of complex diagrams etc. However, if changing the contrast will result in incorrect interpretation of information, i.e. changes the meaning of the subject then those graphics do not need to adhere to the above-mentioned contrast requirements. For example, in the case of science experiments, if by changing the contrast will result in changing the meaning of the topic being explained, then such graphics do not need to adhere to the above-mentioned contrast requirements.

The above requirements were introduced to ensure that low vision users can access all the important information conveyed using non-text content on a web page.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.4.11 Non-text Contrast | AA |

#### How to implement

This section lists the techniques that need to be implemented to ensure sufficient contrast for non-text content in different scenarios.

UI Designers Need to…

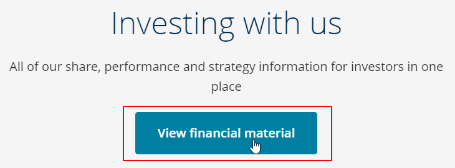
Check the contrast ratio of the graphical object and its surrounding using color contrast testing tools before using them on a web page.

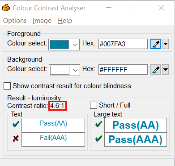
Two colors are considered to have good visibility if the contrast ratio between the graphical object and its surrounding is enough. Darken either the graphical object or the background color to ensure that it passes the contrast requirement outlined by WCAG 2.1.

#### Practices to Apply & Avoid

* Ensure that the color contrast for all user interface controls in their hover state has a contrast ratio of 3:1 with the adjacent content.

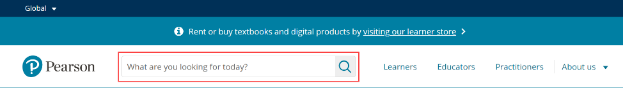
For example, on the Pearson website, color contrast for the “View Financial Material” button in its hover state has a contrast ratio of 4.6:1 with the adjacent content.

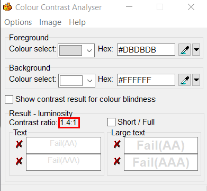




* Do not use colors with insufficient color contrast for all user interface controls in their default as well as active, hover and focus states of less than 3:1 with the adjacent content.

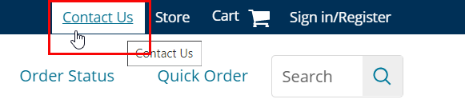
For example, on the Pearson website, the “What are you looking for today?” input field by default has a low contrast of 1.4:1 with the adjacent background.

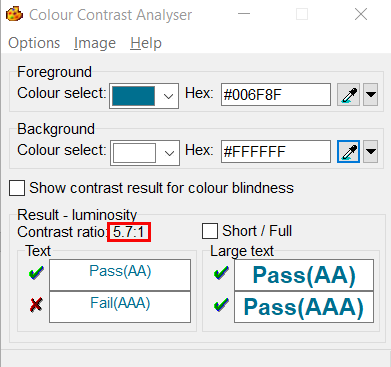




* Ensure link text has enough contrast in hover/focus, visited, active state.

For example, on the Pearson website, the color of the link texts “Contact Us”, “Store”, “Cart” “Sign in/Register” change to blue on a white background on hover with a color contrast of 5.7:1, ensuring it passes the contrast requirement.





#### How to test for Accessibility?

To check the color contrast for Non text content, perform the following steps:

1. Open Color Contrast Analyser (CCA).
2. Open the page for which color contrast of a non-text content needs to be checked in any browser.
3. Use the picker to select the color of the graphical object and its surrounding.
4. Color contrast ratio along with its compliance result will be displayed.
5. If the color contrast ratio is less than 3.0:1 then it is an accessibility violation of WCAG 2.1 success criteria 1.4.11 at Level AA.

### **Color contrast for Images**

Color contrast needs to be enough not only for textual and non-textual content available on web pages but also for all the informative content. The contrast between the informative image and its adjacent content should be higher than 3:1.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.4.11 Non-text Contrast | AA |

#### How to implement

This section lists the techniques that need to be implemented to ensure sufficient contrast for images in different scenarios.

UI Designers Need to…

Check the contrast ratio of the informative graphical object with its surrounding using color contrast testing tools before using them on a web page.

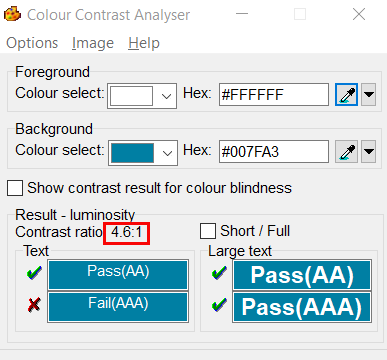
Two colors are considered to have good visibility if the contrast ratio between the graphical object and its surrounding is enough. Darken either the graphical object or its surrounding color to ensure that it passes the contrast requirement outlined by WCAG 2.1.

#### Practices to Apply & Avoid

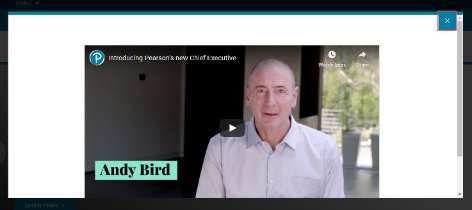
* Ensure informative images have a sufficient color contrast.

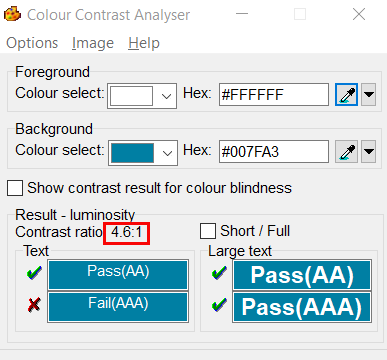
For example, on the Pearson website, the “SORT BY” drop-down image has a good contrast of 4.6:1 with the adjacent color.





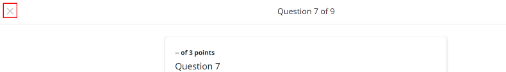
For example, on the Pearson website, color contrast for the “X” (Close) informative image in the modal dialog (that becomes available on activating the “Watch Video” link) has an enough color contrast of 4.6:1 with the adjacent color.

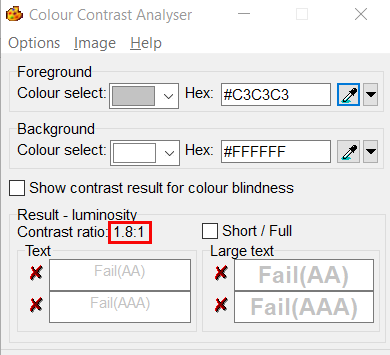




* Avoid having informative images with an insufficient color contrast.

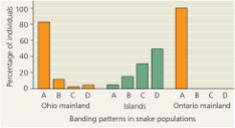
For example, on the Pearson website, color contrast for the “X” (Close) informative image insufficient color contrast of 1.8:1 with the adjacent color.

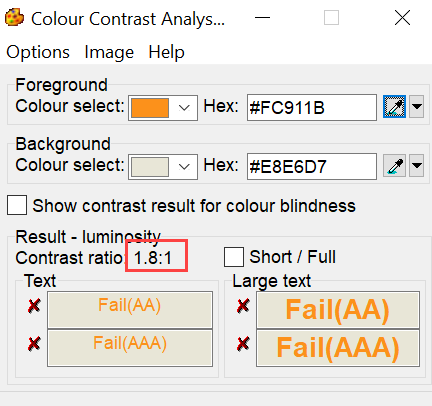


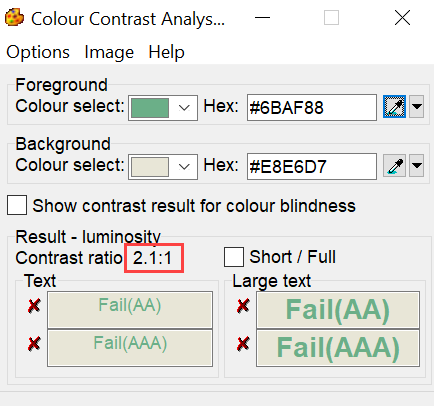


* Avoid having graphs with an insufficient color contrast.

For example, in a Biology course on the Pearson website, ­the two colors used in the graph have an insufficient color contrast of 2.1:1 and 1.8:1.







User Groups Affected

When insufficient color contrast is used, the user group that gets largely affected is users with visual disabilities such as:

* Color-blind users (with protanopia, deuteranopia, tritanopia etc.)
* Low-vision users
* Elderly users

#### How to test for Accessibility?

To check the color contrast for Non text content, perform the following steps:

1. Open Color Contrast Analyser (CCA).
2. Open the page for which color contrast of an informative graphical object needs to be checked in any browser.
3. Use the picker to select the color of the graphical object and its adjacent color.
4. Color contrast ratio along with its compliance result will be displayed.
5. If the color contrast ratio is less than 3.0:1 then it is an accessibility violation of WCAG 2.1 success criteria 1.4.11 at Level AA.

## **CONTENT AND LANGUAGE**

Content is king and it is equally important to make it accessible to ensure that people with disabilities can access the content. Different factors play a crucial role when it comes to making content accessible. Content must be available to users in a meaningful sequence as well as it should be free from use of sensory characteristics (size, shape, location etc.) alone for conveying important information.

Web page content also includes use of abbreviations and acronyms that need to be marked-up correctly so all users can interpret them correctly. HTML mark-up provides web page authors with different elements for displaying content. On a web page content <p> elements can be used to mark-up paragraphs whereas <section> element can be used to mark-up a general region.

Web page authors can use <article> element to include content that is self-contained which can be removed from the page without affecting other content. Examples of content that can be marked-up using <article> element include news item, blog post, etc. Additionally, quotations can be presented using inline quotes or blockquotes. This helps screen reader users to understand where a quote starts or ends.

Now let us see what accessibility considerations need to be kept in mind when it comes to language of web page content.

Language of the web page content must be defined programmatically to ensure that user agents can render the information accurately. User agents, such as web browsers, assistive technologies (screen readers and braille displayers) rely on HTML markup to identify the language of a web page.

The ’lang’ attribute is used to define the language of a web page. Based on the language defined in the markup, screen readers and braille displayers will interpret the content and provide it to their users.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


1.3.3 Sensory Characteristics Level A

Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound.

1.3.2 Meaningful Sequence Level A

When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined.

3.1.4 Abbreviations Level AAA

A mechanism for identifying the expanded form or meaning of abbreviations is available.

3.1.1 Language of Page Level A

The default human language of each Web page can be programmatically determined.

3.1.2 Language of Parts Level AA

The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text.

### **Reading Order**

Reading order of the web page content depends on the order in which the content is presented in the page source. Assistive technologies such as screen readers read the content in left-to-right and top-to-bottom order as it is presented in the source code.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.3.2 Meaningful Sequence | A |

#### How to implement

This section lists the techniques that need to be implemented to ensure that the content is read out in a meaningful sequence.

##### Developers Need to…

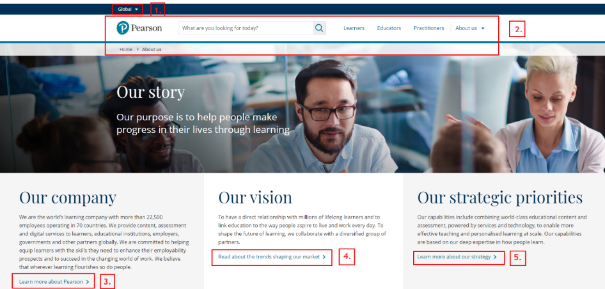
Place the content in the correct sequence in the page source.

#### Practices to Apply & Avoid

* Ensure the reading order of the content is logical.

For example, on the Pearson website, the reading order is logical. The content is read out in the following sequence:

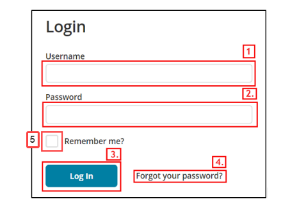
1. “Global” drop-down in the header section.
2. “Pearson logo” image link, “what are you looking for?” input field, “search” button and links such as “Learners”, “Educators”, etc. in the header section.
3. Main content area.
4. Footer section.



* Avoid displaying web page content in a manner that results in an inaccurate reading order.

For example, on the Pearson website, the reading order is inaccurate. The screen reader read the content in an order that did not match the visual reading order of the content. The content is read out in the following sequence:

1. “Username” input field.
2. “Password” input field.
3. “Log In” link.
4. “Forgot your password?” link.
5. “Remember me” checkbox.



User Groups Affected

Blind users

Low-vision users

#### How to test for Accessibility?

To test a web page for reading order, perform the following steps:

1. Open the web page that needs to be tested in a browser.
2. Turn on a screen reader.
3. Check if the reading order of the content is correct by accessing the content using the screen reader – Up and Down Arrow keys.
4. If the condition in step 3 fails and the content is not read out in a meaningful sequence, then it is an accessibility violation as per WCAG 2.1 success criteria 1.3.2 at Level A.

