

cristina.gena@unito.it

Accessibility by W3school

<https://www.w3schools.com/accessibility/index.php>



You are not the user

- Always remember that you are not the user. As designers and developers, we often think that what can be used, accessed and understood by us, also is good enough for others.

Diversity

- There are close to 8 billion people in the world. According to World Health Organization, there are over one billion people with disabilities.
- **That is about 15 % of the population.** Access to information is a human right.



Groups of disabilities

Auditory

- People with hearing loss. **Remember transcripts for audio and captions for video.**

Cognitive, learning and neurological

- People that may have challenges understanding information. **Remember a clear structure, consistent labeling, predictable links and plain language.** This helps everyone regardless of ability or disability.

Physical

- People with limitations in their muscular control, sensations, joint disorders or missing limbs. **Remember to support assistive technologies like switches and voice recognition and to have sufficient target sizes.** Target size is the size of a interactive element, the clickable or tapable area.

Speech

- Some people are not able to speak. Do not make your service voice-based only.

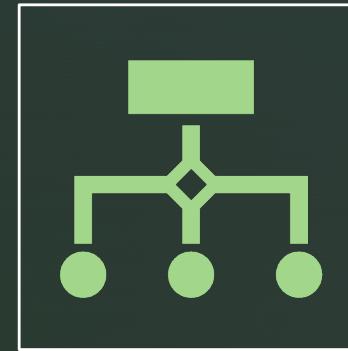
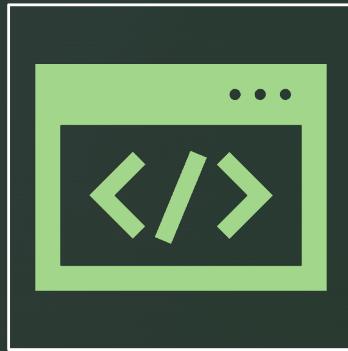
Visual

- People with reduced vision. **Make sure text and images can be enlarged, you use sufficient contrast, that you do not rely on color alone, and that the content supports text-to-speech**

Permanent, temporary and situational disabilities

Touch	See	Hear	Speak
<ul style="list-style-type: none">• A person with one arm has a permanent disability• A person with a broken arm has a temporary disability• A new parent holding a baby on one arm has a situational disability	<ul style="list-style-type: none">• A blind person has a permanent disability• A cataract can be temporary• A distracted driver has a situational disability	<ul style="list-style-type: none">• A deaf person has a permanent disability• An ear infection is temporary• A bartender has a situational disability	<ul style="list-style-type: none">• A non-verbal person has a permanent disability• A person with laryngitis has a temporary disability• A person with a heavy accent has a situational disability

Semantic elements = elements with a meaning.



Provide the user a good way to navigate and interact with your site. Make your HTML semantic.

Semantics is about using the correct element in HTML. There are approximately 110 elements in HTML 5.

Two of them have no meaning – <div> and . They tell us nothing about the content.

They have no built in accessibility features, so we should always check to see if any other elements are better suited.

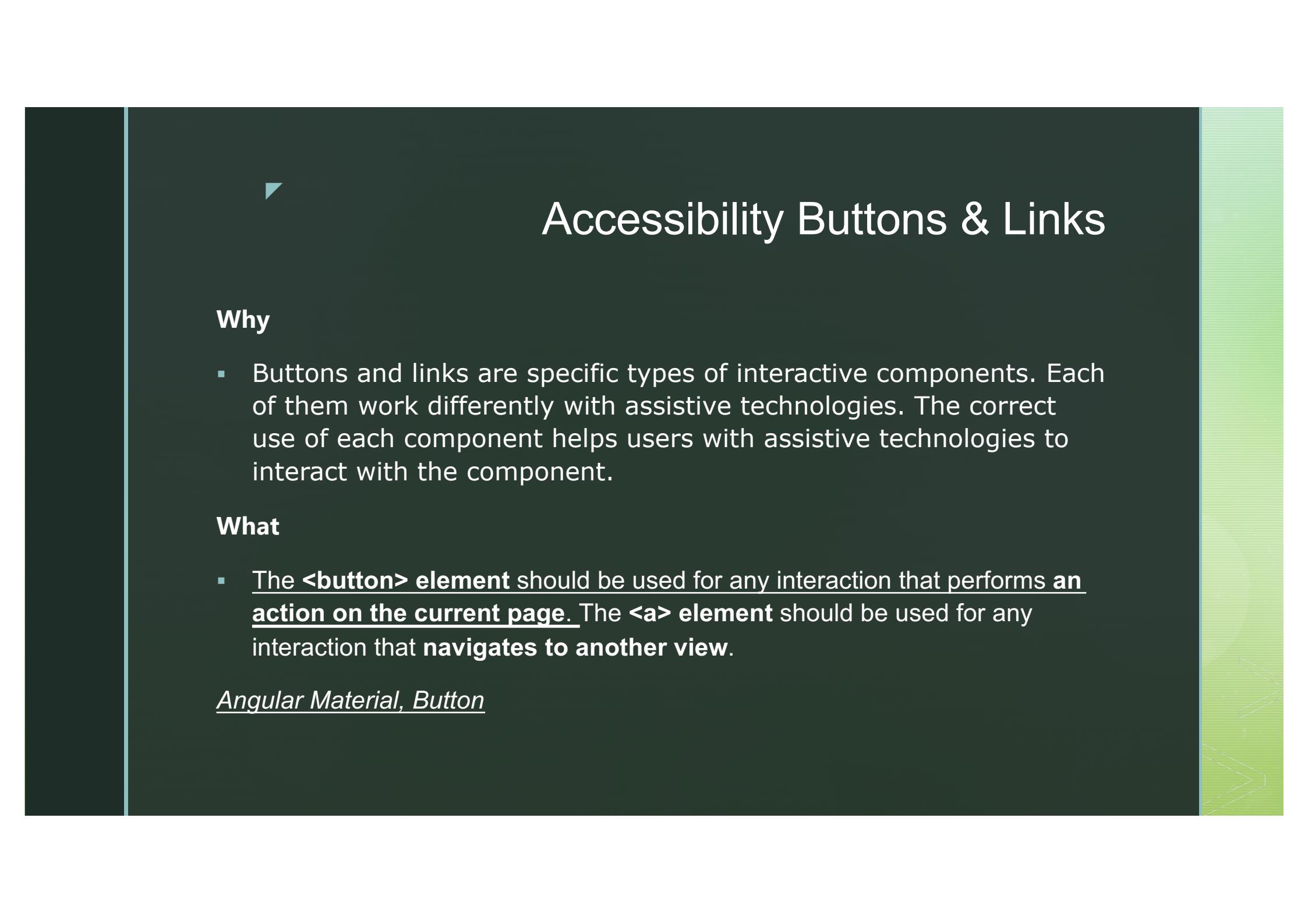
Example of semantic elements: <form>, <table> and <article>. They clearly define the content.

Accessibility Landmarks

- With landmarks, blind users using a screen reader have the ability to jump to sections of a web page.

In HTML there are some semantic elements that can be used to define different parts of a web page:

- <header>
- <nav>
- <main>
- <aside>
- <section>
- <footer>



Accessibility Buttons & Links

Why

- Buttons and links are specific types of interactive components. Each of them work differently with assistive technologies. The correct use of each component helps users with assistive technologies to interact with the component.

What

- The **<button> element** should be used for any interaction that performs an action on the current page. The **<a> element** should be used for any interaction that **navigates to another view**.

Angular Material, Button



Proper links

- Links take users from one page to another, or sometimes to another part of the page. For a link to be accessible, remember to:
 - Use the href attribute to specify the link destination.
 - Use a proper URL in the href attribute. The URL can be absolute or relative.
 - <https://uber.com/about> is an absolute URL.
 - </about> is a relative URL.
 - Not simulate a link with other elements like or <div>.
 - Open the link in the current window. It is not recommended to open links in a new window.

Accessibility Role, Name & Value

Why

User interface components need
a role,
a name
and sometimes a value

to ensure that people using assistive technologies are able to use them.

Examples of assistive technologies are screen readers, switch controls and speech recognition software.

What

There are two cases where we can't use a good HTML element with built-in accessibility features, even though we want to:

There is no native HTML element for what we are trying to achieve.

There are technical limitations that prevents us using the semantically correct element.

In both cases, we need to build a custom control. An important accessibility principle is that a custom control needs a role, a name and sometimes a value.

Role

- A dropdown menu button should be coded as a <button>.
 - What if our framework does not allow us to do that? If it forces us to use an <a> instead? If the navigation component in the library we are using, is built with <a>s?
 - Then we need to add a role.
- This is done with the **role="button"** attribute.
 - Now users of assistive technologies can understand what the custom control is.



WAI-ARIA

approfondimento

WAI-ARIA

- ARIA = Accessibile Rich Internet Applications
- <https://www.w3.org/TR/wai-aria/>
- Insieme di metodi per fornire alle tecnologie assistive indicazioni sul funzionamento di un'applicazione web:
 - indicare il **tipo di widget** presentato ("menu", "treeitem", "slider", "progressmeter")
 - indicare il **ruolo** assunto da un elemento nella struttura della pagina web
 - indicare lo **stato** in cui si trova un widget (es. checked per un checkbox)
 - definire le aree “**live**”, ossia quelle che possono essere aggiornate dall'applicazione a seguito dell'interazione con l'utente
 - gestire interazioni basate su drag and drop
 - fornire una navigazione / uso basati su tastiera per tutto quanto menzionato

ARIA Landmarks

Consentono allo screen reader di identificare delle “aree di navigazione” della pagina, a cui l’utente può accedere agevolmente.

I tag strutturali HTML5 creano automaticamente i landmark principali, anche se può essere utile esplicitarli per ragioni di compatibilità.

