

Device Independence:  
Responsive Design Strategies for Digital Publications

by

Champagne Choquer  
B.A., University of British Columbia, 2008

*Project Report Submitted in Partial Fulfillment  
of the Requirements for the Degree of*

MASTER OF PUBLISHING

in the

Publishing Program

Faculty of Communication, Art and Technology

© Champagne Choquer, 2012  
SIMON FRASER UNIVERSITY  
Summer 2012

*All rights reserved. However, in accordance with the Copyright Act of Canada, this work may be reproduced, without authorization, under the conditions for "Fair Dealing." Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.*

# APPROVAL

NAME: **Champagne Choquer**

DEGREE: **Master of Publishing**

TITLE OF

PROJECT: **Device Independence:  
Responsive Design Strategies for Digital Publications**

SUPERVISORY

COMMITTEE:

---

**John Maxwell**

*Senior Supervisor*

*Assistant Professor*

*Publishing Program*

---

**Roberto Dosil**

*Supervisor*

Senior Lecturer

Publishing Program

---

**Carolyn McNeillie**

*Industry Supervisor*

Program Developer

eBOUND Canada

Toronto, Ontario

DATE APPROVED: \_\_\_\_\_

## Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <[www.lib.sfu.ca](http://www.lib.sfu.ca)> at: <<http://ir.lib.sfu.ca/handle/1892/112>>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library  
Burnaby, BC, Canada

## ABSTRACT

This report discusses and provides strategies for implementing responsive web design solutions for digital publications in HTML-based formats, with the aid of open web standards, media queries, device-specific CSS, and relative measurements. As technological innovations create more diverse reading experiences, presenting content in a way that is dynamic and suitable to each device and user's preference becomes a larger priority in ensuring reader satisfaction and engagement. This report outlines methods for creating digital publications that adapt to the reading environment of a user by embracing device independence principles and open web standards, thereby extending the shelf life of a publication's design and source files, and maximizing publisher investment in digital conversions. The report provides practical methods for implementing responsive design in the form of elastic layouts, built on the relative measurement of the em, which allows the optimization of layouts for display on handheld devices, tablets, and personal computers.

**Keywords:** *responsive web design; publishing; elastic layouts; HTML; CSS; open web standards*

*for my grandmother, Donnie Choquer.  
A woman who, in her seventies, owned a  
Mac, a PC, and an iPod and was always eager  
to learn more about technology.*

## ACKNOWLEDGEMENTS

Thanks to Carolyn McNeillie, my industry supervisor, for providing me with constructive and enlightening feedback pertaining to the needs of the publishing industry.

Thanks to Roberto Dosil for his interest, thoughtful feedback, and passion for design, all of which have inspired me throughout my time in the Publishing Program.

Thanks to Jesse Marchand of WhiteCap Books for her insight during the writing process.

Special thanks to the many developers and designers whose innovative writings and contributions helped shape this report. In particular, thanks to Chris Armstrong (and everyone at Front), Haig Armen, Todd Sieling, Jon Tan, John Allsopp, and Ethan Marcotte.

Many thanks to my senior supervisor John Maxwell, whose curiosity, enthusiasm, and guidance has propelled me in my research and writing of this report, and whose supervision afforded me the opportunity to be extra ambitious.

Thank you to my mother and sisters for their continued love and support.

Finally (and most importantly) thanks to Eric — you're pretty ok, too.

# CONTENTS

II	APPROVAL
III	COPYRIGHT LICENCE
IV	ABSTRACT
V	DEDICATION
VI	ACKNOWLEDGEMENTS
VIII	LIST OF FIGURES
<b>1</b>	<b>Introduction</b>
<b>3</b>	<b>Chapter 1   Overview of Responsive Design for Publishers</b>
<b>5</b>	<b>Chapter 2   Borrowing from the Open Web: Responsive Design and Usability</b>
6	<i>Designing Goal-Oriented Artifacts</i>
7	<i>What is Responsive Web Design?</i>
9	<i>Designing Functionality</i>
10	<i>Responsive Publishing</i>
<b>13</b>	<b>Chapter 3   Becoming Responsive: Implementation of Device Independent Practices</b>
14	<i>Aiming For Device Independence</i>
15	<i>Open Web Standards vs. Proprietary Coding</i>
18	<i>How to Implement Responsive Design</i>
<b>24</b>	<b>Chapter 4   The Elastic Approach to Responsive Design</b>
25	<i>The Case for Elastic</i>
27	<i>Delivering Accessibility: Elastic Layouts and the End User</i>
30	<i>How to Calculate Elastic Layouts</i>
<b>33</b>	<b>Case Study   Fontasizing.com and the Goldilocks Approach to Elastic Layouts</b>
33	<i>Implementing Responsive Design</i>
41	<i>Conceptualizing the Page</i>
48	<b>Conclusion</b>
51	BIBLIOGRAPHY

## LIST OF FIGURES

PAGES	FIGURES	NAME OF FIGURE
27	Figure 1	The Golden Mean Ratio
37	Figure 2	Mobile (Small Screen) Elastic Layout
39	Figure 3	Tablet (Medium Screen) Elastic Layout
40	Figure 4	Personal Computer (Large Screen) Elastic Layout
42	Figure 5	Case Study Typeface Samples
43	Figure 6	Case Study Baseline Ratio
46	Figure 7	SVG Image Sample

## INTRODUCTION

As the predominant format shifts from print to digital, presenting publications that are dynamic and respond to changing reading environments becomes a larger priority for publishers in ensuring reader satisfaction. The focus of this report is the practical application of responsive web design solutions to digital publications to improve the interoperability of digital reading. By adopting contemporary web design practices in conjunction with open web standards and relative measurements (as will be addressed in Chapter 3 of the report), responsive design strategies allow publishers the opportunity to create digital publications that are not limited to one device, screen size, or distribution channel, and better serve readers in terms of functionality, aesthetics, and interoperable reading experiences. The report also discusses the methods and best practices for creating HyperText Mark-up Language (HTML)-based publications that adapt to users' reading environment while embracing device independence, thus extending the shelf life of publication designs and source files, and increasing the market reach across an array of devices.

Although this report is largely written for book publishers, the material herein is applicable to all publishers of digital content. Throughout the report the term “digital publications” is used to refer to publishing formats that utilize HTML and Cascading Style Sheet (css). Due to the increasing similarities in the coding of websites and ebooks, this term refers to self-contained ebook formats, such as the EPUB (.epub), Mobipocket (.mobi), Apple iBooks (.ibook), and Kindle Fire 8 (.kf8), as well as hyperlinked websites such as online magazines, newspapers, and blogs.

The report is divided into four main sections. The first chapter serves as a brief overview of responsive design for publishers. On a contextual level, the second chapter introduces responsive web design and discusses digital publications in regards to open web usability features and Herbert A. Simon's theory for goal-oriented design. Thereafter, the section proposes a strategy for responsive publications. The third chapter discusses device independence strategies, open web standards, and the components essential to implementing responsive design, including css media queries and relative measurements. The fourth chapter is divided into two parts. The first part outlines the elastic approach to responsive design, the report's suggested layout method for applying responsive web design to digital publications. The second part uses the website [Fontasizing.com](http://Fontasizing.com)<sup>1</sup> and the

---

<sup>1</sup> [Fontasizing.com](http://Fontasizing.com) is a website created by the author that has content related to the arguments presented within this report.

## *Introduction*

application of the Goldilocks Approach framework<sup>2</sup> as a specific case study of the implementation of the elastic approach as a strategy for HTML-based publication design. The conclusion summarizes the main points of the report and evaluates responsive design based on the findings of the case study. The objective of the report is to contribute a greater understanding of web design to the publishing industry and aid publishers in the implementation of adaptive web design practices for digital publications.

---

<sup>2</sup> The Goldilocks Approach is an elastic responsive design framework released by Irish design studio Front. The framework can be found at <http://goldilocksapproach.com/>.

## *Chapter 1*

# OVERVIEW OF RESPONSIVE DESIGN FOR PUBLISHERS

With a growing number of readers opting for digital reading platforms and devices, the market is driving publishers to increase production and delivery of content in digital form. These digital publications must then accommodate the many interactive features and reading experiences available to current technology. Due to the prevalence of such Internet-enabled devices, comprised of smart phones, tablets, ereaders, and personal computers, digitally delivered content carries with it expectations from readers based on contemporary web design and usability features. To provide readers with the ability to read over a range of devices, many publishers and designers have to contend with differing device specifications and new, often proprietary, file formats in the realm of ebooks. Increasingly web and book designers realize that designing for each device or screen size is a futile pursuit and that they must adapt their designs to suit a myriad of devices, rather than one or two popular screens.

Responsive design is the practice of a page design, written primarily in HyperText Mark-up Language (HTML) and Cascading Style Sheets (CSS), adapting to the end user's reading environment by resizing and reorienting the publication to scale and display in an optimal style for the user's device constraints. Responsive design strategies are an ideal solution for publication design because they do not rely on one device or application to display content. When used with open web standards established by the World Wide Web Consortium (W3C), the organization responsible for setting guidelines for the web, responsive design allows for source files that are flexible and can be read on many devices, rather than one or two dominant readers.

For those publishers unfamiliar with the coding process, open web standards are the conventions established for web coding and are open source, meaning that anyone can use the coding including all browsers and reading applications, unlike some proprietary codes and formats released by private companies that tend to limit use to one device. Meanwhile, relative measurements are scalable and proportional to an aspect of the design. Unlike absolute units such as inches or pixels, relative measurements of percentages and ems maintain their proportions when a design is resized. By utilizing both open web standards and relative measurements, responsive design has the potential to deliver digital publications to any screen size and display content in a way that is optimal to each device.

## *Overview of Responsive Design*

Considering the predominance of the open web and the many solutions available online that could be readily adopted by publishers to make content more dynamic, such as responsive design, there exists a gap between book publishing and open web development that has yet to be adequately bridged; many publishers lack the HTML and CSS coding knowledge required to create innovative digital publications, while few web designers and developers have joined the ranks of publishers in the practical, day-to-day production of ebooks. Although some publisher will readily adopt fixed-layout EPUBS because they can relate to the fixed output (the same as print — the trusted medium), when it comes to page design that closely resembles something “web-like” trepidation tends to set in unnecessarily. Instead, publishers should embrace web design as the paradigm for future publishing. As digital publication expert Craig Mod notes, “the most pervasive digital book format is undeniably HTML. EPUB and Mobi are effectively subsets of HTML. And woven into EPUB3 is the promise of robust HTML5, CSS3 and enhanced JavaScript<sup>3</sup> capabilities.”<sup>4</sup>

It is now time for publishers to take advantage of the new medium by familiarizing themselves with the capabilities of HTML. Responsive design is a great starting point that takes advantage of the robust web coding standards available to HTML-based formats. For publishers with limited time and budgets it is in their best interest to create digital book files for a range of devices, rather than new source files each time a new device or file format is created. With responsive designs, the HTML and CSS coding can adapt to many devices including smart phones, tablets, and personal computers. Such device-agnostic practices will remain free of the constraints that normally limit source files and will remain flexible and adaptive for the life of the content. These practices also streamline the publishing process by limiting the time spent on future conversions and reformatting. Due to their dexterity and freedom from output constraints, responsive publications will have an increased shelf life, and will remain adaptable should reformatting be required. In this way responsive design strategies can maximize the value of digitization efforts by extending the longevity of source files as the responsive coding delivers publications across multiple platforms and resizes designs to fit the reader’s screen.

---

<sup>3</sup> JavaScript is a scripting language that allows for enhanced interface features and dynamic page designs. JavaScript-specific features will not be covered within this report.

<sup>4</sup> Craig Mod, “Post-Artifact Books and Publishing,” *Craigmod.com*, October 2010, [http://craig-mod.com/journal/post\\_artifact/](http://craig-mod.com/journal/post_artifact/).