### **Abbreviations and Acronyms**

Abbreviations are shortened form of words, name, or phrase, such as W3C, WCAG etc. Acronyms on the other hand are abbreviated forms made from initial letters or other parts of a word, name or phrase which may be pronounced as a word, such as WHO, USA etc.

Expanded form of abbreviations and acronyms need to be provided to ensure their meaning is clear for all users correctly. Assistive technologies such as screen readers support reading of abbreviations and acronyms if they are marked-up using semantic HTML elements. However, by default screen readers are not configured to read expansion of abbreviated words but provide users with options to do so through screen reader settings.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| **3.1.4 Abbreviations** | **AAA** |

Adding expansion for abbreviations and acronyms is required by WCAG at Level AAA, but at Pearson who is the leader in online learning courses it is essential to add expansions for abbreviations and acronyms. Since Pearson produces course content for so many different subjects and learners need the expansion of shortened form of words to understand their meaning.

#### How to implement

This section lists the techniques that need to be implemented to ensure that expansions of abbreviations and acronyms are accessible for all users.

##### Developers Need to…

Use the <abbr> element for marking-up abbreviations, initialisms, and acronyms.

Include the expansion for abbreviations, initialisms and acronyms using the title attribute of the <abbr> element.

Provide users with an option to search for expansion of abbreviated words and acronyms by either linking it to an online dictionary or through site search function.

##### Content Writers Need to…

Specify the expansion of abbreviations or acronyms where they first occur in a document.

Include expansions for abbreviations, initialisms, and acronyms in the Glossary section of online learning course.

#### Practices to Apply & Avoid

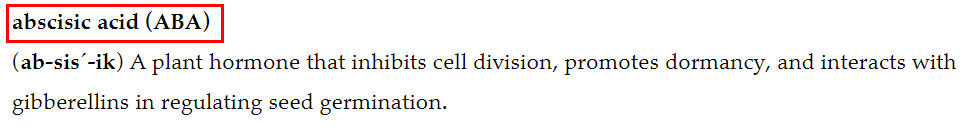
* Avoid using abbreviations without their expansion on a web page.

For example, on the Pearson website, for the heading “How online learning platforms facilitate NACEP accreditation” the expanded form of the abbreviation “NACEP” is not provided.



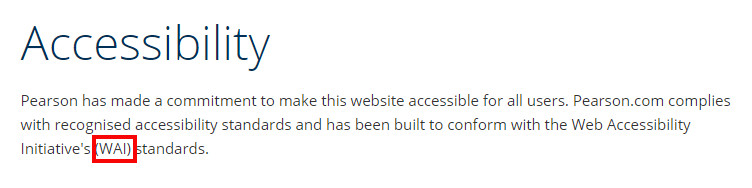
* Provide the full form of an abbreviation.

For example, in the Pearson Mastering Biology course, the expanded form of the abbreviation ABA is provided.



* Ensure to specify acronyms with their expanded form in their first occurrence.

For example, on the Pearson website, the expansion of the acronym “WAI” is specified.



* Avoid using acronyms without their expansion on a web page.

For example, on the Pearson website, the expansion of the acronym HTML is not mentioned in the first occurrence.

Expansion of the acronym "HTML" is specified. 

User Groups Affected

Blind users

Low-vision users

Users with cognitive disabilities

#### How to test for Accessibility?

To test a web page for abbreviations and acronyms, perform the following steps:

1. Open the web page that needs to be tested in a browser.
2. Turn on a screen reader with abbreviations and acronyms reading options enabled.
3. Check if expansion for abbreviations and acronyms is provided either at the first occurrence of their use or through the <abbr> element.
4. If the condition in step 3 fails and there is no expansion provided for abbreviations and acronyms, then it is an accessibility violation as per WCAG 2.1 success criteria 3.1.4 at Level AAA.

### **Sensory Characteristics**

Instructions provided on a web page for understanding and operating content and functionality should not rely only on sensory characteristics such as size, shape, visual location, orientation and sound, since users with disabilities might not be able to perceive information that is presented using sensory characteristics alone.

For example, the instruction “Fill in all the form details and click on the button available at the top right corner to submit the details” will not be understandable for many blind users. As blind users cannot see where the button is, and their assistive technology (e.g. screen readers and Braille displayers) will not provide them with information about the visual location of the button, this instruction will not be of any help to them. Instead, provide the instruction “Fill in the form details and click on the Done button available at the top right corner to submit the details” to ensure it works for all users.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.3.3 Sensory Characteristics | A |

#### How to implement

This section lists the techniques that need to be implemented to ensure that sensory characteristics alone are not used for conveying important instructions.

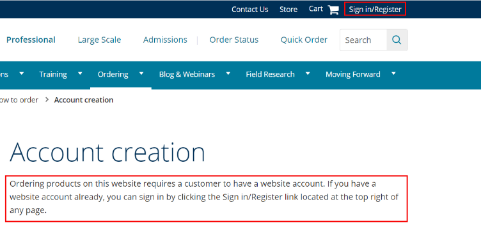
##### Content writers Need to…

Supplement sensory characteristics with textual labels and other visual clues to make the information understandable for all users.

#### Practices to Apply & Avoid

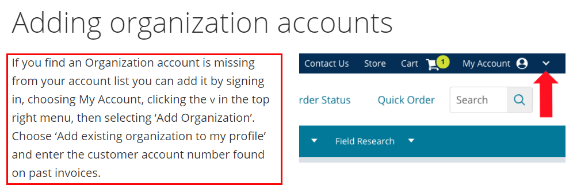
* Ensure instructions that make use of sensory characteristics are supplemented with textual labels or visual clues to make the information available for all users.

For example, on the Pearson website the instruction “sign in by clicking the Sign in/Register link located at the top right of any page” is provided, which is clear for screen reader and braille displayer users.



* Avoid using sensory characteristics alone to convey important information.

For example, on the Pearson website, the instruction "Clicking the 🇻 in the top right menu, then selecting ‘Add Organization’. Choose ‘Add existing organization to my profile’ and enter the customer account number found on past invoices." is provided, which is not clear for screen reader users.



User Groups Affected

Blind users

Low-vision users

Deaf-blind users

#### How to test for Accessibility?

To test web page instructions for use of sensory characteristics alone, perform the following steps:

1. Open the web page that needs to be tested in a browser.
2. Check if sensory characteristics alone is not used to convey important instruction.

If the condition in step 2 fails, then it is an accessibility violation as per WCAG 2.1 success criteria 1.3.3 at Level A.

### **Primary Language**

Primary language is the default language of a web page that is defined using the ‘lang’ attribute on the <html> tag.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 3.1.1 Language of Page | A |

#### How to implement

This section lists the techniques that need to be implemented for defining Primary Language of a web page.

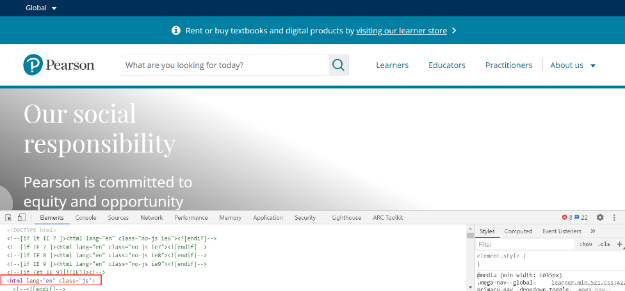
##### Developers Need to…

* Specify the default language of a web page via the ‘lang’ attribute for the <html> element.
* In XHTML, 'xml:lang' attribute is used to define the default language.

#### Practices to Apply & Avoid

* Ensure primary language is defined for the webpage.

For example, on the Pearson website, the primary language is specified in the HTML code.



* Avoid not defining “lang” attribute for any webpage.
* Avoid using incorrect primary language code for any webpage.

For example, on the Pearson website, incorrect primary language “English” is specified instead of “Chinese”.



#### How to test for Accessibility?

To test a web page for Primary language, perform the following steps:

1. Open the web page that needs to be tested.
2. Right click your mouse and select “View Source” or “View Page Source.”
3. Check if “lang” attribute is specified along with accurate language code on the <html> element.
4. If the condition in step 3 fails, then it is an accessibility violation as per WCAG 2.1 success criteria 3.1.1 at Level A.

### **Secondary Language**

Parts of web page content displayed in another language should be defined programmatically through HTML markup. This informs users of assistive technologies that the content is another language and thus they can render it accurately for their users.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 3.1.2 Language of Parts | AA |

#### How to implement

This section lists the techniques that need to be implemented for defining Secondary Language for parts of content on a webpage.

##### Developers Need to…

* Specify the language attribute for parts of content displayed in other language through HTML markup.
* Use the 'hreflang' attribute to inform users that the language of the page that the link points to is different from the primary language of the current page.

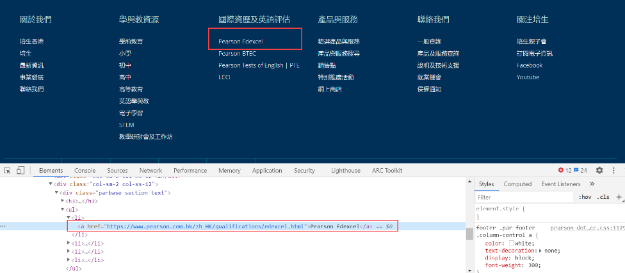
#### Practices to Apply & Avoid

* Ensure the change in language is programmatically identified via code.

For example, on the Pearson website, the change in language is correctly defined via code.



* Avoid missing of “lang”/”hreflang” attribute for parts of content in secondary language on any web page.



User Groups Affected

* Blind users

#### How to test for Accessibility?

To test a web page for Secondary language, perform the following steps:

1. Open the web page that needs to be tested in Google Chrome.
2. Check if the secondary language is defined for the multilingual webpage.
3. To test for secondary language, select the text and inspect it. Check whether “hreflang” and “lang” attributes are provided in case of a link and a plain text, respectively.
4. If the conditions in step 2 and 3 fails, then it is an accessibility violation as per WCAG 2.1 success criteria 3.1.2 at Level AA.

## **FRAME AND PAGE TITLES**

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


2.4.1 Bypass Blocks Level A

A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.

2.4.2 Page Titled Level A

[Web pages](http://www.w3.org/TR/WCAG20/#webpagedef) have titles that describe topic or purpose.

4.1.2 Name, Role, Value Level A

For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

### **Page Titles**

Page titles are displayed at the top of the browser window. In most browsers, page titles are displayed as the names of the browser tabs. Title of a web page generally includes page name, section name as well as company name. Page titles act as the bookmark names when users opt to bookmark a particular page in their browser.

 Tip:

Title is the first piece of information that a screen reader reads when a web page loads.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 2.4.2 Page Titled | A |

#### How to implement

This section lists the techniques that need to be implemented to ensure page titles are meaningful and descriptive.

##### Developers Need to…

* Define the title for the page using the <title> element within the <head> element in a HTML document.

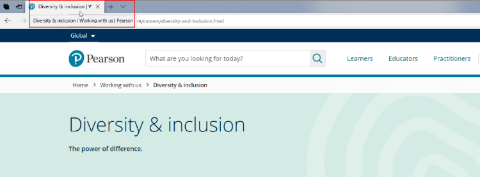
##### Content Writers Need to…

* Use unique and descriptive titles for each web page.
* Use front loading titles, i.e. Name of the page followed by section name and finally website name.

#### Practices to Apply & Avoid

* Make sure the title of the page is unique and descriptive.

For example, on the Pearson website, the title of the page "Diversity & Inclusion | Working with us | Pearson" is unique and descriptive.



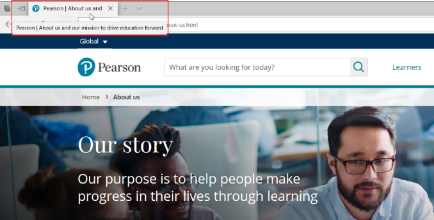
* Page titles should be front-loading (putting the brand name or organization/company name the end of the title).

For example, on the Pearson website, the title of the page "Investor Information | Investing in Pearson | Pearson" is an example of front-loaded title.



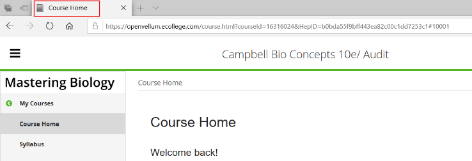
* Avoid using page titles that are not front-loading.

For example, on the Pearson website, the title of the page "Pearson | About Us and Our Mission to drive education forward" is not front-loading.



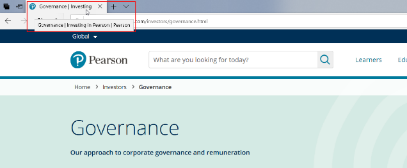
* Avoid using incomplete and non-descriptive page titles.

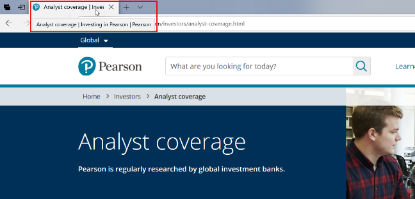
For example, in the Pearson Mastering Biology course, the title of the page “Course Home" is incomplete. This makes it difficult for screen reader users to identify the type of information that will be available on the page.



