WCAG 2.0

Compliancy

- Web Content Accessibility Guidelines -

Client: 23RED
Client contact: Attia Jamil
Authors:Benjamin RICHEBOIS & Thomas Colliers
Version 1 (25/12/2017)

Introduction

Web Content Accessibility Guidelines (WCAG) 2.0 covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make content accessible to a wider range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited movement, speech disabilities, photosensitivity and combinations of these. Following these guidelines will also often make your Web content more usable to users in general.

We ensured to meet W3C compliances throughout the development of *Year of Engineering*.

More specifically, we targeted the **AA level** - and lower. We'll explicit how we meet these expectations in the following document.

Compliancy

Perceivable

Text alternative

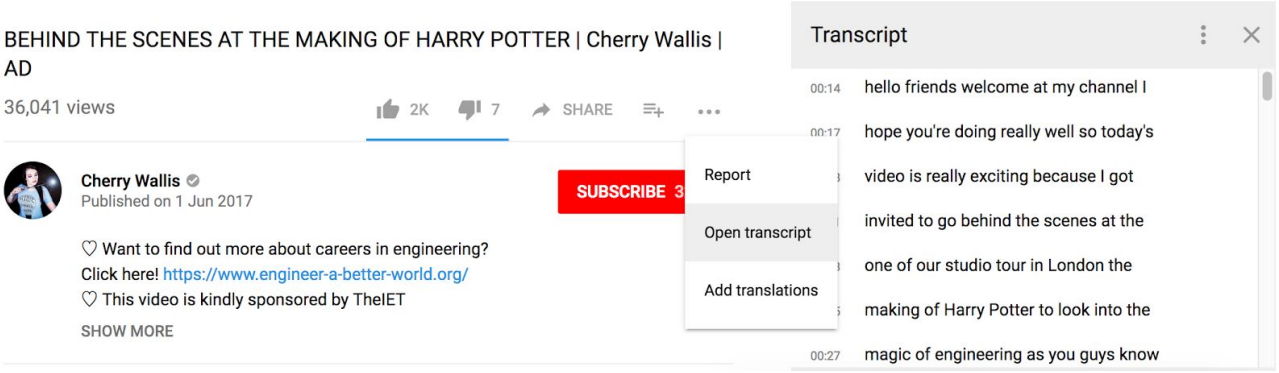
Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

All images and medias (videos in this case) provide labels (*title* attribute) and alternative text (*alt* attribute).

Time-based Media

Provide alternatives for time-based media.

YoE displays Youtube videos. Currently, the website doesn't display transcripts directly. However, Youtube auto generates transcripts and are visible on their website directly (cf image below).



Additionally, we recommend to add a message or a link to the Youtube page for users who want to read transcripts.

Additionally, the auto generated transcription might be not correct; the client should provide captions and/or subtitles.

Adaptable

Create content that can be presented in different ways (for example simpler layout) without losing information or structure.

Hierarchy and region on the page are clearly identify, with distinctive colors (aways the same in every pages).
Content is ordered in a meaningful sequence: header (yellow), navigation menu (red), body(white and grey background) and footer (blue)

Moreover, we used dedicated HTML5 tags to clearly identify each region: `<header>`, `<section>` and `<footer>`. `<main>` is badly supported by IE, that's why we didn't use it.

Distinguishable

Make it easier for users to see and hear content including separating foreground from background.

The website uses colors to structure the content: to identify sections, and also for titles, subtitles, body and links.

Except for youtube videos, YoE doesn't play any audio content. And Youtube videos don't play automatically, only on user's action.

The visual presentation of text and images of text has a contrast ratio of at least 4.5:1

For gray background (*rgba(0,0,0,0.65)* on white background, → being *#E6E6E6*) with blue text (*#1f4b96*): the ratio is 6.7:1. It passes AA level for any size text and AAA for large text (above 18pt or bold above 14pt). Sample:

> The Year of Engineering is a government campaign, which celebrates the world and wonder of engineering.

The content can be resize with the native browser tools.

Operable

Keyboard Accessible

Make all functionality available from a keyboard.

User interface components and navigation are operable, and doesn't require specific timings for individual keystrokes.

The default hotkeys are not changed, and users won't face unexpected behaviors when pressing hotkeys.

Enough Time

Provide users enough time to read and use content.

There is no time limits of any sort in the whole website

No moving, blinking, scrolling, or auto-updating information

Seizures

Do not design content in a way that is known to cause seizures.

Year of Engineering website doesn't contain anything that flashes.

Navigable

Provide ways to help users navigate, find content, and determine where they are.

The only blocks of content that are repeated on multiple Web pages are the header and footer, and we ensured they do not exceed a reasonable space (320px for header + cookie banner on desktop and 334px on mobile)

`<h1>` tags are used on top of each pages to describe topic or purpose of the current page.

We also ensure that ARIA attributes follow best practices - using `[aria-*]` and `[role]` attributes

Every links have a *title* attribute and display a description of the redirection. Moreover, every links open in a new tab.

Most pages can be reached with different ways - the navigation bar, the footer links, the sub menu for School Resources and Partner Resources, and all the internal links.

As mentioned before, headings and labels describe topic or purpose.

Outlines are not disabled and let the user knows where the focus is.

Finally, we added a sitemap page -available in the footer and with the hotkey `Alt + S`

Understandable

Readable

Make text content readable and understandable.

The website is in English, and browsers are able to automatically detect the language, except for proper names or technical terms

The `[lang]` attribute is also specified in the `<html>` tag

Predictable

Make Web pages appear and operate in predictable ways.

The website never forces sudden changes such as scroll, redirection or focus. If it happens, it's in reaction to user's action.

The document does not use `<meta http-equiv="refresh">` to refresh the page without any action from the user.

Navigation (or mechanism that are repeated on multiple Web) stays consistent within all the pages and occur in the same order each time they are repeated. Generally, components that have the same functionality are identified consistently (forms, links, submit buttons and so on).

Input Assistance

Help users avoid and correct mistakes.

Labels or instructions are provided when content requires user input. If an input error is automatically detected (unexpected format), a red caption warns the user about the mistake - and what it is expected to validate the form.

Robust

Compatible

Maximize compatibility with current and future user agents, including assistive technologies.

We worked as close of the W3C restrictions as possible. However, the technology we used - ReactJS - is not fully compatible with W3C recommendations. We ensure to test the website on all the most common browsers to prevent any dysfunction.

Moreover, the DOM is entirely built in javascript, meaning the source code is no appropriated of parsing - for index robots for instance. That's why we set up a *React Rendering Server* to overcome issues tied with DOM built in JS.

Complementary resources

W3C

W3C ARIA Guideline	Link
Web Content Accessibility Guidelines (WCAG)	Link
How to Meet WCAG 2.0	Link
EN 301 549	PDF