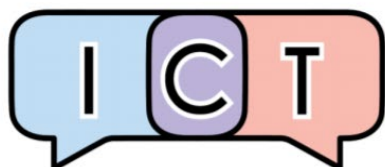




Mobile Web Accessibility Testing Methodology

Step 1: Identify what needs to be tested

Step 2: Conduct specific mobile tests



**Accessibility Testing
Symposium**

Mobile Web Testing Methodology Overview

Please note that this methodology does not include those errors already included in WCAG2. In order to ensure your mobile site is fully accessible, you need to meet WCAG2 and this mobile testing methodology.

Step 1: Identify what needs to be tested

Identify devices

Recommended devices:

- iPhone, Safari
- iPad, Safari
- Android phone, Chrome

Identify the site type and variations of the page

There are three types of sites on the web and each type has different mobile testing requirements:

- **Desktop web sites:** that have only one display, whether viewed on desktop or mobile or tablet device;
- **m.dot sites:** that have a particular display for mobile and tablet sites. The m.dot site must also be tested against the entirety of WCAG2, **in addition** to the standard www version of the site.
- **Responsive web sites:** that change depending on the screen size or other feature as determined by the developer;

Responsive sites contain multiple variations or versions of a page. It is important that each variation of the page is tested and that all functionality is available on all variations of the page. People with disabilities may be restricted to one variation of the page only.

Step 2: Conduct specific mobile tests

In addition to the familiar criteria that are tested on desktop – such as alternatives to images, and coding headings and tables – there are five more types of mobile testing errors:

- a. **Critical issues:** hover trap, touch trap, screen reader swipe trap, on-screen keyboard trap, zoom trap, etc.
- b. **Mobile-specific issues:** orientation, motion actuation, geolocation, scroll-bars, blocking pinch zoom, touch targets, inactive space, navigational aids, etc.

- c. **Mobile assistive technology and feature support:** screen reader support, keyboard behavior, magnification / zoom behavior, inverse colors / grayscale behaviour, etc.
- d. **Mobile and desktop relationship issues:** consistency, restriction of content, choice of content, etc.

Table of Contents

Mobile Web Testing Methodology Overview	2
Step 1: Identify what needs to be tested	2
Step 2: Conduct specific mobile tests.....	2
Table of Contents	4
List of Figures	5
Introduction	7
A note on mobile apps	7
About mobile testing	8
Types of mobile sites	8
Variations of the page (responsive sites only)	20
Testing methods	32
Capturing errors	45
Acknowledgements	48
Relationship to existing Accessibility testing standards	48
ICT Accessibility Testing Symposium Mobile Sub-Committee.....	48
Contacts.....	48
Gian Wild	48
Peter McNally	48

List of Figures

Figure 1: Australian National Botanic Gardens desktop web site at desktop size	9
Figure 2: Australian National Botanic Gardens desktop web site at mobile window size.....	9
Figure 3: www.sephora.com – their desktop site in a desktop-sized window	10
Figure 4: m.sephora.com – their “m.” site in a desktop-sized window	10
Figure 5: www.sephora.com Their desktop site in a mobile-sized window	11
Figure 6: m.sephora.com Their “m.” site in a mobile-sized window.....	11
Figure 7: AccessibilityOz site at desktop width	11
Figure 8: AccessibilityOz site at a mobile width	12
Figure 9: www.stanford.edu at desktop width.....	13
Figure 10: www.stanford.edu at mobile width.....	13
Figure 11: m.stanford.edu at desktop width	14
Figure 12: m.stanford.edu at mobile width	14
Figure 13: Instagram on desktop at desktop width	15
Figure 14: Instagram on desktop at a mobile width	16
Figure 15: Instagram mobile site on an iPhone	17
Figure 16: Standard Amazon site	18
Figure 17: Amazon site aimed at people with disabilities	18
Figure 18: YouTube at 100% zoom	21
Figure 19: YouTube at 200% zoom	21
Figure 20: AccessibilityOz site at desktop width	22
Figure 21: AccessibilityOz site at a mobile width	22
Figure 22: ICT Accessibility Testing Symposium site at desktop width	22
Figure 23: ICT Accessibility Testing Symposium site at mobile width	22
Figure 24: Minimize, Restore down and Close buttons	23
Figure 25: Select the side of the screen and drag to the left.....	24
Figure 26: Variation 1 of the AccessibilityOz web site	25

Figure 27: Variation 2 of the AccessibilityOz web site (the navigation has been replaced by an “Open Menu” button)	25
Figure 28: Variation 3 of the AccessibilityOz web site (Services are now two per row and the controls for the slideshow have changed)	26
Figure 29: AccessibilityOz site in portrait orientation on iPhone	28
Figure 30: AccessibilityOz site in landscape orientation on iPhone.....	28
Figure 31: AccessibilityOz site in portrait orientation on iPad	29
Figure 32: AccessibilityOz site in landscape orientation on iPad.....	30
Figure 33: AccessibilityOz site in portrait orientation on Google Pixel 3+	30
Figure 34: AccessibilityOz site in landscape orientation on Google Pixel 3+ ..	31
Figure 35: Turning on VoiceOver on iOS.....	35
Figure 36: Setting Accessibility shortcuts on iOS	35
Figure 37: Activating zoom on iOS	37
Figure 38: Turning on the Reader view on iOS.....	38
Figure 39: The same article displayed in Reader view	38
Figure 40: Vertical ellipsis in Chrome on Google Pixel 3+	40
Figure 41: Drop-down menu once the vertical ellipsis has been selected.....	40
Figure 42: Article on San Francisco Chronicle with simplified view option visible at the bottom of the screen.....	40
Figure 43: The simplified view of the article	40
Figure 44: Grayscale view on iOS	42
Figure 45: Chrome Developer Tools	43
Figure 46: Chrome Device Toolbar	43
Figure 47: Chrome Developer Tools visible in the browser	44
Figure 48: Close-up view of the Toggle Devices option.....	44
Figure 49: Chrome Device Toolbar	44
Figure 50: Device Size Options	44

Introduction

The ICT Accessibility Testing Symposium has developed a methodology for evaluating the accessibility of mobile web sites. This document is an amalgamation of accepted mobile accessibility testing standards from around the world, including additional developments from the ICT Accessibility Testing Symposium's Mobile Sub-Committee (for more information, see Acknowledgements on page 48).

WCAG2 success criteria are applicable to mobile, however, not all aspects of mobile accessibility are specifically covered by WCAG2. It is the opinion of this committee that merely conforming WCAG 2.1 does not provide for a fully accessible experience for users with disabilities.

Please note that this methodology does not include those issues already included in WCAG2, however does include issues identified in WCAG2.1. This guide was written with the intent to clarify the unique needs of users with disabilities who use mobile websites and to raise the bar for the web development community. This is a work-in-progress, and, as such, we do not make a claim that conforming to these requirements will ensure that your mobile site is fully accessible to all users.

A note on mobile apps

There is a great difference between mobile sites and mobile apps – native apps utilize a completely different codebase. Therefore, the ICT Accessibility Testing Symposium has created a separate document on Native App testing methodology.

About mobile testing

Please note that this methodology does not include those errors already included in WCAG2. In order to ensure your mobile site is fully accessible, you need to meet WCAG2 and this mobile testing methodology.

Types of mobile sites

There are three types of sites on the web:

- **Desktop web sites:** that have only one display, whether viewed on desktop or mobile or tablet device;
- **m.dot sites:** that have a particular display for mobile and tablet sites. The m.dot site must also be tested against the entirety of WCAG2, **in addition** to the standard www version of the site.
- **Responsive web sites:** that change depending on the screen size or other feature as determined by the developer;

Most sites are responsive sites. All these site types need to be tested on mobile devices, however testing is different depending on the site type:

- Mobile and Desktop relationship errors **do not** need to be tested on **Desktop** web sites.
- **Some** Mobile and Desktop relationship errors do not need to be tested on **Responsive** web sites.
- In addition to this mobile testing methodology, **all WCAG2 errors** need to also be tested on **m.dot** sites.

Desktop web site

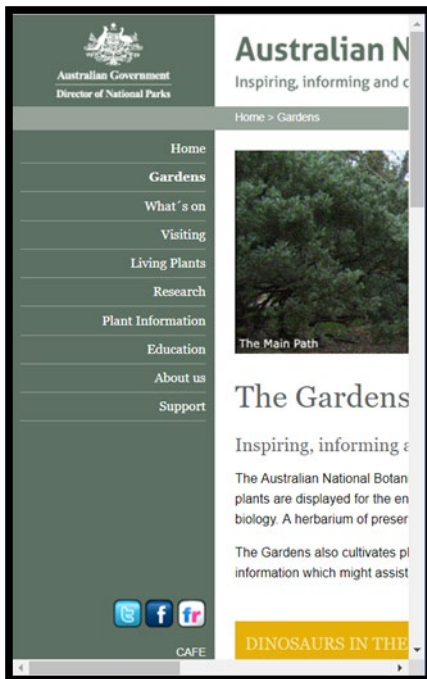
The site is a desktop site if the content does not change as you drag the browser window narrower. In most cases you will see a horizontal scrollbar at the bottom of the page.

Example of a desktop web site

Figure 1: Australian National Botanic Gardens desktop web site at desktop size



Figure 2: Australian National Botanic Gardens desktop web site at mobile window size



m.dot site

In most cases you will need to ask the client if they have an m-dot site. Historically, most mobile sites are the same as the desktop URL but preceded with "m." instead of "www.". The key differentiator with m-dot sites is that they have entirely different code than the desktop sites, so you will need to test the entirety of that site with a blank

slate. Therefore, you can type in the URL preceded by "m." to determine if the site has a mobile version. However, please make sure that the site does not automatically redirect to the "www." site – this can be checked by looking at the URL address bar once the site has loaded. Another way to tell is to open the "m." site in a desktop-sized window and it will still look like a mobile site. You can then check how the site changes when you change the size of the browser window.

In some cases, a site will only display the mobile site if opened on a mobile device. To determine if this is the case, open the site on the desktop and reduce the window size so it is the same width as a mobile browser. Open the same site on your mobile phone and determine if the site is different. If the site is different then there is a separate mobile site.