* Use a unique title for each web page.

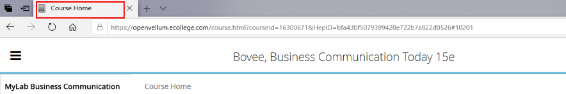
For example, on the Pearson website, the title of pages "Governance | Investing in Pearson | Pearson” and “Analyst coverage | Investing In Pearson | Pearson “are unique.

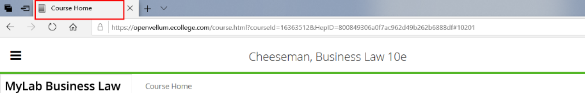




* Avoid using pages with no title for a website or online learning course. They are read out as “Untitled” by screen readers. Also, in the absence of a title, browsers display the page URL as the title.
* Avoid using identical page titles for multiple pages.

For example, in the Pearson courses, an identical page title “Course Home” is used for the homepage of both MyLab Business Communication and MyLab Business Law courses.





 Tip:

Web pages with descriptive titles are ranked higher by search engines and help in the search engine optimization of a website.

#### How to test for Accessibility?

To test a web page’s title, perform the following steps:

1. Open the web page that needs to be tested.
2. Right click your mouse and select “View Source” or “View Page Source.”
3. Check if descriptive title is provided in the <title> tag within the <head> tag.
4. On Windows with a screen reader running (JAWS or NVDA), press “Insert + T” to listen to the page title.
5. Listen if the title of the page is descriptive of the page’s contents.

### **Frame and iFrame Titles**

HTML frames are used to split a browser window into different sections. Authors can opt to load multiple HTML documents using these sections. Splitting a page up using frames is an old style of web design and is rarely used in the modern web content. However, iframes are used extensively for displaying content, embedding 3rd-party content, social media content as well as search engine indexing in modern web design. Assistive technologies such as screen readers identify the frames for their users as well as provide them with keyboard shortcuts to navigate from one frame to another.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| **4.1.2 Name, Role and Value** | A |

#### How to implement

This section lists the techniques that need to be implemented to ensure Frames and Iframes can be distinguished effectively.

##### Developers Need to…

* Use the title attribute to label each frame and iframe element used on a web page.

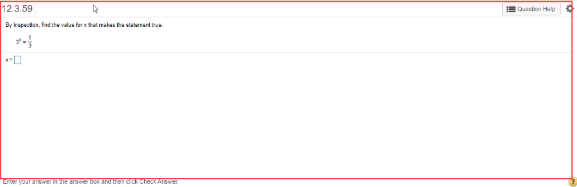
##### Content Writers Need to…

* Specify unique and descriptive titles for frames and iframes.
* Avoid using the word “frame” within the title. A screen reader identifies the element as “frame” for users so adding the word within the title will result in repetition of information.

#### Practices to Apply & Avoid

* Ensure the frame or iframe title is meaningful, i.e. describes the content that is displayed in it.

For example, on the Pearson website, a meaningful title of “Question Viewer” is provided for the iframe.



* Avoid using frames and iframes with identical, inappropriate or no titles.

For example, on the Pearson website, an incomplete title of “A message from Andy Bird” is provided for the iframe.



User Groups Affected

* Blind Users
* Low-vision users
* Deaf-blind users

 Tip:

Hide iframes added in the page source for search engine indexing from users by specifying the CSS 'display:none' property.

#### How to test for Accessibility?

To test a web page for Frame/iFrame titles, perform the following steps:

1. Open the page in Google Chrome web browser.
2. Open the Web Developer extension. The extension is freely available on Chrome store.
3. Click on Outline Frames option available on the Outline tab.
4. All the frames present on the page will be highlighted with a border.
5. Right-click on the heading and choose Inspect.
6. Check if each frame has a title attribute that contains descriptive text that identifies the frame.
7. On Windows with screen reader (JAWS or NVDA) running, press “M” and “Shift + M” to jump to the next and previous frame/iframe.

## **HEADINGS**

Headings act as a title for key sections of a webpage. Headings are defined for the main section of the page as well as sub-sections. Headings form an outline that identifies key sections of a page. They depict the hierarchy of the page’s content structure.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


1.3.1 Info and Relationships Level A

Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.

2.4.6 Headings and Labels Level AA

Headings and labels describe topic or purpose.

### **Heading Structure**

Headings are used to define the content structure of a web page. Topics and subtopics of a web page are marked up using heading level markup. As per HTML specifications, there are six heading levels H1-H6.

Assistive technologies can identify the headings based on the HTML markup and provide users with an option to understand the content structure as well as navigate quickly to different parts of a web page.

Heading levels should be defined in a manner that they depict the content hierarchy of the page. Heading levels should not be skipped, i.e. <h5> should not be used directly in the main content as the first heading of the core topic. If a web page has the main heading (<h1>) at the top all the section headings, such as navigation, footer, sidebars etc. should be <h2>. All the sub-sections can be marked-up as <h3> followed by <h4> depending on page content.

When it comes to headings for sidebars, navigation sections and footer sections they should be use consistently across the course or website. It is recommended to use headings in a sequence but in situations where new section starts, it is acceptable to break the sequence. For example, if on a web page a sub-section has <h2> as the main heading and <h3> and <h4> are sub headings then using <h2> after <h4> is acceptable since the <h2> is heading for a next sub-section.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| **1.3.1 Info and Relationships** | **A** |
|  |  |

#### How to implement

This section lists the techniques that need to be implemented to ensure visual headings are marked-up as Headings.

##### UI Designers Need to…

Avoid the use of only visual indication using font-weight, font-size etc. property to mark-up headings.

##### Developers Need to…

* Use HTML heading tags to define the page structure.
* Use headings in the correct sequence:
* <h1> should be followed by <h2>
* <h2> should be followed by <h3> and so on
* Use <h2> tag for marking up main title of the modal dialog and follow the page structure of the original page in case of non-modal dialogs.
* Use ARIA role “heading” along with appropriate aria-level attribute value for marking up headings in situations where it is not possible to use <h1> to <h6> tags.
* It is a best practice to have only one <h1> element for a webpage that is indicative of the core content of the page.

**Note:** Remove all empty heading tags from the HTML source of a web page, as it gives a false impression to screen reader users that some information is not being read out for them.

##### Content Writers Need to…

* Define the outline of the content by providing short headings to group related content.

#### Practices to Apply & Avoid

* Avoid marking up content using heading tags.

For example, on the Pearson website, the text “The analytics that Revel provides gets me excited. I want to know what my students struggle with because that guides how I teach them.” is marked up using H2 element. This will result in screen reader users finding it confusing to understand the page’s content structure.



* Heading should only be defined for the text that constitutes a heading for a section or subsection on a page.

For example, on the Pearson website, the text such “Our story”, “Our company”, “Our vision” and “Our strategic priorities” is appropriately marked-up as headings.



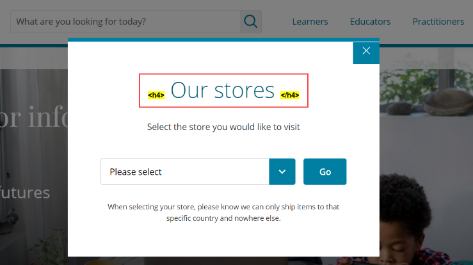
* Use heading tags such as H1-H6 to mark-up headings on a page.

For example, on the Pearson website, the text such as “Be you at Pearson”, “Current vacancies” and “Diversity & inclusion” is appropriately marked up using H2 and H3 elements, respectively.



* Avoid using elements other than <h2> to mark-up the heading for a modal dialog.

For example, on the Pearson website, the text "Our stores” in the modal dialog (that becomes available on clicking "Learners" link in the header section) is marked up using html <h4> element incorrectly.



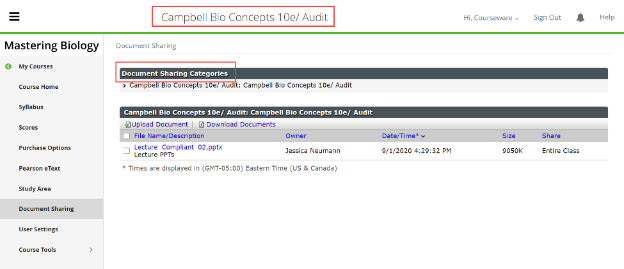
* Avoid skipping heading levels.

For example, on the Pearson website, the text such as “Guide de survie aux réunions à distance”, “Finance d'entreprise”, etc. in the main content area were defined as heading level H4 instead of heading level H3, respectively.



* Avoid using more than one H1 element on any webpage.

For example, in the Pearson Campbell Biology Concepts course, multiple H1 elements are used incorrectly for the text such as “Campbell Bio Concepts 10e/ Audit” and “Document Sharing Categories”.



#### How to test for Accessibility?

To test headings for accessibility, perform the following steps:

1. Open the page in Google Chrome web browser.
2. Open the WAVE extension. The extension is freely available on Chrome store.
3. Click on Structure tab in the left navigation pane.
4. All the headings present on the page will be displayed.
5. Click on the heading and choose “Code”.
6. Check if structural mark-up, such as <h1>-<h6> is used to mark-up the headings.
7. If the conditions in step 6 fails, then it is an accessibility violation as per WCAG 2.1 success criteria 1.3.1 at Level A

### **Heading Text**

Heading text should be unique, clear, descriptive, and consistent across pages of a website. This helps users in understanding what content will be available below the given heading, and users can easily find the information they are looking for.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| **2.4.6 Headings and Labels** | **AA** |

#### How to implement

This section lists the techniques for correct implementation of heading text on a webpage.

##### Content Writers Need to…

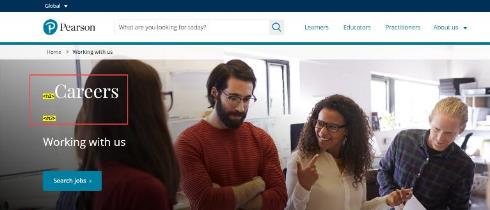
Use unique and descriptive text for displaying web page headings.

Use heading text that describes the content in the section and sub-sections.

#### Practices to Apply & Avoid

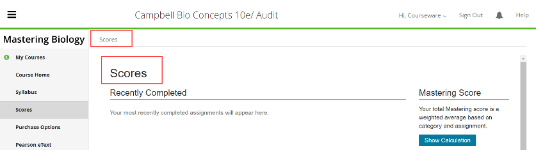
* Ensure heading text is unique and descriptive.

For example, on the Pearson website, unique and descriptive heading text “Careers” is provided.



* Avoid using identical heading text.

For example, in the Pearson Biology course, identical heading text “Scores” is provided for H2 elements.



* Avoid using non-descriptive heading text.

User Groups Affected

Blind users

Low-vision users

Users with learning disabilities

 Tip:

Search engines crawl through web page headings, and descriptive heading text can help a website to achieve a higher search engine ranking.

|  |  |  |
| --- | --- | --- |
| **Function** | **JAWS** | **NVDA** |
| Go to next heading | H | H |
| Go to previous heading | Shift + H | Shift + H |
| Show list of all headings | Jaws key + F6 | NVDA key + F7 |

#### How to test for Accessibility?

To test a web page for heading text, perform the following steps:

1. Open the page in Google Chrome web browser.
2. Open the WAVE extension. The extension is freely available on Chrome store.
3. Click on Structure tab in the left navigation pane.
4. All the headings present on the page will be displayed.
5. Check if headings are clear and descriptive.
6. If the conditions in step 3 fails, then it is an accessibility violation as per WCAG 2.1 success criteria 1.3.1 at Level A

## **LISTS**

Lists are used to group related items on a webpage. Lists can be used to group sequential items as well as non-sequential items. Lists are usually denoted with a “circle bullet” before non-sequential item and numbers, alphabets etc. before sequential items.

Lists are basically of three types, ordered lists, unordered lists, and description lists. Along with these nested lists are used to structure sub-lists for each of the above. Depending on the nature of the content, select the type of list to semantically group information for users.

Ordered lists are used to group sequential content, unordered lists are used to group non-sequential content whereas description lists are used to group related terms and their descriptions programmatically. Different HTML elements, such as headings, paragraphs, anchors, and nested lists can be presented within list items.

Assistive technologies such as screen readers identify lists based on the HTML mark-up and inform users about the number of list items within a list. This makes it easier for users to understand the number of items available to them. Some screen readers also provide users with information about beginning as well as end of a list.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


1.3.1 Info and Relationships Level A

Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.

### **Ordered Lists**

An ordered list is used to group sequential items. On the web, an ordered list is commonly used to present a multi-step form, breadcrumb trail, procedures to follow, etc.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| **1.3.1 Info and Relationships** | **A** |

#### How to implement

This section lists the techniques that need to be implemented to define ordered list.

##### Developers Need to…

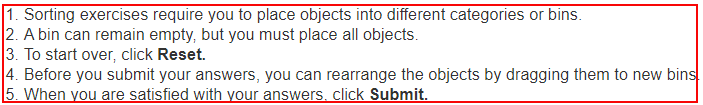
Use <ol> and <li> tags to code ordered lists.

