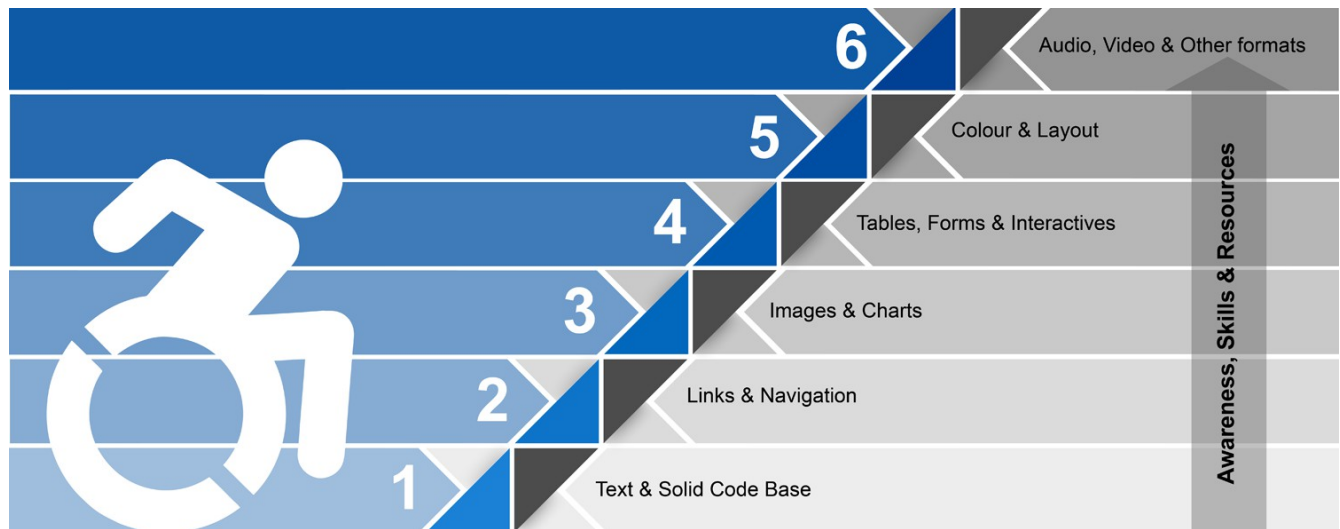


# Ramp it up! Action based guide for creating accessible websites



## 1. Text & Solid Code Base

Electronic text-based content is accessible by nature. It can be rendered visually, auditorily and tactilely.

Use HTML's built-in **semantic structure** [1] - headings, lists, quotes - to code meaning directly into your content.

Keep the **reading level** as simple as possible.

Define the **natural language** & mark language changes.

Ensure there is a **doctype**, **charset** & a **unique page title** in each of your templates for a valid code base.

## 2. Navigation, Links & Landmarks

Navigation, links and landmarks, like stepping stones, help you find your way. Use unique meaningful words for linked text. Keep links DRY [2] (do not repeat yourself over and over and over).

Leave **context changes** - pop-ups, new tabs - in the control of the user.

Use WAI-ARIA [3] **landmark roles** or HTML5 landmark tags to provide direct access to page content (formerly we used skip navigation links).

Design well-planned, **consistent navigation**. Usability helps all users.

## 3. Images & Charts

Provide meaningful text alternatives for all content images. The alt attribute is required, but can be left empty, if appropriate.

For charts or complex images you may need to **link to supplementary content** or use the **longdesc** attribute.

The goal of text alternatives is to **maintain the meaning** of the document whether you can see the image or not.

#### **4. Tables, Forms & Interactives**

Simplify tables wherever possible.

Use the rich **semantic table mark-up** available in HTML - **thead**, **tfoot**, **tbody**. Introduce tables with the **caption** element. Appropriately distinguish table header cells from table data cells (**th**, **td**) on rows and columns.

