DEVELOPMENT AWARD GUIDELINE

You will notice that the judging system is divided into 6 categories, each one with different relative weights:

1. **WPO** *0.20*
2. **RWD / Mobile** *0.20*
3. **Markup / Metadata** *0.15*
4. **Semantics / SEO** *0.20*
5. **Animations / Transitions** *0.15*
6. **Accessibility** *0.10*

Below you can find a guideline / checklist for each category, and a list of recommended Monitoring Tools and Metrics. Be aware that categories and their contents are not in order of relevance or importance.

Being an open and ongoing document, suggestions and contributions are always welcome! Don't hesitate to drop an email to oscar@awwwards.com

1. WEB PERFORMANCE OPTIMIZATION

"First and foremost, speed is more than a feature. Speed is the most important feature. If your application is slow, people won't use it." Fred Wilson

"WPO is similar to SEO in that optimizing web performance drives more traffic to your web site. But WPO doesn't stop there. WPO also improves the user experience, increases revenue, and reduces operating costs" Steve Souders

GOALS

- Best practices and techniques implemented to improve website performance and display contents faster to the end-user, as well as save bandwidth and energy power.
- Optimized content efficiency / data delivery and improved server response time.
- Proper render blocking resources management (minification, deferring scripts, inlining, total HTTP requests)
- Balance between number-based-metrics vs. user metrics and perceived performance.

CHECKLIST

- Enable compression / Gzip.
- Critical rendering path, fetching, parsing and loading order of resources.
- Render-blocking JS, defer parsing, asynchronous scripts, delayed script loading.
- Prioritize visible / above-the-fold content.
- Minify CSS and JavaScript.
- Optimized, compressed and deferred media (image formats, sprite images, videos).
- Use of CDN.
- Cache layers (leverage browser caching, CDN, web server and app server).
- Expires or Cache-Control Header.
- Reduce DNS Lookups.
- Fewer Redirects.
- Fewer third-party plugins.
- No iframes.
- Avoid filters.
- Responsive images. Scale, formats, compression, lazy/deferred loading.
- Inline scripts, styles and images (SVG, Data URI).
- Progressive enhancement / graceful degradation.
- Stylesheets at Top / Scripts at Bottom.
- Conditional loading, Lazy loading.

- No console logs or error messages.
- Rendering performance.
- HTTPS and HTTP/2

RELEVANT METRICS

- Page Load Time and Above The Fold Time.
- Total HTTP requests.
- Overall asset count and weight.
- Media count and weight (Images, Videos, Embeds, iFrames, Objects).
- Total domains.
- Total third-party dependencies (analytics, retargeting, ads, social widgets, frameworks, libraries, plugins, web services).

2. RWD / MOBILE

GOALS

- Consistent user experience across different app modules and widgets, regardless of the user's screen size or features supported on their device.
- Clear and uniformed website design principles, implementation, and appeal across devices.
- Critical content and features accessible across different types of user input methods (e.g. click, swipe, pinch).

CHECKLIST

- Flexible Grid based layout / Liquid UI.
- Flexible Media (Images, Videos, Embeds, iFrames, Objects).
- Responsive images and media optimization.
- Consistency through media queries / breakpoints across devices.
- Performance (speed, loading time, etc).

- Readability and responsive typography.
- Conditional loading / Lazy loading.
- Progressive Enhancement / Graceful degradation.
- Viewport Meta Tag and other <head> related information.
- Feat-detection / UA sniffing.
- Retina and Pixel-density.
- Proper user input.
- Appropriate tap targets size.
- RWD navigation patterns.

3. MARKUP / METADATA

GOALS

- Solid HTML's sectioning model, appropriate use of its elements and their structural and semantic meanings.
- Use of the most meaningful yet minimal markup required to present the styles and needed interaction.
- Right information hierarchy of contents so users, browsers and search engines can more accurately understand and interact with it.

CHECKLIST

- Valid HTML Markup.
- Doctype, language and character encoding.
- Document elements and global attributes (head, title, lang, base, link, meta, style, translate).
- Standard Metadata (description, keywords/concepts, expires, generation, language, viewport, author, content-type, mobile-web-app, etc).
- Structured Metadata Markup and Microdata.
- OpenGraph, Schema.org, Twitter Cards, Pinterest and Social integration.
- Images with captions, alt, width, height and src.