* In HTML 5, use start and type attributes to define the list item prefix, such as “a., b., c.”, “1., 2., 3.”, etc.

#### Practices to Apply & Avoid

* Use <ol> element to define an ordered list.

For example, in the Pearson Mastering Anatomy and Physiology course, ordered list is used to mark-up the text such as “Sorting exercises require you to place objects into different categories or bins.”, “A bin can remain empty, but you must place all objects.”, etc.



### **Unordered Lists**

Unordered list is used to group non-sequential related items. Examples of unordered lists on the web include navigation links, social media icons, news items, list of photographs in a gallery section, etc.

#### How to implement

This section lists the techniques that need to be implemented to define unordered list on a webpage.

##### UI Designers Need to…

Define decorative images, elegant icons, etc., through CSS if the author does not wish to use traditional “disc” bullets.

Hide the “disc” bullets (if required) through CSS to maintain the presentation of the page.

##### Developers Need to…

* Use <ul> and <li> tags to code unordered lists.

#### Practices to Apply & Avoid

* Use <ul> element to define an unordered list.

For example, on the Pearson website, list markup such as unordered lists (<ul>) is used to markup related information such as “Learners”, “Educators”, “Practitioners”, etc.

Content marked correctly using <ul> element

### **Description Links**

A description list consists of two parts: a **term** and its **description**. On the web, common examples of description lists include lists of abbreviations/acronyms and their full form, lists of definitions in a glossary section, etc. Description lists includes definition list as well which was not the case prior to HTML 5. Web content in different situations of use can have single term with its description, multiple terms, and single definition as well as multiple terms with multiple descriptions.

 Tip:

Most screen readers do not identify description list separately other than JAWS which identifies it as a definition list.

#### How to implement

This section lists the techniques that can be used for accessible implementation of description List.

##### Developers Need to…

Use <dl> tag to markup a description list.

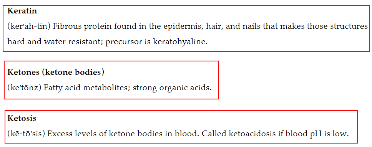
Use <dt> tag to markup a term.

* Use <dd> tag to markup the description.

#### Practices to Apply & Avoid

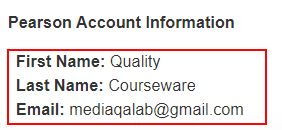
* Use a description list to mark-up lists of terms and corresponding definitions or descriptions.

For example, in the E-text of the Pearson Biology course, a definition list is used appropriately to mark-up the terms "Keratin", "Ketones", “Ketosis”, etc.



* Avoid using description list mark-up to code contents that do not constitute a term and its description.

For example, in the Pearson Mastering A & P Course, “First Name”, “Last Name”, “Email” is marked up as definition list inappropriately.



User Groups Affected

Blind users

* Low-vision users

 Tip:

A screen reader identifies the beginning/end of a list and number of items in an ordered as well as unordered list based on accurate HTML markup.

#### How to test for Accessibility?

To test a web page for ordered list, perform the following steps:

1. Open the web page that needs to be tested in Google Chrome.
2. Open the WAVE extension. The extension is freely available on Chrome store.
3. Click on the “Structure” tab in the left navigation pane.
4. All the lists present on the page will be displayed.
5. Ensure that the following scenarios are satisfied:
   1. Check that content that has the visual appearance of a numbered list is marked as an ordered list (i.e. <ol> and <li>).
   2. Check that content that has the visual appearance of a list (with or without bullets) is marked as an unordered list (i.e. <ul> and <li>).
   3. Check that content is marked as a definition list when terms and their descriptions are presented in the form of a list (i.e. <dl>, <dd> and <dt>).
6. On Windows with screen reader (JAWS or NVDA) running, press “L” and “Shift + L” to jump to the next and previous list; press “I” and “Shift + I” to jump to the next and previous list item.
7. If content that appears to be in ordered, unordered or definition list is not coded or if it is coded inappropriately then it is an accessibility violation as per WCAG 2.1 success criteria 1.3.1 at Level A.

## **TABLES**

Tables are used on the web to organize data and convey logical relationship between two sets of data that is often presented in the form of grids. Traditionally, tables were also used to position blocks of data.

HTML specifications offer a list of tags that can be used to semantically structure table data on the web. Along with HTML, CSS properties should be used to control the presentation of tables. CSS has an even bigger role to play in modern web design when it comes to tables as with responsive design the need to manipulate tabular data has increased a step further.

Semantic use of structural table mark-up ensures that data presented in the form of tables is accessible for assistive technology users. In this section, we will explore what different HTML elements are available to make tables accessible.

Tables on the web are no longer only static in nature. Today web tables allow users to sort the data as well as offer a rich interaction in the form of dynamic grids. To make dynamic tables accessible, JavaScript and ARIA have come to the party.

Broadly tables are of two types, layout tables and data tables. Let us explore each of them in detail now!

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


1.3.1 Info and Relationships Level A

Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.

### **Layout Tables**

Layout tables do not represent any relationships among content. Layout tables are used to position web page content. Designers used to use layout tables to position web page content such as navigation links, search form, site banner, main content, etc. Designing web pages using layout tables is an old style of web design. Modern web designers make use of CSS or table less layout.

When layout tables are used, ensure that the content presented within the tables is linearized correctly, i.e. the order of the content should be meaningful when it is accessed in a linear order. Screen readers read content of a table in a linear order, reading one cell at a time from left-to-right and top-to-bottom. For example, a screen reader will read the table information in the following order:

1. Column 1, row 1
2. Column 2, row 1
3. Column 3, row 1 …
4. Column 1, row 2
5. Column 2, row 2
6. Column 3, row 2 …

Structural table mark-up, such as table headers (<th>), table caption (<caption>), scope attribute, etc. are used for conveying table structure to user agents, and they must not be used in the case of layout tables. Basically, in the case of layout tables, use <table>, <tr>, and <td> elements.

**Note:** The “summary” attribute is deprecated in HTML 5.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


| WCAG Success Criteria | WCAG Conformance Level |
| --- | --- |
| **1.3.1 Info & Relationships** | **A** |

#### How to implement

This section lists the techniques that should be used for accessible implementation of Layout Tables.

##### Designers Need to…

* Use CSS to position web page content.
* Use HTML heading mark-up, such as <h1> to <h6> to structure the content instead of <th> tags.
* Avoid using empty <th> cells. If a cell is blank, use <td> tag to structure it.
* In the case of merge cells, it is recommended to avoid using them in the first place. However, if the table content does not permit this option then make sure that appropriate HTML table attributes, such as “rowspan”, “colspan” are used to indicate the grouping of cells programmatically.

##### Developers Need to…

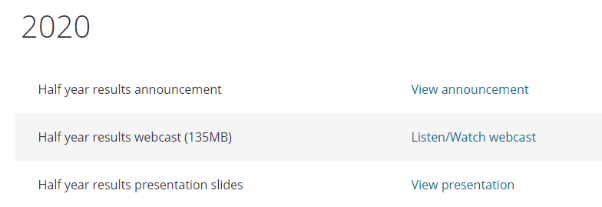
If layout tables are used:

* + Place the content in a logical order to make sure it linearizes correctly.
* Specify ARIA role “presentation” for the <table> element to make sure screen readers ignore the table mark-up.

#### Practices to Apply & Avoid

* Ensure structural table mark-up is not used for layout tables.
* Avoid using <th> element for formatting content presented using layout tables.

For example, in the Statistics course, “2020” is inaccurately marked as table header within a layout table.



In the above table <th> tag structural mark-up is used along with <h2> tag to add the heading.

To make the table accessible, implement the following changes:

* Remove <th> tag from the table’s code.
* Specify role “presentation” for the <table> element.

**Impacts User Groups-**

* Blind users
* Low-vision users

 Tip:

Screen readers do not identify layout tables with their default screen reading settings.

#### How to test for Accessibility?

To test layout tables for accessibility, perform the following steps:

1. Open the page in Google Chrome web browser.
2. Open the Web Developer extension. The extension is freely available on Chrome store.
3. Click on Outline table option available on the Outline tab.
4. All the tables present on the page will be highlighted with a border.
5. Right-click on the table and choose Inspect.
6. Check if structural mark-up, such as <th>, <caption>, scope attribute etc. are not used to mark-up the layout table.

### **Simple Data Tables**

Data tables are used to present related data. A data table consists of data cells and header cells. Table headers are added to provide context for users and if this contextual relationship is not exposed programmatically, it will pose accessibility barriers for people with visual impairments.

A simple data table comprises of either single set of column headers or row headers or both column and row headers. A simple data table does not generally include merged cells but at times might include blank cells. Screen readers identify the table headers and announce them for users as they navigate through the data.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


| WCAG Success Criteria | WCAG Conformance level |
| --- | --- |
| **1.3.1 Info & Relationships** | **A** |

#### How to implement

This section lists the techniques that should be used for accessible implementation of simple Data Tables.

##### Developers Need to…

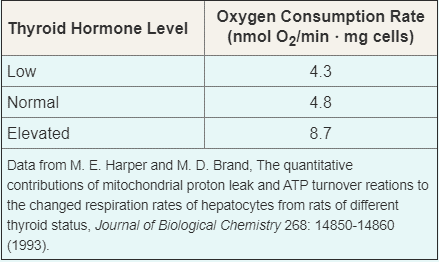
* Title of data table is inside the <caption> element.
* Column headers are inside <th scope="col"> or <th scope="colgroup"> elements.
* Row headers are inside <th scope="row"> or <th scope="rowgroup"> elements.
* Avoid using blank header cells.
* Header cells with text abbreviations that need expansion use the title attribute with the expanded text set as the value.

**Note:** An empty cell should never be marked-up using <th> tag.

#### Practices to Apply & Avoid

* Ensure the table header (<th>) is marked-up semantically and not through formatting.
* Avoid including non-tabular content in the <table> tag.

In the Pearson Biology course, table headers in the below example are conveyed through visual formatting using center alignment and <b> tag. Moreover, non-tabular content is included in the <table> tag.

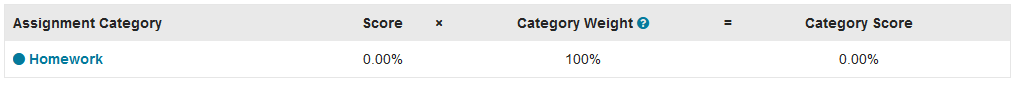


To make the above table accessible, make the following changes:

* Include the <th> tag to mark-up the table headers “Thyroid Hormone Level” and “Oxygen Consumption Rate (nmol O2/min ⋅ mg cells)”.
* Remove the <b> tag from the code as <th> tag by default applies bold style to the data.

For example, in the Pearson Biology course, row and column headers are not marked-up for the table. Moreover, the deprecated summary attribute is specified.

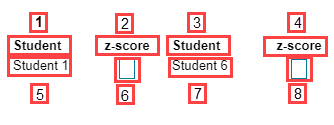
* Ensure to markup row and column header cells using <th> tags.
* Use valid mark-up to present summary information.



To make the above table accessible, implement the following changes:

* Mark-up the row and column headers “Assignment Category”, “Score ×”, “Category Weight”, “Category Score” and “Homework” using the <th> tags.
* Remove the summary attribute from the table mark-up as it is deprecated as per HTML 5.
* Use a hidden <p> tag to include the summary information and specify a unique ‘id’ attribute to associate it with the table via the aria-describedby attribute.
* In the case of tables, it is important to mark-up the tables correctly to ensure that they linearize in a logical order. Screen reader users read one cell at a time in left – to – right and top - to - bottom order.

For example, in the Pearson Sullivan Statistics course, the table fails to linearize in a logical order.



To make the above table accessible, implement the following changes:

* Use 2 columns to present the data instead of 4 columns.
* Alternatively, ensure that “Student 1” should be followed by “Student 2” and so on.

**Impacts User Groups-**

* Blind users
* Low-vision users

#### How to test for Accessibility?

To test a simple data table for accessibility, perform the following steps:

1. Open the page in Google Chrome web browser.
2. Open the Web Developer extension.
3. Click on Outline table option available on the Outline tab.
4. All the tables present on the page will be highlighted with a border.
5. Right-click on the table and choose Inspect.
6. Check if table headers are marked-up using <th> tags.
7. Check if the deprecated “summary” attribute is not used to present important information.

### **Complex Data Tables**

A complex data table comprises of two or more sets of column headers, row headers or column and row headers, as well as data cells.

Tabular data, if not coded keeping accessibility in mind, can pose accessibility issues for users with vision disabilities (who use screen readers). In fact, content presented using simple data tables can yet be understood but in the case of complex data tables, screen reader users often find it difficult to interpret the data.

In the case of complex table along with <th> tags, “headers” and “id” attributes need to be specified for data cells and table header cells, respectively. This helps in explicitly defining the relationship between data cells and their headers. This explicit association is helpful for screen readers in rendering the header cells when users navigate through the table data.