Comparison of a desktop and m dot site

Sephora

Figure 3: www.sephora.com – their desktop site in a desktop-sized window

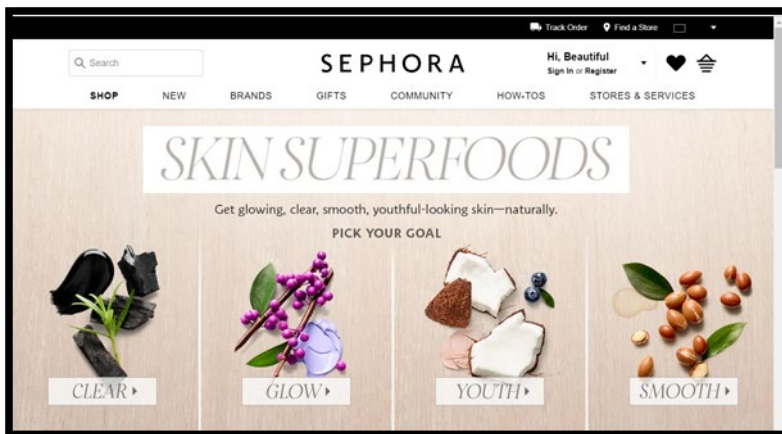


Figure 4: m.sephora.com – their "m." site in a desktop-sized window

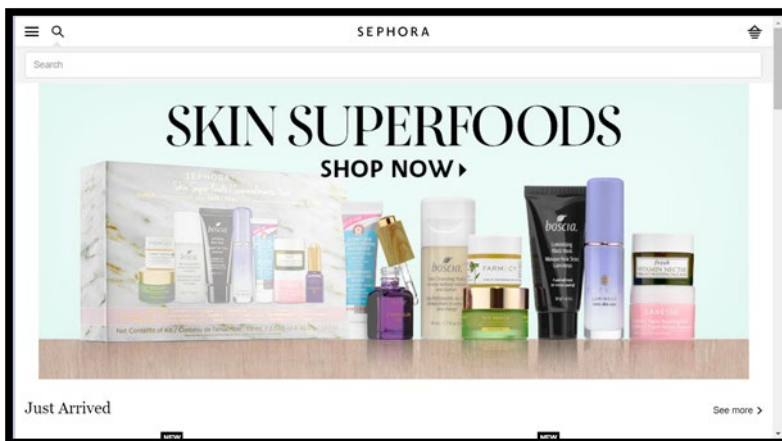


Figure 5: **www.sephora.com**
Their desktop site in a mobile-sized window

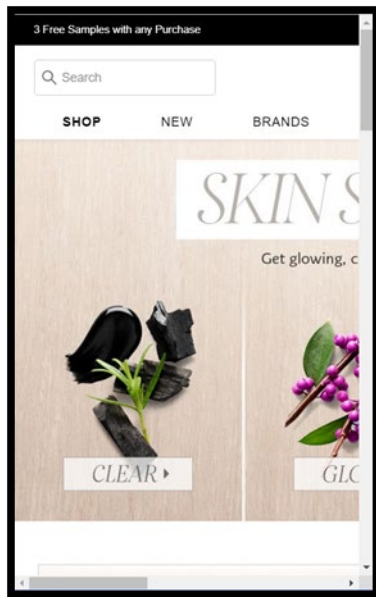
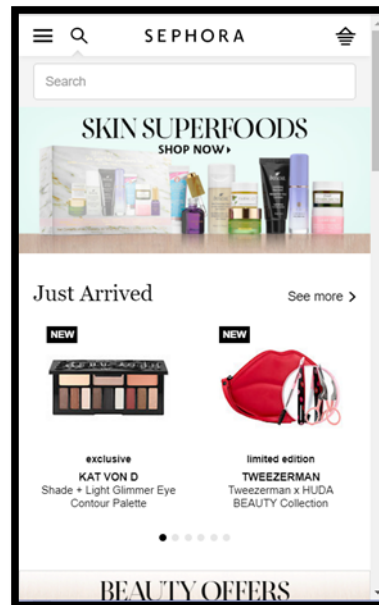


Figure 6: **m.sephora.com**
Their "m." site in a mobile-sized window



Responsive web site

Please note that it is most likely that a site is responsive. If a site is responsive, it is definitely not a desktop site. However, it is possible (but very unlikely) that there is also a mobile web site.

The site is responsive if the layout changes as you change the browser window size. To test this, open the web site in a browser, ensure the window is not maximized, and select the right-hand edge and drag it to the center of the screen. If elements in the page move around, then the site is responsive. Another way to tell is if the navigation disappears and is replaced with a hamburger menu.

Example of a responsive web site

Figure 7: **AccessibilityOz site at desktop width**

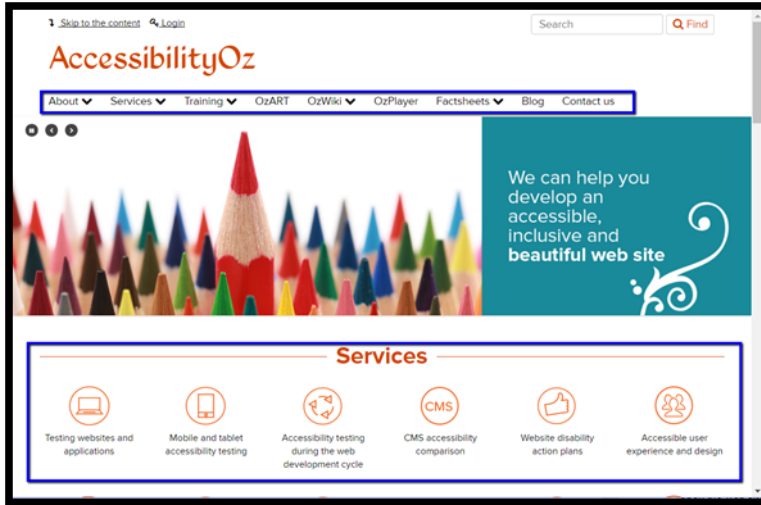
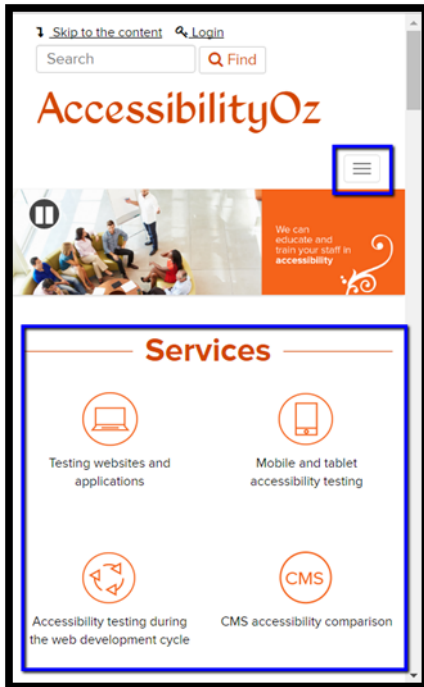


Figure 8: **AccessibilityOz** site at a mobile width



Further examples of mobile sites

Stanford - both responsive and an m.dot site

Stanford University has both a responsive site and an m.dot site. In this case it appears that the m.dot Stanford site serves a different purpose from the main Stanford web site.

Figure 9: www.stanford.edu at desktop width



Figure 10: www.stanford.edu at mobile width



Figure 11: m.stanford.edu at desktop width

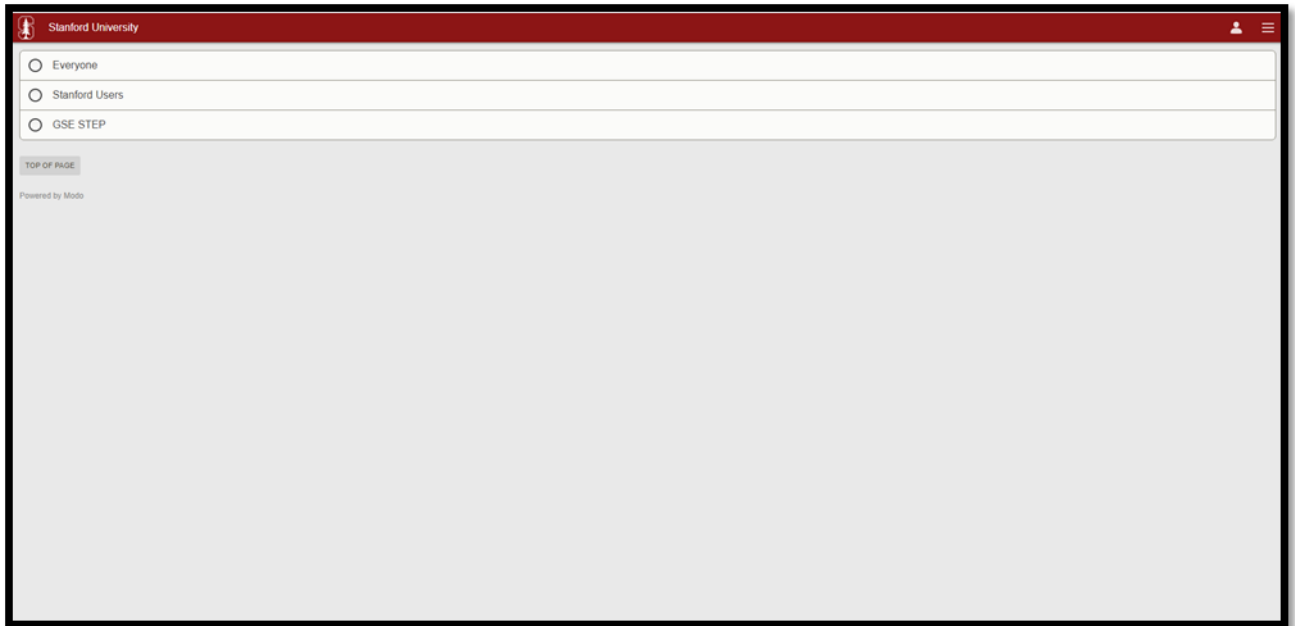
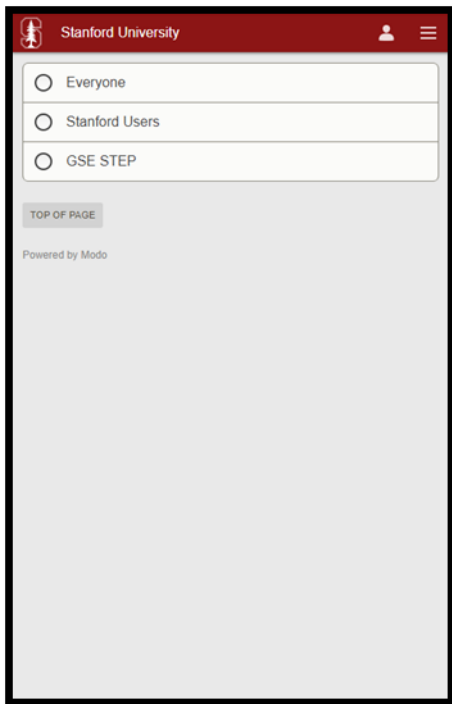


Figure 12: m.stanford.edu at mobile width



Instagram

Instagram is an example of a site that has limited functionality dependent on the operating system. When viewed on a desktop computer you are unable to upload a picture.