- Document structure / organization / cohesion.
- Readability, indentation, use of whitespace and comments.
- Descriptive variable names and semantic naming conventions.
- No div-itis, class-itis, span-mania, tag-soups.
- Favicon.ico at different sizes and cacheable.

TIP

- Isolate website contents and try to project mentally its markup and semantics. By considering content first, you are already getting yourself into a structural and semantic mindset.

4. SEMANTICS / SEO

"Besides SEO, another benefit of semantic markup is that it provides the foundation for an Accessible site" *Sarah Lewis*

GOALS

- Proper information hierarchy of contents so users, browsers and search engines can more accurately understand and interact with it.
- Solid HTML's sectioning model, appropriate use of its elements and their structural and semantic meanings.
- Best practices and techniques implemented in order to help Google and other crawlers to find, index, rank and display the web site and its contents.

CHECKLIST

- Meta-tags: unique, accurate, semantic, brief but descriptive meta-tag definitions, per page (title, description, keywords, expires, language, author, publisher, etc.)
- Improved URL structure (hyphens, lowercase, human-readable, brief but descriptive, use of keywords, no dynamic parameters, etc).

- Performance and loading times.
- Images: "alt" attribute describing image-related information (brief, but descriptive filenames and alt text).
- Suitable anchor text.
- Canonicalization.
- Structured Metadata Markup and Microdata.
- Social Meta Tags / Schema.org (itemscope, itemprop -name, description, image-)
- Social Meta Tags / Open Graph protocol (og:title, og:type, og:image, og:url, og:description, fb:admins; twitter:card,).
- RDFa, Microdata, and Microformat Data Sets
- Language declaration in the HTML element.
- Long Word Count.
- HTML5 Semantic Markup (nav, page, section, footer, etc).
- HTML5 Semantic Headings (h1, h2 ... p, etc) to emphasize relevance.
- HTML Sitemap page on the site.
- XML Sitemap for search engine robots.
- Robots.txt / Restrict crawling: exclusion standards (nofollow, noindex, etc) and most relevant User Agents.
- Rel="next" and rel="prev" for pagination.
- Rel=”alternate” href=”x”.
- Quality experience (speed, reputable, spelling errors, broken links).
- Quality content (mobile-friendly, unique, original, relevant, text-based content).
- Use of text for navigation links and contents.
- Avoid duplicate content (canonicalization or 301 redirect).
- Avoid redirects or use them properly (301, 307, etc)
- Minimize 404 (use 301 redirect).
- Proper Hyperlinks (HTML links over JS, “nofollow” for untrusted or paid content, etc).

ANIMATION / TRANSITIONS

“Animations are most useful when they reflect and reinforce the semantic relationships between elements: for example, ‘this comment belongs to this article’ or ‘these menu items are part of this menu’. You can’t treat individual screens as separate entities: to the user, the entire experience is one continuous space.” Amin Al Hazwani

GOALS

- A consistent animation strategy and fluid execution through all website pages and user-input-interactions, like buttons on click/tap/hover, image sliders, transitions between sections, etc.
- An animation approach that enhances the user experience by telling a story, providing visual feedback and info to users on what just happened (eg. on click, hover, success or failed form submission etc) or what to do next (eg. pre-loaders).

** Please consider only real-time scripting animation, not video or image-sequences.*

CHECKLIST

- Smooth, fast and stable frame-rate.
- Easings, timings, rhythm, appeal.
- Balance and consistency: avoid animations being too slow, too fast, too often, too different.
- 2D / 3D animations and layouts.
- Math, physics and dynamic/random generation techniques.
- Animation libraries / frameworks.
- Garbage collector, GLSL optimization, GPU testing, assets management, memory leaks.
- Parallax scrolling implementation.
- CSS / JS animations.

