

Accessing the mobile web: myth or reality?

Henny Swan

About the author



Henny Swan is a Web Evangelist for Opera, advocating web standards and the open web, with a specialism in both web and browser accessibility as well as the mobile web.

Henny takes an interest in where accessibility standards overlap with mobile best practice and in particular, internationalisation. She is a member of the Web Accessibility Initiative User Agent Accessibility Working Group (UAAG) [<http://www.w3.org/WAI/UA/>] and Co-Lead of the Web Standards Project (WaSP) International Liaison Working Group (ILG) [<http://www.webstandards.org/>].

A major area of interest for Henny is web standards in general and how internationalisation and mobile access complement web accessibility.

Having started out working for a search engine in China in the late 90s, she then went to work for UK charity RNIB as a Senior Web Accessibility Consultant. She speaks at various international conferences and contributes to the Opera Developer Network and blog as well as her own blog (iheni), which looks at accessibility, internationalisation and mobile access.

Outside work, Henny can be found kick-boxing, entertaining and cooking Chinese food, as well as hanging out in Second Life.

Contents

Introduction	4
Leaving no user behind	5
Mobile browser wars?	8
Mobile browser wars?	8
Safeguarding usability and accessibility	9
Mobile web versus full web	10
Progressive enhancement.....	10
Media types and media queries	12
HTML5	13
Accessible websites help mobile optimisation	14
The way of the widget.....	15
Conclusion.....	16
Glossary.....	18

Introduction

After a few false starts, mobile browsing has finally gone mainstream. No longer is it the domain of the geek or the deep-pocketed business man or woman, it is now becoming more integral to how we access the web today.

The mobile web has been knocking on our doors for years but has never quite been attractive, or usable enough, to really take off. What has changed is the demise of WAP¹ and the advent of better mobile devices (smaller, cheaper, faster, sleeker), social networking (Facebook, Twitter, LinkedIn etc.) and cheaper mobile browsing packages. All of these factors have converged to make mobile browsing a practical alternative.

However, while demand for the mobile web is growing, mobile web content is yet to mature, with many problems of usability and accessibility that are reminiscent of desktop web content ten years ago. Added to this are the specific problems associated with mobile browsing such as size of screen display (viewport), handset capability context (being outside, in noisy places, differing light, time restricted), and technology support (lack of JavaScript, Flash, CSS cascading stylesheets² and so on).

While many of these issues are bad for the evolution of the mobile web in general, they are a very real problem for disabled and older users in particular. Given that we not only socialise but also work online using mobile devices and are becoming increasingly reliant on information on the move, the danger of leaving a significant proportion of people behind is a grave one indeed.

This article highlights a few of the problems with mobile access today, considers who is affected by them and looks at how we can overcome these issues by drawing on lessons learnt from desktop technology and how the evolution of accessible web content there can influence the evolution of accessible web content for mobile devices.

¹ See Glossary at the end of the article

² Comparison of stylesheet languages

[\[http://en.wikipedia.org/wiki/Comparison_of_stylesheet_languages\]](http://en.wikipedia.org/wiki/Comparison_of_stylesheet_languages)

Leaving no user behind

“77 per cent of 65-74-year-olds use a mobile, whereas only 36 per cent use the internet.” Andrew Harrop, Age Concern

When developing content it is all too easy to design and build with ‘average’ users in mind, perhaps much like ourselves, forgetting that many don’t fit neatly into the ‘average’ user category.

According to the Disability Rights Commission, there are 8.6 million registered disabled people in the UK – 14 per cent of the population³. In addition, the Government estimates there are 12 million people aged 60 or over – 21 per cent of the UK population – who may also struggle online. As Robin Christopherson of AbilityNet says⁴:

In the UK there are around 1.6 million registered blind people, 1.5 million with cognitive difficulties, six million with dyslexia and a further 3.4 million who have some problem making use of a standard computer difficult or impossible. In addition there is an increasing number of elderly ‘silver surfers’ with failing eyesight or arthritis. These potential internet users represent a spending power in excess of £120 billion.