- <aside role="complementary">
- <footer role="contentinfo">
- <header role="banner">
- <main role="main">
- <nav role="navigation">
- <section role="region">
(con aria-label="" oppure aria-labeledby="")
- <form role="form">
(con aria-label="" oppure aria-labeledby="" oppure title="")
- <form role="search">

ARIA roles

Metodi per comunicare alla tecnologia assistiva quale ruolo semantico abbia un elemento dell'interfaccia, **indipendentemente dal suo markup**

Utili per rendere più accessibili elementi costruiti con markup originale, come controlli UI custom

Esempio:

`Etichetta`

(ma sarebbe meglio usare `<button>` !!!)

button, checkbox, gridcell, link, menuitem, menuitemcheckbox, menuitemradio, option, progressbar, radio, scrollbar, searchbox, separator, slider, spinbutton, switch, tab,tabpanel, textbox, treeitem, combobox, grid, listbox, menu, menubar, radiogroup, tablist, tree, treegrid, application, article, cell, columnheader, definition, directory, document, feed, figure, group, heading, img, list, listitem, math, none, note, presentation, row, rowgroup, rowheader, table, term, toolbar, tooltip

[https://www.w3.org/TR/wai-aria/#roles categorization](https://www.w3.org/TR/wai-aria/#roles_categorization)

ARIA live regions

- Metodo per avvisare l'utente in merito a cambiamenti avvenuti nel DOM
 - https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Live_Regions
- Elemento che controlla l'aggiornamento:
 - <button **aria-controls="validazione"**
- Area che viene aggiornata:
 - <div **role="region"** **aria-live="polite"** **aria-labelledby="validazione_titolo"** **aria-describedby="validazione_desc"** **id="validazione"**>
 <h3 id="validazione_titolo">Errori di validazione</h3>
 <p id="validazione_desc">Si sono verificati...</p>
 </div>

- role =
 - “log”
 - “status”
 - “alert”
 - “progressbar”
 - “marquee”
 - “timer”
- aria-live=
 - “polite”
 - “assertive”
 - “off” (default)
- aria-labeledby=”id_titolo”
- aria-describedby=”id_descrizione”
- aria-atomic=”false”
- aria-relevant=
 - “additions”
 - “removals”
 - “text”
 - “all”

Normally, only **aria-live="polite"** is used. Any region which receives updates that are important for the user to receive, but not so rapid as to be annoying, should receive this attribute. The screen reader will speak changes whenever the user is idle.

For regions which are not important, or would be annoying because of rapid updates or other reasons, silence them with **aria-live="off"**.

aria-atomic: The **aria-atomic=BOOLEAN** is used to set whether or not the screen reader should always present the live region as a whole, even if only part of the region changes. The possible settings are: false or true. The default setting is false.

aria-relevant : The **aria-relevant=[LIST_OF_CHANGES]** is used to set what types of changes are relevant to a live region. The possible settings are one or more of: additions, removals, text, all. The default setting is: **additions text**.

Name

```
<select aria-label="Country calling code" name="countryCode">...</select>
```

- The dropdown is coded with a <select>, which is a correct element for this case.
 - <select name="countryCode">...</select>
- It has a name attribute. **This is not the same as an accessible name.** This is confusing. The name attribute is for computers. In a <form>, it is used as a reference when the data is submitted. This name countryCode will not help any users. It will not be picked up by assistive technologies.
- To give this <select> an accessible name, we must use the attribute aria-label. Normally, we would have connected a visual label to the <select> component. In this case, there is only one visual label for both the components..

Name

- An accordion is considered a custom component. There is no standard HTML element to use here. Each accordion header should be a <button> or role="button":
- <div role="button">When do I get charged for a ride?</div>
- Good. It has the role of a button. It also has a name, the content of the div.
- **To give this button a value, we need to tell assistive technologies that it is closed. This is done with aria-expanded="false":**
- Now, our accordion header has a role, name and a value.

```
<div role="button" aria-expanded="false">When do I get charged for a ride?</div>
```

Accessibility Color Contrast

Why

- Text and graphical components on a web page need good contrast so that we make sure that it is perceivable for users. Some of us have reduced vision. Others will be in a situation where contrast is important, like out in a bright sunlight.

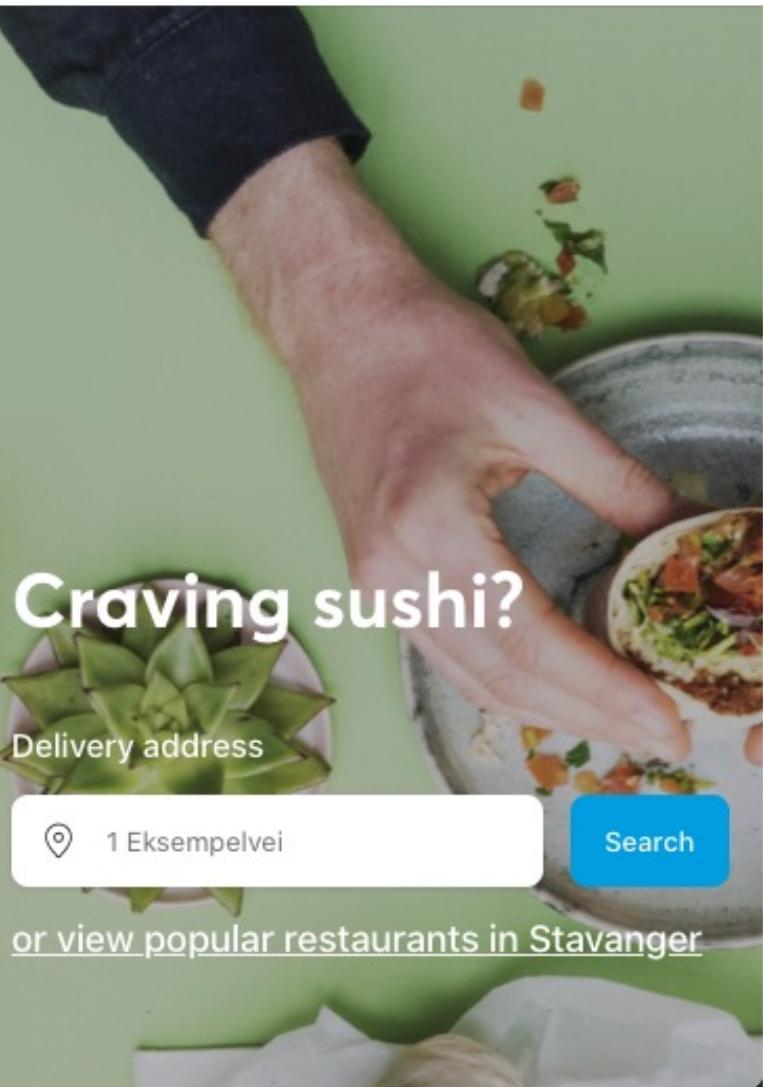
What

- We measure contrast between text or graphics against the background color. This is called contrast ratio. A white text on a white background has a contrast ratio of 1. This is impossible to perceive. Black text on a white background has a contrast ratio of 21.
- There is no perfect ratio. It is not always as high as possible, even though a high contrast is usually more readable than a low contrast. According to Apple, we should strive for a minimum of 4.5, although 7 is preferred.

How

- One way to measure the color contrast is to use a tool like Contrast Ratio. This accepts multiple color inputs, like RGB, HSL and hex. It even supports transparency, like RGBA.

Wolt



Text on images

- To measure contrast on text on top of a background image, we need to find the **brightest or darkest** part of the image. If the text is bright, look for the brightest part and vice versa.
- In this example from Wolt, we have white text on a bright background image.
- Using a color picker on a light green section gives us the hex value `#a1ad95`. This is a contrast ratio of 2.35. Not sufficient.
- One possible improvement is to add a color **overlay** on that part of the picture with text. The overlay can be solid or have a degree of opacity.



Different states

- Any interactive component has different states – hover, focus, active, unvisited, visited and deactivated.
- **Remember to ensure that the states also have good contrast.**
Working with states, we have to think about two scenarios:
 - Color contrast needs to be good for all states.
 - Change of contrast from unfocused to focused state is at least 3.

<https://coolors.co/contrast-checker/>

▼ Accessibility Color Alone as Meaning

Why

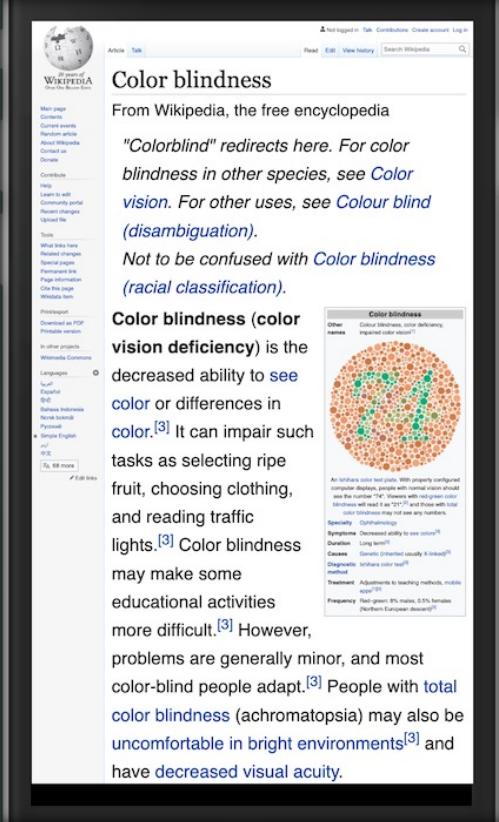
- Not everyone perceives color the same way. Red-green color blindness is the most common form, it affects up to 8% of males. Some use grayscale mode to curb their phone addiction.

What

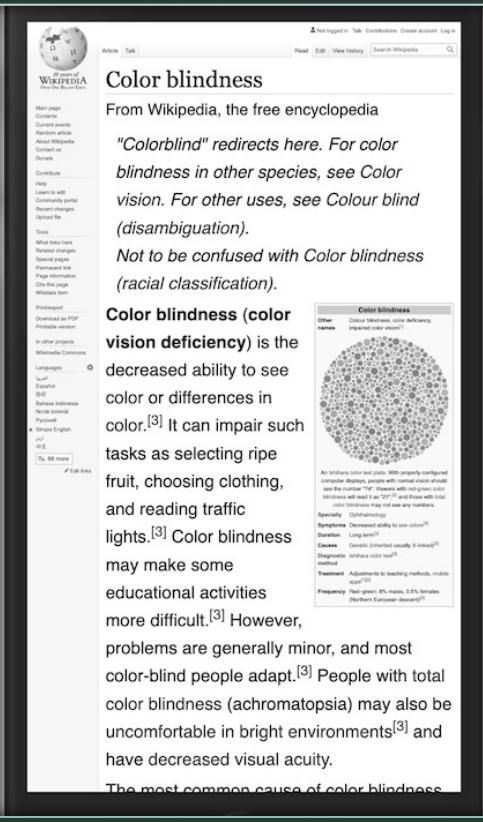
- Do not use color as the *only* visual indicator of a meaning.
- Add text and/or icons to communicate meaning, in addition to color.
- The most common example of this is to style links without underline or border.
- Browsers underline hypertext links by default. It is possible to remove the underline using Cascading Style Sheets (CSS), but this is a bad idea most of the time. Users are accustomed to seeing links underlined.