LINEAR, INTERACTIVE AND TRANSITIONAL

- Linear Animation: keyframe animations that run without user interaction or triggered by simple user interaction -onscroll, onclick, etc-.
- Interactive Animation: dynamic animations based on user interaction -mouse, keyboard, navigation path, motion detection, etc.- or other variables based on complex scriptings including Maths, Physics, Device Features, Context, etc.

- Transition Animation: animations between sections or main website pages. Normally involves a wide use of site related features: AJAX, URL history, a friendly “Loading UI” implementation, etc.

ANIMATION TYPES

- Procedural Animation: classic movement over time.
- Representational Animation: objects changing shape during the animation, like character animation.
- Stochastic Animation: processes to control groups of objects, such as particle systems, like a fireworks animation, Secondary action.
- Behavioral Animation / AI: objects or "actors" are given rules and variables about how they react to their environment.

LOW LEVEL AND HIGH LEVEL TECHNIQUES

- Low level Techniques: techniques that aid the animator in precisely specifying motion, such as shape interpolation algorithms (in-betweening), involving a fairly specific idea of the exact motion that he or she wants.
- High level: typically algorithms or models used to generate a motion using a set of rules or constraints. The animator sets up the rules of the model, or chooses an appropriate algorithm, and selects initial values or boundary values. The system is then set into motion and the motion of the objects is controlled by the algorithm or model.

ACCESSIBILITY

“It may be useful to think of search engines as users with substantial constraints: they can’t read text in images, can’t interpret JavaScript or applets, and can’t “view” many other kinds of multimedia content. These are the types of problems that accessibility is supposed to solve in the first place.” Andy Hagans

GOALS

- Confirm that web site / application is accessible for all kind of users, including those ones under technical, physical, or other constraints, and those on alternative platforms.
- Semantic information about widgets, structures, and behaviors, in order to allow assistive technologies (screen reader software, voice interactive software, Braille output devices, closed captioning, etc) to convey appropriate information to persons with disabilities.
- Custom interactive controls / widgets should be operable through keyboard and meet the WCAG 2.0 [WCAG20] criteria: “All functionality of the content is operable through a keyboard interface”.

CHECKLIST

- Supported user input methods (keyboard / mouse / touch navigation support).
- Labeled interactive elements.
- Viewport configuration.
- Appropriate tap targets size.
- Focus / active management.
- Tabbing order (tabindex on focusable elements).
- Wrap buttons and <a> within one large single clicking area (no multiple <a> for the same link or interaction)
- Legible font sizes.
- Colour contrast text / background.
- No relevant information only on hover.
- Text alternatives and labels for any non-text elements.
- Videos with audio: provide captions, full text transcription and keyboard controls.
- aria-hidden="true" for non interactive or decorative elements.
- Browser history (next / prev / refresh).
- WAI-WCAG Design Principles: Perceivable, Operable, Understandable, Robust (http://bitsofco.de/2015/the-accessibility-cheatsheet/).
- WAI-ARIA Landmark Roles (banner, navigation, search, main, etc).
- WAI-ARIA Widget Roles (progressbar, slider, button, tree, textfield, etc).
- WAI-ARIA Abstract Roles (input, section, widget, select, etc).
- WAI-ARIA Live Regions Roles (alert, log, timer, etc).
- WAI-ARIA Document Structure Roles (article, figure, list, img, text, etc).

- WAI-ARIA Labels on custom elements (https://www.w3.org/TR/wai-aria-1.1/#dfn-accessible-name)
- WAI-ARIA States and Properties (https://w3c.github.io/aria/aria/aria.html#states_and_properties).
- Language declaration.
- Pause controls for any animation that starts automatically and plays for more than 5 seconds. Auto-updating content and animated carousels could likely fall under this category.
- Content accessible with no CSS.
- Content accessible with no JS.