Figure 13: **Instagram on desktop at desktop width**

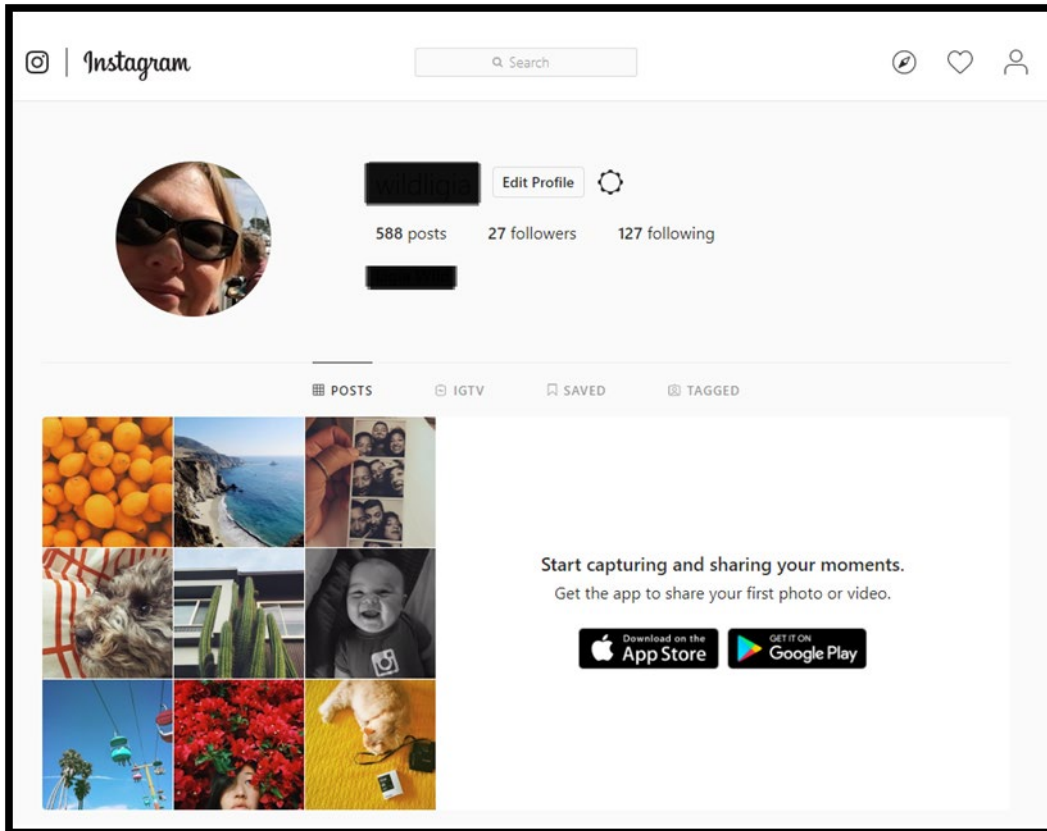


Figure 14: Instagram on desktop at a mobile width

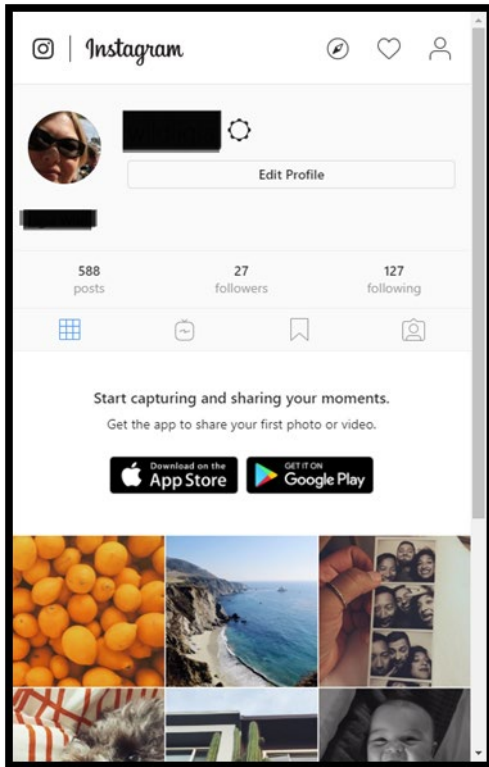


Figure 15: Instagram mobile site on an iPhone



Amazon

Amazon maintains a separate, simplified view for its customers with disabilities. This is a controversial practice; some people with disabilities prefer the ease of navigation while others believe that having a separate site is discriminatory. The debate is outside the scope of this document, and we include the example as a means of showcasing the differences between a responsive and an m-dot site.

This is what is read by the screen reader when accessing www.amazon.com:

Link- Welcome to Amazon.com. If you prefer a simplified shopping experience, try the mobile web version of Amazon at www.amazon.com/access. The mobile web version is similar to the mobile app. Stay on Amazon.com for access to all the features of the main Amazon website.

Figure 16: Standard Amazon site

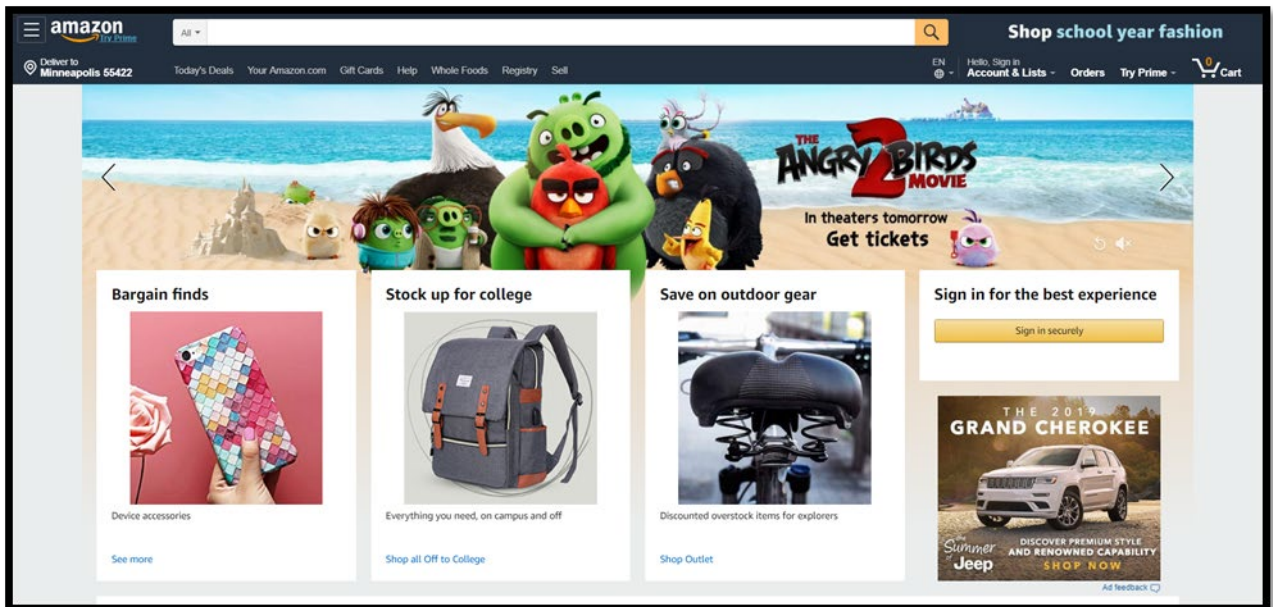
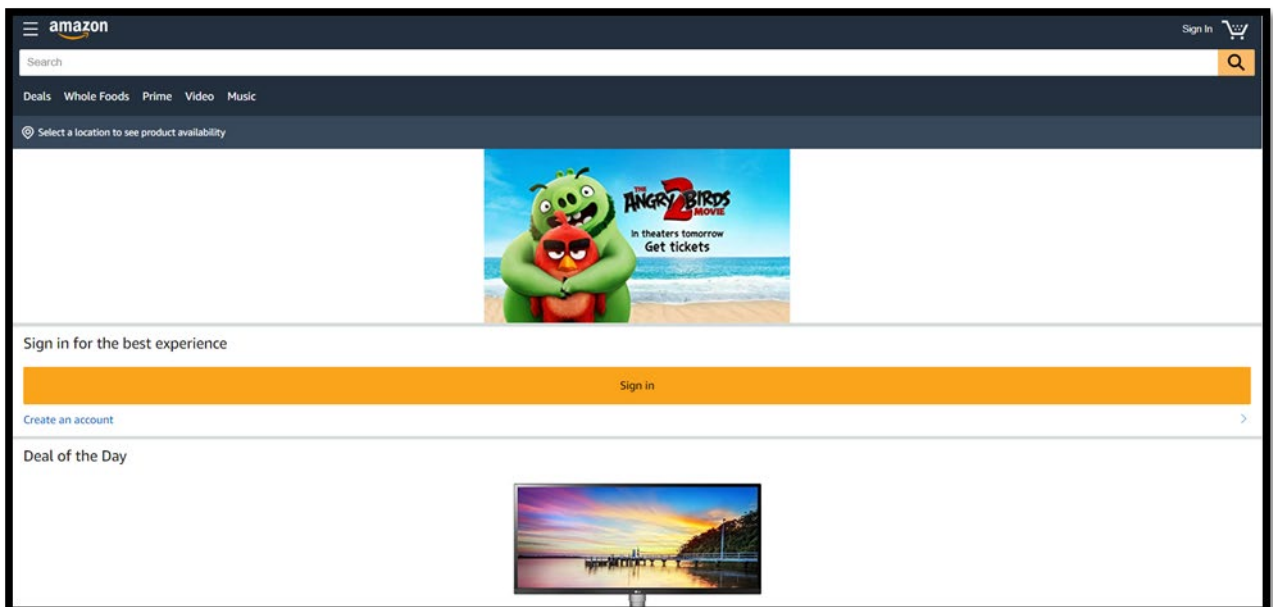


Figure 17: Amazon site aimed at people with disabilities



How to identify the type of mobile site

It is incredibly unlikely that a desktop site has a responsive site, however it is possible for the following site combinations:

- A desktop site and an m.dot site
- A responsive site and an m.dot site

There are eight steps to identifying the type of mobile site.

1. Open a browser. Type in the web site URL. Ensure the screen is maximized. Take a screenshot. This should be called "www.desktop".
2. Restore down the window to a mobile-sized window (for detailed instructions see How to test variations of a page on page 31). Take a screenshot. This should be called "www.desktop.mobile"
3. Maximize the window.
4. Replace the "www" with "m" in the URL address bar. If there is no "www" add "m." to the beginning of the site URL. If this is not a 404 error, take a screenshot. This should be called "m.desktop"
5. Restore down the window to a mobile-sized window (for detailed instructions see How to test variations of a page on page 31). Take a screenshot. This should be called "m.desktop.mobile"
6. Open a browser on your phone. Type in the URL and take a screenshot (for detailed instructions on how to capture screenshots on various devices see page 46). This should be called "www.mobile"
7. Replace the "www" with "m" in the URL address bar. If there is no "www" add "m." to the beginning of the site URL. If there is not a 404 error, take a screenshot. This should be called "m.mobile"
8. Compare the screenshots to decide the site types.
 - a. If www.desktop and www.desktop.mobile are the same site, just with the content cut off, then this is a desktop site.
 - b. If www.desktop and m.desktop are different sites then this is an m.dot site.
 - c. If www.desktop, m.desktop and www.desktop.mobile are the same site, but m.mobile is different then this is an m.dot site that only serves an m.dot site to a mobile or tablet device.
 - d. If www.desktop.mobile is the same site as www.mobile but they are both different to www.desktop, then this is a responsive site
 - e. If www.desktop.mobile is the same site as www.mobile but they are both different to www.desktop and different again to m.mobile and/or m.desktop.mobile, then this is a responsive site with an m.dot site

Variations of the page (responsive sites only)

Why provide different variations of a page?

Responsive sites contain multiple variations or versions of a page. It is important that **each variation of the page is tested** and that **all functionality is available on all variations of the page**. There may be many reasons that a site owner wants to provide variations of one particular web page, for example:

- Highlighting particular content on mobile, such as phone details;
- Hiding particular content on mobile, such as image galleries; and
- Hiding functionality that does not work on mobile, such as a drag and drop feature.