WebAIM: Links and Hypertext

Wikipedia I



The screenshot shows the English Wikipedia article on color blindness. The page title is "Color blindness". The content starts with a redirect notice: "'Colorblind' redirects here. For color blindness in other species, see [Color vision](#). For other uses, see [Colour blind \(disambiguation\)](#)". Below this, a note clarifies the difference from "Color blindness (racial classification)". The main text discusses "Color blindness (color vision deficiency)" as a decreased ability to see color or differences in color, mentioning its impact on tasks like fruit selection and traffic light reading. It also notes that color-blind people may have some educational difficulties. A sidebar on the right provides a brief summary of symptoms, duration, causes, diagnostic methods, and treatment.



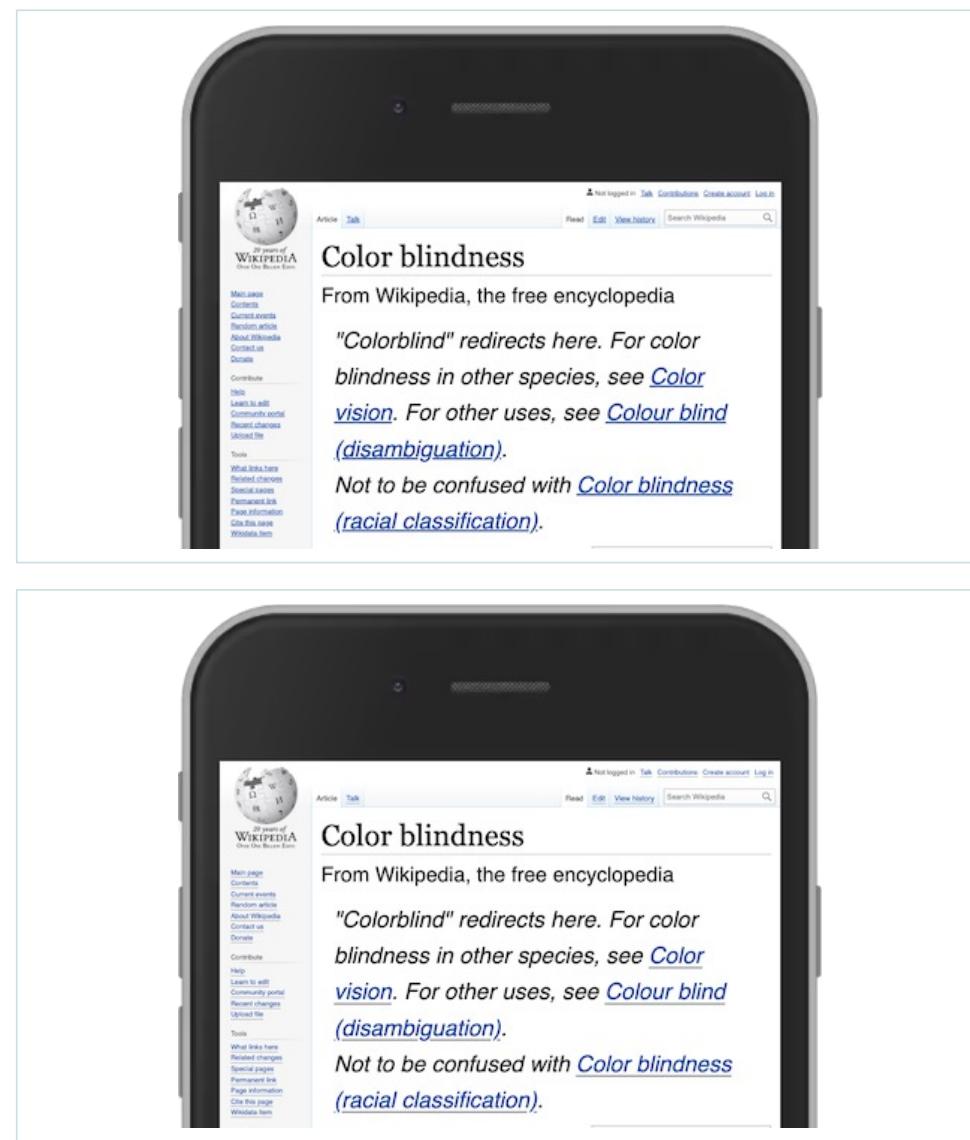
This screenshot shows the same Wikipedia article on color blindness, but with color restored. The text remains identical to the grayscale version. The key difference is the color-coded text links: the word "Color" in the redirect notice and the "Color blindness" section title are blue, while the "Color vision deficiency" section title is red. The sidebar also contains colored text links for "Symptoms", "Duration", "Causes", "Diagnosis", "Treatment", and "Frequency".

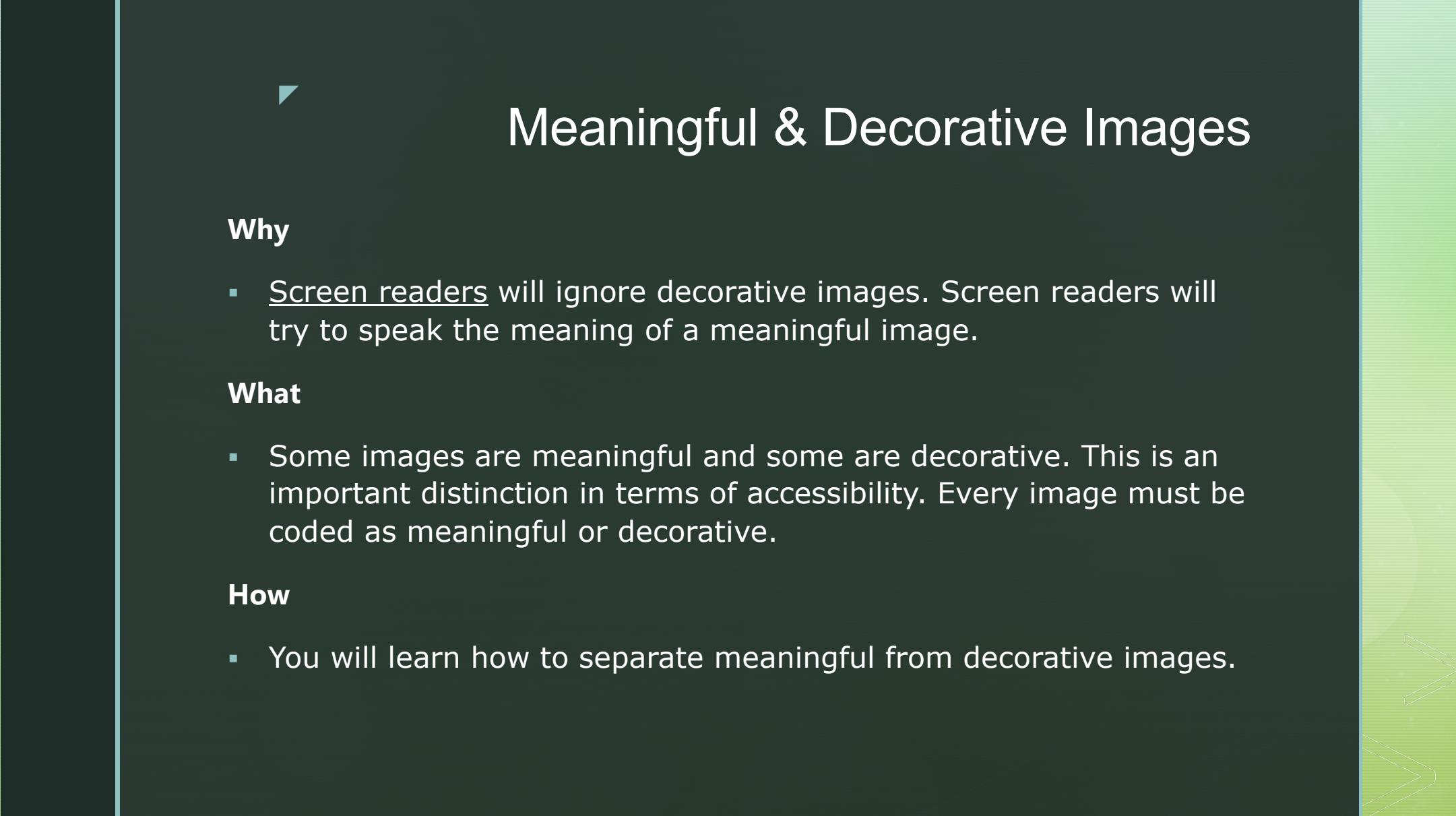
Wikipedia is one example where color alone is used for styling links.

In the grayscale version of the site, it is not possible to see what is plain text and what is a link.

Wikipedia II

- Add underline to links.
- Keep in mind that they can reduce readability.
- To improve readability, we can use CSS properties like
 - `text-decoration-color`.





Meaningful & Decorative Images

Why

- Screen readers will ignore decorative images. Screen readers will try to speak the meaning of a meaningful image.

What

- Some images are meaningful and some are decorative. This is an important distinction in terms of accessibility. Every image must be coded as meaningful or decorative.

How

- You will learn how to separate meaningful from decorative images.