## *Chapter 2*

### **BORROWING FROM THE OPEN WEB:**

#### **Responsive Design and Usability**

There is little question that the preferred medium for published content is undergoing a massive shift from print to digital. This change is directly related to the devices used to access and read digital content. According to consumer statistics company Nielsen, 36 percent of the global market currently uses smart phones and approximately 50 percent of those queried were intent on buying a tablet within the next year.<sup>5</sup> Meanwhile, the American year-over-year increase in ereader ownership from 2010 to 2011 was 100 percent.<sup>6</sup> With such proliferation of handheld devices there will be an increase in demand for digital publications. In particular, publications that can be easily accessed on smaller screens and are capable of being read on more than one device. Considering the many digital reading environments currently available and those likely to emerge in the future, digital publications must be flexible enough to perform optimally for every device available — without requiring publishers to write different coding for each ebook format for a single title. To accomplish this, publication source files require a level of adaptability and longevity built into their coding and design in much the same way websites currently do. Publication design that enables access over a range of devices, from smart phones to personal computers, will cater to readers and their preferences for consuming digital content. Such dynamic publications will broaden the market and extend the shelf life of digital publications, making the initial investment for digital conversions worthwhile for publishers.

The following section provides a theoretical foundation for functional, goal-oriented design based on the writing of Herbert A. Simon and introduces the practice of responsive web design as a concept and indicates how responsive web design can and should be applied to digital publications given its ability to provide usability features common to the open web. Meanwhile, the specific, practical implementation of responsive design techniques in terms of HTML and CSS coding will be discussed further in Chapter 3 and 4 of the report.

---

5 “Global Online Consumers and Multi-Screen Media: Today and Tomorrow,” Nielsen.com, May 2012, <http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2012-Reports/Nielsen-Multi-Screen-Media-Report-May-2012.pdf>.

6 Lee Rainie, Kathryn Zickuhr, Kristen Purcell, Mary Madden, and Joanna Brenner, “The rise of e-reading,” PewInternet.org, April 4, 2012, <http://libraries.pewinternet.org/2012/04/04/the-rise-of-e-reading/>.

## Designing Goal-Oriented Artifacts

Prior to addressing the practical reasons for designing responsive publications, it is helpful to first uncover the theoretical motivations for improved design. In the book *The Sciences of the Artificial*, Herbert A. Simon presents a theory for design based on the artifact, defined within the work as artificial, man-made objects “produced by art” rather than nature.<sup>7</sup> Artifacts, Simon suggests, are designed to accomplish a goal or fulfill a specific purpose.<sup>8</sup> Broadening the definition of design, Simon also declares that “everyone designs who devises courses of action aimed at changing existing situations into preferred ones.”<sup>9</sup> This premise proposes that design is not merely a practice of aesthetics, but instead an ongoing process of problem solving. This expanded view of design lends well to the application of responsive design to digital publishing, as it is largely concerned with improving the performance of digital publication files over a range of devices.

For Simon, a designed artifact has an inner environment and exists in conjunction with its outer environment. The inner environment is its structure, where the organization and operation of components that make up the artifact exist, whereas the outer environment consists of external factors that impose upon and interact with the object.<sup>10</sup> A successfully designed artifact must have an inner environment that is conducive to the external variables and requirements of the outer environment. Where the two environments meet is the “interface,” where goals of the artifact are achieved. The inner environment is essentially the little black box that ensures the artifact’s functionality. To use Simon’s example, if a clock is to be used on a ship, the clock’s inner environment of cogs and machinery must be immune to the buffeting of waves of the outer environment. If it is not, then the interface of the clock will not fulfill its purpose of telling time. Comparatively, for digital publications, the artifact has an inner environment of HTML and CSS coding that fulfill users’ goals of reading on multiple devices, within the outer environment of Internet-connected devices. The digital interface of these artifacts is then where the goals of reading and interactivity come to fruition.

Considering the man-made aspect of the artifact, in conjunction with Simon’s theory that designed artifacts achieve goals, human interactions play a prominent role in determining an artifact’s success. In much the same way human interactions are central to the success of digital publications; users

---

<sup>7</sup> Herbert A. Simon, *The Sciences of the Artificial* (Cambridge: MIT Press, 1996), 4.

<sup>8</sup> *Ibid.*, 5.

<sup>9</sup> *Ibid.*, 111.

<sup>10</sup> *Ibid.*, 6.

## *Borrowing from the Open Web*

must be satisfied with the functional components of the product and have the ability to read a publication on their device of choice. To accomplish this, digital publications must recreate Simon's triad and have inner environments that achieve the goals of human action present in the outer environment by way of the interface. Digital publications must then have inner environments of HTML and CSS coding that are dynamic and respond to an array of reading environments, rather than just one, to reach the goal of interoperable reading. Achieving these ends is precisely what responsive design can do by displaying publications on a range of mobile, tablet, and computer devices.

### What is Responsive Web Design?

As outlined in the 2010 seminal treatise "Responsive Web Design" for *A List Apart*, Ethan Marcotte introduces web designers to the idea of creating "responsive" design solutions. Marcotte borrows the term "responsive" from an emerging subset of architecture, "responsive architecture," that experiments with structural and environmental components that change shape, lighting, and temperatures to adapt to the number of visitors within a space via motion sensors.<sup>11</sup> Inspired by the dynamic and responsive structures of innovative architecture, Marcotte notes that similar flexibility can be built into the design of websites. Specifically, he notes that designs should be optimized to suit the device and screen size of the end user.<sup>12</sup> This translates to text and image content that will resize and reconfigure to present in a way that is optimal to each end user's device. In essence, responsive design consists of layouts that respond and cater to the "user's behaviour and environment based on the device's screen size, platform and orientation."<sup>13</sup> In practice, responsive design currently consists of a series of relative measurements, flexible images, and adoption of specific CSS directives, such as media queries, to alter a page layout to resize relative to a user's screen.<sup>14</sup>

As responsive web design was first introduced in 2010, these dynamic solutions are still in the minority when it comes to web design. When responsive

---

11 Ethan Marcotte, "Responsive Web Design," *A List Apart*, May 25, 2010, <http://www.alistapart.com/articles/responsive-web-design/>.

12 Ibid.

13 Kayla Knight, "Responsive Web Design: What It Is and How To Use It," *Smashing Magazine*, January 12, 2011, <http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>.

14 Ibid.

## *Borrowing from the Open Web*

layouts are not utilized, most sites will use fixed layouts. Unlike responsive design, fixed web design is fixed in place like print, with all elements fixed on screen by pixel placement and built to the constraints of the intended viewing environment. Websites are most commonly designed for laptop and desktop devices but mobile-only sites have also become popular in recent years. Fixed layouts are notable for the increased amount of control over the final design; however, these designs are an absolute size and dependent on certain screen resolutions to display properly. The text content does not necessarily reflow well and most images are a static size. Due to the dependence on larger screens and lack of adaptability of fixed-layout design, users viewing content on a handheld device will likely have to zoom and scroll to find information on a website, which many times results in frustration. Instead, websites and other HTML and CSS designs should automatically adjust, as responsive designs do, to the dimensions of the viewing environment.

What differentiates responsive design from fixed-layout methods is not necessarily the specific coding employed to create adaptive layouts, but the approach to design itself. Rather than have static designs limited to certain viewing environments, responsive design holds the potential for ideal output to all devices with page designs that alter to suit each platform. The technologies in place for browser and ereading applications now allow for device and screen resolution constraints to communicate with the layout, permitting designs to be optimized relative to the user's device and preferences. This allows one source file to have varied display features that can be selected for some viewing environments and ignored for others. Comparing responsive solutions to previous methods where multiple websites were created for each viewing environment, the old, fixed-layout approach appears short-sighted and costly.

To achieve dynamic page designs there are two distinct methods for adopting responsive design solutions: *elastic layouts* and *fluid layouts*. Both methods have the added advantage of optimally displaying content to a user's viewing environment. Compared to fixed layouts, fluid and elastic approaches to responsive design are dynamic because of their use of relative measurements that scale page elements according to the reading environment. Fluid designs are built on *percentages* of the screen size whereas elastic design use *ems*, a measure relative to type size, that scales according to user interface controls and screen size. Due to the dexterity of both responsive design methods, implementing either elastic or fluid layouts will extend the shelf life of a design. However, the elastic approach is well suited for digital page design due to its potential for device independence, scalability of the text in service of the reader, and measurements proportional to the text size.

## Designing Functionality

Due to the pervasiveness of the web, cloud computing, and the constant-connectivity available through a range of mobile devices, users' expectations for digital publications and long-form reading will likely follow the norms established online. Twenty years of web development predate the recent adoption of single-file digital publications, and during this time some features available online have become fixtures of the digital medium. To account for user expectations established online and how to best incorporate web trends into digital publications, it is important to note what works on the web. According to information systems scholar Jonathan Palmer, usability is closely tied to website success.<sup>15</sup> Web "usability" is essentially ensuring that an interface works well, meaning that the average person can use it without frustration.<sup>16</sup> While usability sounds basic, its importance should not be underestimated. Good usability on the web increases markers of success including traffic, site visits, and customer satisfaction.<sup>17</sup> Based on performance metrics there are specific characteristics associated with improved functionality. As Palmer explains "usability includes consistency and the ease of getting the Web site to do what the user intends to do, clarity of interaction, ease of reading, arrangement of information, speed and layout."<sup>18</sup> Publishers can transfer the characteristics of optimal websites and online usability to digital publications by implementing similar features. All of the usability criteria outlined above can be achieved with the inclusion of responsive design practices: digital publications can be optimized to each device, thus increasing the clarity of interaction, arrangement of information, ease of reading, as well as optimized layouts suited to each screen size. Once the design of the interface responds to a device's screen size, ideal presentation of content will follow. In this regard, responsive design is an invisible component that increases the usability of a publication and optimizes the design of a site or file to the constraints of a screen.

In his book on website usability, Steve Krug notes that the first law of web usability is simply "don't make me think."<sup>19</sup> While browsing online, individuals do not want to contemplate their actions and typically want information as fast as possible. Layouts that facilitate such interactions are seen as ideal. An updated

---

15 Jonathan W. Palmer, "Web Site Usability, Design, and Performance Metrics," *Information Systems Research* 13, no. 2 (2002): 163.

16 Steve Krug, *Don't Make Me Think!* (Indianapolis: New Riders, 2000), 5.

17 Jonathan W. Palmer, "Designing for Web Site Usability," *IT Systems Perspective* (2002): 102.

18 Palmer, "Web Site Usability, Design, and Performance Metrics," 153.

19 Krug, 11.

## *Borrowing from the Open Web*

version of this law, specific to mobile and tablet devices with touch screens, could very well be “don’t make me ‘pinch.’” Thinking, and thinking that requires follow up action such as pinching (the zoom feature on many touch screens), is detrimental to the reading experience due to the added time and frustration placed on the user. One way to prevent unnecessary actions and delays is to utilize design solutions that display content optimally for each device from the outset. Responsive design solutions fit content to each screen and thus improve the functional aspects of an interface including ease of reading, arrangement of information, and clarity of interaction.

### Responsive Publishing

For traditional publishers, book design evolved within the constraints of the printed page, with text and images existing within the confines of paper’s physical dimensions. New technologies have liberated content from such constraints, and there is no longer a need to implement static page designs. Perhaps due to the substantial shift in output from finite paper sizes to the expansive digital canvas, publication designers are intimidated by the many possibilities for digital output and default to popular screen dimensions. As a result there are many designers who begin with an Apple device and build a digital canvas to suit the resolution of that particular screen. This is quite obviously a myopic view, as a succession of devices will inevitably be released to take its place. Rather than design reading experiences that have a limited shelf life due to eventual device obsolescence, there is a need to build designs that will cater to the reader and last for an extended period of time. This necessitates design that will adapt to changing screen dimensions and resolutions rather than static page designs that cater to one device.<sup>20</sup> Computing technology is now at a point where devices can communicate their parameters with source files, which allows content and page designs to dynamically adapt to device constraints and user preferences. The next logical step for digital publishing is to harness these design solutions to the benefit of both readers and publishers.