Use **tables only for tabular data** – not content layout!

Forms are complex by nature, so **keep them simple** – don't ask for data you don't need.

**Use & associate labels** for every form control. Use the **button element** when you need a button.

**Group form controls** to create meaningful organization in forms - **legend**, **fieldset**, **optgroup**. Organization is good for all users.

Design **clear error identification & intuitive error handling**. Clearly tell the user where they have gone wrong, how to fix the problem and when they have succeeded.

Provide **reasonable time-outs** (think user control).

Employ **WAI-ARIA** [3] standard to define roles and behaviours that HTML cannot describe. This is particularly important for interactive structures like forms.

*Note: most legal cases have happened around in-accessible tables & forms [4].*

#### **5. Colour & Layout**

Don't rely on colour alone for meaning.

**Choose colours wisely**. Avoid confusing colours The *Brewer Palette* is a good resource.

Test for **sufficient contrast ratios**.

Use **CSS for layout & consistency**. Using **CSS & HTML together properly** is best method in making your content accessible & easier to maintain.

#### **6. Audio, Video & Other formats**

Do not auto-play anything (think user control).

Provide open or closed **captions** for audio and video.

**Describe video** when the meaning cannot be understood by the soundtrack alone. Video description can be open or closed.

**Providing transcripts**, in some cases (e.g. talking heads & interviews), makes better sense than captions.

For other formats, such as PDF or Word documents, first ask is there a reason not to use HTML? Follow the WCAG for content in Word documents and PDFs. **PDFs must be tagged** to be accessible [6].

## **Summary**

When web design teams ramp up a website for users of assistive technology and other edge cases, the payoff happens at many levels. The website will likely be easier to find, be of a higher quality, be easier to maintain, last longer and work better for everyone - all of which contribute to a better return on the investment. Just like accessible ramps at building entrances, the virtual ramp built into websites benefits more than the intended audience.

## **References**

[1] (2013) Semantic Structure. WebAIM.org.

<http://webaim.org/techniques/semanticstructure/>

[2] Don't repeat yourself. Wikipedia.org. [http://en.wikipedia.org/wiki/Don%27t\\_repeat\\_yourself](http://en.wikipedia.org/wiki/Don%27t_repeat_yourself)

[3] WAI-ARIA Overview. W3C.org. <http://www.w3.org/WAI/intro/aria.php>

[4] Monsebraaten, L. (2010, Sep 19) Blind woman says federal websites discriminate against the visually impaired. *Toronto Star*.

[http://www.thestar.com/news/gta/2010/09/19/blind\\_woman\\_says\\_federal\\_websites\\_discriminate\\_against\\_the\\_visually\\_impaired.html](http://www.thestar.com/news/gta/2010/09/19/blind_woman_says_federal_websites_discriminate_against_the_visually_impaired.html)

[5] Brewer, C. Harrower, M. & Pennsylvania State University. Color Brewer 2.0  
<http://colorbrewer2.org/>

[6] Clark, J. (2005) Facts and Opinions About PDF Accessibility. A List Apart.  
[http://alistapart.com/article/pdf\\_accessibility](http://alistapart.com/article/pdf_accessibility)

## **Resources**

- Clark, J. (2003) Building accessible websites. Indianapolis: New Riders.
- Hendren, S. & Glenney, B. (2012) Accessible Icon Project:  
<http://www.accessibleicon.org/>
- Horton, S. & Quesenbery, W. (2014) A Web For Everyone.
- <http://rosenfeldmedia.com/books/a-web-for-everyone/>
- WCAG 2.0: <http://www.w3.org/WAI/WCAG20/glance/Overview.html>
- WCAG Samurai: <http://www.wcagsamurai.org/>

- WebAIM: <http://webaim.org/>
- WAVE Accessibility Evaluation Tool: <http://wave.webaim.org/>

### ***Acknowledgements***

Thanks to Amita Pharshy for her comments.