Decorative images

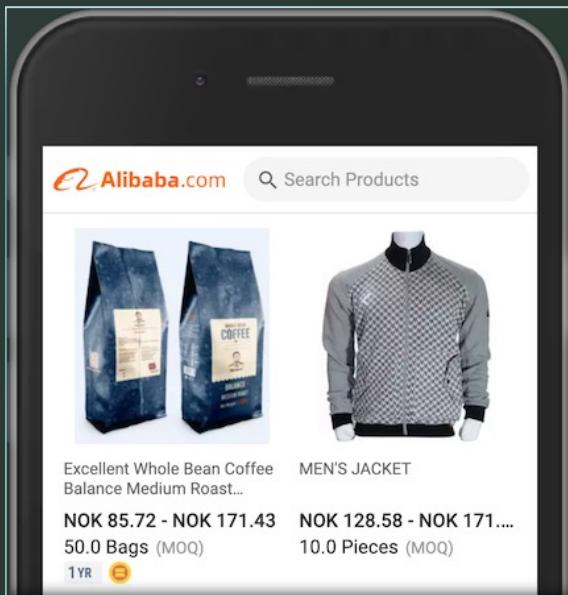
- If an image is not important for a user to understand the functionality or the content of a web page or app, it is considered decorative. Can you remove it with no impact? Then it is a decorative image.
- Empty alt attribute
- The basic way to set an image as decorative is to use an empty alt attribute.

```

```

- Assistive technologies, like a screen reader will then ignore the image. Without the empty alt attribute, a screen reader may read the file name. This will make no sense, and will confuse the user.

Descriptive alt attribute



- The alt attribute provides an alternative text for an image, if the user for some reason cannot view it. The reason can be a slow connection, an error with the image file, or if the user uses a screen reader.
- The value of the alt attribute should describe the image, or even better: the **intention** of the image. You will learn what to write in the page Descriptive texts for images.

```

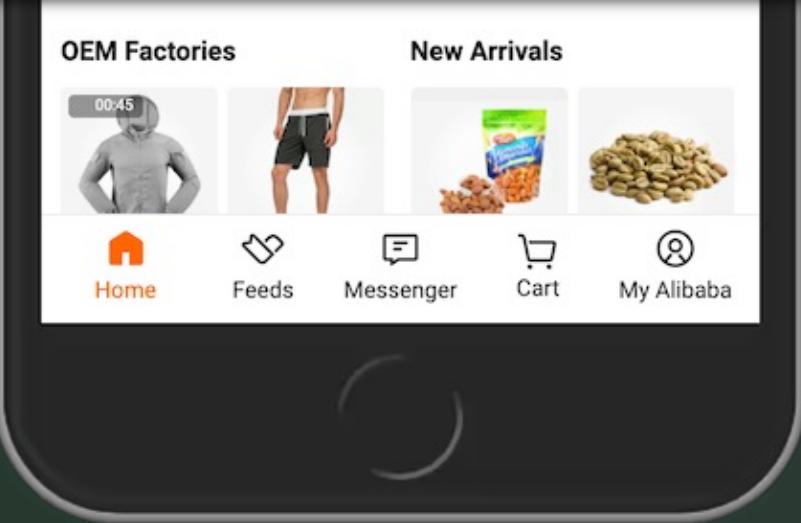
```

Background images

- Another method for decorative images is to add them using the CSS background-image property.
- This is common when we create hero images.



Font icons



- At the bottom of the mobile version of Alibaba, we have five links that are combinations of icons and text – *Home*, *Feeds*, *Messenger*, *Cart* and *My Alibaba*. Since the site is still readable if we remove the icons, they are decorative. The icons are created with font icons. No `` element and no background image. Add `role="img"` and `aria-hidden="true"`:
- **`<i class="navbarIcon" role="img" aria-hidden="true"></i>`**
- With this code, we add some semantics to the `<i>` with the image role. User agents now understand that this is an image. Screen readers also understand that they should ignore the image.

Accessibility Link States

Why

- Different link states help users interact with the links. A visited state can help a person with **short-term memory loss** to understand which content has been read. A hover state can help a person with **reduced muscle control** to know when to click. A focused link helps **keyboard users** to know which link they are about to activate.

What

- Links hardly needs an introduction. They are the heart of the web. A link has many states. Here are five of the most common states. In CSS terms, these are pseudo-classes:
 - Unvisited
 - Visited
 - Hover
 - Active
 - Focus

How

- To make sure that all link states are accessible, we must remember these three tips.

Accessibility Link States: three tips

1. Underline

- Links are underlined by default. Removing the underline from a link *in body text* is a bad idea most of the time. We learned this in the section about color alone. This is most important for unvisited and visited links.

2. Contrast and focus

- All states must have sufficient contrast, as we learned in color contrast. In addition, a focused link must have sufficient contrast to the unfocused state

3. Hover

- A clear hover state is helpful for everyone, especially people with motor impairments.

QUICK LINKS

[Home](#)

[What's new](#)

[Contact](#)

[Media Contacts](#)

[Working for the ICRC](#)

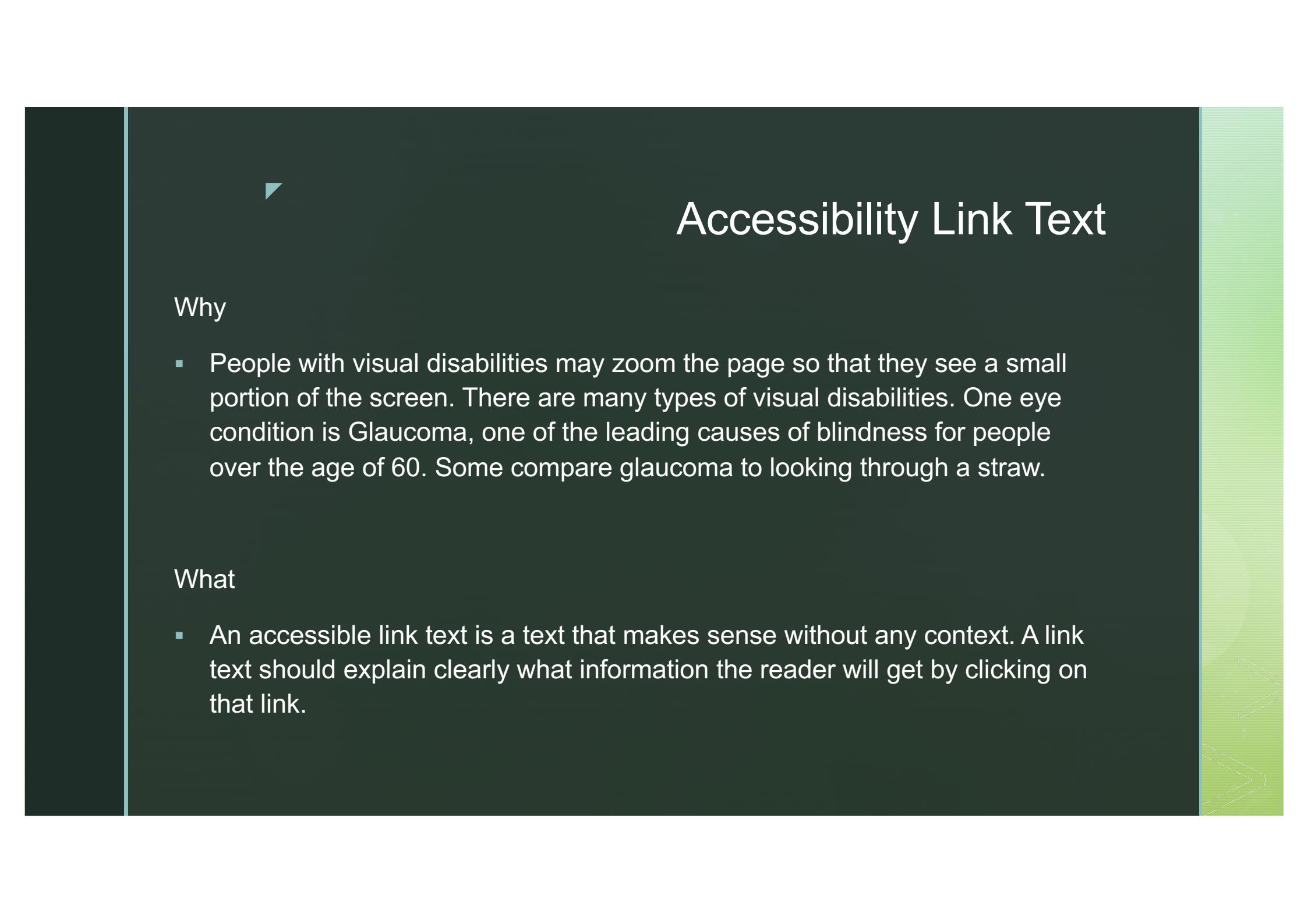
[Other languages](#)

[Doing business with the ICRC](#)

[Reporting Misconduct](#)

[Access to information and
accountability](#)

[!\[\]\(18e1df0c276e9f4cd314213298ee2803_img.jpg\) Extranet for donors](#)



Accessibility Link Text

Why

- People with visual disabilities may zoom the page so that they see a small portion of the screen. There are many types of visual disabilities. One eye condition is Glaucoma, one of the leading causes of blindness for people over the age of 60. Some compare glaucoma to looking through a straw.

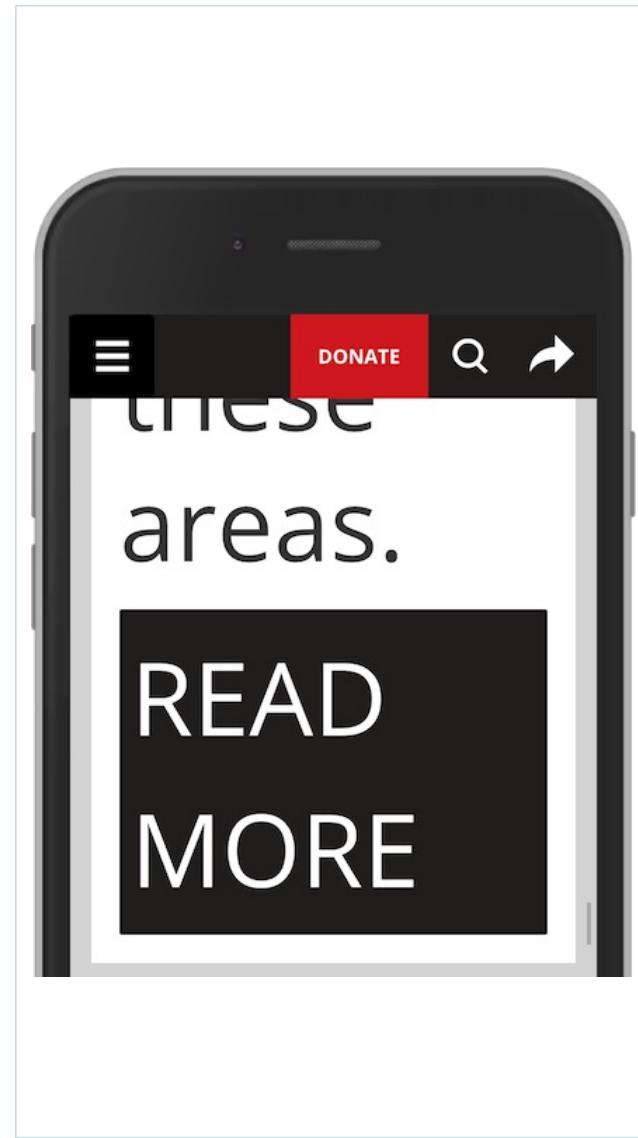
What

- An accessible link text is a text that makes sense without any context. A link text should explain clearly what information the reader will get by clicking on that link.