Echoing Herbert A. Simon’s definition for design, user experience designer Todd Sieling says that the goal for product design is usually to “make things better.”<sup>21</sup> If the design at hand is a publication, the goal will typically be improving

---

<sup>20</sup> John Allsopp, “A Dao of Web Design,” *A List Apart*, April 7, 2010, <http://www.alistapart.com/articles/dao/>.

<sup>21</sup> Todd Sieling, “The Curation of Future Moments,” February 21, 2012, slide 48.

## *Borrowing from the Open Web*

the reading experience for readers.<sup>22</sup> This is precisely what responsive design for digital publications aims to do: if ease of reading is to be enhanced, the design of a publication has to correspond to the parameters of a device and adapt to achieve optimal layouts for long-form reading. By changing the aesthetics and layout of a page relative to the output device, the reading experience will be clear and ideal for the setting.

Publishers, however, have been slow to adopt the skill set necessary to carry out the many presentational features that are available to the open web within digital publications. To date responsive web design strategies have not been applied to single-file digital publications. Although ebooks are self-contained websites that carry many of the same presentational features as the web, publishers have not harnessed the flexibility of responsive design. Hopefully this is on the verge of changing. In October 2011, the International Digital Publishing Forum (IDPF) released the third iteration of the electronic publication format (EPUB3) that included web standards from the World Wide Web Consortium (W3C), the organization responsible for determining the conventions of the Internet. The new format included HTML5 and CSS3 as well as other standardized coding and practices from the open web. This iteration of the format incorporated many of the conventions existent on the open web to the EPUB, which now makes it possible to institute responsive designs within the format. As a result many ereaders and applications now subscribe to the format. As of May 2012, Apple iBooks can display the EPUB3 and as of July 2012 so can the Barnes and Noble Nook and Kobo for iOS.<sup>23,24</sup>

Although the biggest individual player in the digital book world, Amazon, still subscribes to a proprietary format, currently the Kindle Fire 8 (.kf8), it, too, is becoming more web compliant by subscribing to HTML5 and CSS3 as defined by the W3C.<sup>25</sup> This means that although the two most popular ebook formats are different, the coding they subscribe to, in the form of standards compliant HTML and CSS, is the same. As a result HTML5 and CSS3 open-source coding can and should serve as a foundation for publishers to produce digital publications with the clean,

---

<sup>22</sup> Sieling, (2012).

<sup>23</sup> Liz Castro, "Creating pop up footnotes in EPUB3 (and thus in iBooks)," *Pigs, Gourds, and Wikis*, May 22, 2012, <http://www.pigsgourdsandwikis.com/2012/05/creating-pop-up-footnotes-in-epub-3-and.html>.

<sup>24</sup> Liz Castro, "EPUB3 has finally arrived," *Pigs, Gourds, and Wikis*, July 3, 2012, <http://www.pigsgourdsandwikis.com/2012/07/epub-3-has-finally-arrived.html>.

<sup>25</sup> "Kindle Format 8," Amazon.com, 2012, <http://www.amazon.com/gp/feature.html?docId=1000729511>.

## *Borrowing from the Open Web*

flexible coding necessary to implement responsive designs, which can then be reformatted to suit the output requirements of specific distributions channels.

Giving a mainstream vote of confidence in favour of adaptive layouts, on June 6, 2012, Google Webmaster Trends officially endorsed responsive design as the preferred layout method for mobile-enabled sites.<sup>26</sup> In conjunction with the IDPF's new adherence to open web standards, and to a lesser extent Amazon's subscription to HTML5 and CSS3, Google's endorsement gives responsive design a solid basis for application within digital publications. As such, it is only a matter of time before publishers harness these strategies for extending the shelf life and interoperability of their products. Giving non-fixed layout designs a larger focus, publishers can impact the book market with designs that surpass the basic functionality provided by ereaders and create layouts that adapt to the user directly.

---

<sup>26</sup> Barry Schwartz, "Google Finally Takes A Clear Stance On Mobile SEO Practices," SearchEngineLand.com, June 6, 2012, <http://searchengineland.com/google-finally-takes-a-clear-stance-on-mobile-seo-practices-123543>.

## Chapter 3

### BECOMING RESPONSIVE:

#### Implementation of Device Independent Practices

Each quarter it seems a new tablet or handheld device is released and with it new dimensions and resolutions for designers to accommodate, as well as new file formats for publishers to adhere to in the realm of ebooks. Among popular devices, physical screen size and resolution differ greatly which can result in designs that display well on one device or screen size, but not others. For example, the iPhone 3 has a pixel resolution of  $480 \times 320$  (163 pixels per inch (ppi)), the iPhone 4  $960 \times 640$  (326ppi) and the Samsung Galaxy Nexus a resolution of  $1280 \times 720$  (316ppi), while in the tablet category the iPad 3 has a resolution of  $2048 \times 1536$  (264ppi) and the Nexus 7 a resolution of  $1280 \times 800$  (216ppi).<sup>27</sup> Designing aesthetically enhanced digital publications (not just plain-text EPUBs) that display optimally for all of these devices requires dynamic coding that can alter to the constraints of each screen. Web and book designers are beginning to realize that designing individual layouts for each device is an expensive and futile pursuit and that they must adapt a single design to suit many devices, rather than one or two popular screens. For ebooks specifically, designing for one ereader is analogous to designing for one browser on the open web—a short-sighted approach that will exclude a large portion of the market. Instead, fighting single platform dominance in favour of open standards for all digital reading will simplify workflows and allow access to wider audiences over a range of devices.

The responsive design approach allows digital publications to attain a level of longevity during a time of diverse reading environments. Adhering to open web standards used in conjunction with media queries and relative measurements will allow for flexible publications that can be read on any screen. Implementing responsive design requires a few additional lines of coding to HTML and CSS files within a publication, and a change of measurements (commonly written in pixels) to relative units of measurement, such as percentages or ems. Together these changes allow layouts to adjust to differing screen widths and, when paired with screen size-specific style sheets, create single publication designs that adapt to display across multiple devices.

---

<sup>27</sup> “List of displays by pixel density,” *Wikipedia*, February 21, 2012, [http://en.wikipedia.org/wiki/List\\_of\\_displays\\_by\\_pixel\\_density](http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density).

## *Becoming Responsive*

The following section discusses the components involved in creating responsive designs, particularly in terms of adherence to open web standards for coding and practices, CSS media queries, and relative measurements. The existing constraints of digital output in terms of resolution and pixel measurements are also examined to address how design solutions should remain free of attachment to devices.

### Aiming For Device Independence

The benefit that responsive design has over other layout methods is that the design is dynamic and changes to suit the output device. The coding that implements these layouts is open-source HTML and CSS and can be harnessed by any reading application or browser. The advantage with such solutions is that publications will not be sidelined to certain platforms or devices and, as a result, new markets will be able to purchase content with similar design attributes over many different devices.

Although the practice of device independence is relatively new to design, it has been established for content on the open web for some time. In 2001, the W3C, the governing body responsible for developing web standards, first approached the issue of device independence formally by way of the Device Independence Principles Working Group as a guide for web development. The Device Independence Principles argued for consistent accessibility for all content across a wide array of devices including phones, flat screen televisions, home appliances, and personal computers. This instituted the view that devices act as an intermediary that facilitates the connection of users to online content, and that content should not be restricted to display on certain devices. Once the principles of device independence were established for web content, the next natural step was creating device independent design over the same range of devices. In 2000, John Allsopp's article "A Dao of Web Design" affirmed the value of device independence for design and implored designers to "make pages which are accessible, regardless of the browser, platform or screen."<sup>28</sup> With these driving concepts in place, many web designers and developers recognized the desire to make the web and its design features accessible everywhere. However, the implementation of flexible, single-site web designs did not become a common approach to web design until the recent rise of the mobile web and Ethan Marcotte's "Responsive Web Design" article. In it, Marcotte touts the

---

<sup>28</sup> Allsopp, (2010).

## *Becoming Responsive*

same principles Allsopp had proposed years earlier, particularly that web design should be “future proof” and flexible enough to transition seamlessly from one screen to the next.<sup>29</sup> As defined by Marcotte responsive design advocates creating dynamic web layouts that *respond* to the user and their viewing environment, including their screen size, resolution, platform, orientation (portrait or landscape), and behaviour.

Although layouts can be responsive without necessarily being device independent, it is much more cost effective to have designs that are free of any limitations in terms of device resolution or proprietary constraints. This makes for desirable design efforts that adapt to the many devices that will be released in the future. Unlike web designs that use pixel measurements that reflect a device’s resolution, device independent design is reliant upon the use of measurements that are not defined by the specific resolution of a device<sup>30</sup>, or proprietary set of devices. Device independence takes place in flexible coding that is founded on the premise of interoperable, open-source coding information, as well as layouts that are not restricted to output on certain devices. Instead, device independence for design is very similar to device independence for content — it strives for similar user experiences for layout on any access mechanism.

### Open Web Standards vs. Proprietary Coding

The International Digital Publishing Forum’s (IDPF) October 2011 release of the EPUB3 specification was innovative in its decision to include more of the W3C’s open web standards, including HTML5, CSS3, Scalable Vector Graphics (SVG), and the Web Open Font Format (WOFF) to its format specifications. The W3C’s standards are fundamental to coding practices for the Internet as it exists today and the IDPF’s decision was significant because it made a format for digital books consistent with much of the coding on the web. In conjunction with the IDPF’s decision to open up the EPUB format to web standard coding, Amazon introduced the Kindle Fire 8 format, which also harnesses open-source standards in the form of HTML5 and CSS3, despite remaining a proprietary file format. This further established both the EPUB and the Kindle as dominant formats for digital books as they are now more open and adept at delivering interactive and dynamic experiences.

---

<sup>29</sup> Marcotte, (2010).

<sup>30</sup> Chris Armstrong, “The Goldilocks Approach to Responsive Design,” Designbyfont.com, April 29, 2011, <http://www.designbyfont.com/workinprogress/article/the-goldilocks-approach-to-responsive-design>.

## *Becoming Responsive*

Now that the two leading ebook formats utilize web standards, publishers should take a vested interest in the open web standards that impact their products most and use them to guide future publication practices. The W3C's standards for HTML, CSS, Graphics, and Mobile Web will be central to innovative and adaptive publication design, while standards for audio, video, and JavaScript will be foundational in extending the media richness of future publications. Publishers should also take note of some of the other standards outlined by the W3C, particularly the Device Independence Principles. The W3C Device Independence Group recommends, "the goal is that a functional user experience should be possible via any access mechanism,"<sup>31</sup> meaning *any* device or browser should be able to access any and all online content. As such, publishers should hold their digitally published content to the same standard as web content. Ideally, a publication would be readable on any device that subscribes to the format, as well as any web browser, and would not be limited to certain ereaders or reading applications, regardless of their popularity. What impedes publishers from being able to publish in such a way is the monopolistic aspect of large distributors, such as Apple and Amazon, that limit a publication's display on other devices. These distributors commonly require proprietary formats, such as Amazon's Kindle Fire 8 (.kf8), or fixed-layout ebooks, such as Apple's fixed-layout iBooks, which can only be read on one device. Fixed-layout EPUBs and proprietary formats are problematic because new devices will inevitably be released take the place of those on the market today.