A disabled user could be anyone who has a visual, hearing, cognitive or motor impairment, or any combination of these. Typical barriers that people face in using technology – and mobile devices in particular – broadly fall into the following categories:

Keyboard access: users can find website navigation that relies on a mouse impossible to use if there is only an alphanumeric keypad available.

Fallback content / Alternative content: some users of assistive technology, such as blind users with screen readers, may find it hard to access content delivered using Flash, Canvas or some types of JavaScript. This means that content and/or functionality is unavailable. Equally, on a mobile device that does not support Flash or JavaScript, mobile users won’t be able to access content and functionality, so alternative content should be used. For example, if a form validates client-side, programmers should always add in a server-side validation for those that do not have JavaScript enabled.

³ Quoted in *Benefits of an accessible site Part 1* by Web Credible
[\[http://www.webcredible.co.uk/user-friendly-resources/web-accessibility/benefits-of-accessible-websites-1.shtml\]](http://www.webcredible.co.uk/user-friendly-resources/web-accessibility/benefits-of-accessible-websites-1.shtml)

⁴ *State of the eNation Reports: Disabled people favour accessible sites*
[\[http://www.abilitynet.org.uk/enation9\]](http://www.abilitynet.org.uk/enation9)

Complex content: lengthy text, poorly spaced layout, inconsistent navigation, poorly formed link text, wordy headings and copy can all contribute to making a site less readable for many users. Good use of images, colour and layout can often help facilitate readability on desktop web pages and is equally – or even more – relevant when it comes to mobile web access.

Many of the barriers a disabled user encounters on a desktop are also felt by non-disabled users on mobile devices. The W3C's *Shared Web Experiences: Barriers Common to Mobile Device Users and People with Disabilities*⁵ describes the crossover and how fixing web content for desktop access can help the usability and accessibility of mobile web content. Knowing that making your web pages accessible for disabled users also helps mobile access can also help when developing a business case for your organisation to build in accessibility or mobile support. The additional users this can accommodate is not an insignificant number.

Disabled users are therefore not the only group to benefit from good design. Combined, disabled and older users account for a significant part of our population, roughly 48 per cent (although some belong in both categories). Given we are all ageing, this is a market that is more about us than we may realise, especially as life expectancy increases.

Many older users come to the web today with little or no experience of the digital world and computing, and many of the more experienced have never surfed with a mobile device, so facilitating ease of access is crucial. Arguably this will not remain the case as years go by and today's standard web users age, but expectations to be able to use the web fully will not dwindle with age. It is crucial therefore that we safeguard the accessibility of the web from mobile devices for both disabled users and the elderly.

Another significant group of people overlooked are global users. Opera's 'State of the Mobile Web' reports⁶ analyse usage of Opera Mini across the globe and provide some crucial insight to mobile browsing habits globally. In developing countries, where the infrastructure is poor and access to hardware is scarce, there is evidence that reliance on mobile web access is significantly higher as people are more able to afford mobile devices than computers which in turn need access to a phone line or Wi-Fi and a general infrastructure to support desktop browsing.

⁵ <http://www.w3.org/WAI/mobile/experiences>

⁶ State of the Mobile Web, October 2009 [<http://www.com/smw/2009/10/>] and November 2009 [<http://www.opera.com/smw/2009/11/%5D>]

The fact that some people's main, or only, web browsing experience is on mobile devices could add weight to the idea that with the growing popularity of mobile browsing, designing for mobile access could influence desktop design. This includes not just web pages but also web browsers. Ease of access, efficiency and usability are at a premium on mobile devices where users are restricted by small screens, high network charges and varying support for fonts and colours. Where usability and accessibility are relatively easy to ignore on a desktop, they are essential on mobile devices because small screen size or difficult navigation have an impact on all users, not just those with sensory impairments.

Given the diversity and range of mobile web users and growing reliance on mobile browsing to communicate and work, it is essential that we accommodate disabled and older users, and those global users for whom access through a mobile device is the only means available.

Mobile browsing growth in Africa

In Africa triple-digit percentage growth in mobile Web usage was observed 2008–2009, in just one year. Page views in the top 10 countries increased by 374%, unique users increased by 177%, and the amount of data transferred increased by 183%.

