

List of Level A Accessibility Guidelines as per WCAG 2.1

// Success Criteria 1.1.1 Text-Alternatives

```
accessibleCodeExample =
``;
inaccessibleCodeExample =
`          <!-- The alternative text
should be provided so users can understand the content even if they cannot
see the image. -->`;
```

// Success Criteria 1.2.1 Captions for videos

```
accessibleCodeExample =
`<video controls>
    <source src="video.mp4" type="video/mp4">
    <track kind="captions" label="English" src="captions.vtt"
srclang="en">
</video>`;
inaccessibleCodeExample =
`<video controls>          <!-- Subtitles to be video should be
provided. -->
    <source src="video.mp4" type="video/mp4">
</video>`;
```

// Success Criteria 1.3.1

```
accessibleCodeExample =
`<article>
    <h1>Main Article Title</h1>
    <p>Article content goes here.</p>
</article>`;
inaccessibleCodeExample =
`<div>          <!-- Semantic HTML elements provide meaning and
context to the content, making it more accessible. -->
    <span>Main Article Title</span>
    <p>Article content goes here.</p>
</div>`;
```

// Success Criteria 1.3.2 Meaningful sequences

```
accessibleCodeExample =
`<article>
  <h2>Article Title</h2>
  <p>Article content goes here.</p>
</article>`;
inaccessibleCodeExample =
`<article>                                <!-- Heading of the article should appear before
the content. -->
  <p>Article content goes here.</p>
  <h2>Article Title</h2>
</article>`;
```

// Success Criteria 1.3.3 Don't solely rely on sensory characteristics of the content such as shape, color, size, visual location, orientation, or sound.

```
accessibleCodeExample =
`<div id="wrong-answer" style="background-color: red; width: 100px;
height: 100px;">X</div>`;
inaccessibleCodeExample =
`<div id="div-to-show-wrong-answer" style="background-color: red; width:
100px; height: 100px;"></div>                                <!-- Don't solely rely on
sensory characteristics of the content such as shape, color, size, visual
location, orientation, or sound. -->`;
```

// Success Criteria 1.4.1 Ensure information conveyed by color is also conveyed without color

```
accessibleCodeExample =
`<h2 style="color: red;"><span style="border: 2px solid red;">Important
Heading</span></h2>
  <p>This is some important information.</p>`;
inaccessibleCodeExample =
`<h2 style="color: red;">Important Heading</h2>                                <!--
Information conveyed by color should also be conveyed without color. -->
  <p>This is some important information.</p>`;
```

// Success Criteria 1.4.2 Provide controls to pause, stop, or adjust the volume of audio content

```
accessibleCodeExample =
`<audio controls>
  <source src="audio.mp3" type="audio/mp3">Your browser does not
support the audio element.
</audio>`;
```

```
inaccessibleCodeExample =
`<audio autoplay>           <!-- Controls to pause, stop, or adjust
the volume of audio content should be provided to the user. -->
    <source src="autoplay-audio.mp3" type="audio/mp3">Your browser
does not support the audio element.
    </audio>`;
```

// Success Criteria 2.1.1 All Content should be focusable

```
accessibleCodeExample =
`<button onclick="myFunction()" tabindex="0">Click me</button>`;
inaccessibleCodeExample =
`<div onclick="myFunction()">Click me</div>           <!-- All contents
should be focusable. -->`;
```

// Success Criteria 2.1.4 Avoid the use of single-character key shortcuts

```
accessibleCodeExample =
`<button onclick="submitForm()">Submit</button>`;
inaccessibleCodeExample =
`<button onclick="submitForm()" accesskey="s">Submit</button>
<!-- Use of single-character key shortcuts should be avoided. -->`;
```

// Success Criteria 2.2.1 Allow users to adjust or disable session timeouts.

```
accessibleCodeExample =
`<meta http-equiv="refresh" content="600;url=logout.html">`;
inaccessibleCodeExample =
`<meta http-equiv="refresh" content="300;url=logout.html">
<!-- Users should be allowed adjust or disable session timeouts. -->`;
```

// Success Criteria 2.2.2 Automatic refresh is set to 0 seconds, meaning it won't automatically refresh. Additionally, a button is provided, allowing users to control when they navigate to the specified URL.

```
accessibleCodeExample =
`<button onclick="location.href='example.com'">Go to Example</button>`;
inaccessibleCodeExample =
`<meta http-equiv="refresh" content="5;url=example.com">
<!-- Automatic redirections after specific time can be disorienting and
frustrating for users, especially if they weren't expecting it. -->`;
```

// Success Criteria 2.3.1 No part of content should flash more than three times per second

```
accessibleCodeExample =
`<div style="animation: flashing 1s infinite;"></div>`;
```

```
inaccessibleCodeExample =
`<div style="animation: flashing 0.25s infinite;"></div>
<!-- No part of content should flash more than three times per second.
-->`;
```

// Success Criteria 2.4.1 Navigation is easy

```
accessibleCodeExample =
`<a href="#main-content" class="skip-link">Skip to important section</a>
  <div id="main-content">Important content here</div>`;
inaccessibleCodeExample =
`<div>Important Section</div> <!-- Provide navigations to
important sections. -->`;
```

// Success Criteria 2.4.2 Add descriptive and concise title for the page

```
accessibleCodeExample =
`<title>Accessible Page Title</title>`;
inaccessibleCodeExample =
`<title></title> <!-- Add descriptive and concise title for
the webpage. -->`;
```