Unfortunately in an attempt to sell books publishers commonly succumb to distributor pressure and adhere to the requirements of each distribution channel in terms of coding and format, even in instances where such decisions may pigeonhole content to one distribution channel for the life of the content. While solely publishing in open-source formats, such as the EPUB3, would be ideal, it is simply not feasible due to the current market share of certain distributors. However, now that both the EPUB and the .kf8 format subscribe to open web standards, publishers should take a decisive stance in favour of web standard coding for all publications regardless of the final format. This would allow publishers to have source files built on open-source coding (essential to publishing in the EPUB3 format) that could then be stripped down and repackaged within other formats, such as the .kf8, without having to rewrite the majority of a publication's HTML and CSS coding. By deciding to use open-source mark-up for publications, publishers can sidestep some of the tedious alterations that are sometimes required of certain distributors' proprietary coding.

---

<sup>31</sup> "Device Independence Principles: W3C Working Group Note 01 September 2003," W3C, September 1, 2003, <http://www.w3.org/TR/di-princ/>.

## Becoming Responsive

An example of coding that publishers should avoid can be seen in the January 2012 release of Apple's iBooks textbook format. With this format, Apple diverged from adhering to the w3c's open web standards for css in an effort to corner more of the market and capitalize on content by limiting ebook distribution to their iBooks channel. Apple's textbook format included an array of css extensions that are not standards compliant, such as:

```
-ibooks-gutter-margin-left: 50.0pt;  
-ibooks-gutter-margin-right: 25.0pt;  
-ibooks-head-height: 660.0pt;32
```

These css classes are only supported within the iBooks application and do not comply with open web standards. With such alterations to standardized css, Apple requires textbook publishers to create files specific to their devices alone, which cannot be easily altered to comply with other ereaders or applications, and also demands additional hours of coding and design to produce.

Unfortunately, these practices are not new. On the open web, before css features are officially adopted by the w3c, or in some instances as a way to use features not even proposed to the w3c for css, browsers will adopt shortcuts and prefixes to use unreleased css3 features. Just as Apple has included the -ibooks-prefix within the iBooks format, Google Chrome, Apple Safari, and Mozilla Firefox browsers have also include similar pseudo-css: Chrome and Safari use the css prefix of -webkit-, where as Firefox uses the prefix -moz-.<sup>33</sup> What needs to be addressed in such situations is that these features are *not* css; they are proprietary shortcuts that evade the standards compliance process required of css.<sup>34</sup> By adopting these tricks, companies like Apple, Google, and Mozilla allow designers to utilize proprietary css extensions to improve design features for their browsers and reading applications. These features wow users while undermining the practice of standards under the guise of *pseudo-css*.

By introducing proprietary css, beginning with the textbook format, Apple

---

32 Baldur Bjarnason, "The iBooks 2.0 textbook format," baldurbjarnason.com, January 19, 2012, <http://www.baldurbjarnason.com/notes/the-ibooks-textbook-format/>.

33 For example, to use the anticipated— but not finalized—css3 feature of bevelled corners for graphics in Chrome, Safari, and Firefox the following pseudo-css is used: `-webkit-border-radius: 20px; -moz-border-radius: 20px;` Source: Zoey Mickley Gillenwater, *Stunning CSS3: A project-based guide to the latest in css*, (Berkeley: New Riders, 2011), 52-53.

34 Lea Verou, "Every Time You Call a Proprietary Feature 'css3,' a Kitten Dies," *A List Apart*, February 14, 2012, <http://www.alistapart.com/articles/every-time-you-call-a-proprietary-feature-css3-a-kitten-dies/>.

## *Becoming Responsive*

has started a push for *device dependence*, a move counter to the guiding principles of the web. Over the short term, this means profits for digital distributors, but in the long term it will mean a decline in user experience as readers will be tied to a brand's device for the life of the content. Meanwhile, for publishers, this results in countless hours of reformatting source files to comply with a single distributor's coding specifications. Limiting output to one device alone, such as the iPad, or catering to each distributor's proprietary format is a step in the wrong direction. Such decisions will cause publishers to create several ebook files for each title, or recreate the wheel every five years should a competitor's tablet or ereader emerge to become the leading device. This is an unnecessary and costly burden that publishers should reject outright.

The solution to this problem is for publishers to be aware of open web standards and use these standards as a guide for determining their publication and coding practices. By investing in web standard coding and conventions established by the w3c, publishers will have source files that are largely independent of proprietary codes and formats and will therefore be free of distributor's control and better equipped to stand the test of time and device obsolescence.

### How to Implement Responsive Design

Creating responsive design requires addressing different screen sizes in HTML code and creating CSS that reflect specific screen sizes, be they smart phones, tablets, or personal computers. With well over 400 different devices on the market today, mobile screen dimensions can range from 128 pixels (px) × 128 px to 2048 px × 1536 px with varying aspect ratios (height to width ratio) and pixel densities.<sup>35,36</sup> If aspect ratios were consistent across all devices, simply changing pixel measurements to relative measurements would allow layouts to scale from device to device without additional HTML and CSS coding. However, when device widths are variable some presentational features must be adapted for optimal presentation. For example, an ideal layout for a laptop will not scale well to the portrait view on mobile phones. In these instances changes to design need to be made to ensure that optimal reading is possible and alterations in coding allow designs to adapt to an optimum width for each device. To implement responsive

---

<sup>35</sup> Morten Hjerde, "Mobile screen size trends," *Sender 11*, April 15, 2008, <http://sender11.typepad.com/sender11/2008/04/mobile-screen-s.html>.

<sup>36</sup> "List of displays by pixel density," *Wikipedia*, February 21, 2012, [http://en.wikipedia.org/wiki/List\\_of\\_displays\\_by\\_pixel\\_density](http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density).

## *Becoming Responsive*

layouts, there are three components that are needed within the design: *css media queries*, the *meta viewport tag*, and *relative measurements*. The media queries and meta viewport tags are lines of code added to the HTML of a design and act to call on CSS files with style features specific to broad categories of screen sizes, such as mobile phones (small), tablet (medium), and personal computers (large). The application of relative measurements for all elements of a design within the CSS of a layout allows the page to dynamically adjust depending on the specific resolution and size of each device. These components, together with web standard coding, allow for layouts that are functional across a range of devices.

### *CSS Media Queries*

With the growing number of mobile devices available to access content, a feature called the media query (written in HTML code) works to inspect the properties of a device and pull the correct formatting and styling from a device-specified CSS file.<sup>37</sup> With the aid of the media query feature, in conjunction with specified style sheets, a given page's content can scale to the optimal constraints for a device. This is achieved by coding a media query within an HTML document's <head> tags:

```
<head>
<link rel="stylesheet" type="text/css" media="screen
and (max-device-width: 480px)" href="mobile.css"/>
<link rel="stylesheet" type="text/css" href=
"default.css"/>
</head>
```

In this example, the code “media=“screen and (max-device-width: 480px)” href=“mobile.css”/>” asks the device if a maximum device width of 480 pixels (on the X axis) is present. If the device is equal to, or smaller than 480 pixels, the CSS file containing style sheets specific to smaller mobile screens, such as the iPhone4 (a device with a 480px width), then the mobile.css file will load. This CSS file will contain specific coding for small devices that presents content in an optimal way for those particular screen sizes. If, on the other hand, the device

---

<sup>37</sup> For those unfamiliar with the media queries, the directive is also used when printing web content. To optimize what is on screen for the printed page, different style sheets are used by specifying a media query for print outs.

## *Becoming Responsive*

has a screen size larger than 480 pixels, the mobile css file will be ignored and the default.css file, intended for optimal reading on larger screens, will be used.

As a point of comparison, on standard fixed-layout webpages that do not define mobile- or tablet-specific css, there will be one css file used to define the layout for all devices. The code is similar but lacks css styling specific to mobile device constraints:

```
<head>
<link rel= "stylesheet" type= "text/css" href=
  "desktop.css" >
</head>
```

Without the media query directive, the above example will not display content dynamically. Instead, when browsing on mobile devices the standard, large format style sheet desktop.css will be used to display the same page designs for smartphones and desktop computers, which will require mobile readers to zoom, pinch, and scroll to read content on smaller screens.

Media queries can define an array of components in addition to device widths including resolution, height, width, orientation (portrait or landscape), and device-aspect-ratio.<sup>38</sup> Publishers will likely recognize coding media queries from fixed-width books for Apple's iBooks, which requires the directive for portrait and landscape specifications.<sup>39</sup>

### *The Meta Viewport Tag*

For media queries to function properly on handheld devices the code must also be paired with the meta viewport tag. The meta viewport tag essentially gives an optimized width for the device at hand. The meta viewport code communicates with the device as to how large the visual portion of the design should be. Currently most smartphones automatically resize and enlarge web content to display properly on smaller screens due to the differences in physical size and resolution of most mobile devices compared to personal computers. Once a design is optimized for viewing on handheld devices (and not personal comput-

---

<sup>38</sup> Matthew Diener, "Media Queries in ePUBs," *epub Secrets*, February 9, 2012, <http://epubsecrets.com/media-queries-in-epubs.php>.

<sup>39</sup> Ibid.

## *Becoming Responsive*

ers), as will be the case for all responsive designs, the browser must be told *not* to zoom in. By setting the meta viewport to the device width, the design tells the browser or reading application to fit the design to the device constraints. To adjust this aspect of the layout, include the following code:

```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0" >40
```

If this is not included, as the Quirksmode blog notes, the responsive design will “misfire horribly.”<sup>41</sup> The meta viewport tag includes the “initial-scale=1.0” directive to tell mobile devices with built-in presentation adjustments (automatic zoom and scaling for web content) that the initial scale of the site should be presented at 100 percent, rather than much larger. If this is not included the site can zoom unintentionally and make small portions of an already ideal sized mobile page appear huge on handheld devices.

### *Relative Measurements*

To optimize publication files for device independence, relative units of percentages or ems are best as they are proportional and adapt well to differing screen sizes. The application of relative measurements within the CSS allows a design to dynamically adjust depending on the specific resolution and size of each device. One aspect that contributes to the interoperability of digital page design is removing all aspects related to the device from its coding. Currently designers commonly limit the shelf life of a website or publication by building existing resolution constraints into the HTML and CSS of a design. The most common association designs have to devices is the use of pixel measurements within source files. This is problematic because it ties designs to measurements that become rapidly out of date on mobile devices. The following information is provided to clarify why pixels are not an ideal measurement for digital publications and that relative measurements should be used instead.

---

<sup>40</sup> Note the use of commas within the meta viewport tag. If semicolons are used instead the layout can misfire on certain devices.

<sup>41</sup> Peter-Paul Koch, “A pixel is not a pixel is not a pixel,” *Quirksmode*, April 20, 2010, [http://www.quirksmode.org/blog/archives/2010/04/a\\_pixel\\_is\\_not.html](http://www.quirksmode.org/blog/archives/2010/04/a_pixel_is_not.html).

## *Becoming Responsive*

On the open web and in ebooks, designers use pixel measurements for many, if not all, aspects of a design. This practice has origins from the beginning of web design as pixels are the smallest element of a device display. However, at some point in time the term “pixel” began to have two different meanings. Pixels can be used to note the exact number of pixels on a device’s screen, and they can also be used to measure page features, such as type size, within a design. Most people, including many designers, assume that in both instances the term “pixel” is indicative of individual pixels on screen. Although, when it comes to most CSS measurements this is not the case. When measuring page elements, such as type size, the pixel measurement is inconsistent across devices. For example, a 16px font on an iPhone can be quite small when compared to a 16px font on a personal computer.<sup>42</sup> The pixel value in such instances is an arbitrary measure, dubbed by some as “CSS pixels,” or, as the Quirksmode Blog refers to them, “essentially an abstract construct created specifically for...web developers.”<sup>43</sup> This disparity has occurred because of the varied pixel densities and screen resolutions on the market today. Due to the variability in resolution over the range of devices used to access web content, pixel measurements could no longer be indicative of literal pixels on a display as this would make for devastating browsing experiences on small screens and low-resolution devices. With major discrepancies in physical size and screen resolution from handheld devices to personal computers, mobile devices required scaling to insure that output from the web is relatively consistent when compared to larger displays. As a result, the W3C redefined the pixel as something that user agents (browsers, or in the case of ebooks, reading platforms) must rescale:

“The user agent should rescale pixel values. It is recommended that the reference pixel be the visual angle of one pixel on a device with a pixel density of 96dpi and a distance from the reader of an arm’s length.”<sup>44</sup>

In an attempt to rectify the discord that emerges from “CSS pixels” and absolute screen pixel measurements, the W3C notes that 96 pixels are now equal to one

---

42 According to Armstrong (2011), 16px text on an iPhone appears to be approximately 60 percent of the size of 16px text on personal computers.