Developers can identify one or more of the following four features to determine which content to include or display on a page:

- The **device** (e.g. iPhone, desktop, Android, etc.);
- The **operating system** (e.g. Windows, iOS, OS, etc.);
- The **browser** (e.g. Safari, IE 11, Chrome, etc.); and
- The **screen size** (e.g. 280 by 720, 1920 by 1080, 320 by 480, etc.).

A note on variations due to screen size

Low vision users (who use the zoom function inherent in the browser) are often restricted to a mobile view of the site on their desktop. As part of WCAG2, zooming to 200% should already be included in regular testing (and therefore is not included in this methodology). It is essential that functionality is not removed due to a variation in the page.

For example, previously in YouTube (this has now been fixed), the upload and notifications buttons were visible at 100% screen size but not at 200% screen size. This would mean that people browsing at 200% screen size would not be able to upload a video or view their notifications.

Figure 18: YouTube at 100% zoom

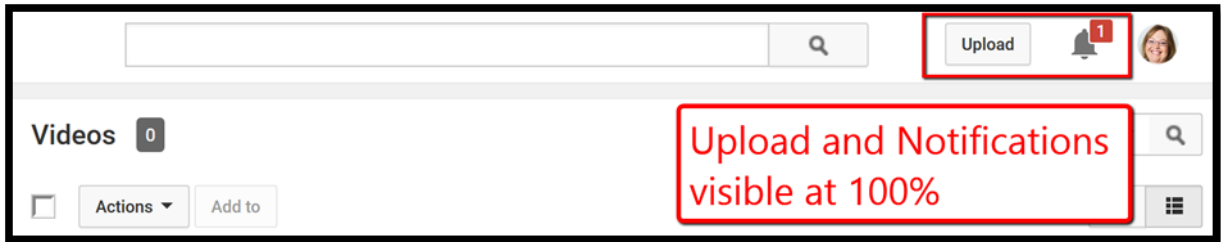
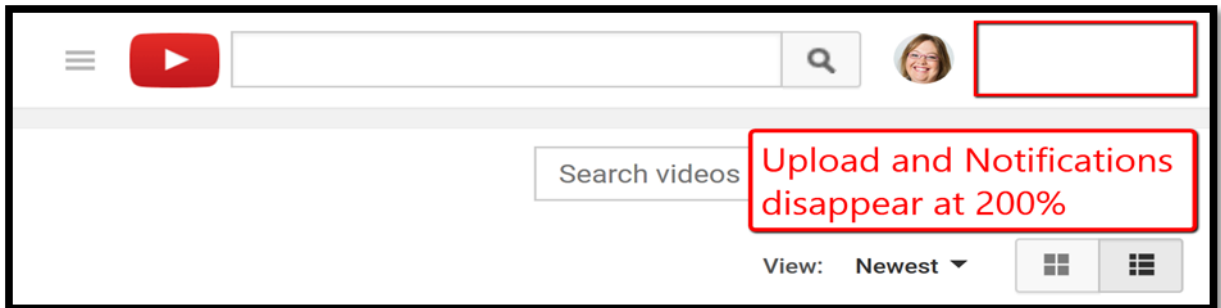


Figure 19: YouTube at 200% zoom



Pages should not require two dimensional scrolling, except for content that requires the scrolling for meaning, such as a data table. The [Success Criteria 1.4.10: Reflow](#), additionally refers to a screen magnification of 400%, but this is not a viable setting for a mobile device.

Types of variations of a page

One type of variation of a page covers differences in content and code. This is where the content is varied dependent on a particular feature (device, operating system, browser, screen size). In the following example, the AccessibilityOz web site shows top-level navigation at a desktop screen size. At a mobile screen size this is replaced with a hamburger menu.

Figure 20: **AccessibilityOz site at desktop width**

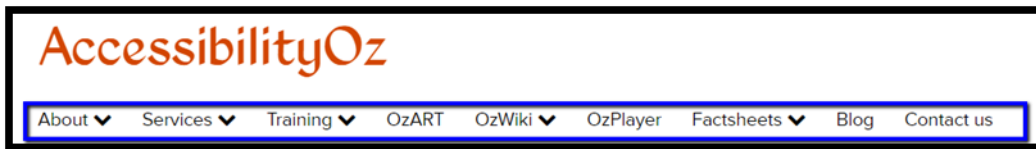


Figure 21: **AccessibilityOz site at a mobile width**



Another type of variation of a page relates to differences only in the layout of content. This is where the content does not change, but the presentation of that content does. In the following example, the ICT Accessibility Testing Symposium navigation appears in a single row of four items at a desktop screen size. At a mobile screen size this changes to a single column of four items, in a row.

Figure 22: **ICT Accessibility Testing Symposium site at desktop width**

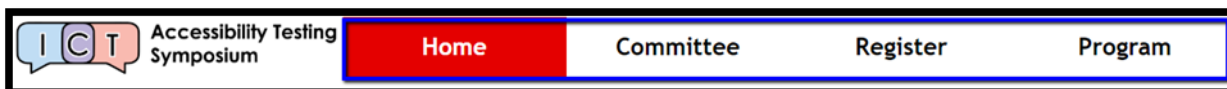
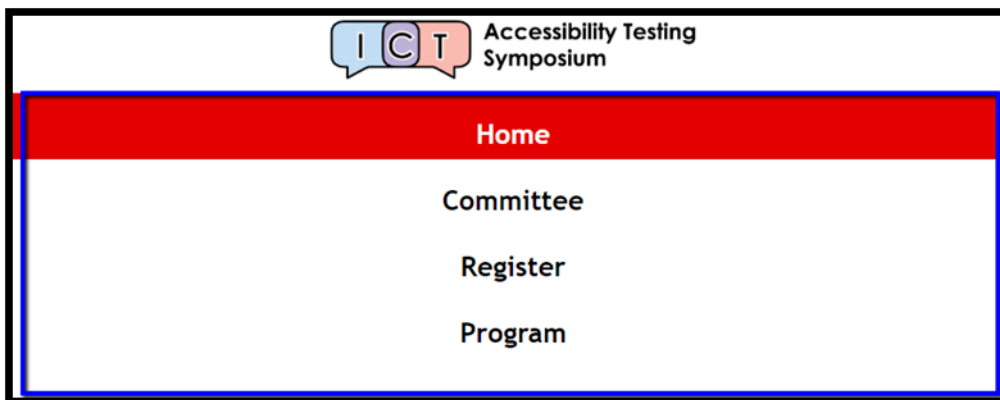


Figure 23: **ICT Accessibility Testing Symposium site at mobile width**



Which variations of a page need to be tested

Every variation of a page needs to be tested, if it contains a variation in content or variation in presentation. For more information see [WCAG2.1: Conformance](#).

People with disabilities may be restricted to one variation of the page only and may not be able to access other variations due to restrictions based on their accessibility needs. For example, people can be dependent on one particular device, orientation (people

with vision impairments relying on screen readers often lock the screen in portrait mode), screen size or use of zoom.

How to find variations of a page

There are two steps to finding variations of a page. However, we always recommend that you **ask the site owner** for the variations of a page. However, we still strongly recommend that you undertake the following two steps as well:

- Resize a window on desktop manually or with a browser extension
- Open the site in different devices and change orientation

Resizing a window on desktop manually

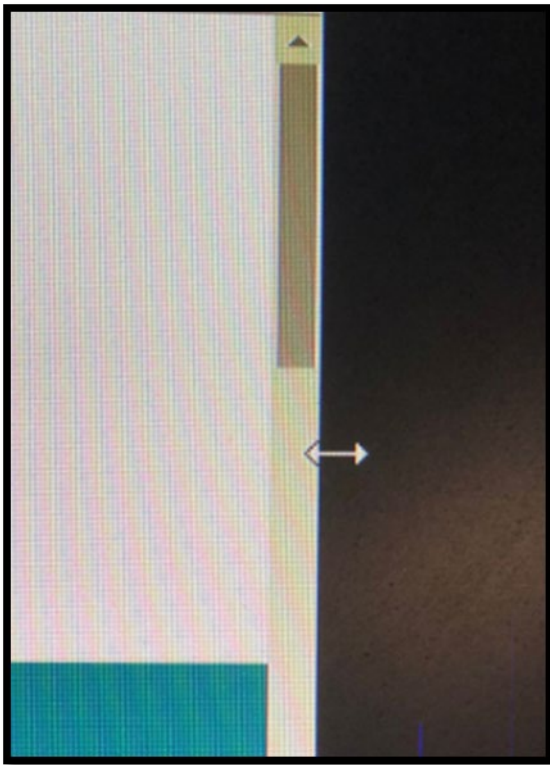
Open the web site on the desktop at fullscreen. Take a screenshot of the page. Resize the window using the “Restore down” button in the top-right hand corner.

Figure 24: **Minimize, Restore down and Close buttons**



Resize the window so that you see the same display as the screenshot you first took. Then select the right-hand side and drag it to the left.

Figure 25: **Select the side of the screen and drag to the left**



Watch to determine if items change or move. Each time the items change or move this is another variation of the page. Take a screenshot of each variation.

Figure 26: **Variation 1 of the AccessibilityOz web site**

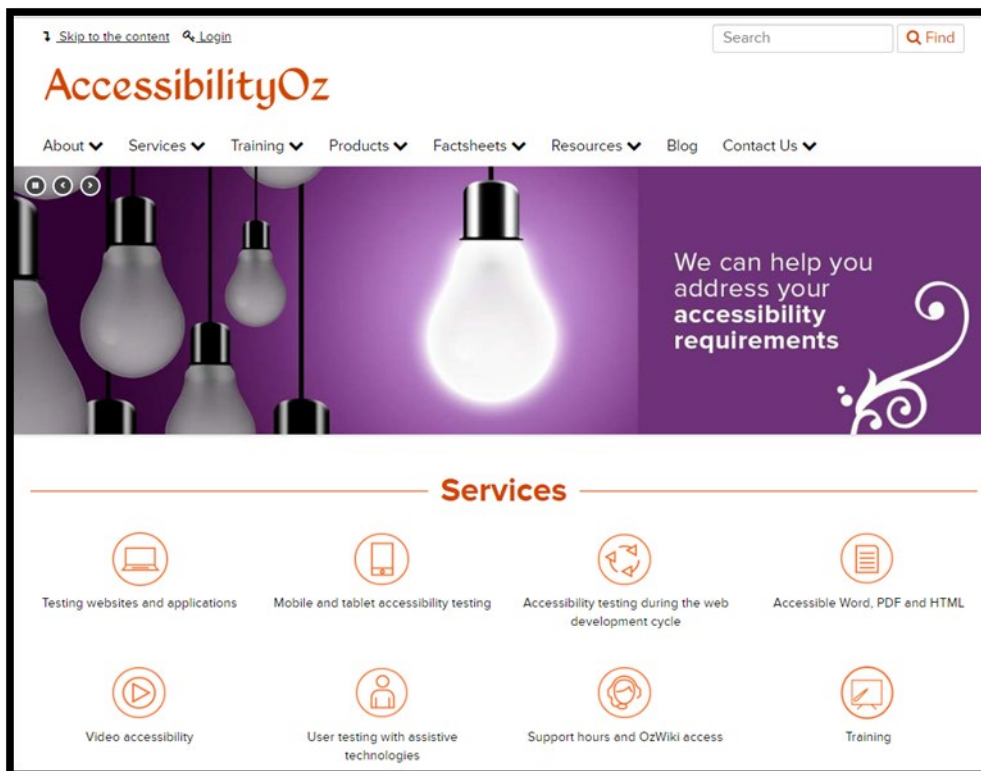


Figure 27: **Variation 2 of the AccessibilityOz web site (the navigation has been replaced by an "Open Menu" button)**