// Success Criteria 2.4.3 Meaningful order of focus

```
accessibleCodeExample =
`<form>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username">
  <label for="password">Password:</label>
  <input type="password" id="password" name="password">
  <button type="submit">Submit</button>
</form>`;
inaccessibleCodeExample =
`<form> <!-- Form contents should be organized in a
meaningful order. -->
  <button type="submit">Submit</button>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username">
  <label for="password">Password:</label>
  <input type="password" id="password" name="password">
</form>`;
```

// Success Criteria 2.4.4 Link purpose must be stated clearly

```
accessibleCodeExample =
`<a href="/contact">Contact Us</a>`;
inaccessibleCodeExample =
`<a href="/page1">Click here</a>`           <!-- Link purpose must be
stated clearly. -->`;
```

// Success Criteria 2.5.1 Functionality triggered by a pointer gesture is also available through a single pointer without requiring a path-based gesture

```
accessibleCodeExample =
`<div onpinch="openMenu()">Pinch to Open Menu</div>
  <button onclick="openMenu()">Open Menu</button>`;
inaccessibleCodeExample =
`<div onpinch="openMenu()">Pinch to Open Menu</div>`           <!--
Functionality triggered by a pointer gesture should also be available
through a single pointer without requiring a path-based gesture. -->`;
```

// Success Criteria 2.5.2 Provide alternatives for content that relies on motion, like autoplaying videos

```
accessibleCodeExample =
`<video controls>
  <source src="video.mp4" type="video/mp4">
  <p>Your browser does not support the video tag.</p>
</video>`;
inaccessibleCodeExample =
`<video autoplay>`           <!-- Users may find autoplay annoying,
especially if they're not expecting it or if it consumes bandwidth
unnecessarily. -->
  <source src="motion-video.mp4" type="video/mp4">
  <p>Your browser does not support the video tag.</p>
</video>`;
```

// Success Criteria 2.5.3 Label purpose must be clear

```
accessibleCodeExample =
`<label for="search">Search:</label>
  <input type="text" id="search" name="search">`;
inaccessibleCodeExample =
`<label for="search">Field1:</label>`           <!-- Label purpose must
be stated clearly. -->
  <input type="text" id="search" name="search">`;
```

// Success Criteria 2.5.4 Provide alternatives for functionality that relies on motion or requires specific gestures.

```
accessibleCodeExample =
`<div onshake="shakeDevice()">Shake to Activate</div>
  <button onclick="shakeDevice()">Shake Device</button>`;
inaccessibleCodeExample =
`<div onshake="shakeDevice()">Shake to Activate</div>                                <!--
Alternatives for functionality that relies on motion or requires specific
gestures must be provided. -->`;
```

// Success Criteria 3.1.1 Language of page must be mentioned

```
accessibleCodeExample =
`<html lang="en">`;
inaccessibleCodeExample =
`<html>                                <!-- Language of page must be mentioned. -->`;
```

// Success Criteria 3.2.1 Inconsistency in navigation

```
accessibleCodeExample =
`<nav>
  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/about">About</a></li>
    <li><a href="/contact">Contact</a></li>
  </ul>
</nav>`;
inaccessibleCodeExample =
`<nav>                                <!-- Ways to navigate should be consistent. -->
  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/about">About</a></li>
    <li><div onclick="toggleContact()">Contact</div></li>
  </ul>
</nav>`;
```

// Success Criteria 3.3.1 Form submission is only allowed if there are no errors

```
accessibleCodeExample =
`<form action="/submit" method="post" onsubmit="return checkForErrors()">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username">
  <button type="submit">Submit</button>
</form>`;
```

```

inaccessibleCodeExample =
`<form action="/submit" method="post">                                <!-- Form submission
should be allowed only if there are no errors. -->
    <label for="username">Username:</label>
    <input type="text" id="username" name="username">
    <button type="submit" onclick="checkForErrors()">Submit</button>
</form>`;

```

// Success Criteria 3.3.2 Provide labels for form controls using the label element

```

accessibleCodeExample =
`<label for="name">Name:</label>
  <input type="text" id="name" name="name">`;
inaccessibleCodeExample =
`<input type="text" id="name" name="name">                                <!-- Labels for
form controls should be provided using the label element. -->`;

```

// Success Criteria 3.3.7 If same question is repeated, make sure that input is autopopulated

```

accessibleCodeExample =
`<label for="userInput">Enter something:</label>
  <input type="text" id="userInput">
  <br>
  <label for="userInputAgain">Enter something:</label>
  <input type="text" id="userInputAgain"
onblur="storePreviousEntry(this)" onfocus="autopopulate(this)">`;
inaccessibleCodeExample =
`<label for="userInput">Enter something:</label>                                <!-- If
same question is repeated, make sure that input is autopopulated. -->
  <input type="text" id="userInput">
  <br>
  <label for="userInputAgain">Enter something:</label>
  <input type="text" id="userInputAgain">`;

```

// Success Criteria 4.1.2 User interface components must have accessible names and roles

```

accessibleCodeExample =
`<button aria-label="Close" onclick="closeDialog()">X</button>`;
inaccessibleCodeExample =
`<button onclick="closeDialog()">X</button>                                <!-- User
interface components must have accessible names and roles. -->`;

```