43 Koch, (2010).

44 “Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification,” W3C, June 7, 2011, <http://www.w3.org/TR/CSS2/syndata.html#length-units>.

## *Becoming Responsive*

inch. This is done to adjust the lag between designers' measurements and the rapid release of devices with ever-increasing pixel densities.

With such incongruences in the measurement, pixels should be replaced with a unit of measurement that translates to something less arbitrary. Just as it would be impractical to have units of inches or centimetres on a site or ebook given the variability of screen output sizes and pixel densities, it is just as dissonant to build sites or books with pixels. A practical approach is to adopt units that are consistent across devices and platforms. For responsive design to be device independent, relative units of measurement must be used. With relative measurements of percentages and ems, when a page increases or decreases in size, the proportional relationship between elements will remain as dimensions change.<sup>45</sup> When designing for an array of devices that require more flexibility, relative units are variable enough to accommodate changing outputs.

Both fluid and elastic approaches to responsive design use relative units. Fluid layouts measure some design features in percentages, while the elastic approach uses em measurements for all page elements. Converting measurements to relative units is essential to creating responsive page design and leaves all components of a design flexible to resize proportionally regardless of the screen size. Em units, which are an essential component of elastic layouts, are recommended for publishers because all measurements become tied to the size of text. Applying this unit to digital publications allows all aspects of a page, including graphics and images, to proportionally scale to suit the visibility needs of the reader.

---

<sup>45</sup> James Felici, *The Complete Manual of Typography* (Berkeley: Peachpit, 2012), 24.

## Chapter 4

### THE ELASTIC APPROACH TO RESPONSIVE DESIGN

Individuals are increasingly reading on more than one device, be it smart phone, tablet, or personal computer, and each has differing physical dimensions and resolution capabilities. The end goal of responsive design is to build a site or publication that intuitively reacts to the constraints of each device seamlessly. As previously mentioned, there are two methods for implementing responsive design solutions: both fluid layouts and elastic layouts address issues of interoperability of designs and benefit publishers by increasing the longevity of source files with the use of relative measurements. However, elastic layouts are the recommended layout method for publishers due to the reciprocal relationship that elastic layouts have to the text. Elastic layouts use em measurements for all elements of a page within CSS files. The em unit is representative of the size of the text and all other page features are sized in proportion to the type size. Elastic layouts use both CSS media queries and the em unit to adapt to the dimensions of mobile, tablet, and personal computing screens to establish a consistent and intuitive user experience. To date the elastic approach has primarily been applied to designs on the open web, but there is no reason that the same method for dynamic design and device independence cannot come to the forefront of the ebook debate. Currently publishers use the em to measure type sizes in EPUBS but then break from the unit for all other elements on screen. Rather than use ems once arbitrarily, ems can instead be adopted as the measurement for all page features and give readers an optimized experience on each device.

The following section is divided into two parts. The first part outlines the elastic approach to responsive design, the report's suggested method for applying responsive design to digital publications. It addresses issues of aesthetics including type and ratios as well as the visual accessibility of readers. The second part uses the website [Fontasizing.com](http://fontasizing.com) and its application of the Goldilocks Approach framework<sup>46</sup> as a case study for implementing the elastic approach to responsive design for HTML-based publications. The case study covers the responsive solutions of CSS media queries, relative units, and responsive graphics, as well as aesthetic aspects of digital page design.

---

<sup>46</sup> The Goldilocks Approach is an elastic responsive framework released by Irish design studio Front. The framework can be found at <http://goldilocksapproach.com/>.

## The Case for Elastic

The elastic approach to responsive design uses em units and ratios — rather than pixels — to determine the size of an object or block of text and their relationship to other features on a page. Elastic layouts differ from other web and EPUB designs in that they are free of any constraint that is tied to the device or output on screen.<sup>47</sup> Instead, the em measurement is determined by the text size. This makes the design of a page self-referential due to all features being a proportion of the text. When the end user increases or decreases the type size, the layout and images will not break as many EPUBS currently do. Without the involvement of pixels or other absolute units, the elastic interface is free to scale up or down in size as needed. This allows all devices to display content because the page can scale to accommodate the output of each screen size. Compared to the prevailing coding practices for EPUBS, which use varying units of measure, elastic layouts only adhere to the em, which is unrelated to screen resolution. Elastic layouts are advantageous to publishers because in the future when devices inevitably increase in pixel density there will be no need to repeatedly update source files with new pixel measurements. This essentially makes elastic layouts a device and resolution independent solution.<sup>48</sup>

With a variety of measurements to choose from for digital style sheets, the rationale behind the elastic approach's use of ems is founded on its proportionality and ability to size elements relative to type on screen. On another level, the em is a logical choice for digital publications due to its relationship to typography beginning with early printing methods. The em is a fundamental unit of typography with a history in lead type.<sup>49</sup> The em is a perfect square that measures the width and height of a font's bounding box in digital type and the height of the metal printing block, or "body", in lead type. These determine the space that the characters will take up on a page or screen.<sup>50</sup> As a relative unit, the em remains *proportionally* the same for each font size. Be it measured in points, pixels, centimetres, millimetres, or inches, an em's size, be it height or width, is always reflective of the font size: a font of 12 points (pt) will have an em of 12pt; a font of 16px will have an em of 16px.

Like all page designs, elastic layouts must reflect a balanced relationship between space, type, and image. Without standard page dimensions or

---

<sup>47</sup> Armstrong, (2011).

<sup>48</sup> Ibid.

<sup>49</sup> Felici, 24.

<sup>50</sup> Ibid., 23.

## *The Elastic Approach to Responsive Design*

measurements as a guide, to achieve a sense of continuity on an elastic page ratios are used to create aesthetically satisfying page designs. Ratios and measured intervals are commonplace in graphic design; however, when designing elastic layouts in particular, proportions become the glue that keep elements in relationship when the parameters of a page frame are variable. Measured proportions give a page a sense of rhythm and flow, and on an unconscious level, keep the reader engaged with the content.

The most common approach to developing a digital page with relative units is by adopting well-established ratios from mathematics and music. The most common proportion used in graphic design is the Golden Section. This universal proportion is celebrated for its aesthetic appeal in nature and man-made works. Also referred to as the golden rule, it is a division of two where the “smaller is to the larger as the larger is to the whole”(See Figure 1).<sup>51</sup> The golden section is a ratio of 1 : 1.618... but is commonly simplified to 1 : 1.6 in typography. Deciding how proportions and ratios, such as the golden section, will be applied to the layout is ultimately the designer’s responsibility. On a given page, important ratios can present themselves in any number of ways. Areas where measured intervals are most apparent are between section components, for example between articles and sidebars, paragraphs and headings, or type and leading. The one facet that brings the page an overall sense of rhythm is by establishing a baseline grid, commonly done by selecting a leading (“line-height” in css)<sup>52</sup> for the main text and setting subsequent headings on the same leading. This establishes a baseline ratio for the rest of the page to adhere to and creates a sense of balance on the page.

In conjunction with proportional measurements and ratios, the elastic approach, while functional, also lends itself to the aesthetic and typographic pursuits of the publishing industry. For this reason, in addition to the em measurement’s inherent relation to the text, elastic layouts are an attractive solution for designing digital publications.

---

<sup>51</sup> Robert Bringhurst, *The Elements of Typographical Style*, (Vancouver: Hartley & Marks, 2008), 148.

<sup>52</sup> Typographically, “leading” is the measurement of vertical space between one line of type and the next.

### Ratios: The Golden Mean

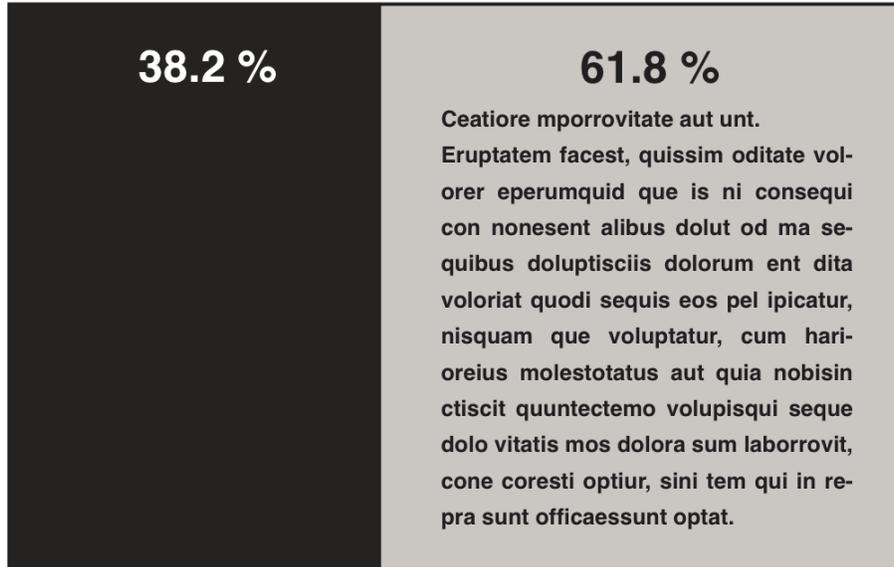


FIGURE 1: The golden mean, a ratio of 1:1.618, can be used as a measured interval for many aspects of page design. As Robert Bringhurst notes in *The Elements of Typographical Style*, the smaller part of the golden section is approximately 38.2 percent, while the larger is 61.8 percent. The golden ratio in this example is also present within the baseline of the text. The type is 14pt while the leading is 19.4pt, recreating the ratio of 1:1.618.

### Delivering Accessibility: Elastic Layouts and the End User

Due to the proportional aspects of elastic layouts, the approach is also well suited to the needs of readers because all page components easily resize in conjunction with the text. This is advantageous because it makes publications more accessible to those with limited vision. Such benefits follow the accessibility norms established online. In 1997, the w3c launched the Web Accessibility Initiative and since that time have been committed to accessibility on the Internet.<sup>53</sup> This

<sup>53</sup> “World Wide Web Consortium (w3c) Launches International Web Accessibility Initiative,” w3c, April 7, 1997, <http://www.w3.org/Press/WAI-Launch.html>.

## *The Elastic Approach to Responsive Design*

web standards practice has made for web development and design that satisfies a diverse range of abilities. The elastic approach to web design is one method that caters to the accessibility needs of those with low vision as the page design scales along with the text. Although there are many kinds of visual disabilities, elastic layouts are most beneficial to those defined by the w3c as having low vision (including poor acuity, tunnel vision, clouded vision, and central field loss) where an increase in text size typically results in the ability to read despite visual limitations.<sup>54</sup>

As elastic layouts are built on type size, the text and the typeface are given a prominent role in the layout and can resize easily. By building layouts that adhere to this principle, *all* design features can scale to suit a comfortable reading size for the end user. Currently basic plain-text EPUBS also have the ability to scale text, but do not offer the same features when it comes to images and other page design features. Due to the proportional relationship of all elements in an elastic layout, when a user increases or decreases the size of the text *all* design features on a page, including images and graphics, will scale accordingly. This appeals to users because the entire layout scales with the size of the text, rather than the size of the browser widow. For those with visibility issues, not only will the text respond to their visual requirements but the whole layout will adjust in unison with the text. On the open web, elastic layouts currently scale easily according to users' text preferences. When the type size is increased with the browser controls of [control] and [+] or [-] the whole layout scales seamlessly, and many ereaders carry the same text resize features.