Some examples of good and bad link text.

Bad

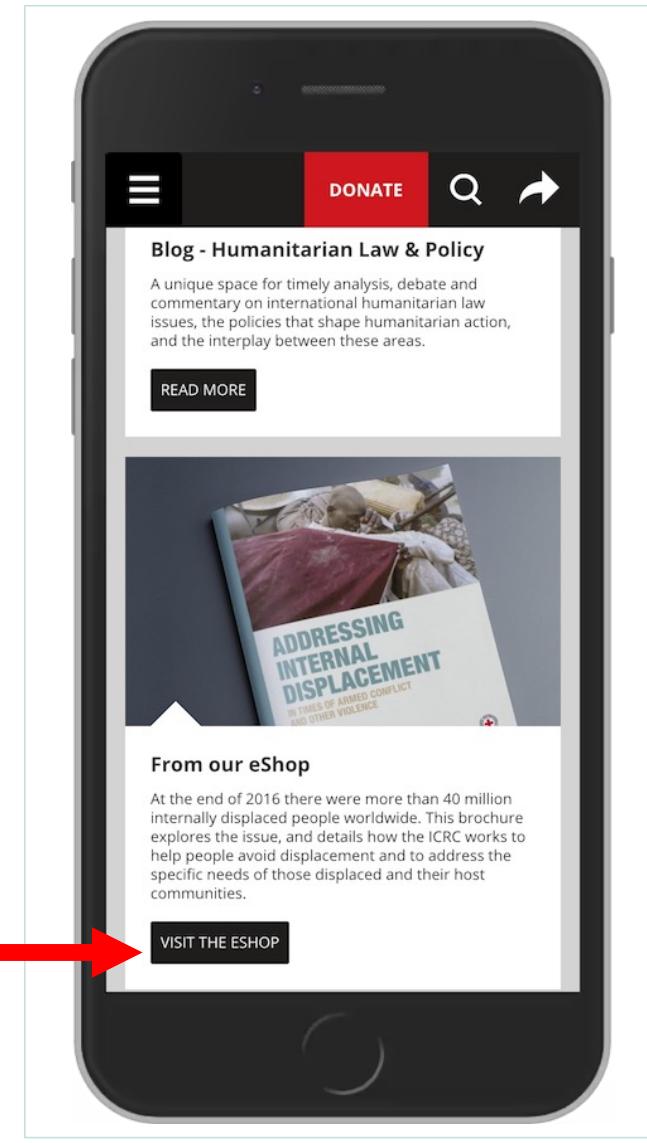
- Click here
- Read more
- Buy tickets to Mars here



Some examples of good and bad link text.

Good

- Find out more about the HTML language
- Read more about how to eat healthy
- Buy tickets to Mars here



Accessibility Headings Introduction

- A heading is text that describes the content that follows it.
- What you might not know, is that headings are important for accessible navigation:
 - Sighted users scan a web page to understand the structure of the page.
 - The same way, screen reader users use headings to navigate and scan the page.
- The headings must be clear, both visually and use clear wording. The heading structure of a page forms the outline of the page. You might think of this like the skeleton of the page.



<https://youtu.be/vAAzdi1xuUY>

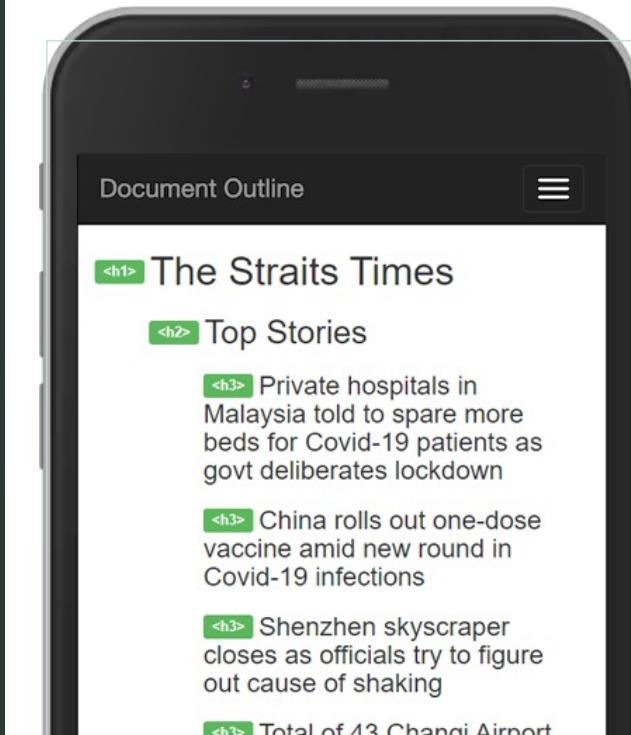
Accessibility Headings

Why

- People use the heading structure to scan the page and get an understanding of the main content. This is true for both sighted users and screen reader users.

What

- Headings are defined with the `<h1>` to `<h6>` tags. Users skim your pages by its headings.
- It is important to use headings to show the document structure and the relationships between different sections. `<h1>` headings should be used for main headings, followed by `<h2>` headings, then the less important `<h3>`, and so on.



1. Download the [browser extension Web Developer](#). It is available for Chrome, Firefox and Opera.
2. Open [The Straits Times](#).
3. Open Web Developer. Under the tab *Information*, click *View Document Outline*.
4. Scan through the document outline.

.sr-only

- Sometimes it makes sense to add content just for screen readers. This is such a case. A common practice is to use a CSS class .sr-only, where sr means screen reader:

```
<h2 class="sr-only">Briefings</h2>
```

and add this styling to put it off screen:

```
.sr-only {  
    position: absolute;  
    left: -10000px;  
    top: auto;  
    width: 1px;  
    height: 1px;  
    overflow: hidden;  
}
```

- CSS class .sr-only that is only accessible for screen readers:

Accessibility Keyboard and Assistive Technologies

People with disabilities use technology in a variety of ways. Here are some examples:

- People with hand tremors cannot grip a mouse. They use the keyboard to navigate.
- People with mobility disabilities use their voice to control the computer or mobile device.
- People with mobility disabilities use eye tracking to move the screen cursor.
- People with mobility disabilities use switch devices to operate the computer or mobile device.
- People who are blind use screen readers, braille displays or speech recognition software.
- People with low vision use screen magnification.



Accessibility Visual Focus

Why

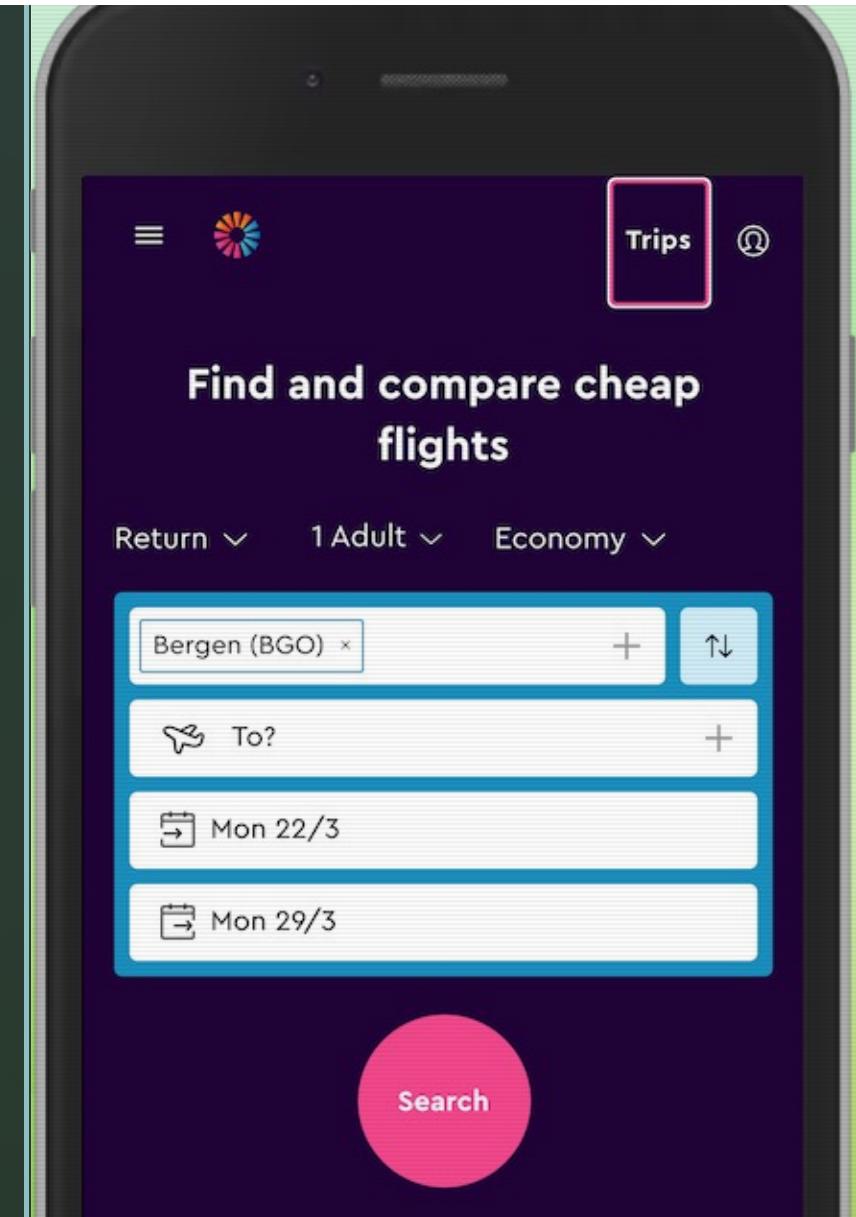
- Visual focus is crucial for all users that rely on keyboard and switch devices.