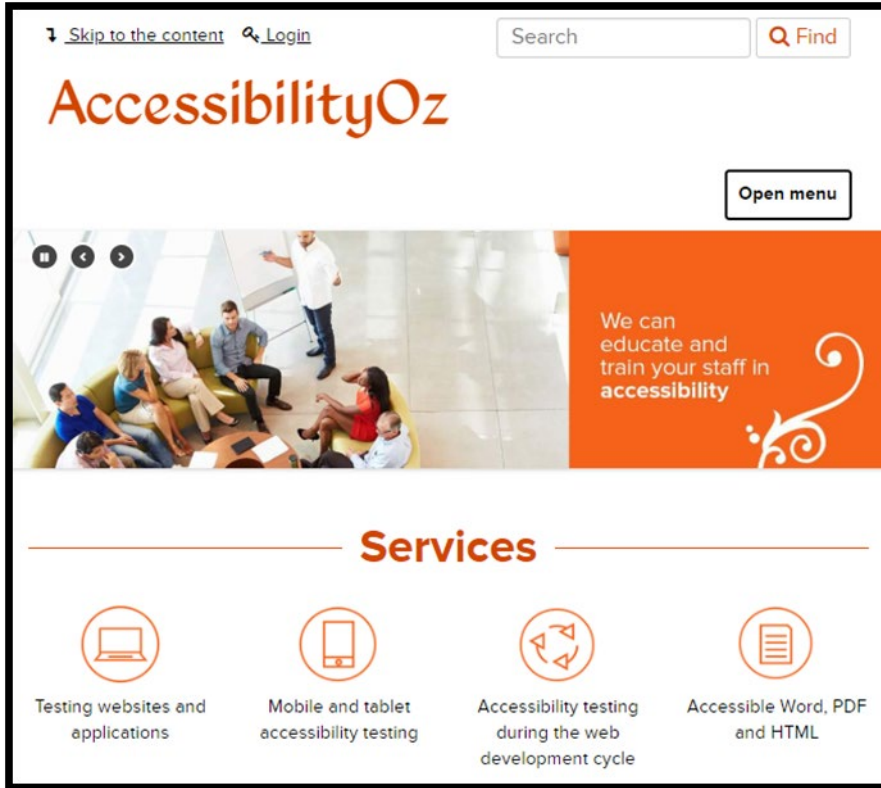
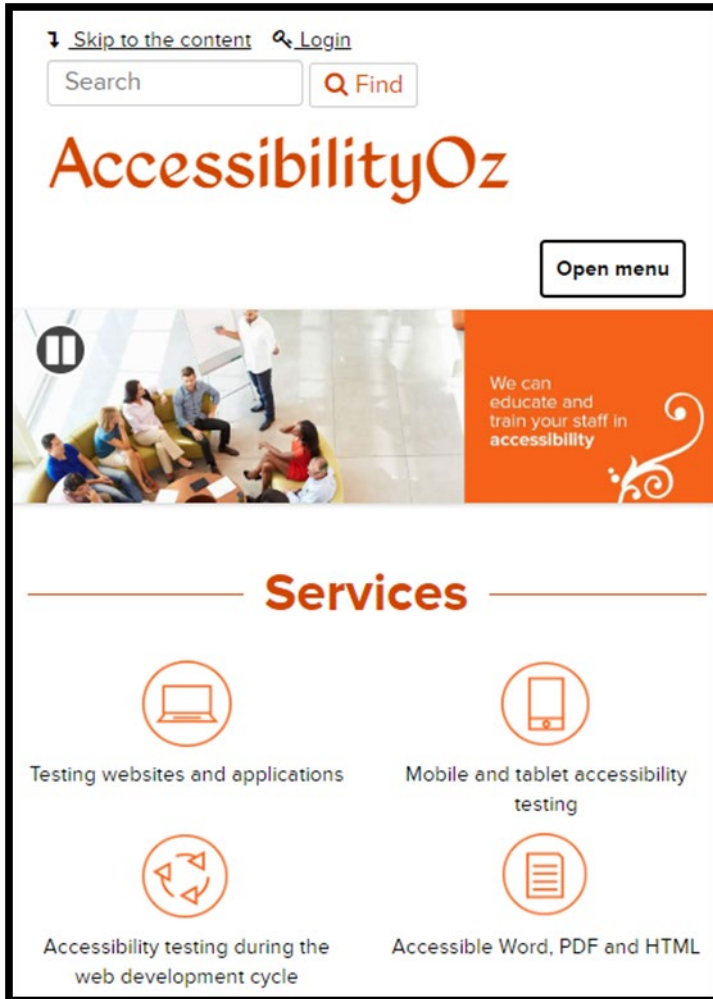


Figure 28: **Variation 3 of the AccessibilityOz web site (Services are now two per row and the controls for the slideshow have changed)**



Opening the site in different devices and change orientation

Open the site in each of your devices. Take a screenshot in portrait mode and in landscape mode. How do they compare to the screenshots you have already taken?