In order to make complex tables accessible, it is recommended to define the table caption and summary as well. Table caption is visible for all users whereas summary should be provided to orient users with visual impairments about how the data is organized in the table. Alternatively, complex tables can also be made accessible by breaking them in to multiple simple data tables. Simple tables are easy to comprehend by users with visual impairments.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


| WCAG Success Criteria | WCAG Conformance level |
| --- | --- |
| **1.3.1 Info & Relationships** | **A** |

#### How to implement

This section lists the techniques that should be used for accessible implementation of complex Data Tables.

##### Developers Need to…

* Title of data table is inside the <caption> element.
* Column headers are inside <th scope="col"> or <th scope="colgroup"> elements.
* Row headers are inside <th scope="row"> or <th scope="rowgroup"> elements.
* Avoid using blank header cells, use a <td> tag to structure blank cells.
* Header cells with text abbreviations that need expansion use the title attribute with the expanded text set as the value.
* Mark-up table header cells using <th> tag and define “id” attributes that can be referenced via the “header” attribute of the respective data cells.
* Use <col> and <colgroup> tags to programmatically define groups of columns within complex tables.
* Use <thead>, <tfoot> and <tbody> to programmatically define groups of rows.
  + <thead> and <tfoot> can be defined only once within a <table> tag whereas <tbody> can be defined multiple times.
* Use “rowspan” and “colspan” attributes for merged cells.
* Split a complex table into multiple simple data tables.

Tip:   
Accessibility testing tools for web pages do not flag errors for blank as well as merge cells.

Tip:   
Specify value for the “scope” attribute as “colgroup” or “rowgroup” to define the grouping of columns and rows programmatically.

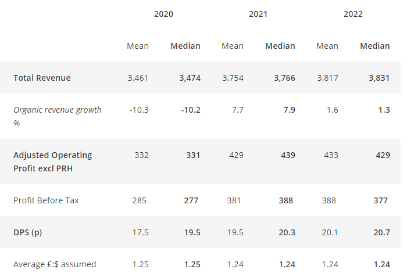
##### Content writers need to …

* Split data of a complex table into multiple simple tables.
  + Use one table per topic to keep things simple.
  + Repeat header text of merged cells within column and row headers and thereby eliminate the need to add a spanned/merged header in the first place.
  + Use “-“, “Nil”, “N/A” etc. instead of blank cells. This will remove ambiguity for all users.
* Use <caption> tag to add a visible heading/title to the table.
  + <caption> tag should be the first child element of <table> element.
* Define summary for the complex table using:
  + Hidden text before the table and referencing it via the aria-describedby attribute of <table> tag.
  + Use <details> and <summary> tags to display the table summary for a complex table in the form of a disclosure widget.

Tip:   
A good summary information should not repeat the caption information.

#### Practices to Apply & Avoid

* Ensure “header” and “id” attributes are defined for complex data tables.



For example, in the below table, “header” and “id” attributes are not defined and table headers are not marked-up correctly.

To make the above table accessible, implement the following changes:

* Use <th> tags to mark-up table headers.
* Ensure “header” and “id” attributes are defined to explicitly associate the table header and data cell information.
* Define “scope” attribute for the column and row headers.
* Add a <caption> tag to include a visible title for the table.
* Add summary information using a hidden <p> tag and include a unique ‘id’ attribute in its code. Reference the value of the ‘id’ attribute via the aria-describedby attribute of the <table> tag.

**Impacts User Groups**

* Blind users
* Low-vision users

#### How to test for Accessibility?

To test a complex data table for accessibility, perform the following steps:

1. Open the page in Google Chrome web browser.
2. Open the Web Developer extension.
3. Click on Outline table option available on the Outline tab.
4. All the tables present on the page will be highlighted with a border.
5. Right-click on the table and choose Inspect.
6. Check if structural mark-up, such as <th>, <caption>, scope attribute etc. as well as summary information is available for the complex table.
7. Check if “header” and “id” attributes are specified for data cells and header cells, respectively.

The below table lists keystrokes for accessing tables using the two leading screen readers on the Windows operating system:

| **Function** | **Jaws**  **Commands** | **NVDA**  **Commands** |
| --- | --- | --- |
| Go to Next Table | T | T |
| Go to Previous Table | Shift + T | Shift + T |
| List of Tables | Insert + Control +T | – |
| Move to next cell | Ctrl + Alt +Right Arrow | Ctrl +Alt +Right Arrow |
| Move to previous cell | Ctrl + Alt+ Left Arrow | Ctrl+ Alt+ Left Arrow |
| Move Down cell | Ctrl+ Alt+ Down Arrow | Ctrl + Alt + Down Arrow |
| Move Up cell | Ctrl + Alt + Up Arrow | Ctrl + Alt + Up Arrow |

### **Accessible Tables and ARIA**

Tables with rich interaction can be made accessible by using ARIA roles, states, and properties. ARIA provides attributes such as “aria-sort” for sorting tables and roles such as “grid”, “columnheader”, “rowheader”, “gridcell” etc. to programmatically define component information to assistive technologies.

Let us discuss different ARIA attributes related to tables in brief.

* aria-sort: used for tables with sorting functionality.
  + List of values include “ascending”, “descending”, “none” and “other”.
  + Aria-sort should be specified to roles such as columnheader, rowheader and the native <th> element.
* Role “grid” for implementing layout grids as well as data grids.
  + A grid can have different child elements from native HTML table elements as well as role “row”, “column”, “rowgroup”, “colgroup”, “gridcell” etc.
  + A grid is a composite widget and one element in the grid can receive tab focus. Once the grid is focused, authors should ensure that users can navigate between the gridcells using the Arrow keys.
  + Applying the role “grid” sets the screen reader in application mode.
  + Row and column headers are not required to receive keyboard focus.
* Based on the functionality that the grid offers, other ARIA attributes can be also used, such as   
  aria-readonly, aria-selected, aria-colindex, aria-rowindex, aria-colcount, aria-row-count etc. This is helpful when grids provide users with an option to add or delete rows and columns and therefore it is not possible for assistive technologies to compute the number of columns and rows.

Tip:  
A layout grid can be used to present non-tabular content whereas a data grid is used for presenting tabular content.

**Note:** In the case of sorting tables, it is recommended to use an image with updating alternate text to convey the sorting information instead of aria-sort as support for aria-sort is not uniform across different screen readers and browser combinations.

## **LINKS AND NAVIGATION**

**Links:**

Links are used on a website to help users navigate from one web page to another, one section of a web page to another and even to external websites.

**Navigation:**

Navigation mechanisms on a website should be used consistently throughout the website. Also, if icons, symbols, or images are used to identify navigation options, they should be used consistently throughout the website. This is found very helpful by users with visual and learning disabilities as they can identify the options easily and navigate across the website quickly.

Users should also be provided with a means to quickly locate information of interest. Multiple methods of locating content of interest should be given on a website, such as a sitemap, A to Z index, search mechanism, etc. This is found useful by all users, including those with disabilities, as they do not have to click on several links to find the content of interest.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


2.4.1 Bypass Blocks Level A

A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.

2.4.4 Link Purpose (In Context) Level A

The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.

### **Descriptive Link Text**

Link text used on a website should be descriptive of its target. Descriptive link text helps users to understand which page will get activated if they decide to follow the link.

Screen readers identify links for their users based on the text used to present the link. They also provide users with an option to access the links in the form of a list, whereby only the link text is displayed and not the surrounding text. Thus, it is recommended to use descriptive link text to ensure screen reader users can understand which page will get activated even when they are read out of context.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 2.4.4 Link Purpose (In Context) | A |

#### How to implement

This section lists the techniques that should be used for accessible implementation of descriptive Link Text.

##### UI Content Writers Need to…

Use unique text for describing links leading to different destination pages.

**Note:** The purpose of the link should be available from the link text itself or from the combination of the link text and the link’s programmatically determined context (paragraph, sentence, table header or list item).

##### Developers Need to…

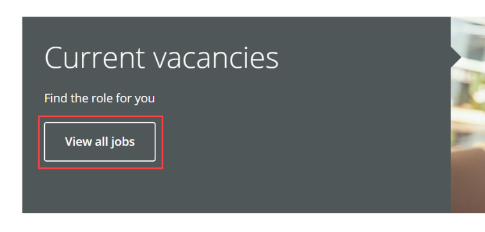
Use descriptive text to present links:

* + In the case of textual links, this can be done by adding descriptive text between the opening and closing <a> element.
  + In the case of image links, this can be done by specifying descriptive alt attribute value for the <img> element.
* Use aria-label, aria-labelledby and aria-describedby attributes to provide descriptive link text for screen reader users, if the design of the web page does not permit too much text to be added on the page itself.

#### Practices to Apply & Avoid

* Ensure that link text is meaningful within context.

For example, on the Pearson website, descriptive link text “View all jobs” is provided, thus making it clear to screen reader users where the link points to.



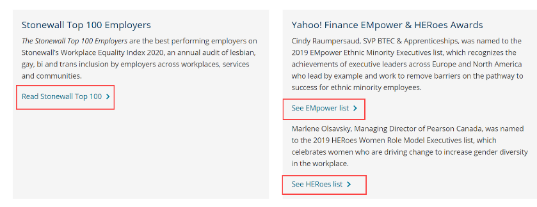
* Avoid using non-descriptive, identical link texts such as "Learn More”, “Read More”, etc.

For example, on the Pearson website, a non-descriptive link text “Learn More” is provided, thus making it unclear to screen reader users where the links point to.



* Ensure that link text is meaningful and unique within context.

For example, on the Pearson website, descriptive link text “Read Stonewall Top 100”, “See Empower list” and “See HERoes list” is provided, thus making it clear to screen reader users where the links point to.



User Groups Affected

Blind users

Low-vision users

Users with learning disabilities

Users with mobility disabilities – Speech input users

#### How to test for Accessibility?

To test a web page for descriptive link text, perform the following steps:

1. Open the web page that needs to be tested with screen reader running.
2. Navigate to the link by using the Tab key or pull up links list and check if the links are descriptive.
3. If the link text is not descriptive then it is an accessibility violation as per WCAG 2.1 success criteria 2.4.4 at Level A.

 Tip:

In JAWS and NVDA, link list can be pulled up using Insert +F7 keystroke.

### **Skip Links**

Skip links are in-page links that are added on a page to help keyboard users navigate to different parts of the page quickly. A few examples of skip links used on the web include “skip to main content”, “skip to navigation”, “skip to footer”, etc.

A “skip to main content” link is very useful to users with mobility and vision disabilities (who use a keyboard to access the web) as it helps them to bypass repetitive navigation links available at the top of web pages and directly access the core content of the page.

Web page authors can either use visible or invisible skip links depending on the visual presentation of the page. Alternatively, authors can opt to make the skip links visible when they receive keyboard focus by applying appropriate styles. However, it is recommended to use visible skip links, as it helps users know that skip links are available on the page.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 2.4.1 Bypass Blocks | A |

#### How to implement

This section lists the techniques that should be used for accessible implementation of skip links.

##### Developers Need to…

Add a visible “skip to content” link at the top of the page:

**Link:**

<a class="skip" href="#content">Skip to Content</a>

Point the link at the beginning of the main content of the page:

**Anchor:**

<div id="content">

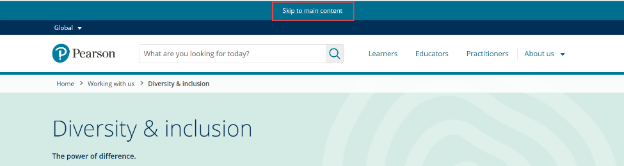
<h1 >Diversity and Inclusion</h1>

The above code will work on some browsers. In order for the skip links to work uniformly across desktop and mobile browsers, provide tabindex=”-1” to the beginning of the main content along with JavaScript .focus(); method. This will ensure that the focus moves to the bookmarked content for keyboard-only users rather than the page scrolling visually.

#### Practices to Apply & Avoid

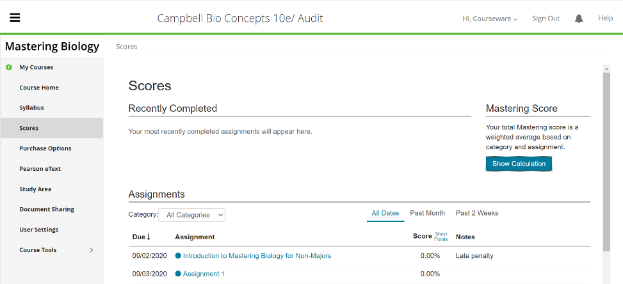
* Add a "skip to content" link at the beginning of each web page of a website.

For example, on the Pearson website, “Skip to main content” link is provided, thus making it easy for keyboard-only and screen reader users to skip pass repetitive navigation links and jump to sections of the page quickly.



* Avoid missing the "Skip to content" link on a web page.

For example, in the Pearson Campbell Biology Concepts course, the "Skip to content" link is not available across different pages.



User Groups Affected

Blind users

Low-vision users

Users with mobility disabilities

 Tip:

Position the skip link off-screen via CSS to hide it, rather than using ‘display:none’.

#### How to test for Accessibility?

To test a web page for the presence of skip link, perform the following steps:

1. Open the web page that needs to be tested.
2. Navigate to the page using keyboard and check the following conditions:

* Are skip links visible on page by default?
* If no, are skip links visible on keyboard focus?
* Do the skip links work appropriately?