State of the Mobile Web, November 2009

<http://www.opera.com/smw/2009/11/05/>

‘1.6 billion people are online, yet more than 4 billion people – two out of every three people on Earth – have a mobile device or access to one. By making the Web accessible on mobile devices, we can usher in a communications revolution on an unprecedented scale and pace.’ Jon S. von Tetzchner, Co-Founder Opera Software, State of the Mobile Web, October 2009 □

Mobile browser wars?

1999 was the heyday of the browser wars on desktop with the web largely subject to the desires of two vendors, Microsoft and Netscape, as their browsers Internet Explorer and Netscape Navigator went head-to-head in the battle for supremacy. The cost to the web in general was significant as proprietary technologies protecting the vested interests of these two vendors were pushed to the detriment of web standards. Interoperability, accessibility and usability were the casualties as developers were forced to take sides and design for one browser, while users found that their favourite websites were not guaranteed to work in their browser of choice.

Web standards provide a formal framework, made up of technical specifications and best practices that define how we build web pages. They are interdependent, vendor neutral and intended to work across different browsers and platforms to ensure interoperability, accessibility and usability of web content.

Well-known examples of technologies that fall into the web standards category are HTML⁷ and CSS⁸. All browsers – desktop or mobile – as well as assistive technologies are designed to work with HTML and CSS, two technologies that can be considered the backbone of the web.

Proprietary web technologies, unlike web standards, are vendor specific and therefore can be problematic to implement across different platforms and browsers. A good example of this is Flash. On the desktop computer, Flash content can be made to a large extent accessible but keyboard access into and out of Flash content from the web page itself is not possible with Firefox, Safari, Opera, Google Chrome or other browsers, but only with Internet Explorer which in turn uses another proprietary plug-in, ActiveX, to enable keyboard access⁹.

As a result, keyboard-only users miss out on Flash content on the desktop. Similar problems are presenting on mobile devices, as Flash is not supported as well as HTML or CSS. This affects all users and is an example of how both interoperability and accessibility are damaged when proprietary technologies are used.

The web is too vital – for commerce, for business, and for society – to be in the hands of any one vendor. While on the desktop the web is becoming increasingly open, the mobile web is at an important crossroads where mobile platform and browser lock-in *could* threaten to splinter development of mobile content as developers feel they have to choose one platform to develop for. This means they may have to develop using proprietary technologies rather than open web standards and technologies that can be deployed cross-platform and browser-independent.

⁷ HTML5 Web Forms [<http://www.whatwg.org/specs/web-apps/current-work/multipage/forms.html>]

⁸ CSS Zen Garden [<http://www.csszengarden.com/%5D>]

⁹ Henny Swan, 'Flash and keyboard access across browsers' [<http://www.iheni.com/flash-and-keyboard-access-across-browsers/>]

Safeguarding usability and accessibility

Looking ahead, what lessons can be learnt from desktop web development and what technologies and best practices should we be looking towards to advance the mobile web while safeguarding its usability and accessibility?

We need first to consider those issues that are barriers to usability and accessibility – and the constraints on web design and authoring:

Display screen size (Viewport) On a desktop we contend with varying screen sizes but guidance exists as to what sizes should be accommodated. On mobile devices there are dozens of different screen sizes and resolutions, making it extremely difficult to know the sizes for which content needs to be designed.

Handset capability All handsets are not equal. Some, such as the iPhone, may offer a good range of colours, fonts and styles, whereas others may have limited options. This being the case it is tricky to know, as the web page author, what baseline set of colours, fonts and styles should be used. You do not want to pick styles that do not render well on mobiles devices but, equally, you do not want to constrain design on phones that have advanced styling support.

Technology supported Not all handsets can support all technologies. Flash and JavaScript are obvious technologies that fall into this category. Web page authors need to think about using alternative content (also known as fallback content) so that mobile devices without Flash or JavaScript can still access this content using HTML replacements.

Context This is probably the hardest area to allow for on mobile devices as it is the least easy to define and predict. Context affects the browsing experience more on mobile devices than desktops as users find themselves accessing content on the move, often with little time and not always in the best of conditions. There may be poor light, too much glare, noise, poor signal, prohibitive page download costs or keypads (touch or otherwise) that are difficult to use.