CUSTOM CONTROL ACCESSIBLE DEVELOPMENT CHECKLIST:

- Keyboard operable - Can you use the control with the keyboard? Keyboard Navigation
- Touch operable - Can you use the control with touch gestures? With assistive technology enabled?
- Expected operation - Can you operate the control using the standard keys and/or touch gestures for the control type? ARIA Widget Design Patterns
- Focusable - Can you get to the control via the keyboard? Providing Keyboard Focus
- Clear indication of focus - Can you easily see it when the control has focus? WCAG2 Visible Focus
- Label - The control has a text label that is exposed as an accessible name in accessibility APIs
- Role - The control has an appropriate role exposed in accessibility APIs
- States and properties - The control has any UI states and properties that it has exposed in accessibility APIs
- Color contrast - The control label/description/icon is perceivable/usable for low vision users (Use a color contrast checker.)
- High contrast mode - The control is perceivable/usable when High Contrast Mode is enabled (e.g. Windows HC mode)

TIPS

- High accessibility overlaps heavily with effective white-hat SEO.
- Keep in mind some of the disabilities that might keep someone from enjoying your full customer experience, like color blindness, blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, vestibular disorder, limited physical dexterity-such as the inability to use a keyboard or mouse, and more.

APPENDIX. MONITORING TOOLS / METRICS

"If you can't measure it, you can't improve it" H. James Harrington

PRIMARY TOOLS

- Page Speed https://developers.google.com/speed/pagespeed/insights
- YSlow http://developer.yahoo.com/yslow/
- Website Test http://websitetest.com/
- Pingdom http://tools.pingdom.com/fpt/#!/mjmicGfYX/
- Web Page Test http://www.webpagetest.org/compare
- GT Metrix http://gtmetrix.com/
- W3C Validator https://validator.w3.org/
- Web Page Test Mobile http://www.webpagetest.org/mobile

SECONDARY TOOLS AND RESOURCES

- LoadsIn http://loads.in/
- SiteSpeed.io http://www.sitespeed.io/
- SpeedCurve http://speedcurve.com/
- Browser Shots http://browsershots.org/
- Responsive Px http://responsivepx.com/
- Varvy SEO https://varvy.com/
- Varvy Tools https://varvy.com/tools/
- Varvy gWebGuidelines https://varvy.com/
- Varvy PageSpeed https://varvy.com/pagespeed/
- W3C CSS Validator http://jigsaw.w3.org/css-validator/
- 3PO http://www.phpied.com/3po/
- Souders Episodes http://stevesouders.com/episodes2/
- Woorank https://www.woorank.com/
- HTML5 Outliner https://gsnedders.html5.org/outliner/
- SEO Browser http://www.seo-browser.com/

- Wave Accessibility Tool http://wave.webaim.org/
- Prettydif https://github.com/prettydiff/a11y-tools
- Social - Twitter Validation Tool https://cards-dev.twitter.com/validator
- Social - Facebook OG Debugger https://developers.facebook.com/tools/debug/
- Social - Google Structured Data Testing Tool

http://www.google.com/webmasters/tools/richsnippets

- Social - Pinterest Rich Pins Validator - http://developers.pinterest.com/rich_pins/validator/
- Security Headers https://securityheaders.io/
- Structured Data Markup Helper https://www.google.com/webmasters/markup-helper/u/0/
- http://www.real-user-monitoring.com/the-complete-list-of-end-user-experience-monitoring-tools/
- http://sixrevisions.com/tools/10-excellent-tools-for-testing-your-site-on-mobile-devices/
- http://www.strangeloopnetworks.com/web-performance-optimization-hub/topics/tools-and-analytics/
- https://webmaster.yandex.com/

CHROME EXTENSIONS

- Wave Accessibility Tool

https://chrome.google.com/webstore/detail/wave-evaluation-tool/jbbplnpkjmmeebjpijfedlgcdilocofh

- Google Accessibility Dev Tools

https://chrome.google.com/webstore/detail/accessibility-developer-t/fpkknkljclfencbdbgkenhalefipecmb

- Axe https://chrome.google.com/webstore/detail/axe/lhdoppojpmngadmnindnejefpokejbdd
- Color Contrast Analyzer

https://chrome.google.com/webstore/detail/color-contrast-analyzer/dagdlcijhfbmgkjokkjicnnfimlebcll/related

Revision dates:

- v0.7 - 14/12/2016

Authors:

- Luciano Borromei - @alectortoatoa
- Jessica Travieso - @jessicatv

- Alexandre R. Gomes - @alexrgomes
- Greg Matyszewski - @gregmatys