As the latest EPUB specification aligns with the w3c standards for HTML, ebooks will subscribe to the same standards for visual accessibility as pages on the web. In anticipation of users with limited vision, the w3c recommends, “do not specify the font-size in pt, or other absolute length units for screen style sheets. They render inconsistently across platforms and can't be resized by the User Agent (e.g. browser),” and to “use relative length units such as percent or (better) em.”<sup>55</sup> By using a relative unit such as ems, those with limited vision are free to resize text without restrictions that sometimes occur when absolute units like points are used. In an elastic interface, the text and images will easily scale in ems when

---

54 “Diversity of Web Users,” W3C, February 20, 2012, <http://www.w3.org/WAI/intro/people-use-web/diversity.html#visual>.

55 “Quality Assurance: Care With Font Size,” W3C, February 20, 2012, <http://www.w3.org/QA/Tips/font-size>.

## *The Elastic Approach to Responsive Design*

user interface controls readjust the text size.<sup>56</sup> By building scalable features into the design of a page, publishers will ensure that regardless of reading application the whole publication, including images and design elements, will scale to suit the reader's visual needs.

“Accessible design is good design,” as Steve Ballmer CEO of Microsoft says, and when it comes to publishing, accessibility it is not only good design, it is good for business.<sup>57</sup> While scaling text has been the norm on the web for many years, the first handheld device to bring hardware controls of font size adjustments was in November 2007, when Amazon's first generation Kindle was released. Such built-in text resizing controls support readability and affirm the user's desire for personal control. It is likely that such hardware controls have also propelled elderly readership to choose on-screen reading over print for this reason. In the musings of one 67 year old, ereaders are “‘old people’ friendly devices” because the text scales to suit their sight limitations.<sup>58</sup> According to Nielsen, the second quarter of 2011 reported that 30 percent of ereadership is 55 years of age or older, which is an increase of 5 percent on the previous year.<sup>59</sup> With the average age of ereader users increasing, issues of limited vision play a larger factor in delivering accessibility through design. For those publishers with an eye on aging demographics, it is estimated that one in seven Canadians and over 14 million Americans currently suffer from some form of vision loss, with this number assumed to increase in the coming years with an aging baby boomer generation.<sup>60</sup><sup>61</sup>

Perhaps it is obvious that if a reader has trouble viewing text at an average size they will also have trouble seeing design features and images at a similar scale.

---

56 The previous CSS2 standards for fonts had browsers increase absolute text sizes (ie. “small”, “medium”, “large”) at a fixed ratio of 1:1.2. However, in the most recent release of CSS Fonts recommendation, the W3C no longer advocates for fixed ratios for text resizing and text sizes will be determined by the browser or ereader.

57 Bergel et al., (2005).

58 “eReaders for the elderly – A useful gadget for older people,” eBookAnoid, March 3, 2010, <http://www.ebookanoid.com/2010/03/03ereaders-for-the-elderly-a-useful-gadget-for-older-people/>.

59 “Changing Demographics of Tablet and eReader Owners in the US,” NielsenWire, August 25, 2011, [http://blog.nielsen.com/nielsenwire/online\\_mobile/changing-demographics-of-tablet-and-ereader-owners-in-the-us/](http://blog.nielsen.com/nielsenwire/online_mobile/changing-demographics-of-tablet-and-ereader-owners-in-the-us/).

60 “Study reveals many Canadians unnecessarily living with vision loss,” CNIB, August 5, 2012, [http://www.cnib.ca/en/Pages/Study\\_reveals\\_many\\_Canadians\\_unnecessarily\\_living\\_with\\_vision\\_loss.aspx](http://www.cnib.ca/en/Pages/Study_reveals_many_Canadians_unnecessarily_living_with_vision_loss.aspx).

61 Marguerite Bergel, Ann Chadwick-Dias, Lori LeDoux, and Tom Tullis, “Web Accessibility for the Low Vision Use,” Fidelity Investments, 2005, [http://www.eastonmass.net/tullis/presentations/Web\\_Accessibility\\_for\\_Low\\_Vision.pdf](http://www.eastonmass.net/tullis/presentations/Web_Accessibility_for_Low_Vision.pdf).

## *The Elastic Approach to Responsive Design*

When using elastic layouts, the em unit scales text and design features of a page seamlessly according to the end user's desired text size. By allowing other design elements to shift with the text size those with limited vision can also appreciate page aesthetics and images at larger sizes.

### How to Calculate Elastic Layouts

To create elastic layouts it is essential to understand how inheritance works within Cascading Style Sheets. As the name implies, the styles “cascade” from one element to the next, allowing for styling to apply to more than one element on a page. Inheritance allows elements to propagate the same styling properties from a parent element to a child element without the need to redefine the style for each element on a page. Some (but not all) css properties are inherited by default. For example, the body text selector's font properties, such as colour and size, will be inherited by subsequent child elements like headings and paragraphs without the need to write out the recurring font features. With elastic layouts, all feature sizes (including items such as images and borders) are tied to the font size — *in ems*—of a parent element. The child of that element will then inherit the same font size unless the child element defines a different font size. Once the size of the em is altered by a child element, some calculations will need to be made to ensure that element sizes are uniform throughout a page's design.

In the following examples, pixels will be used as a comparative unit due to their widespread use in web design. This is done to simplify conversions to the em. Initially, an em will be whatever the base font size is calculated to be. Most browsers have a default font size of 16px to facilitate average reader accessibility. If the body text is set to 1em, the em unit will remain 16px unless altered by a child element. Where this becomes complicated is when child elements cascade multiple times over and subsequent child elements require differing font sizes.

One method for simplifying the calculation of ems is proposed by web designer Jon Tan. This formula uses pixels as a guide for simplifying child elements:

$$1 \div \text{parent font size (in pixels)} \times \text{required font size (in pixels)} = \text{em value}^{62}$$

---

<sup>62</sup> Jon Tan, “The Incredible Em & Elastic Layouts with css,” Jontangerine.com, September 25, 2007, <http://jontangerine.com/log/2007/09/the-incredible-em-and-elastic-layouts-with-css>.

### *The Elastic Approach to Responsive Design*

With this method, if a paragraph (p) with a height of 12px is desired, divide 1 by the default browser font size of 16px then multiply it by 12px, resulting in an em unit of 0.75:

$$1 \div 16 \text{ px} \times 12 \text{ px} = 0.75 \text{ em}^{63}$$

The style sheet for this example would then be as follows:

```
body {
    font-size: 1em; }
p {
    font-size: 0.75em; }
```

This alteration to the font size changes the em size for subsequent child elements to 0.75em. The p element, and its child elements, now have an em size of 0.75em that equates to an absolute unit of 12px.

If at this point a photo was desired within the p paragraph, the image becomes a child element of the paragraph and will inherit the computed em value (0.75em) based on the paragraph's font size. Using the above formula to calculate the photo size, establish the desired photo dimensions in pixels and then convert to ems based on the parent element (the p paragraph property). For a square photo with a desired output of 120px, the calculations would be as follows:

$$1 \div 12 \text{ px} \times 120 \text{ px} = 10 \text{ ems}$$

```
img {
    height: 10em;
    width: 10em;
}
```

With these examples the text size is less than the browser standard of 16px, and for this reason it is possible that the reader may be inclined to increase the text size. If and when they do, all elements on the page, including the image, will scale in accordance with the user's newly defined text size.

---

<sup>63</sup> Ibid.

### *The Elastic Approach to Responsive Design*

When beginning elastic layouts these calculations can be a bit tedious, but with familiarity they will become routine. With book designers already familiar with proportions, ratios, and the em, the transition to using ems to create similar layouts digitally is not as difficult as it may initially seem. Where elastic layouts hold an advantage over other layout methods is in its ability to instill device and resolution independence within the coding. This in itself is well worth the basic calculations needed to implement designs that are flexible, increase the shelf life of publication source files and have the potential for reaching wider markets.

## *Case Study*

### **FONTASIZING.COM AND THE GOLDBLOCKS APPROACH TO ELASTIC LAYOUTS**

For the purpose of exploring future possibilities of publication design, this case study uses the website Fontasizing.com and the application of the Goldilocks Approach framework for the implementation of the elastic layouts as a strategy for HTML-based publication design. The application of responsive design solutions were made possible for this site with the aid of the Goldilocks Approach, an open-source responsive framework. The open web is used for output for this case study, however, the underlying principles, coding, and approach to design remain the same for all other publication formats that subscribe to HTML, such as the EPUB format. The website example is used with the intention of addressing the possibilities of HTML and CSS and to show how responsive design practices can and should be applied to all digital publications. The following case study discusses how to approach the process of implementing a responsive design framework by employing CSS media queries, em measurements, ratios, and dynamic page features including Scalable Vector Graphics (SVG).

#### **Implementing Responsive Design**

##### *Unit Conversions*

When first designing elastic layouts it may be difficult to conceptualize the relationship between objects without the help of an absolute unit. Developing wireframes for a design is helpful in the planning stage and can also aid in converting all units to ems. In web design the standard is to design in pixels. To make the transition to ems simpler, it is advisable to first design a wireframe using pixels due to the number of conversion charts and calculators available online for pixel to em conversion. A great resource is [PXTOEM.COM](http://PXTOEM.COM), which presents a conversion chart for pixels, ems, percentages, and points and calculates pixel units to ems based on the default font size of a page design, an essential component for designing elastic layouts. For those print designers more comfortable designing in picas and points, it is possible to convert to pixels given that PostScript picas, points, and pixels are all fractions of an inch: picas are 1/6 of an inch, points 1/72 of an inch, and pixels 1/96 of an inch. However, this will require an additional

## *The Elastic Approach to Responsive Design*

set of calculations when converting to ems. For those already comfortable with the em unit and its variability, an alternative is to build an elastic grid in ems from the outset. One online resource to aid in this pursuit is [Gridinator.com](http://Gridinator.com).

### *Responsive Frameworks*

There are several solutions available for designing responsive layouts, most of which are open-source frameworks. To facilitate responsive layout with an elastic interface, the framework selected for [Fontasizing.com](http://Fontasizing.com) is the Goldilocks Approach by the Irish design studio Front.<sup>64</sup> This framework is free and readily available for download at the Goldilocks Approach website.<sup>65</sup> The advantage of this framework is that all aspects are measured in the em unit. This means that output resolution does not play a factor in the coding of a site or publication. The Goldilocks Approach presents three distinct reading environments: small, medium, and large. These three interface sizes are based on general sizes for smart phones, tablets, and personal computers, respectively. These sizes are measured in ems and can be scaled based on the reader's text size preferences. Among other responsive frameworks, Goldilocks stands out for its simplicity of presentation and design. Written in CSS, it communicates with devices via media queries that are measured in em units rather than device pixels (as was shown in Chapter 3 of the report). Unlike other JavaScript-based responsive frameworks, the Goldilocks Approach is comprised of HTML and CSS that remain active and adaptive even when devices such as tablets go offline.<sup>66</sup>

### *Media Queries*

Most responsive design solutions are created by defining media queries. Media queries facilitate optimization by communicating with the device to establish the screen width. With other responsive frameworks these tend to be measured in pixels. Goldilocks, however, is device independent and measures device

---

<sup>64</sup> <http://www.designbyfront.com>

<sup>65</sup> <http://goldilocksapproach.com>

<sup>66</sup> For those following trends in publishing interfaces, the Goldilocks framework is reminiscent of Craig Mod's *Bibiotype* project which also has three distinct size settings. However, *Bibiotype* use JavaScript whereas the Goldilocks Approach only uses HTML and CSS coding.