Figure 29: AccessibilityOz site in portrait orientation on iPhone

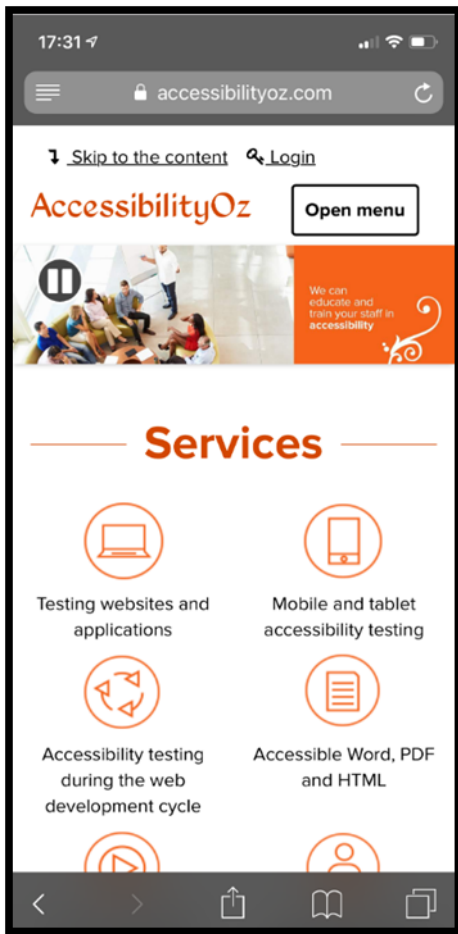


Figure 30: AccessibilityOz site in landscape orientation on iPhone

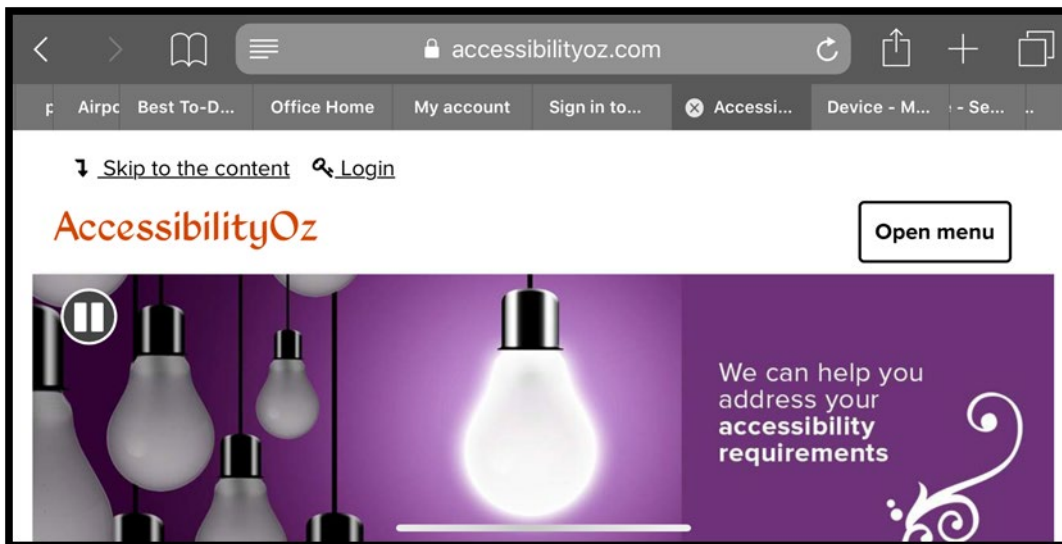


Figure 31: AccessibilityOz site in portrait orientation on iPad

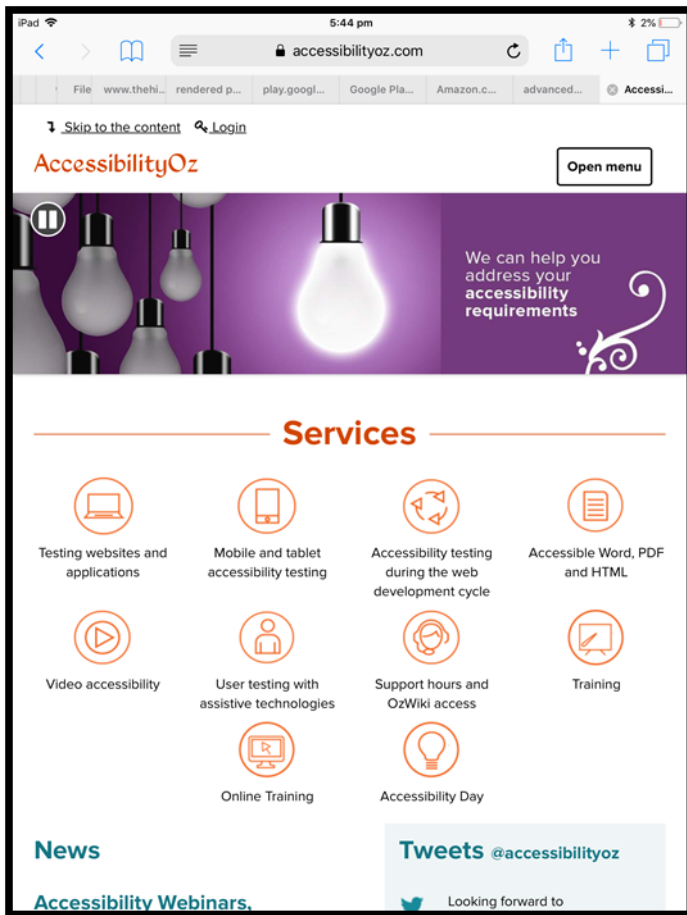


Figure 32: AccessibilityOz site in landscape orientation on iPad

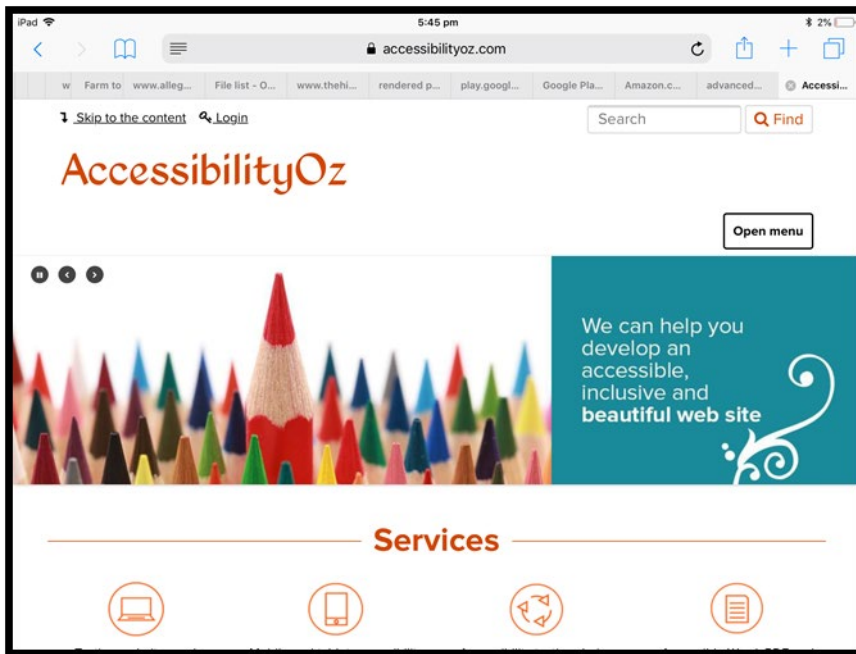
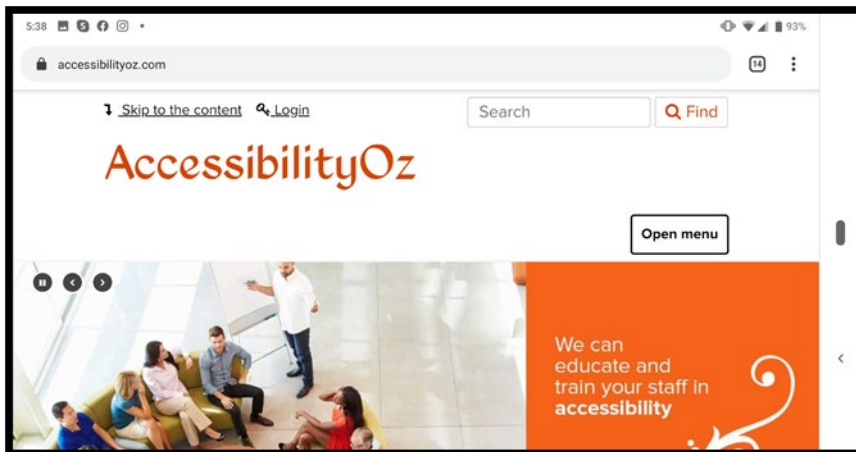


Figure 33: AccessibilityOz site in portrait orientation on Google Pixel 3+



Figure 34: **AccessibilityOz site in landscape orientation on Google Pixel 3+**



How to test variations of a page

Although all functionality must be available on all versions of a page, the functionality does not need to be displayed in the same way. For example, content can be hidden via an expandable menu (e.g. a hamburger menu), included on another page via a link (e.g. a map moved to a secondary page with a link provided) or the functionality could be replaced with a different feature (e.g. a drag and drop feature replaced with a form that provides the same functionality). For more information see the [WAI Tutorial page](#), when you zoom in to 300% the left-hand navigation moves to the bottom of a page, and a link "Jump to the navigation" is added.

Testing methods

There are four main testing methods in mobile testing:

- **Device:** test on mobile and tablet devices
- **Device with assistive technology:** test on mobile and tablet devices with assistive technologies
- **Responsive Window:** test on different sized window on desktop
- **Desktop:** test on desktop

Devices

Please note that “Internet” browser that comes pre-packaged with most Samsung phones is very dependent on the operating system itself and it is a better representation to test with Chrome.

Note: an assumption is that mobile sites will support touch access because that is the dominate mode use by typical users.

Due to the popularity of the Android system, there are tens of thousands of Android operating systems and browser combinations available. It is not possible to test on all these systems.

Recommended devices and browser combinations

- iPhone, Safari
- iPad, Safari
- Android phone, Chrome

Determining which devices to test on

In the United States and other western countries, iOS devices are most popular. In Asia and other eastern countries, Android devices are most popular. To best find which devices to test on, review the Google Analytics or other analytics system for the requisite web site.

Other assistive devices

Where a site is directly aimed at people with particular kind of disability it is worth considering including assistive devices and/or other assistive technologies used by potential users. Including these devices will help to ensure that a tested page is not only accessible, but also user-friendly. A good example of such a site is [Bookshare](#) – an online library for people with print disabilities. Among the users of this service you can

find users of braille notetakers and other mobile devices and their user experiences are likely to be different from the experience of standard screen reader users.

Device with assistive technologies

Even if the site is a desktop site, people will still use the site on mobile with various assistive technologies. In most cases, assistive technologies are inherent to the device. It is important to remember that even assistive technologies that are available on desktop and devices may behave differently on each system, and therefore they still need to be tested on mobile and tablet devices.

For the latest information on screen reader usage, please see the [WebAIM Screen Reader Survey](#).

Samsung includes an additional screen reader called "Voice Assistant," however TalkBack is still available as part of the Accessibility Suite. Amazon Fire also utilizes a different screen reader called "Voice View". These may need to be incorporated into testing.

The following assistive technology / device combinations should be tested.

iOS assistive technology / mobile features and device combinations

Please note that iOS 13 features, such as Dark Mode and Voice Control are not included in this list. Please note: As of July 2019, "dynamic type" on iOS does not change the appearance of web pages in Safari.

iPhone

- VoiceOver
- Keyboard
- Keyboard and switch
- Zoom
- Invert colors
- Grayscale
- Reader view

iPad

- VoiceOver
- Keyboard
- Keyboard and switch
- Zoom
- Invert colors
- Grayscale

- Reader view

Android assistive technology / mobile features and device combinations

- TalkBack
- Keyboard
- Keyboard and switch
- Magnification
- Invert colors
- Grayscale
- Increase text size
- Simplified view

Using VoiceOver on iOS devices

VoiceOver users include blind and low-vision users, as well as users with some cognitive or learning difficulties. To enable VoiceOver, navigate to **Settings** → **General** → **Accessibility** → **VoiceOver**

Then, tap the white **Off** button (beside the word *VoiceOver*) to the right to enable VoiceOver. Once in the On position, the button changes color to indicate VoiceOver is enabled. See the Apple documentation for [keyboard commands with VoiceOver](#).

Figure 35: Turning on VoiceOver on iOS

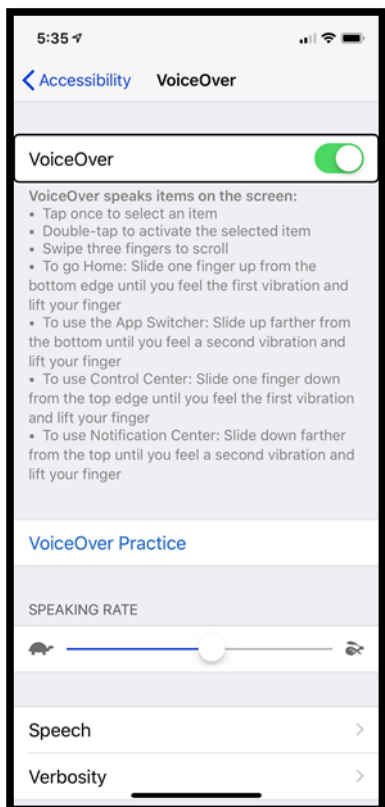
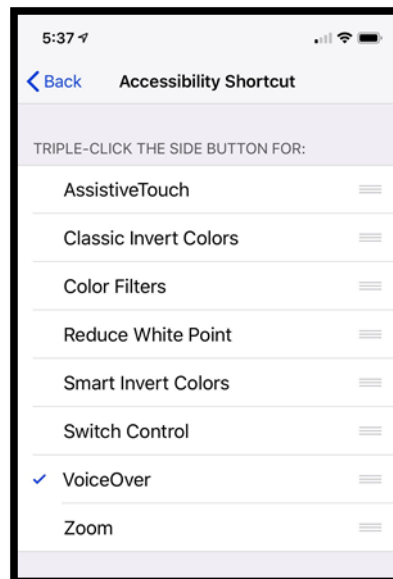


Figure 36: Setting Accessibility shortcuts on iOS



You may find it easier to toggle VoiceOver by setting the “Accessibility Shortcut” to VoiceOver and/or the other accessibility options. Once set up, triple-click the home or side button of the device to enable the menu. (If only one accessibility setting is enabled, the device won’t show a menu and will immediately toggle the setting.)

Using TalkBack on Android devices

To enable TalkBack, navigate to **Settings** → **Accessibility** → **TalkBack** then switch the button to **On**. A “Use TalkBack” confirmation screen will appear. When used for the first time a tutorial will be launched. Ensure the device volume is turned up as TalkBack doesn’t automatically adjust the volume if muted. See the [Android documentation on getting started on Android with TalkBack](#).

Using keyboard on iOS devices

iOS 12.3 and lower only support keyboard use when VoiceOver is activated. Apple is projecting that iOS 13 will support keyboard use without VoiceOver.

Connect a Bluetooth keyboard keyboard to your iOS device, activate VoiceOver, and navigate to the page in Safari as you would on desktop.

Using keyboard on Android devices

Connect a Bluetooth keyboard to your Android device, and navigate the page in Chrome as you would on desktop. Press TAB to navigate from one actionable item (buttons, links and form controls) to the next, use SPACEBAR to scroll the page, and press ENTER to activate the currently-selected item.

Using keyboard on Android devices and Switch Access

To enable switch access, navigate to **Settings** → **Accessibility** → **Switch access** and then switch the button to **On**.

Switch access allows you to control your device using configurable key combinations. This means you can use an external keyboard or switch device to execute core keyboard commands to navigate and interact with web sites. As with the use of TalkBack, this is a useful feature for testing keyboard access and content order.

To use switch access, you'll need a keyboard or a switch device paired with the tablet or mobile device. Then you'll need to assign keys to actions through **Settings** → **Accessibility** → **Switch access** → **Settings**. Select **Assign switches for scanning** to map a keystroke to the action. Assign a key/switch to **Auto-scan** as well as **Select** (Note: this will override that key's original action).

If the on-screen keyboard is not working properly with Switch Access, check that your on-screen keyboard is enabled and use the GBoard keyboard if the default keyboard doesn't behave correctly.

Using keyboard on iOS devices and switch

To enable switch control, navigate to **Settings** → **General** → **Accessibility** → **Switch Control** and then switch the button to **On**.

Switch control allows you to control your device using configurable key combinations. This means you can use an external keyboard or switch device to execute core keyboard commands to navigate and interact with web sites. As with the use of TalkBack, this is a useful feature for testing keyboard access and content order.

To use switch control, you'll need a keyboard or a switch device paired with the tablet or mobile device. Then you'll need to assign keys to actions through **Settings** → **General** → **Accessibility** → **Switch Control**. Select **Switches** to map a keystroke. Assign a key/switch/screen to auto-scan (Note: this will override that key's original action).