The challenge of content development for mobile devices is arguably more difficult than for desktop given the variety and disparity of mobile devices available today, and their support capability, plus the fact that handsets change and are upgraded frequently. The last point can work in favour of the web page author, as the expectation is that mobile devices will improve quickly. However, with the global market in mind, many people in developing countries do not upgrade mobile devices as frequently as in the developed world.

Mobile web versus full web

Given the issues outlined above, there is much debate about whether the full web can exist on mobile devices. This is also referred to as 'one web'. If a recent article in CNN is to be believed, the advent of mobile access is spelling the end of the web as we know it so it is no longer the 'internet' but the 'splinternet':

For many years, the Internet was relatively simple: Everyone surfed the same Web. Fast forward to 2010 and the idea of a one-size-fits-all Web is a quaint memory, thanks to the rise of the iPhone, Kindle, BlackBerry, Droid and of course, the much-hyped iPad.¹⁰

The issue is that the current trend is to build websites and apps that work for specific platforms using specific technologies. This very much reflects the problems on desktop ten years ago, and as we saw then, this was a major obstacle in terms of developing and evolving the web.

Combined with this is the concern that the full web cannot be realised on devices owing to hardware limitations, context, display space and technology supported. The recent launch of Apple's iPad has sparked much debate, as Apple does not support Flash on iPad (nor does it on iPhone)¹¹.

Given the positive reaction to the launch of iPad, the fact that it cannot support the 'full web' is significant. However, the response to those that lay claim to the full web on mobile devices being a myth in itself, is that if standards and best practices are followed then 'one web' should remain the goal.

The next few sections look at some of the standards and best practices that we can follow to ensure that we, as users, can enjoy not only the full web on mobile devices but also an accessible one.

Progressive enhancement

'Progressive enhancement' is an umbrella term that first came about when web technologies on the desktop had differing levels of support by desktop browsers and assistive technologies, depending on their functionality.

These varying levels of support meant that web content developers could never be sure that any given user on any combination of platform, browser or assistive technology could access their content. The solution therefore was to build a basic

¹⁰ CNN, 'End of the Internet as we know it' [http://money.cnn.com/2010/02/03/technology/Web_splintering/index.htm?postversion=2010020313]

¹¹ Jeffrey Zeldman 'Flash, iPad and standards' [<http://www.zeldman.com/2010/02/01/flash-ipad-standards>]

version of a site where all content and functionality could be accessed using HTML then overlaid with JavaScript, advanced CSS and other technologies for those browsers, assistive technologies and devices that could handle them.

A simple example is to have a form which validates server-side when a user clicks 'submit' and returns information about any errors in a fresh page (should there be any) for those that do not have access to JavaScript. Users with JavaScript, however, are able to validate forms client-side, making pages faster to use. The aim is to have one website that works regardless of what technologies your browser, device or assistive technology supports.

Some usability commentators, such as Jakob Nielsen¹², suggest building 'special mobile versions' of sites as the best way to accommodate for varying technology supported by mobile devices. This leads us to a similar situation to that which we saw on the desktop ten years ago, where various versions of websites appeared – such as text-only and graphical versions – with the former intended for users with visual impairment. This effectively compartmentalised disabled users and, had the practice remained, would have left them behind entirely as the web evolved.

The website CSS Zen Garden [<http://www.csszengarden.com>] is an interesting example of how CSS can be used for progressive enhancement. It is built using HTML and then has multiple different CSS assigned, giving each version a unique design and layout.

The image on the following page shows HTML text using web fonts, text-shadow, border-radius, box-shadow, transform and a transition so you can see the appearance changes as you progress through different versions of Opera desktop and with it what each version can handle. As you can see the text is accessible and readable throughout but more decorative and designed as the browser becomes more sophisticated. While displayed in Opera, note that the concept works across different browsers, desktop or mobile.

Progressive enhancement on mobile web content is the obvious way to ensure that no user is left behind while technology is given the freedom to innovate and develop.