## *The Elastic Approach to Responsive Design*

width in ems. The media queries evaluate the width of a device by inspecting the font size of the browser and establishes the layout's width in ems based on that font size. The css will then cater to a screen's width in ems resulting in optimized page design.

To implement multiple screen designs, the framework template includes two different css files to define styles specific to small, medium, and large screens. The layout.css file contains style sheets for both tablet and personal computer sizes (medium and large screens), and the global.css file contains styling for smart phones (small screens), as well as design features that are applied universally to all layouts regardless of screen size.

Within the Goldilocks framework the screen sizes defined within the css are:

Small screens (phones): max-width: 30 em;

Medium screens (tablets): (min-width: 30 em) and (max-width: 60 em)

Large Screens (most personal computers): min-width: 30 em;

Both HTML and CSS coding is used to distinguish which style sheets will be used for a device. The HTML specifies with the use of media queries which style sheets should be selected and the linked CSS files define the page characteristics specific to the general screen size.

The media queries denote which CSS file should be used depending on the user's device constraints. In the <head> of an HTML file, both CSS files are linked, with the global.css defined for all sites and small screens:

```
<link rel="stylesheet" href="css/example/global.css"
      media="all" >
```

And the layout.css file for medium and large screens:

```
<link rel="stylesheet" href="css/example/layout.css"
      media="all and (min-width: 33.236 em)" >67
```

---

<sup>67</sup> The added dimensions to the screen min-width (ie. 33.236em instead of simply 30em) are suggested within the Goldilocks files to prevent horizontal scrolling.

## *The Elastic Approach to Responsive Design*

A second level of formatting occurs within the CSS files. Within the layout.css file medium screens are defined with a secondary media query to differentiate tablet-sized layouts from personal computing layouts with the following code:

```
@media screen and (min-width: 30em) and (max-width:
63.236em) { }
```

Here, all CSS element stylings that differ from large screen sizes will be placed within the { } brackets.

Within the framework, the em width measurements are a general guide that allows the layout to work optimally for each device; however, should the end user alter the font size or resize their browser window, the elastic layout will respond to the new dimensions.

Once the three general size settings are established, each page design that is created on top of the framework will define styles that are optimal for each type of device. The style sheets will determine the width of the wrapper and other page elements that also need to conform to a device's constraints. It is at this point that the publication designer is able to craft designs that enhance the readability of a publication for different device settings.

When adapting the Fontasizing.com website to suit different screen sizes, three main components were selected to optimize designs to each device. The elements selected include asides, also referred to as sidebar content, the navigation panel and column widths. Depending on what reading environment the site is viewed in these features will alter slightly to allow for an optimum reading.

The Fontasizing.com style sheets are as follows:

Small screens (from the global.css file):

```
#page {
padding: 0.416em;
max-width: 34em;
margin: auto;
}

.nav {display: none;}
```

## The Elastic Approach to Responsive Design

```
aside { width: 100%; }
```

For small screens (see figure 2) the text has smaller left and right margins to allow for optimal reading environment that maximize the screen space, and the sidebar (aside) content normally to the left of the main text on larger screens is shifted to be full width on smaller screen. This is one instance within the Goldilocks Approach that percentages are used to align text rather than ems. To allow for clear and optimum reading of the text the margins are reduced for smart phone devices, as screen real estate is limited. The navigation is also removed for small screen reading for the same reason.

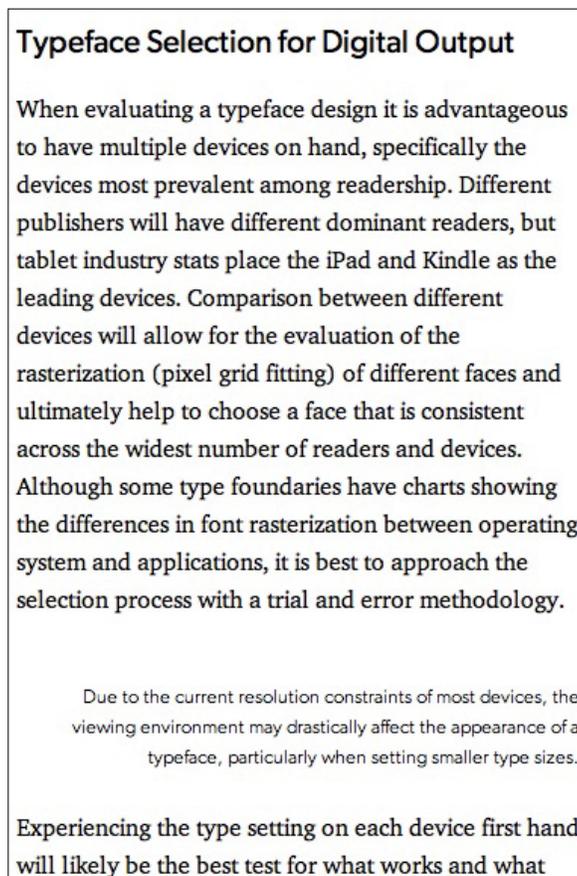


FIGURE 2: Mobile (portrait) view of the elastic layout for Fontasizing.com. Note that no navigation menu is present and the sidebar content, such as pull quotes and asides (see sans-serif text block), is set to the full-width of the display.

## *The Elastic Approach to Responsive Design*

Medium Screens (from the layout.css file):

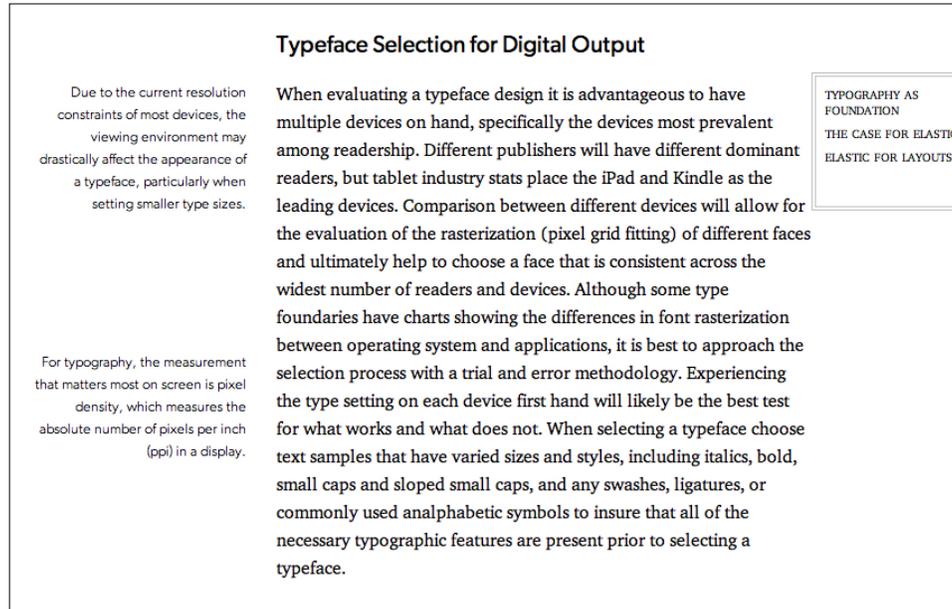
```
@media screen and (min-width: 33em) and (max-width:
  63.236em) {

#page { width: 30em; }

.nav{
  float:right;
  position:fixed;
  right:-1em;
  top:4em;
  width:9em;
  z-index:98;
}
aside {
  float: left;
  width: 17em;
  height: auto;
  margin: 0 1.667em;
}
}
```

For medium screens (see figure 3), the design is much larger and can accommodate differentiated content in terms of space on the page. To allow for the sidebar content to stand out and draw extra attention, the content is set in a left side column with the text aligned to the right. To allow for more white space for the main column of text, the centered page wrapper (#page) is used. To allow for ease of navigation between sections, a right side navigation panel is used.

## The Elastic Approach to Responsive Design



**FIGURE 3:** Tablet view of the elastic layout for *Fontasizing.com*. Note that aside content is now located to the left of the main column and the navigation panel is present to the right.

Large Screens (from layout.css file):

```
#page{
    padding: 1.667em;
    margin: 0 auto;
    max-width: 60em;
}

aside {
    float: left;
    width: 18em;
    height: auto;
    margin: 0 2.5em;
}
```

## The Elastic Approach to Responsive Design

```
.nav{  
    float:right;  
    position:fixed;  
    right:-1em;  
    top:5em;  
    width:10em;  
    z-index:98;  
}
```

For larger screens (see figure 4), there were few dramatic differences from the medium screen page design. For larger screens, the design allows for more white space, which gives the content room to breathe and draws in the reader's attention. All aside elements remain flushed right and are wider than the medium screen version. The navigation menu is also slightly larger in size for ease of navigation on personal computers.

### Typeface Selection for Digital Output

Due to the current resolution constraints of most devices, the viewing environment may drastically affect the appearance of a typeface, particularly when setting smaller type sizes.

For typography, the measurement that matters most on screen is pixel density, which measures the absolute number of pixels per inch (ppi) in a display.

When evaluating a typeface design it is advantageous to have multiple devices on hand, specifically the devices most prevalent among readership. Different publishers will have different dominant readers, but tablet industry stats place the iPad and Kindle as the leading devices. Comparison between different devices will allow for the evaluation of the rasterization (pixel grid fitting) of different faces and ultimately help to choose a face that is consistent across the widest number of readers and devices. Although some type foundaries have charts showing the differences in font rasterization between operating system and applications, it is best to approach the selection process with a trial and error methodology. Experiencing the type setting on each device first hand will likely be the best test for what works and what does not. When selecting a typeface choose text samples that have varied sizes and styles, including italics, bold, small caps and sloped small caps, and any swashes, ligatures, or commonly used alphabetic symbols to insure that all of the necessary typographic features are present prior to selecting a typeface.

By designing for the content, it was imperative to uncover a ratio relevant

TYPOGRAPHY AS  
FOUNDATION  
THE CASE FOR ELASTIC  
ELASTIC FOR LAYOUTS

**FIGURE 4:** *Personal computer view of elastic layout for Fontasizing.com. Note that compared to the tablet size, aside content takes a larger width and the navigation panel is slightly larger.*

## Conceptualizing the Page

### *Column Widths*

With no page or device constraints as a guide, the Goldilocks framework breaks down design features to the paragraph level. Typographically this is a logical choice as there is a long-standing tradition in print publishing for optimal measure (average line length) in characters. Much like elastic layouts, the ideal measure is relative and will first require a typeface and type size to calculate the ideal length. As typographer James Felici notes, the average minimum measure is generally 27 characters, with 40 characters being the ideal, and 70 the standard maximum length.<sup>68</sup> With characters having different face widths, 40 characters generally equates to a measure of approximately 30ems due to the longer width of the em character; therefore an appropriate width for a justified column will be 30ems. For ragged text 30ems will be an ideal length, but text will likely fall narrower than the defined width.

For the main serif typeface Fontasizing.com uses a measure of 30ems:

```
p, ul, ol, dl, blockquote {
    font-size: 1em;
    line-height: 1.667em;
    margin-bottom: 1.667em;
    max-width: 30em;
}
```

### *Selecting a typeface*

Type is an essential building block for creating an elastic layout as all other design elements will be proportional to the typeface's size. When choosing a typeface it is essential to evaluate its design and how its features are impacted by an increase or decrease in size on screen. A typeface with a large x-height<sup>69</sup> may require a

---

<sup>68</sup> Felici, 122.