What

- You learned a bit about visual focus when we talked about link states. Let us dig deeper.
Visual focus is sometimes called keyboard focus or tab focus. It is a visual indicator on a interactive component that has keyboard focus.
The effect is often a border or outline.

How

- You will learn not to remove the focus, and two options for styling the focus.





Do not remove or hide the focus

- This is the most important takeaway from this module. Whatever you do, do not remove the focus. This CSS line is ruining the accessibility for a lot of people:

outline: 0;

- Another common method for hiding the focus that the parent element is too small to show it, in combination with:

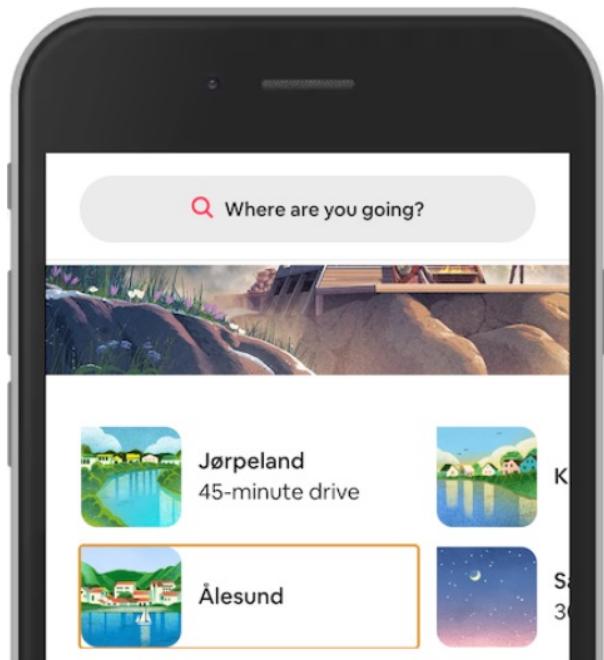
overflow: hidden;

- Most browsers use the outline property to show the visual focus of an interactive element. We have two options. Leave it or customize it. Removing it is not an option.

Option 1: Use the default

The easiest way to handle visual focus is to leave it for the browser to handle.

- Users that rely on the visual focus, recognize it easily.
- You don't have to code anything.
- Users don't get any surprises.

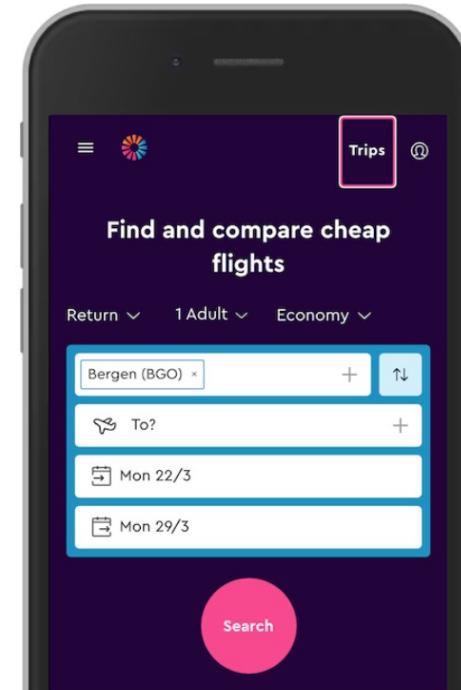


Option 2: Customize the visual focus

We also have some challenges with the default focus.

- The default styling may not align with the color palette of the site.
- The default styling is similar to the color palette of the site.
- The default styling is not visible enough in all browsers.

The travel site Momondo has a vivid color palette. They can pick a color from their palette



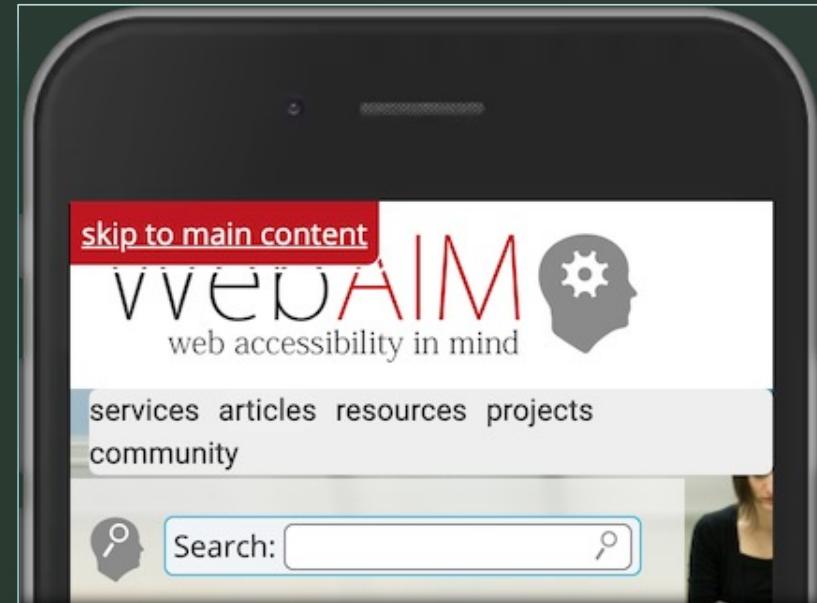
Accessibility Skip links

Why

- People using keyboard, screen readers, switch controls and other assistive technologies use skip links to reach main content or other important sections easier and faster.

What

- The most common skip link is the first interactive element on a page. It takes the user to the main content, past the global elements like the logo, search and navigation. It is almost always hidden until it receives focus.



Accessibility Skip links

```
<header>
<a href="#main" class="skip">Skip to main content</a>
...
</header>
...
<main id="main">
```

```
.skip {
  position: absolute;
  left: -1000px;
  top: auto;
  width: 1px;
  height: 1px;
  overflow: hidden;
}
```

```
.skip:focus {
  position: static;
  width: auto;
  height: auto;
}
```

Accessibility Screen Readers

- Screen readers are necessary for blind people, important for partially-sighted users and helpful for people with reading disorders.

Mobile

- For mobile devices, Apple has the biggest share of screen reader users. The screen reader **VoiceOver** is built in on iOS. The second most popular is **TalkBack** for Android, also built in on all Android devices.
- Making sure your site works well with these two is a good starting point. Before we proceed, read these articles:

Desktop and laptop

- For desktop and laptop computers, there are two screen readers you should be aware of – NVDA and JAWS. If you have to choose one for testing, go for NVDA. It is free and its popularity is growing. Both are only available for Windows.

Screen Readers Language

- For the screen reader to speak the correct language, it needs to know what language your content is. This is done with the lang attribute in the `<html>` element. The following example specifies English as the language:
- `<!DOCTYPE html>
<html lang="en">`

Language of parts

- Sometimes parts of your content is in another language. To make screen readers change their language in the middle of the page, we use the same lang attribute. Check the source code of the link to Bahasa Indonesia on the [english page about Dyslexia](#):

```
<a href="https://id.wikipedia.org/wiki/Disleksia" lang="id"  
    hreflang="id">Bahasa Indonesia</a>
```

- Now the screen reader understands that the words "Bahasa Indonesia" should be read in the language Bahasa Indonesia, not English. It also understand that the target page is in Bahasa Indonesian because of the hreflang attribute.



Accessibility Forms - labels

- The most important element to make sure your form is accessible is a label: [learn about labels](#).

Why

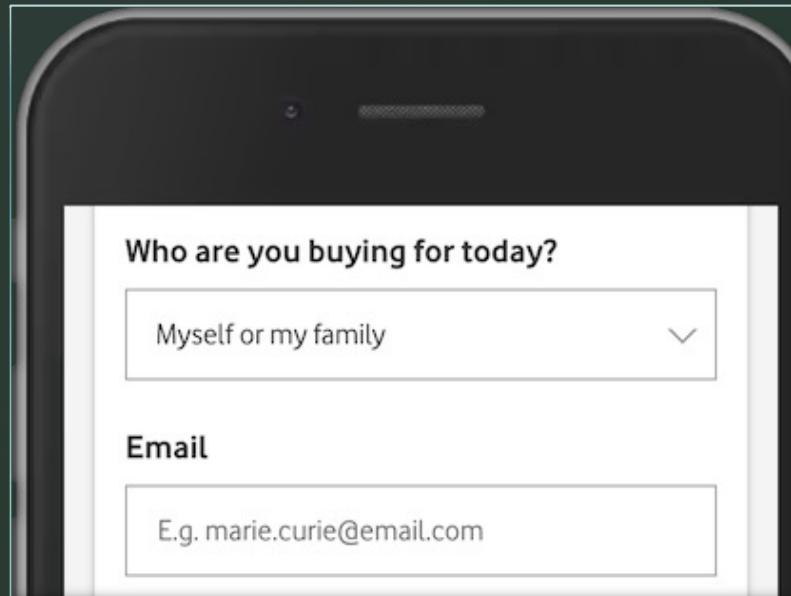
- Labels are critical for blind users, user with low vision, users with mobility disabilities and users with memory loss. Missing labels will make a form inaccessible for many users.

What

- Visual labels are text near a form control that describe what information belongs in a given form field or a group of fields. Each label must be programmatically associated with the form control or group of controls. Labels are not necessarily the <label> element.

Accessibility Forms - labels

```
<label for="customerType">Who are you buying  
for today?</label>  
<select name="customerType"  
id="customerType">
```



The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

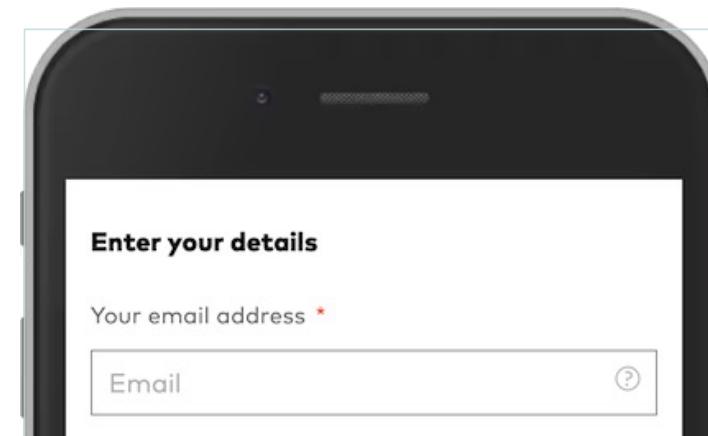
The `<label>` element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the `<label>` element, it toggles the radio button or checkbox.