¹² Jakob Nielsen, 'Mobile Web 2009 = Desktop Web 1998' [<http://www.useit.com/alertbox/mobile-2009.html>]

PHOTO REDACTED DUE TO THIRD PARTY RIGHTS OR OTHER LEGAL ISSUES

(Image by kind permission of Daniel Davis [<http://people.opera.com/danield/css3/freebeer/>])

Media types and media queries

One example of progressive enhancement is how CSS can be used to tailor HTML content to suit the particular needs of the mobile platform and/or browser being used. The hundreds of mobile devices available today, with their varying viewport sizes and support for fonts, colours and layouts, make developing accessible content that is readable across various mobile platforms and browsers difficult.

Using CSS 2 Media Types¹³ and CSS 3 Media Queries¹⁴ a developer can tell a mobile device what CSS to use to render fonts, colour, layout and so on, based on the mobile browser and platform capabilities. This allows the developer to use more advanced features for more advanced devices without ruining the experience for users on less sophisticated devices. Just one source of HTML content is needed and the concept of 'one web' prevails over designing for multiple sites.

The benefit to accessibility is that all content and functionality is available on whatever browser or device you are using in combination with any number of

¹³ W3C Media types [<http://www.w3.org/TR/CSS2/media.html>]

¹⁴ W3C Media Queries [<http://www.w3.org/TR/css3-mediaqueries/#5D>]

assistive technologies. While this concept has formed the basis of accessible web design and universal access on the desktop, it is of such vital importance that it is core to the development of mobile web access.

HTML5

HTML5 is the much anticipated update to HTML 4.01 which is now over ten years old. In that time the web has evolved exponentially to become more focused on user-generated content, video, web applications, social networking – and mobile access, a very different landscape today than when HTML 4.01 was published.

HTML5 aims to define the existing language in more detail as well as extending it to better support web applications. A positive by-product of this is that mobile web access should also benefit.

Consider video on the web, for example. Currently video is rendered using proprietary technologies or plug-ins such as Flash, QuickTime and RealPlayer. All of these are non-standard and have certain limitations in that they may be inaccessible, available in only some browsers, or difficult to install. On mobile devices this is an even bigger issue.

HTML5 introduces a new element: `<video>`¹⁵ which allows video content to be embedded directly within the page without relying on proprietary technologies or plug-ins. Native support in HTML for video means that developers will no longer need to worry about what technology the desktop or mobile browser supports, while users will benefit from easier access and standardisation across browsers. Another benefit for mobile accessibility resulting from HTML5 is Web Forms¹⁶. Error handling, date pickers, auto-focus and more have all traditionally been done using JavaScript. The issue here is that not all users – on either the desktop or mobile devices – can access, or have support for, JavaScript. Web Forms deliver this functionality via HTML5 so that mobile devices do not have to have support for JavaScript. In addition to better accessibility and usability, users will be able to enjoy a degree of consistency in how forms work across desktop and mobile browsers.

¹⁵ HTML5 `<video>` element [<http://www.whatwg.org/specs/web-apps/current-work/multipage/video.html>]

¹⁶ HTML5 Web Forms [<http://www.whatwg.org/specs/web-apps/current-work/multipage/forms.html> - forms]

PHOTO REDACTED DUE TO THIRD PARTY RIGHTS OR OTHER LEGAL ISSUES

How an HTML5 datepicker renders in Opera. (Image by kind permission of Bruce Lawson [<http://www.brucelawson.co.uk/2009/redesigning-with-html-5-wai-aria/>])

Overall the advent of HTML5 has the potential to improve access beyond just the desktop and mobile. As HTML5 is a web standard, games consoles (such as Nintendo Wii and DSi) and other hand-held devices will also provide better accessibility and usability.

HTML5 is still a work in progress, so web authors should check which browsers have support for new HTML5 elements.

Accessible websites help mobile optimisation

A key factor underlying mobile accessibility is that content that has been built with accessibility in mind for the desktop will also be optimised for mobile devices.

The World Wide Web Consortium (W3C) has published resources highlighting the relationship between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG)¹⁷. The former provides guidance on how to optimise content for mobile and the latter on optimising content for accessibility on desktops based on common issues encountered by users with disabilities.