Using Zoom on iOS devices

People with low vision often use Zoom. Users with cognitive impairments, such as autism, may also use Zoom to focus in on key content and reduce the amount of visual

clutter on the screen. To enable full screen Zoom, select **Settings** → **General** → **Accessibility** → **Zoom**.

Then, touch and slide the white **Off** button (beside the word *Zoom*) to the right to enable Zoom. Once in the **On** position, the button changes color to indicate Zoom is enabled.

Triple-tapping the screen with three fingers will display the Zoom menu. You can adjust the following settings with the Zoom menu:

- Choose Region: select Full Screen Zoom or Window Zoom.
 - Full Screen Zoom: You can magnify the entire screen
 - Window Zoom: Magnify only part of the screen with a resizable lens. You can move the lens around to zoom in on different sections of the screen.
- Choose Filter: select Inverted, Grayscale, Grayscale, or Low Light.
- Show/Hide the Controller: show/hide the zoom controller. The Zoom controls provides a quick way to pan the screen or zoom in or out.
- Use the slider to control the zoom level.

Figure 37: **Activating zoom on iOS**

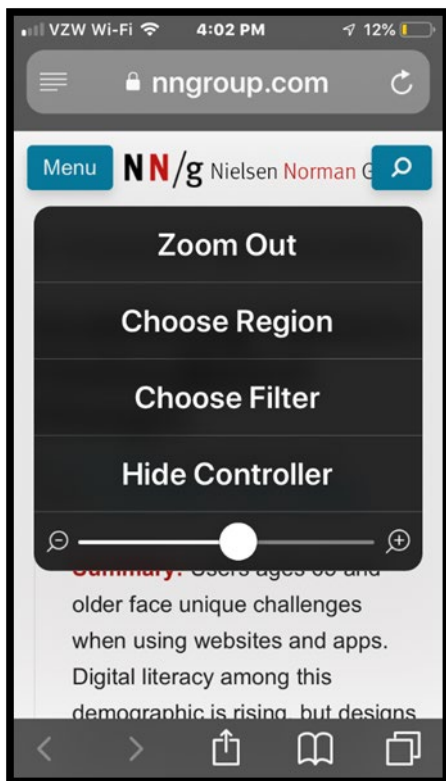


Figure 38: **Turning on the Reader view on iOS**

Figure 39: **The same article displayed in Reader view**

Standard iOS gestures—flick, pinch, tap and rotor—still work when the screen is magnified. Invert Colors and Grayscale also work with Zoom. There are some differences in commands when Zoom is run with VoiceOver.

Using Magnification on Android devices

To enable Zoom, navigate to **Settings** → **Accessibility** → **Magnification**

Choose Magnify with triple-tap or Magnify with button by selecting the item and then turning on the Use service button **On**.

When magnification gestures are enabled, you can zoom, pan across the screen and pan content:

- Zoom in or out by triple-tapping the screen. Then, while zoomed in, you can:
 - Drag two or more fingers to pan across the screen.
 - Pinch two or more fingers together or spread them apart to adjust the zoom level.

Pinch two or more fingers together or spread them apart to adjust the zoom level.

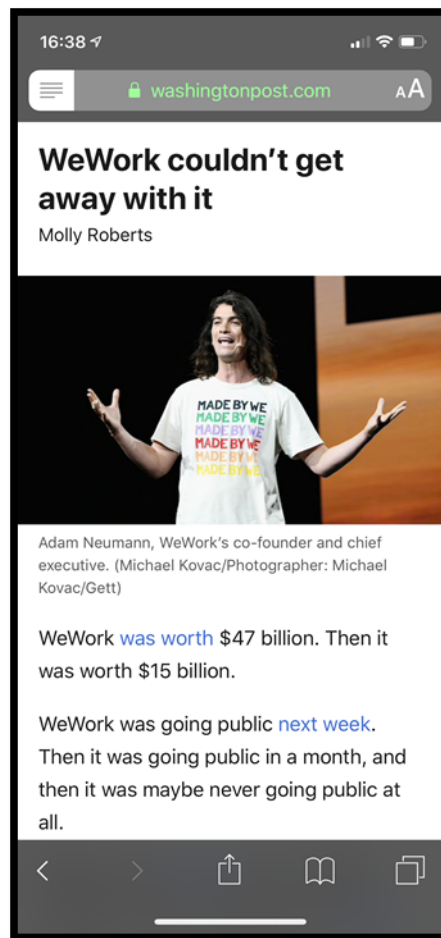
When magnify with button is on, you can use the Accessibility button at the bottom of the screen to quickly magnify. To zoom, tap the Accessibility button, then tap anywhere on the screen. Drag 2 or more fingers to scroll. Pinch 2 or more fingers to adjust zoom.

To zoom temporarily, tap the Accessibility button, then touch & hold on the screen. Drag to move around the screen. Lift finger to zoom out.

You can't zoom in on the keyboard or navigation bar.

Using Reader view on iOS

Using the Reader view within Safari on iOS is quite straightforward, as it is a main-stream feature. When Safari has enough information to create a reader view, you can see the Reader view icon to the left of the domain name (URL) at the top of the screen.



Using Simplified view on Android Chrome

To turn on Simplified view on Android Chrome you need to navigate to Chrome and activate the vertical ellipsis in the top right corner. Select **Settings** and then **Accessibility**. Select the option "Simplified view for web pages". When a simplified view of a page is available it will appear at the bottom of the screen.

Figure 40: **Vertical ellipsis in Chrome on Google Pixel 3+**



Figure 41: **Drop-down menu once the vertical ellipsis has been selected**

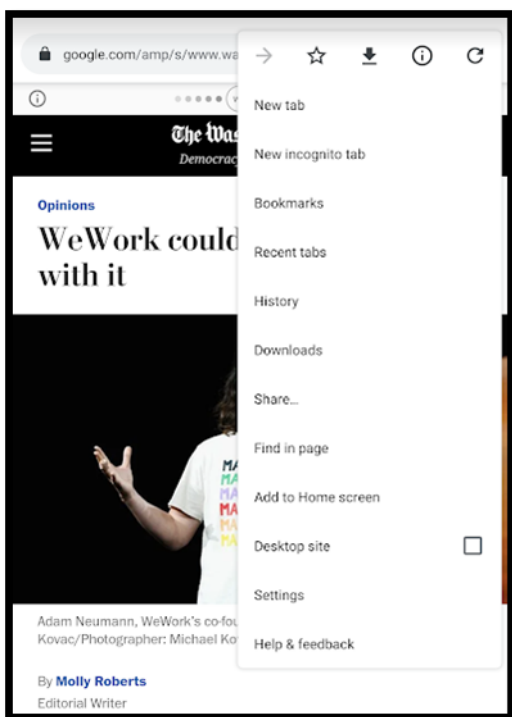


Figure 42: **Article on San Francisco Chronicle with simplified view option visible at the bottom of the screen**

Figure 43: **The simplified view of the article**



Using Invert colors with iOS

To enable Invert Colors, first navigate to **Settings** → **General** → **Accessibility** → **Display Accommodations** and find **Invert Colors** in the Vision list.

Then, tap the white **Off** button (beside the words *Invert Colors*) to enable the feature. Once in the On position, the button changes color and display colors are inverted.

The color inversion is applied to all screens until the feature is disabled by switching the button to the **Off** position.

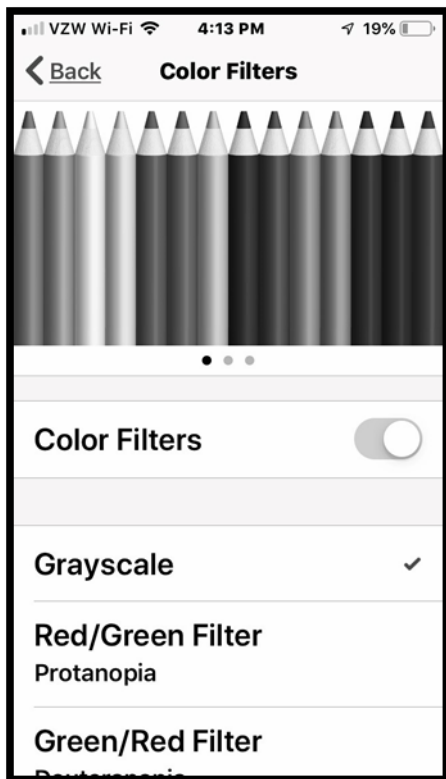
“Smart Invert,” a recent addition to iOS, will invert the colors of text and solid backgrounds while preserving the colors of graphics, media, and some apps that support dark mode. When enabled, images on your site should not be inverted so that they remain clear and consistent.

Invert Colors can be used with VoiceOver, Zoom and Grayscale.

Using Grayscale view with iOS

To enable Grayscale, first navigate to **Settings** → **General** → **Accessibility** → **Display Accommodations**. Select **Color Filters** and toggle **Color Filters** to the **On** position. **Grayscale** is listed as the first color filter in the list and should be checked (on) by default. If not, select Grayscale so it is checked. Once Grayscale is enabled colors across iOS and all apps are in grayscale.

Figure 44: Grayscale view on iOS



Grayscale is applied to all screens until the feature is disabled by toggling the button to the **Off** position. Grayscale can be used with VoiceOver, Zoom and Invert Colors.

Using Grayscale view with Android

To enable Android's grayscale view, first **enable Developer Mode** on your device. (This is a one-time process for each device.)

- Go to **Settings** → **About Tablet** (which might also be labelled "About Phone (or your phone's nickname)" or "About Device").
- Locate the Build number section and tap on the phrase "Build number" 7 times.
- Go back to the main Settings page and tap on "Developer Options."
- Enable the Developer Options switch.

From then on, you can enable Android 's grayscale view (Lollipop 5.0 onwards) by going to **Settings** → **Developer Options** → **Simulate color space (select)** a number of radio button selections appear, choose "**Monochromacy.**" Once you've made that selection, display colors will be replaced by shades of gray.

Grayscale is applied to all screens until you turn off the feature, which you can do by going back to **Settings** → **Developer Options** → **Simulate color space** and selecting "**Disabled.**"

Android's grayscale view can be used alongside any of Android's other modes or applications, including TalkBack and Shade, among others.

Using invert colors with Android

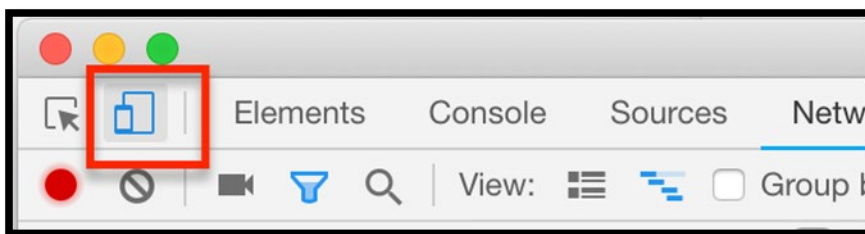
Open Settings (gear icon) → Accessibility → Color Inversion → toggle on by tapping once

Responsive Window

Some errors can be tested on desktop by resizing the browser window.