The `for` attribute of the `<label>` tag should be equal to the `id` attribute of the `<input>` element **to bind them together**.

Accessibility Forms - labels

Required fields

- Form labels often contain a "*" or the word "required" to indicate that the field is required. Both of these methods are fine.
- However, it is recommended to add the required and aria-required="true" to the form control if you use an asterisk (*):

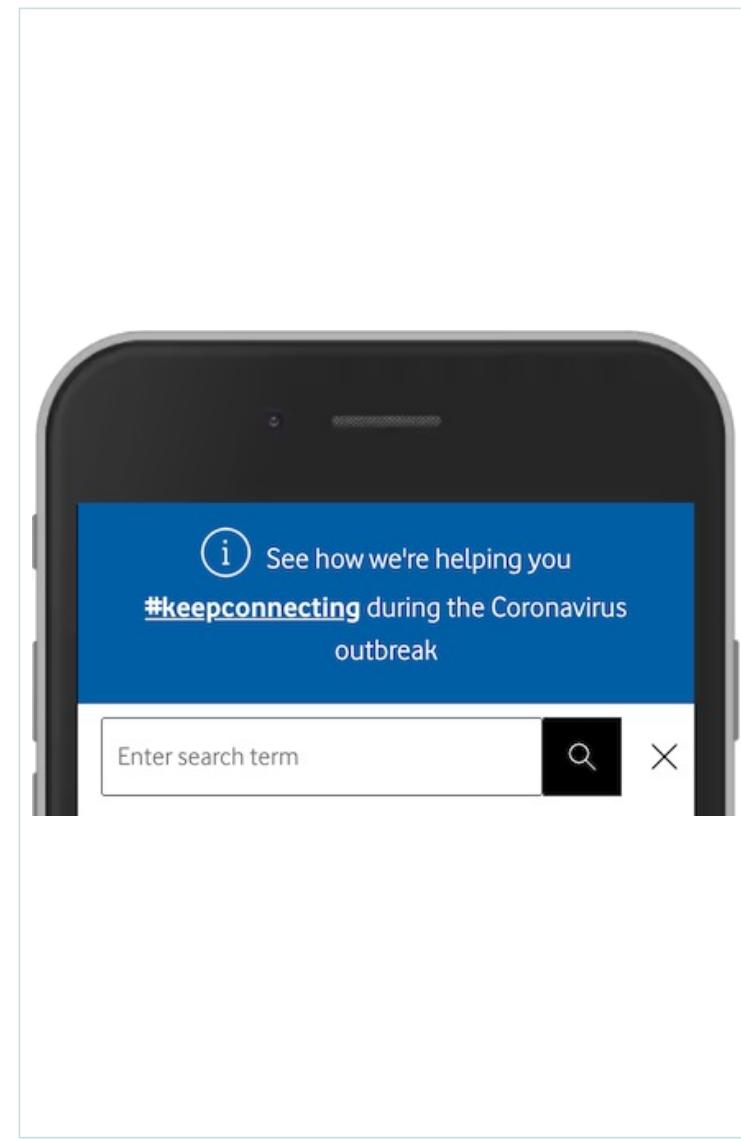


```
<label for="email">Your email address  
<span  
class="mandatory">*</span></label>
```

```
<input id="email" name="email" required  
aria-required="true" placeholder="Email"  
required="">
```

The aria-label

- This search field has a placeholder, but no label. A placeholder is not a valid accessible name. You can not rely on it as a substitute. An easy solution here is to add `aria-label="Enter search term"`:
- `<input placeholder="Enter search term"`
- `aria-label="Enter search term">`



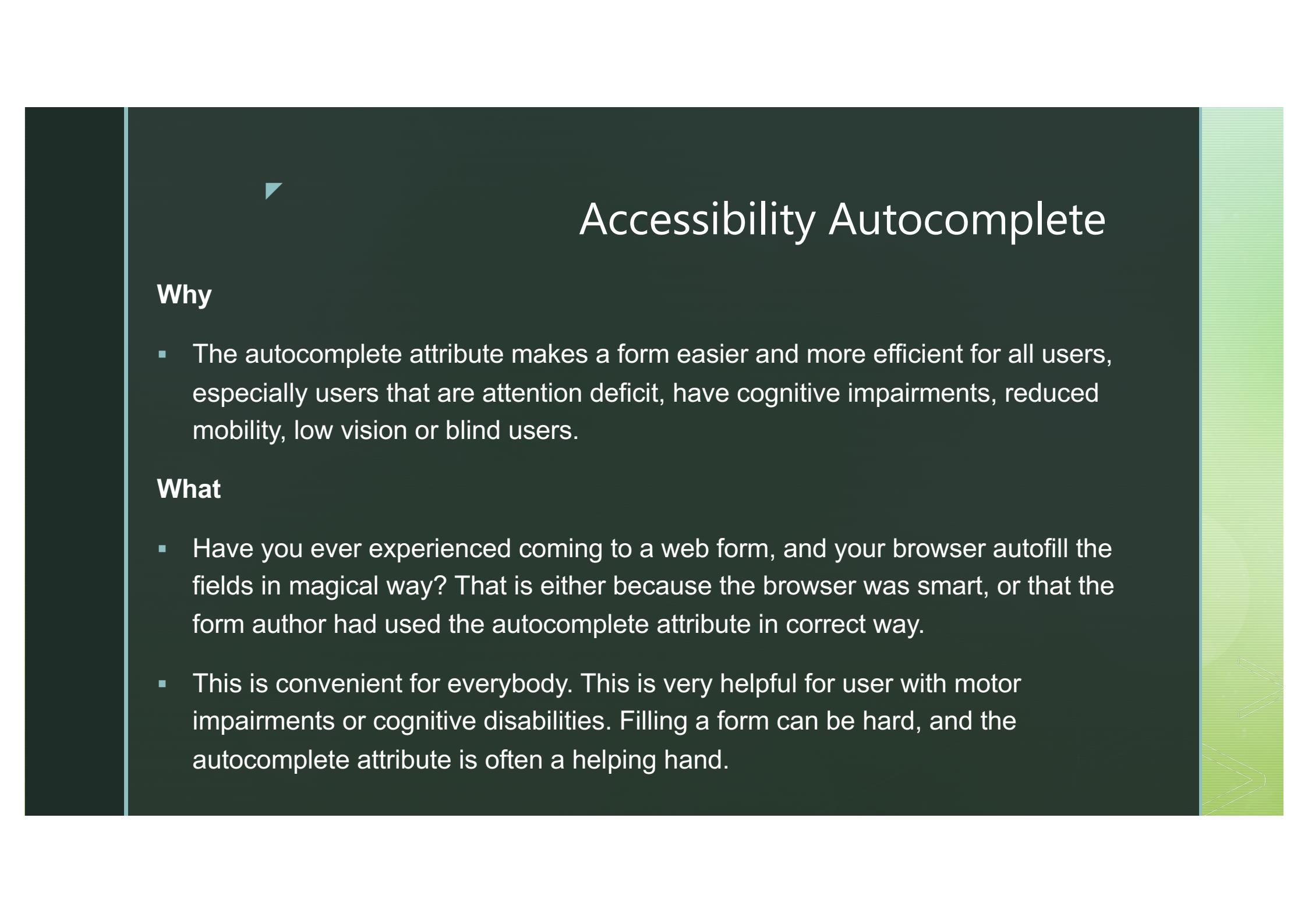
Legend

```
<fieldset>
  <legend>Your date of birth</legend>
  <label for="dobDay">Day</label>
  <select id="dobDay">...</select>
  <label for="dobMonth">Month</label>
  <select id="dobMonth">...</select>
  <label for="dobYear">Year</label>
  <input id="dobYear" type="text" placeholder="YYYY">
</fieldset>
```

Your date of birth *

Day Month Year

DD MMM YYYY



Accessibility Autocomplete

Why

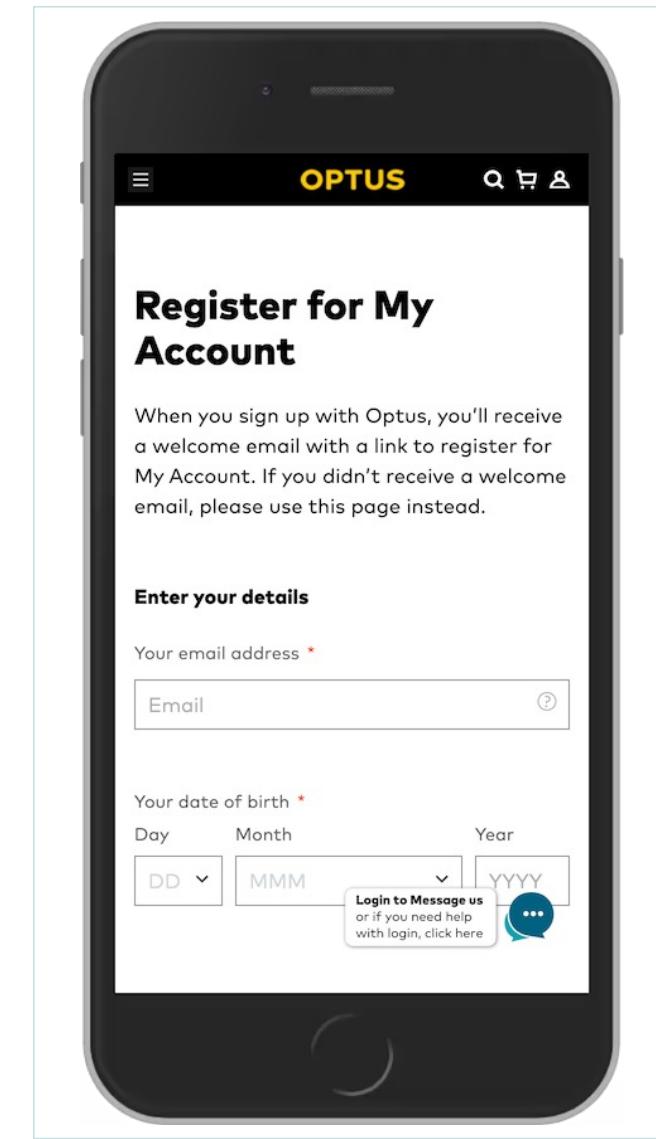
- The autocomplete attribute makes a form easier and more efficient for all users, especially users that are attention deficit, have cognitive impairments, reduced mobility, low vision or blind users.