Key areas include:

Zooming – scaling text and images to suit user needs in various viewport sizes can make or break a website whether on the desktop or on mobile devices. In many ways it is the job of the browser to provide tools and features to allow users to customise content. Most modern desktop browsers have a facility where the user

¹⁷ *Relationship between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG)* [<http://www.w3.org/TR/mwbp-wcag/>]

can scale full pages, as do modern mobile browsers such as Mobile Safari, Opera Mobile, Opera Mini and Internet Explorer for Mobile. Older mobile browsers do not allow scaling, however, so it is the responsibility of the page author to ensure that page content remains legible, uncluttered and readable by making good use of colour, fonts and images.

Progressive enhancement – in combination with media types and media queries as described above.

Colour, images and fonts – progressive enhancement in combination with media types and media queries can help page authors control how content is rendered on different devices with different capabilities. The Mobile Web Best Practices recommend a Default Delivery Context¹⁸ which is essentially a baseline of what low-end devices can support in terms of colours, images, fonts and more. This is a useful foundation that can be used for basic styles that can then be enhanced via media queries for more sophisticated devices.

Keyboard access – by following best practices on desktop, all elements must be focusable via the keyboard, in a logical tab order and preferably with a focus outline. If the right measures have been taken to do this, content should be keyboard accessible on a hand-held device – which is essential.

JavaScript and plug-in support – variable support for JavaScript and plug-ins by access technologies and some browsers creates issues across desktop and mobile devices. For example, text-based browsers such as Lynx raise these issues. HTML5 will eventually lower some of the barriers while progressive enhancement ensures that today's content remains accessible. An accessible site for desktop means you are building web pages using web standards which by definition should work across all platforms and devices.

The way of the widget

Sometimes, no matter what, it is extremely difficult to streamline web content so that pages are not cluttered and unusable on small screens. One important consideration is that a large part of a page's content that may be relevant on a desktop site may not be as useful when viewing in context on mobile devices. In cases like this, widgets, or web apps, are a good alternative.

Let's look at an example of a train timetable website. On the desktop you may be interested to see links to travel updates, places of interest, hotel information, seasonal deals and so on. On mobile devices, when you are out and about and have a need for a specific piece of information at a specific time (not to mention restricted due to browsing costs) you may only want to know about train times and travel

¹⁸ Mobile Web Best Practices, Default Delivery Context [<http://www.w3.org/TR/mobile-bp/#ddc>]

updates.

A widget is a trimmed-down version of a web page with only key information that is stand-alone outside the mobile, or desktop, browser. The benefits here are that you can get asynchronous updates, faster and simpler access to information.

Most people know widgets as web apps that you get on the iPhone. While these do indeed serve a purpose, they are restricted in that they are built using proprietary technologies rather than web standards, so web apps for iPhone can only work on iPhone. If they were built using web standards, however, they could be used on many more mobile devices, not to mention desktop platforms as well as TVs and games consoles. As we have already touched upon, web standards from W3C are designed to be accessible so the likelihood of all users being able to access them is higher too.

W3C widgets¹⁹ and Opera widgets²⁰ are built using web standards such as HTML, CSS and JavaScript, making them both cross-platform compatible and accessible should you choose to build them with both WCAG and MWBP in mind.

Conclusion

While we may today be at a crossroads where mobile web development is as underdeveloped and precarious as desktop web development ten years ago, we're fortunate in that the web has been around long enough for us to be able to learn from mistakes, and successes, from early on in its history.

From this we can see that the key to safeguarding the accessibility and usability of mobile web access is to ensure that web standards are used over proprietary technologies together with progressive enhancement, and recognise that guidelines such as WCAG and MWBP go a long way to supporting both, and will produce an accessible and mobile web that is both robust and future-proofed.

If we do the above we ensure that we avoid the mistakes of the desktop in 1999 when the browser wars were at their height and developers and users alike were forced to take sides. By working towards 'one web' and the 'full web', we ensure that our web content is available on multiple devices and platforms as we find ourselves browsing, networking and working away from the desktop using different technologies. If we fail to do this then the web could suffer the same fate as the Tower of Babel. The web today is too valuable a commodity to working, commerce, education and society to allow this to happen by letting it be owned by one organisation or driven by one browser.