1. If any of the conditions in step 2 are not fulfilled, then it is an accessibility violation as per WCAG 2.1 success criteria 2.4.1 at Level A.

### **Landmarks**

ARIA Landmarks programmatically identify sections of a page. This in turn helps assistive technologies such as screen readers to quickly navigate to and access the content available in each of those sections.

Accessible names should be specified for landmarks to help screen reader users identify the page sections easily and can differentiate between different navigation landmarks on a page. Accessible names can be specified using either aria=label or aria-labelledby attributes. If visible text is present, use aria-labelledby and in the absence of visual text use aria-label.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 2.4.1 Bypass Blocks | A |

#### How to implement

This section lists the techniques that need to be implemented to ensure Landmarks are defined and coded appropriately.

##### Developers Need to…

Add landmarks in the page source using the WAI ARIA role attribute. For example,

<div role="main">

Apply the following landmark roles on a web page:

* + banner
  + complementary
  + contentinfo
  + main
  + navigation
  + search
  + region

HTML 5 sectioning elements are mapped with the ARIA landmark roles:

|  |  |
| --- | --- |
| **WAI ARIA Landmark Role** | **HTML5 element** |
| Banner | <header> |
| Complementary | <aside> |
| Contentinfo | <footer> |
| Main | <main> |
| Navigation | <nav> |
| Search | None |
| Region | <section> |

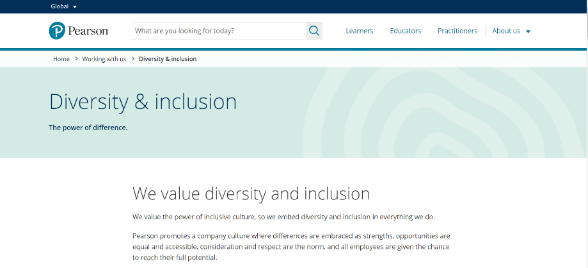
**Note:** A web page can have only 1 main and 1 banner landmark.

#### Practices to Apply & Avoid

* Add landmark roles to define sections on a web page.
* Add accessible names for landmarks when they are used multiple times on a web page.

For example, a page might have multiple navigation and region landmarks and it is recommended to specify accessible name for each landmark. This can be done using either “aria-label” or “aria-labelledby” attribute.

For example, on the Pearson website, landmark roles such as header, main and footer are defined appropriately for webpages.



* Avoid specifying landmark roles for empty elements as it adds up to the listening for screen reader users and defeats their purpose.

User Groups Affected

Blind users

Low-vision users

#### How to test for Accessibility?

To test a web page for landmark roles, perform the following steps:

1. Open the web page that needs to be tested with screen reader running.
2. Navigate to the page or pull up element list-landmarks and check if the landmarks are defined programmatically and are defined accurately.
3. If the landmark roles are defined inappropriately or multiple <header> and <main> elements are used, then it is an accessibility violation as per WCAG 2.1 success criteria 2.4.1 at Level A.

 Tip:

The elements list- landmarks can be pulled using insert + Control + R in JAWS and NVDA key (Insert) + F7 in NVDA.

## **STEM**

Science Technology Engineering Mathematics (STEM) content is one of the most challenging types of content as far as accessibility is concerned. At the same time, it is crucial to make it accessible to ensure learners with different types of disabilities can access the content using their Assistive Technology.

STEM education is important as it teaches logic and critical thinking. STEM education is also important as it is required in different job profiles on offer in the market. However, learners with disabilities are often at a disadvantage since STEM content is not created keeping accessibility in mind. STEM content consists of equations, subscripts, and superscripts as well as mathematic formulas to a large extent. It also includes simulations, experiments, diagrams, and highly interactive content.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


4.1.2 Name, Role, Value Level A

For all user interface components (including but not limited to: form elements, links and components generated by script), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

### **Math ML**

MathML is a mark-up language used for displaying mathematical equations on the web. Displaying math equations using MathML allows users to access the content using their assistive technologies rather than using images to present math content. Math content displayed using images does not scale and requires detailed textual descriptions for screen reader users to access them.

MathML on the other hand displays the complex equations semantically (stores the equation in the form of text) allowing users with low vision to scale them. Screen reader users can access the equations marked-up using MathML by reading parts of equation in a logical order which helps them to interpret complex equations as well.

Support for MathML content is present across most browsers. However, some of the browsers support it natively whereas others require MathJax. Most leading screen readers include support for MathML with and without MathJax. Math player is used along with browsers to access math content by screen reader users.

MathJax is a set of open source JavaScript libraries used for uniform display of MathML content across different browsers. It is useful for displaying math content on browsers that do not support MathML natively.

MathJax allows low vision users to scale the math content and offers flexibility to screen reader users for navigating complex math equations in parts. Implementing MathJax is relatively simple and all it needs is to insert JavaScript in the head section of the HTML document:

<script type="text/javascript"

src="http://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?config=TeX-AMS-MML\_HTMLorMML">

</script>

After inserting the script, use the <math> tag to display the math content. It is recommended to display math content inline with other web page content. However, authors have an option to use the attribute display:”block” to display the math content on a separate line. Additionally, it is a best practice that the <math> tag includes a namespace attribute (xmlns="http://www.w3.org/1998/Math/MathML").

Simple math content that includes subscript and superscript can be marked-up using the <sub> and <sup> tags. Screen readers with their default reading settings might not read the content marked-up using <sub> and <sup> tags. However, screen readers provide users with an option to configure their screen reader to read subscripts and superscripts on the web.

#### Practices to Apply & Avoid

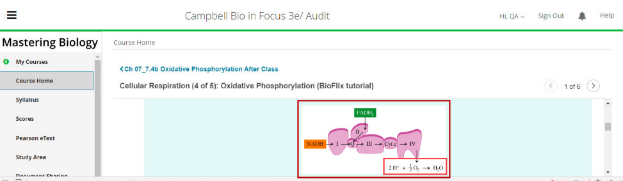
* Ensure subscripts are marked-up using the <sub> tag.

For example, in the Pearson Welding Level 1 course, “O2”, “N2” and “CO2” are marked up correctly using the <sub> tags and are announced as “O subscripts 2”, “N subscripts 2”, “CO subscripts 2” appropriately.

Content marked up correctly using the <sub> tag

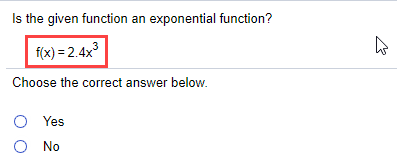
* Avoid marking up subscripts without the <sub> tag.

For example, in the Pearson Biology course the elements such as O2, H2O, etc. are not marked up using a <sub> tag and are announced as “O2”, “H2O”, etc. inappropriately.



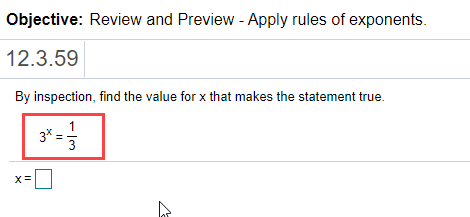
* Ensure superscripts are marked-up using the <sup> tags.

For example, in the Pearson Algebra Course, this equation is marked correctly and is announced as f(x) equals 2.4 x cubed appropriately.



* Avoid marking up superscripts without the <sup> tag.

For example, in the Pearson Algebra course, “3x =1/3” is not marked using the <sup> tag and is announced as “3 x Baseline equals one third” inappropriately.



## **ANIMATION**

Animation is used extensively on the web to create rich and engaging experiences as well as draw user’s attention. Sometimes animation is found in the form of images, auto-rotating carousels, flickering advertisements, important announcements etc. Animation is present on most online content today and thus it is essential to make it available to users in an accessible format.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


2.2.2 Pause, Stop, Hide Level A

For moving, blinking, scrolling, or auto-updating information, all of the following are true:

* Moving, blinking, scrolling: For any moving, blinking or scrolling information that (1) starts automatically, (2) lasts more than five seconds, and (3) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it unless the movement, blinking, or scrolling is part of an activity where it is essential; and
* Auto-updating: For any auto-updating information that (1) starts automatically and (2) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it or to control the frequency of the update unless the auto-updating is part of an activity where it is essential.

2.3.1 Three flashes or below Threshold Level A

Web pages do not contain anything that flashes more than three times in any one second period, or the flash is below the general flash and red flash thresholds.

2.3.2 Three Flashes Level AAA

Web pages do not contain anything that flashes more than three times in any one second period.

### **Moving and Blinking content**

Authors often use moving, blinking, scrolling and auto-updating content on a web page to attract a user’s attention. Even though such presentation techniques are visually appealing, they can pose accessibility issues for users with disabilities.

A few examples of moving, blinking, scrolling and auto-updating content include:

• **Moving content:** auto-rotating slideshows, carousels, etc.

• **Blinking content:** animated gif images, etc.

• **Scrolling content:** news items, weather reports for different cities, etc.

• **Auto-updating content:** live scorecard for tennis, soccer and other sports, stock prices, etc.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 2.2.2 Pause, Stop, Hide | A |

#### How to implement

This section lists the techniques that need to be implemented to make the moving, blinking, scrolling and auto-updating content accessible.

##### Developers Need to…

• In the case of moving, blinking and scrolling content, if the movement starts automatically and lasts for more than 5 seconds and is displayed along with other static content on a page:

* Provide users with an option to control the movement through start, stop and pause buttons or links.
* Inform screen reader users, in the form of hidden instruction, about the presence of moving content in advance, to help them access the information with ease.

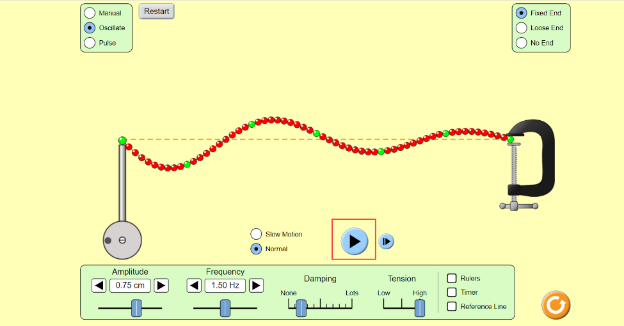
• In the case of auto-updating content, if the content starts updating automatically and is displayed along with other static content on a page:

* Provide users with options to start, stop, pause, and adjust the frequency of auto-updating content, in the form of buttons or links.
* Inform screen reader users, in the form of hidden instruction, about the presence of auto-updating content in advance, to help them access the information with ease.

#### Practices to Apply & Avoid

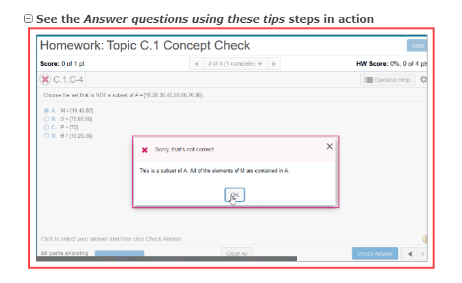
* Provide users with an option to play/pause moving and blinking content.

For example, in the Pearson Physics course, an animation based on string oscillation has a mechanism to play/pause it.



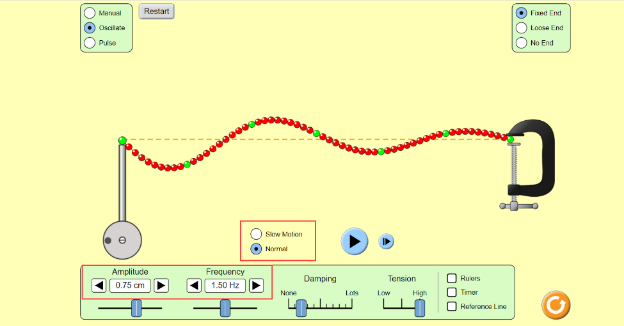
* Avoid displaying moving and blinking content that does not provide users with an option to play/pause the movement.

For example, on the Pearson website, an animated gif with “Tips on how to answer questions” is presented which has no mechanism to play/pause the same.



* Provide users with a means (in the form of increase, decrease buttons or links) to adjust the speed frequency of auto-updating content.

For example, in the Pearson Physics course, an animation based on string oscillation is presented with a means to adjust the motion of the oscillating string (i.e. slow motion and normal) as well the amplitude and frequency of the same.



* Avoid displaying auto-updating content that does not provide users with an option to adjust the updating frequency.

User Groups Affected

Users with attention deficit disorder (ADD)

Blind users

Low-vision users

#### How to test for Accessibility?

To test a web page for moving, blinking, scrolling and auto-updating content, perform the following steps:

1. Open the web page that needs to be tested in a browser.
2. Check if moving, blinking, scrolling and auto-updating content is displayed.
3. Check if pause / play button or link is provided for the moving, blinking and scrolling content.
4. Check if decrease and increase buttons or links are provided for adjusting the speed at which content is auto-updating.
5. If any of the conditions in step 2, 3 and 4 are not fulfilled, then it is an accessibility violation as per WCAG 2.1 success criteria 2.2.2 at Level A.