What

- Have you ever experienced coming to a web form, and your browser autofill the fields in magical way? That is either because the browser was smart, or that the form author had used the autocomplete attribute in correct way.
- This is convenient for everybody. This is very helpful for user with motor impairments or cognitive disabilities. Filling a form can be hard, and the autocomplete attribute is often a helping hand.

Accessibility Autocomplete

- <input id="email" **autocomplete="email"** name="email" aria-required="true" placeholder="Email" required>
- <select id="dobDay" **autocomplete="bday-day"** aria-required="true" required>
- <select id="dobMonth" **autocomplete="bday-month"** aria-required="true" required>
- <input id="dobYear" **autocomplete="bday-year"** placeholder="YYYY" aria-required="true" required>



Accessibility Errors

Why ▾

- Everyone make mistakes. When we do, we need to understand why we have failed, to be able to correct ourselves. An accessible form needs error messages that is perceivable and understandable for people who is color blind, who is blind or low vision, and people with limited cognitive abilities.

What

An accessible error message is

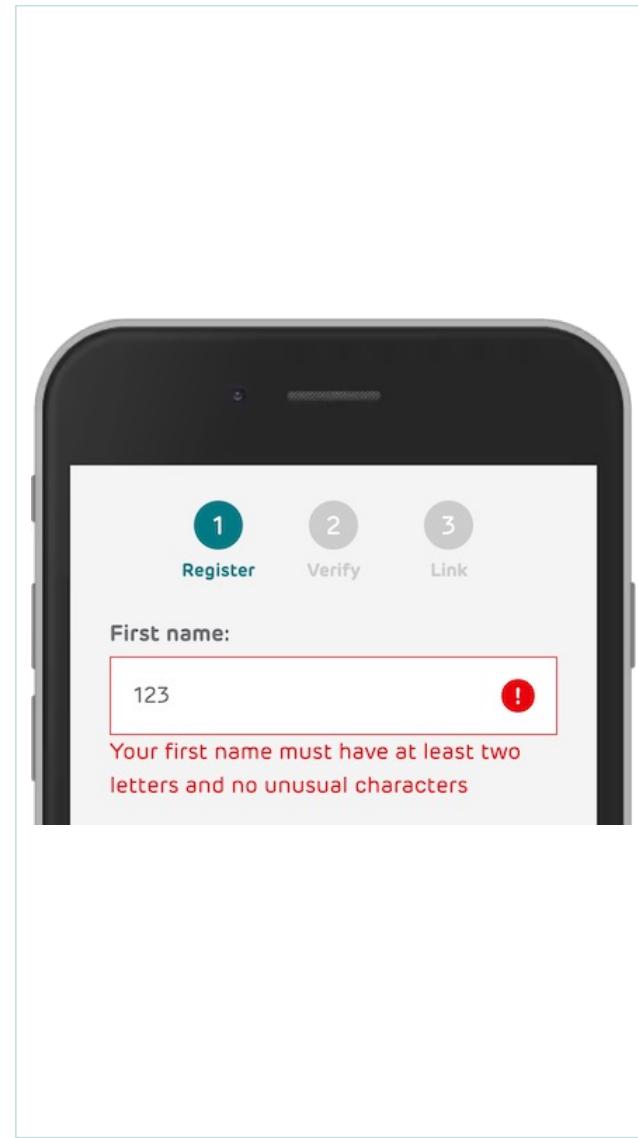
- written *in text*. Color and icon can be used, but not alone.
- close to the element that has failed.*
- informative, helping the user.*
- associated to the failed element in the code.*

In addition, it is helpful to *move the focus* to the form control that has failed.

Accessibility Errors

```
<input name="firstName" id="firstNameInput"
type="text" pattern="[^.]*?>
aria-describedby="firstName-length-error"
aria-invalid="true">

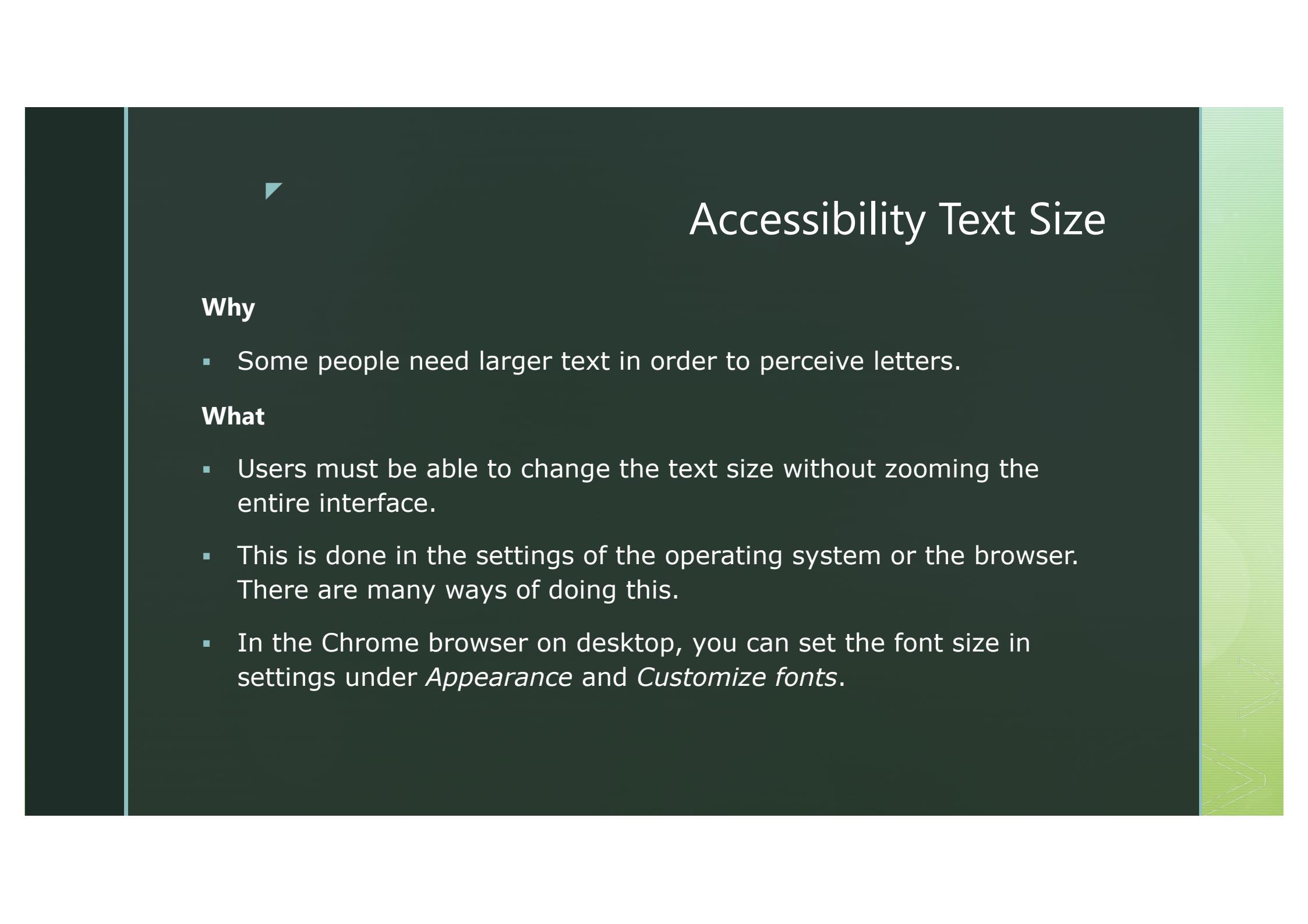
<p id="firstName-length-error" role="alert">
Your first name must have at least two
letters and no unusual characters
</p>
```





Accessibility Zoom Introduction

- Users with low vision zoom in on content to be able to perceive it. Users with good vision zoom in on content to increase the readability of the content. There are three main ways of zooming:
 1. Screen magnifier
 2. Text zoom
 3. Page zoom



Accessibility Text Size

Why

- Some people need larger text in order to perceive letters.

What

- Users must be able to change the text size without zooming the entire interface.
- This is done in the settings of the operating system or the browser. There are many ways of doing this.
- In the Chrome browser on desktop, you can set the font size in settings under *Appearance* and *Customize fonts*.



MOST POPULAR

EXCITING OFFERS FOR YOU

NEWEST



LG
Dishwasher
with True

★★★★★ 3.4 (7)
Rs.57990 Rs.69990
17% Off



LG Tone
Free
Wireless

★★★★★ 4.3 (10)
Rs.19990 Rs.29990
33% Off



10.5Kg/7.0
Kg AI
Direct

★★★★★ 4.0 (4)
Rs.74490 Rs.84990
12% Off



LG Wing
5G*

★★★★★ 4.7 (29)
Rs.59990



Accessibility Text Size

Use relative measure

```
.model-name {  
    font-size: 1.125rem;  
    line-height: 1.2;  
    ...  
}
```

Accessibility page zoom

 https://www.w3schools.com/accessibility/accessibility_page_zoom.php

- People with low vision need to zoom the content in order to use the page.
- Avoid horizontal scrolling.
- All content is available.
- All functionality is available.
- Avoid text in images.
- Provide space for key content.
- Available means that nothing is clipped, truncated or obscured.
- Page zoom often triggers mobile view on responsive sites, which is good.



Links

- <https://material.angular.io/components/button/overview>
- [Get started on Android with TalkBack](#)
- [Turn on and practice VoiceOver on iPhone](#)
- <https://chrome.google.com/webstore/category/ext/22-accessibility>
- <https://chrome.google.com/webstore/detail/web-developer/bfbameneiokkgbdmiekhjnmfkcnldhhm?hl=it>
- Chrome vox:
<https://support.google.com/chromebook/answer/9032490?hl=it>