¹⁹ W3C Widgets [<http://www.w3.org/TR/widgets/>]

²⁰ Opera Widgets [<http://widgets.opera.com/>]

Ultimately it's the end user (you and I) who could lose out, and disabled users in particular. By using web standards we reduce the risk of leaving disabled, older and global users behind, and allow the widest possible usage of the web as it becomes an essential technology supporting communication today – on desktop, mobile and other alternative devices.

It is essential that we build on the foundation that the web standards movement has created to ensure a more open web, as mobile access increasingly features in how we access the web on a day-to-day basis. As Jeffrey Zeldman said: “Any website build without web standards in its DNA has a limited shelf life.”²¹

²¹ The .Net Awards (PDF) [http://www.thenetawards.com/NET184.f_awards.pdf]

Glossary

Auto-focus When an element, such as the first field in a form, automatically receives focus without the user interacting with the page.

Canvas ACD Canvas [<https://developer.mozilla.org/en/HTML/Canvas>] is an element of the HTML language which can be used to draw graphics using scripting (usually Javascript). It can, for example, be used to draw graphs, make photo compositions or create animations.

CSS Cascading Style Sheets enable the separation of document content (written in HTML or a similar mark-up language) from document presentation, including elements such as the layout, colours and fonts. Its most common application is to style web pages written in HTML and XHTML, but the language can be applied to any kind of XML document.

Date pickers Found in forms, date-pickers are pop-up calendars that allow the user to select a day of the month that in turn gets fed into a form element as text.

Focus outline When an element receives focus there is a visual indicator that the element has focus. For example, in Opera elements are highlighted with a blue line.

Focusable When a page element (image, link, form element or button) is highlighted via the mouse or via the keyboard, it has received focus and is therefore focusable.

HTML HyperText Markup Language is the main mark-up language for web pages. It provides a means to create structured documents using structural semantics for text such as headings, paragraphs, lists etc. as well as for links, quotes, and other items. It enables images and objects to be embedded and can be used to create interactive forms.

Media types Style sheets for different media types may share a property, but require different values for that property. For example, the 'font-size' property is used both for desktop and mobile but may need different sizes defined. Therefore, it is necessary to express that a style sheet, or a section of a style sheet, applies to certain media types. This is a key part of progressive enhancement.

Media queries Media queries extend the functionality of media types by allowing more precise labelling of style sheets. Features that can be used in media queries are 'width', 'height', and 'colour'. This is particularly useful when designing for 'one web' and ensuring content renders well on both a desktop and mobile device regardless of their capability in terms of support for fonts, colours and styles.

MWBP Mobile Web Best Practices is a W3C document which specifies best practices for delivering web content to mobile devices [<http://www.w3.org/TR/mobile-bp/>].

Native support A feature that already exists in the environment in which an

application is run or that works out of the box, without the need to install extensions or plug-ins.

One web Making, as far as is reasonable, the same information and services available to users irrespective of the device they are using, their ability or disability.

Progressive enhancement This uses web technologies in a layered fashion that enables everyone to access the basic content and functionality of a web page, using any browser or internet connection, while also providing those with better bandwidth or more advanced browser software an enhanced version of the page.

Render How images, colours, font and other decorative features are displayed on screen. Different types of device will typically have different capabilities when it comes to rendering content.

Viewport The viewport size is the size of the screen on any given device.

W3C (World Wide Web Consortium) The World Wide Web Consortium (W3C) is an international community where member organisations, a full-time staff, and the public work together to develop web standards.

WAP A wireless access point (WAP) is a device that enables wireless communication devices to connect to a wireless network using Wi-Fi, Bluetooth or related standards such as GPRS (General packet radio service). GPRS is more advanced than WAP, which can only be found on first-generation mobile phones and did not handle mobile browsing well.

WCAG Web Content Accessibility Guidelines, produced by W3C, cover a wide range of recommendations for making web content more accessible [<http://www.w3.org/TR/WCAG20/>].

Web application/app – see **Widget**.

Widget Also known as Web Apps (Applications), widgets are stand-alone chunks of code that can be installed and executed in various devices. They sit apart from the browser and are typically made up of HTML, CSS and JavaScript as well as some proprietary technologies.