### **Flickering Content**

The W3C Web Content Accessibility Guidelines includes specific technical requirements for determining whether content flashes or flickers at an unsafe level. In general, if content flashes more than three times per second, it is unsafe. However, the W3C provides a more precise technical formula for calculating [general flash and red flash thresholds](http://www.w3.org/TR/WCAG20/#general-thresholddef). The Trace Center at the University of Maryland has developed a [Photosensitive Epilepsy Analysis Tool (PEAT)](https://trace.umd.edu/peat) for measuring whether web or computer applications are likely to cause seizures.

The best technique for addressing this issue is to avoid using content that flashes or flickers. Not only can it cause seizures, but it is likely to be annoying or distracting for users in general. If you must use content that flashes or flickers, test your content using methods described below to be sure your content flashes or flickers at a safe level.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 2.3.1 Three Flashes or Below Threshold  2.3.2 Three Flashes | A  AAA |

#### How to implement

This section lists the techniques that need to be implemented to make the flickering content accessible.

##### Developers Need to…

• Ensure that no component of the content flashes more than three times in any 1-second period.

• Keep the flashing area small enough.

• Use a tool to check if the content violates the general flash threshold or red flash threshold.

For more information on the above-mentioned techniques, refer:

<https://www.w3.org/WAI/WCAG21/Techniques/general/G19.html>

<https://www.w3.org/WAI/WCAG21/Techniques/general/G176.html>

<https://www.w3.org/WAI/WCAG21/Techniques/general/G15.html>

#### Practices to Apply & Avoid

* Avoid displaying flickering content that flashes more than three times in any one second period.

User Groups Affected

Users with attention deficit disorder (ADD)

#### How to test for Accessibility?

To test a web page for flickering content, perform the following steps:

1. Open the web page that needs to be tested in a browser.
2. Check if flickering content that flashes more than three times in any one second period is displayed.
3. If the condition in step 2 is not fulfilled, then it is an accessibility violation as per WCAG 2.1 success criteria 2.3.1 at Level A and 2.3.2 at Level AAA.

## **CSS AND ACCESSIBILITY**

Cascading Stylesheets (CSS) are used for presentation of web pages. Web designers use different type of stylesheets for presenting the web content. In web design, it is recommended to use HTML for displaying information and use CSS to control the presentation.

Information when presented using CSS will be ignored by assistive technologies and thereby result in users not being able to access the information. For example, if an informative image is defined through CSS alternate text cannot be specified through CSS. This will result in users with visual impairments missing out on important information.

When information is displayed using CSS, it is required to provide textual description through the mark-up to ensure that assistive technologies can render it for their users. Some experts might recommend not to include informative content through CSS. However, this is not always possible in modern web design. So, adding a fall back through HTML ensures that neither web page aesthetics nor accessibility is compromised.

CSS also helps to include information for the benefit of assistive technologies, such as screen readers. When hidden instructions are required to orient screen reader users about functionality of a user interface element it can be achieved through CSS. CSS if used correctly can enhance the accessibility of a web page.

Visual aspects of a web page, such as color, focus indicator, selected state of an user interface element, text resizing that are beneficial for users with low vision, colorblindness as well as people with learning impairments can be controlled through CSS. In this section, we will learn how CSS impacts web page’s accessibility.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


****1.4.4 Resize text:**** Level AA

Except for [captions](http://www.w3.org/TR/WCAG20/#captionsdef) and [images of text](http://www.w3.org/TR/WCAG20/#images-of-textdef), [text](http://www.w3.org/TR/WCAG20/#textdef) can be resized without [assistive technology](http://www.w3.org/TR/WCAG20/#atdef) up to 200 percent without loss of content or functionality.

1.4.10 Reflow Level AA

Content can be presented without loss of information or functionality, and without requiring scrolling in two dimensions for:

* Vertical scrolling content at a width equivalent to 320 CSS pixels.
* Horizontal scrolling content at a height equivalent to 256 CSS pixels.

1.4.12 Text Spacing Level AA

In content implemented using markup languages that support the following text style properties, no loss of content or functionality occurs by setting all the following and by changing no other style property:

* Line height (line spacing) to at least 1.5 times the font size.
* Spacing following paragraphs to at least 2 times the font size.
* Letter spacing (tracking) to at least 0.12 times the font size.
* Word spacing to at least 0.16 times the font size.

1.3.4 Orientation Level AA

Content does not restrict its view and operation to a single display orientation, such as portrait or landscape, unless a specific display orientation is essential.

2.4.7 Focus Visible Level AA

Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.

### **Text Resize**

Users should be able to resize text up to 200% using the browser's zoom feature. When the text is resized, content and functionality of the page should not be lost. Text resizing is useful to users with mild vision loss (low vision) who do not necessarily require assistive technologies to access the page content.

The requirement of text resizing was first introduced in WCAG 2.0 which was later enhanced in WCAG 2.1 under the success criteria of Reflow. We shall discuss about Reflow in the later section. When users zoom the web page up to 200%, make sure the following requirements are met:

* All the content and functionality should be available.
* All the text including form controls should resize
* Web page content should not overlap with each other
* Text should not get cut off

As per WCAG success criteria 1.4.4 at Level AA, text displayed in the form of captions and images of text are excluded from meeting the text resizing requirement.

@ Tip:

Websites can opt to provide users with text resizing controls to allow them to scale the content and access the content with ease.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.4.4 Text Resize | AA |

#### How to implement

This section lists the techniques that need to be implemented to ensure that content and functionality is not lost when the text is resized up to 200% using the browser's zoom feature.

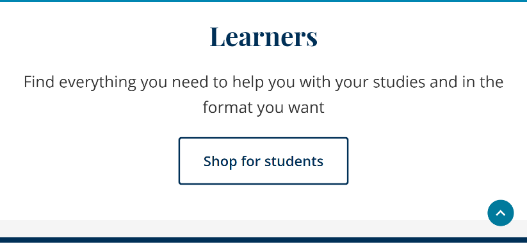
##### UI Designers Need to…

* Use relative font sizes, such as %, em, rem, etc., to display textual information.
* Use relative units, such as %, em, rem, etc., for defining container sizes.

#### Practices to Apply & Avoid

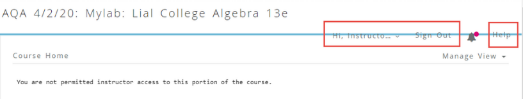
* Ensure that all the page content is available when the browser's zoom feature is used.

For example, on the Pearson website, the content such as “Learners Find everything…” resizes as per user requirements when zoomed up to 200% using browser feature.



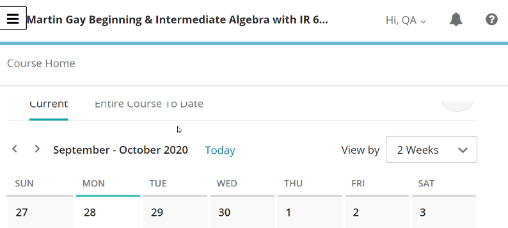
* Avoid defining container sizing using fixed units, as the content will overlap when viewed with the browser's zoom feature turned on.

For example, on Pearson course, content overlaps when viewed with browser's zoom feature turned on.



* Avoid defining container sizing using fixed units, as the content might not be available when viewed with the browser's zoom feature turned on.

For example, on Pearson website, content in main content area is unavailable with the browser's zoom feature turned on.



User Groups Affected

Low vision users

Elderly users

#### How to test for Accessibility?

To test a web page for text, resize, perform the following steps:

1. Open the web page that needs to be tested in Google Chrome.
2. Zoom the page up to 200% using browser’s zoom feature.
3. Check if all the content and functionality of the page are available appropriately.
4. If the conditions in step 3 fails, then it is an accessibility violation as per WCAG 2.1 success criteria 1.4.4 at Level AA.

### **Reflow**

Low vision users access the web content at high zoom levels so they can read the content in a single column. When the browser is zoomed up to 400%, it reflows, and the content is displayed in a single column. This helps users with low vision to read the content displayed in a linear format without requiring them to scroll horizontally.

All the web page content and functionality should be available when reflowed. WCAG 2.1 reflow requirement is not applicable for content that requires two-dimensional layout for usage or meaning, such as maps, images, diagrams, data tables, videos, presentations etc. When the web page content is reflowed-

* A horizontal scrollbar should not appear
* Content should not overlap with each other
* Content should not get cut off etc.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.4.10 Reflow | AA |

#### How to implement

This section lists the techniques that need to be implemented to ensure that content is available in a single column when the content is scaled to 400% using browser’s zoom feature.

##### UI Designers Need to…

Use relative units, such as %, em, rem, etc., for defining container sizes.

Use media queries or CSS grid layout or CSS Flexbox to ensure that the content reflows accurately.

Ensure that content reflows accurately when zoom to 400% with screen resolution as 1280\*1024.

#### Practices to Apply & Avoid

* Ensure that all the page content is visible in a single column when zoomed in.

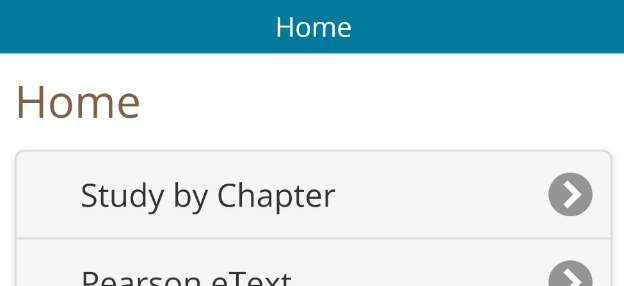
To find out the implementation of reflowing the content refer:

Media queries: <https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_mediaquery>

Grid layout: <https://www.w3schools.com/css/tryit.asp?filename=trycss_grid>

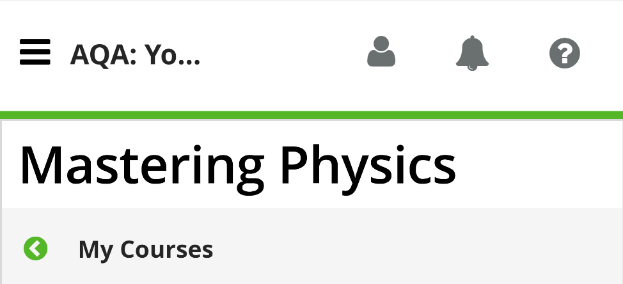
Flexbox: <https://www.w3schools.com/css/tryit.asp?filename=trycss3_flexbox_align-content_space-between>

For example, in the Physics course, the webpage content reflows accurately without any loss of content and functionality.



* Avoid using fixed units to define font and container sizes in CSS, as the content will not reflow when viewed with the browser's zoom feature turned on.

For example, in the Pearson Physics course, some content on the webpage is partially lost.



User Groups Affected

* Low vision users
* Elderly users

#### How to test for Accessibility?

To test if a web page reflows correctly, perform the following steps:

1. Open the web page that needs to be tested in Google Chrome.
2. Zoom the page up to 400% using browser’s zoom feature also set screen resolution as 1280\*1024.
3. Check if the content of the page reflows properly.
4. If the conditions in step 3 fails, then it is an accessibility violation as per WCAG 2.1 success criteria 1.4.10 at Level AA.

### **Text Spacing**

Text spacing requirement of WCAG 2.1 was introduced to assist users with low vision as well as users with learning and cognitive disabilities read the content with ease. When text spacing is supported by websites, users can increase the space between letters, words, sentences/lines, and paragraphs.

Low vision users and those with learning disabilities make use of custom style sheets to browse the web. When text spacing styles are implemented correctly, they can access the content with ease without the text being cut off or overlapping with other blocks of text.

WCAG 2.1 has outlined the text spacing requirements, such as:

* Line height (line spacing) to at least 1.5 times the font size.
* Spacing following paragraphs to at least 2 times the font size.
* Letter spacing (tracking) to at least 0.12 times the font size.
* Word spacing to at least 0.16 times the font size.

Human languages or scripts that do not make use of the above-mentioned text style properties in written text can confirm to the requirements by using the properties that exist for that language and scripts.

Video captions that are embedded directly in the video’s frame are not required to conform to the text spacing requirements whereas captions embedded using a separate caption file should conform to the above-mentioned text spacing requirements. Similarly, text created as an image (images of text) is not required to conform to the text spacing requirements.

**Note:** Canvas implementation of text is considered as images of text by WCAG 2.1 text spacing success criterion.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.4.12 Text Spacing | AA |

#### How to implement

This section lists the techniques that need to be implemented to ensure that content and functionality is available when text spacing styles are applied by users.

##### UI Designers Need to…

Define the text and container sizes using percent (%), em and rem units.

* Use CSS letter-spacing to control spacing within a word.
* Specify line spacing in CSS.

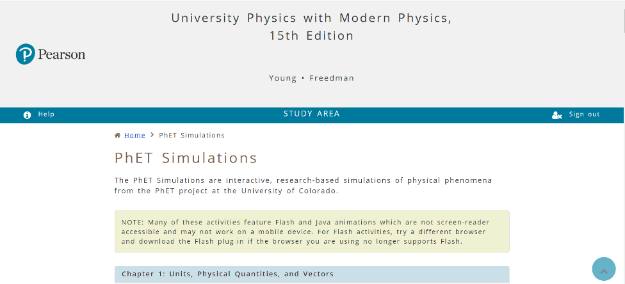
Practices to Apply & Avoid

* Ensure that all the page content and functionality is available when text spacing styles are applied by users.