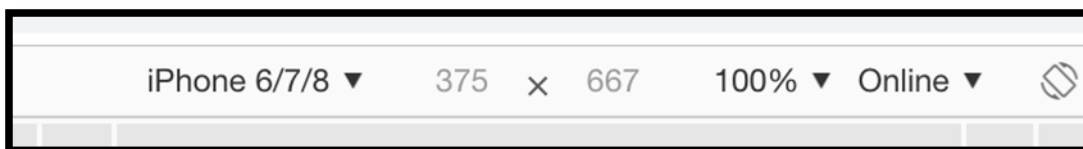
Chrome's developer tools also offer "device simulation", which can be used for window testing. Open the developer tools (View -> More Tools --> Developer Tools), then select the "Toggle Device Toolbar" option on the sidebar menu's top toolbar:

Figure 45: **Chrome Developer Tools**



This will cause the Device Toolbar to appear in the main browser window. From here you can select a device size to apply, along with other useful options:

Figure 46: **Chrome Device Toolbar**



It says Responsive. Select Responsive and a drop down appears with a number of mobile choices.

Responsive Window

Some errors can be tested on desktop by resizing the browser window.

Chrome’s developer tools also offer “device simulation”, which can be used for window testing. Open the developer tools by selecting the “more vertical icon” (which is displayed as a column of three dots with alt text “Customize and Control Google Chrome”) -> **More Tools** --> **Developer Tools**, then select the “Toggle Device Toolbar” option on the sidebar menu’s top toolbar.



Figure 47: **Chrome Developer Tools visible in the browser**

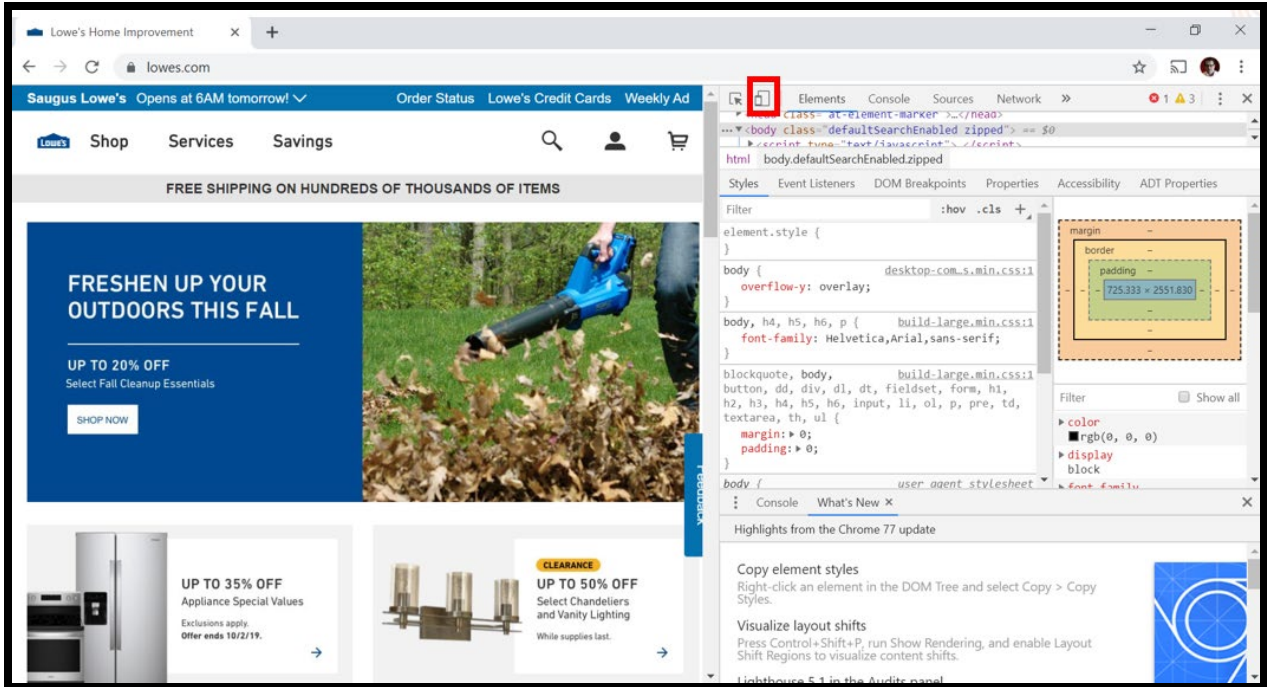
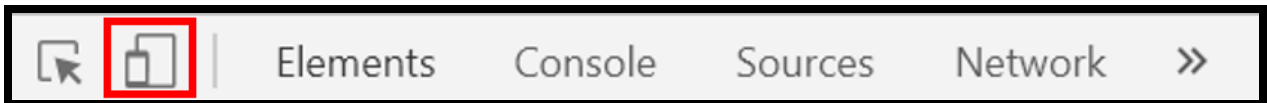


Figure 48: **Close-up view of the Toggle Devices option**



This will cause the Device Toolbar to appear in the main browser window. From here you can select a device size to apply, for example: Responsive, Galaxy S5, Pixel 2, iPhone 6/7/8, iPhone X, or iPad, etc. along with other useful options:

Figure 49: **Chrome Device Toolbar**

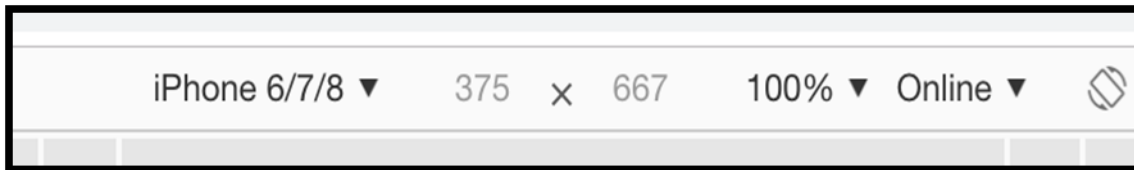
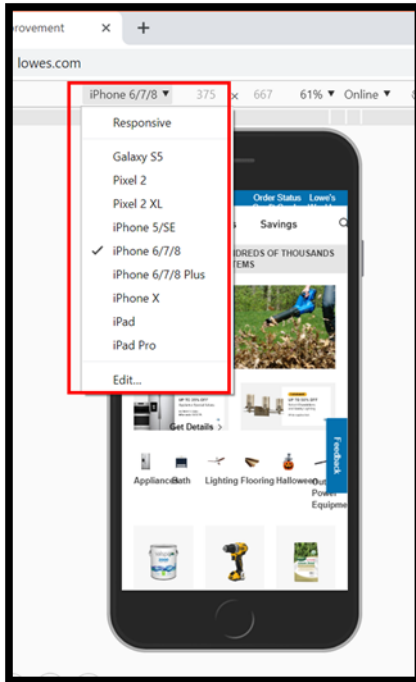


Figure 50: **Device Size Options**



Selecting a size such as iPhone 6/7/8 will change the screen to appear rendered as that device or set of devices. If Responsive is selected then you can adjust the screen size as if it was responsive site.

Desktop

Some errors can be tested on desktop, using standard accessibility methods.

Capturing errors

There are two main methods of capturing mobile errors:

- Capturing code of the page
- Capturing screenshots or video of the page on the device

Once these have been captured, you can use standard automated accessibility testing tools to test the code and screenshots.

Capturing code of the page

The mobile DOM can be captured via:

- iOS: Developer tools on Safari
- Android: Developer mode
- Browser extensions

Viewing the DOM on iOS

To view the DOM of a page with iOS, you'll also need a Mac connected to your iPhone via USB. On your iPhone, navigate to **Settings** → **Safari** → **Advanced** and enable **Web Inspector** and **JavaScript**. On your Mac, open Safari and navigate to **Safari** → **Preferences** → **Advanced**. Enable **Show develop menu** and close the pane. Between the **Bookmarks** and **Window** menus, you should see a menu called **Develop**. Choose your iPhone from that menu, and you'll see a familiar set of developer tools.

Viewing the DOM on iOS Android - Remote Debugging

Remote Debugging can be used to inspect the Document Object Model (DOM) of a web page on a mobile device from a connected desktop or laptop computer. Steps to enable this can be found in the following guides:

- [Remote debugging on Android devices with Google Chrome](#)
- [Remote debugging on iOS devices with Safari](#)

Capturing screenshots or video of the page on the device

There are three main methods for capturing screenshots or video of the page on the device

- Taking a screenshot with the device and emailing it
- Replicating the screen of the device on a desktop and taking screenshots via a desktop app
- Recording or photographing the screen of the device with another device

Taking a screenshot with the device and emailing it

- **iPhone X, XR:** To capture a screenshot, press and release the side button and the volume up button at the same time.
- **iPhone 6, 7, 8:** To capture a screenshot, press and briefly hold both the power button and the home button at the same time.
- **iPad:** To capture a screenshot, press and hold the Sleep/Wake (on/off) button on the top of your iPad. Quickly click the Home button at the bottom of the screen.
- **Samsung Galaxy phone:** To capture a screenshot, press and hold the Power and Volume down buttons at the same time (for approximately 2 seconds).
- **Samsung Tab A:** To capture a screenshot, simultaneously press and hold the Power button (located on the top-left edge) and the Home button.
- **Google Pixel:** To capture a screenshot, simultaneously press and hold the Power

then tap the Screenshot icon. Alternatively, press and hold the Power and Volume Down buttons.

Direct recording from the device (iOS and iPad with iOS 11 and later)

From iOS 11 you can **directly record video from the device**. First go to **Settings** → **Control Center** and select **"Customize Controls"**. Add **"Screen recording"** to controls' list. Now the **"Screen recording"** control will be available in Control Center. Select it to start and stop recording.

Replicating the screen of the device on a desktop and taking screenshots via a desktop app

- **Reflector** works on iOS and Android. Once your phone screen is reflected on PC, you can take screenshot any time you want. Choose where you would like to save the screenshots by clicking the third icon under the Reflector. Afterwards, just click the camera icon under the Reflector.
- **SideSync** works on Samsung devices. Open SideSync on your computer and phone to have them connected. On your computer's SideSync, click "Phone screen." Tap "v" on your phone screen window and select Capture. The screenshot images are saved to the folder that you've set in the SideSync settings.
- **ChromeCast** works on Android. ChromeCast is a mobile app that can mirror the device on a desktop or other screen.

Recording or photographing the screen of the device with another device

- **Taking a photo with another phone:** take a photo and then email it
- **IPEVO, HoverCam or Mr Tappy:** sends live video from your mobile device to a computer which can be recorded.

Acknowledgements

Relationship to existing Accessibility testing standards

This document is based on:

- W3C Web Content Accessibility Guidelines, Version 2.0
- W3C Web Content Accessibility Guidelines, Version 2.1
- BBC Mobile Accessibility Guidelines
- AccessibilityOz Mobile Testing Methodology
- TPG Mobile Testing Guide

ICT Accessibility Testing Symposium Mobile Sub-Committee

This document was developed by the ICT Accessibility Testing Symposium Mobile Sub-Committee. Members include: Gian Wild (Co-Chair), Peter McNally (Co-Chair), Brent Davis, Corbb O'Connor, Karen Herr, Kathryn Weber-Hottleman, Kathy Eng, Laura Renfro, Megha Rajopadhye, Mona Rekhi, Morgan Lee Kestner, Rafal Charlampowicz, Ryan Pugh, Steve Sawczyn, Sunish Gupta, Tom Lawton.

Contacts

Gian Wild

Email: gian@accessibilityoz.com

Peter McNally

Email: pmcnally@bentley.edu