For more information on implementation refer:

<https://www.w3.org/WAI/WCAG21/Understanding/text-spacing>

For example, in the Pearson Physics course, the page content and functionality are available when text spacing styles are applied by users.



* Avoid using fixed dimensions for text spacing, line spacing and paragraph spacing on a web page.

For example, in the Pearson Human Anatomy and Physiology course, the button text such as "Hi, Quality", “Sign Out” and “Help” and the heading “AQA 09/03/20: Mastering: Marieb: Human Anatomy and Physiology” is only partially available when users try to access the page with text spacing styles applied.



User Groups Affected

Low vision users

Dyslexic users

Users with cognitive disabilities

#### How to test for Accessibility?

To test a web page with text spacing styles applied, perform the following steps:

1. Open the web page that needs to be tested in a browser.
2. Run the following scripts on the page.

* [Text Spacing Bookmarklet](https://developer.paciellogroup.com/blog/2018/05/short-note-on-getting-spaced-out-with-sc-1-4-12-text-spacing/)

1. Check if the content and functionality of the page is available appropriately.
2. If the conditions in step 3 fails, then it is an accessibility violation as per WCAG 2.1 success criteria 1.4.12 at Level AA.

### **Orientation**

Websites and web-based applications are often designed to work in a particular orientation, i.e. landscape or portrait. However, from accessibility point of view it is essential to ensure that the website or web-based application is capable to display the content as per the user’s orientation setting. WCAG 2.1 requires that content of a website does not restrict to a specific orientation unless it is essential to display the content in a specific orientation.

Users with mobility disabilities tend to mount their mobile devices on the arm of their wheelchair and prefer to view the content based on their orientation setting. Similarly, users with low vision prefer to view the content in a landscape orientation as they can access more content at any given time.

Few examples of content where particular display orientation might be essential are a bank check, piano application, slides for a projector or television etc. Websites should be designed to work as per user’s display orientation preferences and should not lock the orientation to be either landscape or portrait for any content and functionality. All the web page content and functionality should be available in both landscape and portrait display orientations.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 1.3.4 Orientation | AA |

#### How to implement

This section lists the techniques that need to be implemented to ensure that the entire content of a page is available in both the orientations.

##### UI Designers Need to…

Use CSS to set the orientation to allow both landscape and portrait.

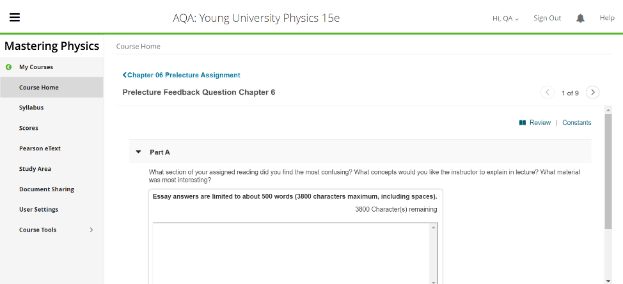
Use show/hide controls to allow access to content in different orientations.

#### Practices to Apply & Avoid

* Ensure entire web page content is displayed as per user’s device orientation and the content should adapt with change in device orientation.

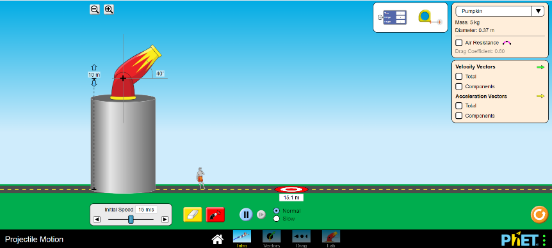
For more information on implementation refer:  
<https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_mediaquery_orientation>

For example, in the Pearson Physics course, the webpage content can be viewed as per user’s device orientation.



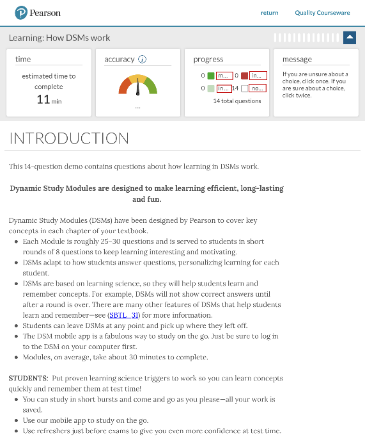
* Ensure that content of a web page is not displayed using fixed orientation.

For example, in the Pearson Physics course, the webpage cannot be oriented in mobile landscape mode.



* Ensure that there is no loss of content or functionality due to change in device orientation.

For example, in the Pearson Biology course, some content is partially lost when the webpage is viewed in mobile portrait mode.



User Groups Affected

Low-vision users

Users with mobility disabilities

#### How to test for Accessibility?

To test a web page for orientation, perform the following steps:

1. Open the web page that needs to be tested in a browser on a mobile device.
2. Change the orientation of the device.
3. Check if the orientation of the page changes as per the device orientation.
4. Check if there is no loss of content and functionality due to change in orientation.
5. If the conditions in steps 3 and 4 fail, then it is an accessibility violation as per WCAG 2.1 success criteria 1.3.4 at Level AA.

### **Keyboard Focus**

On screen, visual focus indication helps users find out about their location on a web page. In the absence of clear focus indication, users would find it difficult to understand where they are on a given web page. Clear focus indication is very useful to users with mobility and visual disabilities as they rely largely on a keyboard to interact with a web page.

W3C logo 
Web Content Accessibility Guidelines (WCAG) 2.1


|  |  |
| --- | --- |
| WCAG Success Criteria | WCAG Conformance Level |
| 2.4.7 Focus Visible | AA |

#### How to implement

This section lists the techniques that need to be implemented to ensure Keyboard focus is clearly visible on interactive elements.

##### UI Designers Need to…

Every browser displays a default focus outline when users navigate between interactive web page elements, such as links, buttons, form controls, image links, etc., using the “Tab” key.

Use CSS focus pseudo class to add a custom visual focus indicator for different interactive elements.

Specify value for CSS opacity property that is greater than 0.

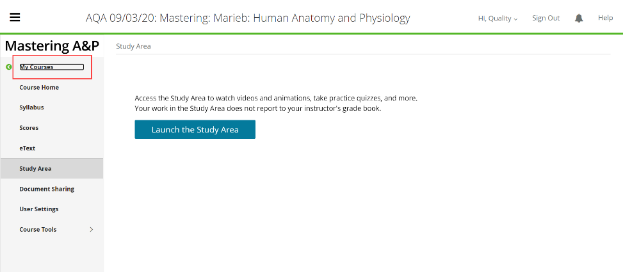
Use CSS outline property value that is clearly visible.

**Note:** Use CSS ‘display:none’ property to hide the content rather than positioning it off-screen. Content positioned off-screen might receive keyboard focus if it includes <a>, <button> or <input> elements.

#### Practices to Apply & Avoid

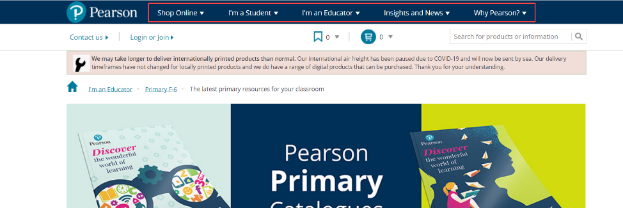
* Ensure that focusable elements do not have negative tabindex values specified. This will ensure that the elements are not removed from tab focus.
* Ensure that focus is clearly visible for all the interactive elements on a page.

For example, in the Pearson Human Anatomy and Physiology course, when accessing the links such as “My Courses”, “Course Home”, “Syllabus”, etc. through the keyboard, it is easy to navigate through the links, as the focus is clearly visible, and it is possible to identify the user’s current location on the page.



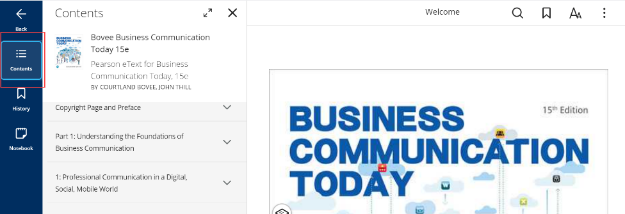
* Avoid applying “outline:0” or “outline:none” via CSS to hide the focus indicator.

For example, on the Pearson website, no focus indicator is visually available as the user tabs through the menu links such as “Shop Online”, “I’m a student”, “I’m an educator”, etc. using a keyboard.



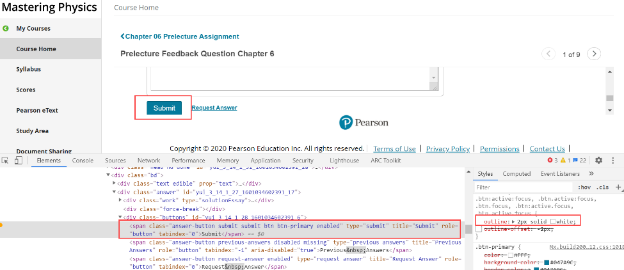
* Ensure that the color contrast of the focus indicator with its background is sufficient.

For example, in the Business Communication e-text on the Pearson website, the contrast for the focus indicator for the "Contents", “Notebook” and “History” buttons is sufficient.



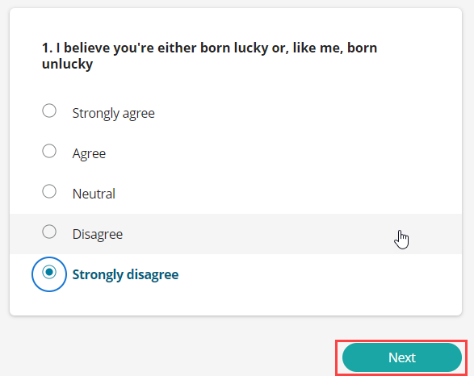
* Avoid using identical colors for the focus indicator and background color.

For example, in the Physics course, identical colors are used for the focus indicator and the background.



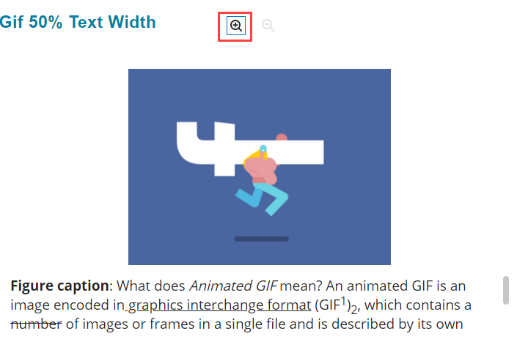
* Avoid using CSS property as opacity less than 0.

For example, on Pearson course, opacity property is used for displaying focus and is less than 0.



* Ensure opacity property of CSS has a value greater than 0 if it is used for indicating focus.

For example, on Pearson course, opacity property is used for displaying focus and is greater than 0.



User Groups Affected

Users with mobility disabilities: keyboard-only users

Low-vision users

Elderly users

#### How to test for Accessibility?

To test a web page for focus indicator, perform the following steps:

1. Open the web page that needs to be tested.
2. Using Tab key navigate to the page and check if focus is clearly visible on the interactive elements as we tab through them.
3. If the focus indicator is not visible, then it is an accessibility violation as per WCAG 2.1 success criteria 2.4.7 at Level AA.

## **APPENDIX**

|  |  |
| --- | --- |
| **Accessibility Resource** | **Where it can be found?** |
| Color Contrast Analyzer | [http://www.paciellogroup.com/resources/contrast-analyser.html](https://www.w3.org/WAI/tutorials/images/complex/) |
| Captioning tools | [ncam.wgbh.org/invent\_build/web\_multimedia/tools-guidelines/download-magpie](file:///C:/Users/WayneI/Downloads/ncam.wgbh.org/invent_build/web_multimedia/tools-guidelines/download-magpie) |
| Alt decision Tree | <https://www.w3.org/WAI/tutorials/images/decision-tree/> |
| Web Content Accessibility Guidelines 2.1 | <https://www.w3.org/TR/WCAG21/> |
| WAI Web Accessibility Tutorial | <https://www.w3.org/WAI/tutorials/> |
| WCAG 2.1 Checklist | <https://a11y.pearson.com/wp-content/uploads/2020/10/WCAG-2.1-Checklist.docx> |
| How people with disabilities use the web? | <https://www.w3.org/WAI/people-use-web/> |
| Pearson Accessibility Guidelines | <https://www.pearson.com/accessibility-guidelines.html> |
| Web Developer extension for Chrome | <https://chrome.google.com/webstore/detail/web-developer/bfbameneiokkgbdmiekhjnmfkcnldhhm> |
| Wave Extension | <https://chrome.google.com/webstore/detail/wave-evaluation-tool/jbbplnpkjmmeebjpijfedlgcdilocofh> |
| Bookmarklets - Paul J. Adam | <http://pauljadam.com/bookmarklets.html> |