<sup>69</sup> Typographically, an "x-height" is the distance a font takes up from the baseline to the mean line and most is easily examined with the *x* character. Generally typefaces with larger x-heights, such as Georgia, are more readable on digital devices.

*The Elastic Approach to Responsive Design*

smaller base font size of 0.875em (14px) instead of the standard 1em (16px), or it may require more leading. Due to the current resolution constraints of most devices, the viewing environment may drastically affect the appearance of a typeface, particularly when setting smaller type sizes. When evaluating a typeface it is advisable to have multiple devices on hand — specifically the devices most common among readers. Comparison between different devices will allow for the evaluation of different faces and ultimately help to choose a typeface that is consistent across the widest number of readers and devices (see Figure 5).

rica reports second-quarter	th forth now: stroke y
admits failure; Iowa broker	have no ears to his rec
golden egg turns lead	I with you; for little
fense budget cuts worry	Majesty commended
slam Afghan war as	my lord, I have be
\$3 billion of new cost	madam. My lady c

FIGURE 5: Samples of typefaces used on the Fontasizing.com site. On the left is the Gibson sans-serif font family and on the right the Charter serif font family. The Charter typeface was well suited for the standard font size of 1em (16px), while the Gibson typeface benefits from a larger type size.

## The Elastic Approach to Responsive Design

### Ratio Selection

For Fontasizing.com a measured interval of a major sixth, with a proportion of 3:5, was selected because it establishes a comfortable balance of type and space for long-form reading. With a ratio of 1:1.667, the 3:5 ratio was a well-suited match to the Charter typeface and leading. This ratio was used for interval measures of both horizontal and vertical space. To implement the baseline grid, the body text was set to 1em, and the leading (line-height) to 1.667 (see figure 6). This ratio selected for the leading is influential in creating the vertical rhythm of the page:

```
body {  
    font-family: Charter;  
    font-size: 1em;  
    line-height: 1.667em;  
}
```

#### Typeface Selection for Digital Output

When evaluating a typeface design it is advantageous to have multiple devices on hand, specifically the devices most prevalent among readership. Different publishers will have different dominant readers, but tablet industry stats place the iPad and Kindle as the leading devices. Comparison between different devices will allow for the evaluation of the rasterization (pixel grid fitting) of different faces and ultimately help to choose a face that is consistent across the widest number of readers and devices. Although some type foundries have charts showing the differences in font rasterization between operating system and applications, it is best to approach the selection process with a trial and error methodology.

#### Optimizing Text Readability

Experiencing the type setting on each device first hand will likely be the best test for what works and what does not. When selecting a typeface choose text samples that have varied sizes and styles, including italics, bold,

**FIGURE 6:** *The ratio of 3:5, or 1:1.667 can be seen most prominently in the leading of the body text. The Charter typeface used for the body text is set to 1em while the leading, referred to as line height in the CSS, was set to 1.667. This creates a baseline for the page that embodies the 3:5 ratio.*

*Including Responsive Graphics*

When creating a responsive layout that scales to accommodate any screen size, the question of placing images tends to be problematic because most images have an absolute resolution at which point they become pixelated. For this reason it is essential to find image solutions that will accommodate the scalability of a page and not become pixelated should a layout scale up in size. One such format is Scalable Vector Graphics (SVG). SVGs can be used to display graphics, charts, illustrations, and fonts.<sup>70</sup> Scalable vector graphics are most commonly created in the Adobe Illustrator program, and consist of paths, circles, and rectangles written in XML.<sup>71</sup> SVGs are computed as mathematical expressions, or vectors, and can be used anywhere GIFs, JPEGs and PNGs are used.<sup>72</sup> Because SVG files consist of paths, they are ideal for all graphics, illustrations, and line art. Where they are not recommended is for images that require bitmapped pixels, namely for photographs. In all other instances SVG is an attractive choice for elastic layouts and has the added benefit of being a compact file size regardless of the size of a graphic on screen. Small SVG files also have the advantage of keeping bandwidth low and EPUB file sizes reasonable (see figure 7).<sup>73</sup> For publishing to the web, the SVG format is supported by Chrome, Firefox, Safari, and Opera. For ebooks, SVG is supported in Apple iBooks and has been added to the EPUB3 specification meaning the Kobo and Nook also support it, and more ereaders will support the format in the future.<sup>74</sup>

For instances where photographs are necessary and SVG will not suffice, there are a few CSS workarounds for other image formats that allow for scalability. The most common of which being the CSS shortcut introduced by Richard Rutter:

```
img { max-width: 100%; }75
```

---

70 Although Apple iBooks and other web browsers allow SVG fonts, they tend to have disastrous results when applied to a layout or web design.

71 Shelley Powers, "Using SVG For Flexible, Scalable, and Fun Backgrounds, Part I," *A List Apart*, January 26, 2010, <http://www.alistapart.com/articles/using-svg-for-flexible-scalable-and-fun-backgrounds-part-i/>.

72 Ibid.

73 Ibid.

74 "EPUB Content Documents 3.0: Recommended Specification 11 October 2011," IDPF, October 11, 2011, <http://idpf.org/EPUB/30/spec/EPUB30-contentdocs-20111011.html>.

75 Ethan Marcotte, "Fluid Images," *A List Apart*, June 7, 2010, <http://www.alistapart.com/articles/fluid-images/>.

## *The Elastic Approach to Responsive Design*

This rule instructs all images (JPEG, PNG, GIF, etc.) on a page to have a maximum width of the container in which the image is set. This CSS shortcut automatically resizes images to adhere to this rule.<sup>76</sup> To ensure all screens have crisp imaging, the highest resolution image is required when using this work around. With the use of this CSS code large images will then be scaled down to the appropriate size. This is an instance where percentages rather than ems are used within elastic layouts as a practical fix. Although each individual image could be defined to have a maximum width in em units, defining all images with a max-width of 100 percent saves time and added calculations. The only problem with this solution is the large amount of bandwidth required on mobile browser when viewing large image files optimal for larger screens. One less than perfect work around for this problem is to include smaller image files in the CSS file optimized for small screens.

For the Fontasizing.com, as a nod to traditional publishing, all images used on the website were created to appear as if done in woodcuts in Adobe Photoshop. Therefore, all graphics used could be converted to scalable vector graphics. These graphics were converted to SVG files within Adobe Illustrator and included in the site's HTML (see figure 7).

To include SGVs, there are currently three possible methods for encoding graphics to a page: the <embed> tag, the <object> tag, and the <iframe> tag. The <embed> tag is a deprecated feature for HTML5 and is not recommended as a result, and the <object> tag does not allow for JavaScript, which may be an issue for future publications should they require scripting features. As a result, the <iframe> tag was selected for including SVG graphics within the HTML:

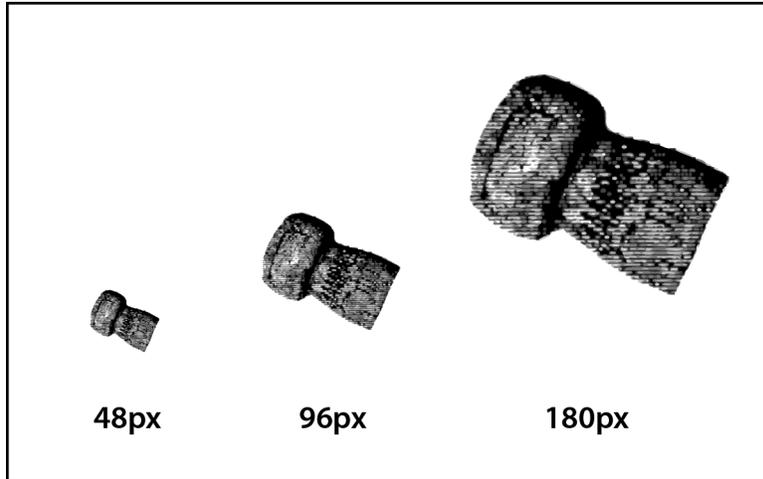
```
<iframe src="cork_woodcut.svg" ></iframe>
```

By utilizing the SVG format, when page proportions are increased or decreased the images on a page readily scale with the elastic layout and maintain their quality at every level.

---

<sup>76</sup> Marcotte, "Fluid Images," (2010).

*The Elastic Approach to Responsive Design*



**FIGURE 7:** *This Scalable Vector Graphic (SVG) of a cork, done in the woodcut style, is a single file shown at three different sizes. This single graphic is displayed at 48, 96, and 180 pixels squared with the potential to be increased many times over with no decrease in image quality. This file is 418kb and keeps bandwidth low even when the image is displayed at large sizes.*

When creating digital publications, if a framework like the Goldilocks Approach is used, many of the requirements for implementing a device independent, responsive design are available out of the box. This framework facilitates all of the necessary components to achieve device independence including web standard coding, relative units (the em), and media queries that allow for specific layouts that correspond to small, medium, and large screens. Once these components are established the functional components of the design are taken care of and the designer can move forward and work on the aesthetics of the design process. By using media queries there become many design configurations that can occur and carry with them the opportunity for designers to flex creative muscles that simply do not exist when designing for a single print edition. While there can be some minor setbacks when creating an elastic layout, particularly in terms of maintaining consistent em measurements and the sometimes tedious calculations that this task requires, they are relatively small pains when compared to the hours and dollars that would otherwise be spent coding new publications every three to five years if responsive, device

### *The Elastic Approach to Responsive Design*

independent strategies are not used. By implementing publication designs that can be read on a range of devices publishers will extend the life of their source files for the foreseeable future and open up publications to new markets across a growing number of mobile and handheld devices.

## CONCLUSION

This report focuses on responsive design as it relates to the interoperability and functionality of reading experiences across a range of devices within HTML-based publications. It is intended as a primer for responsive web design for those within the publishing industry, and not intended to replace the inspired writings of the many web designers and developers referred to within the report actively involved in expanding the theoretical and practical constructs of the web and its responsive components.

As Herbert A. Simon's theory of design notes, an artifact has to have an inner environment that is conducive to the goals of its users. With more readers opting for digital reading across tablets, smart phones, and personal computers, the inner coding of digital publications must be conducive to achieving interoperable reading. By embracing responsive design in the form of open-source coding established by the World Wide Web Consortium (w3c) beginning with HyperText Mark-up Language 5 (HTML5) and Cascading Style Sheets 3 (CSS3), publishers can develop products that deliver the same usability features available on the open web.

With more readers opting for on-screen reading experiences, it is imperative that publishers examine the means by which they are designing and releasing content. The publishing process is growing more detailed and complex in terms of the coding and languages used to express content and page design, and it is no longer enough to simply convert the printed page to the screen. Rather than bend to market pressures that may have temporary solutions for on-screen layouts based on screen resolutions and pixel measurements, publishers have a greater chance of attaining ebook longevity and increasing market reach by maintaining designs that are responsive, device independent, and compliant with open web standards. By adopting open web standards for coding in conjunction with relative units of measurement, publishers stand to maximize the investment of digitization efforts and succeed in increasingly connected reading environments.

Practical first steps for publishers are to choose which method of web design best suits their digital content. It is recommended that publishers begin to look at their publications from the perspective of responsive and adaptive layouts that serve optimal reading on multiple platforms and devices, and implement the use of elastic layouts that use ems as a unit of measure for all aspects of digital page design. For graphics it is the SVG format that is recommended for use in all instances when photographs are not required. Using elastic layouts and SVG images, in conjunction with CSS media queries, will result in pages that allow for optimal scalability and are independent of screen resolutions.

## Conclusion

For those publishers hesitant to apply responsive designs to their publishing schedule, one suggestion would be to begin with publications that are expected to be perennial titles. This will allow for additional time to be spent on creating beautiful and durable elastic layouts, and the open source coding used to produce them will be flexible should future reformatting be required.

Ideally, all publishers would disband from proprietary formats and coding and embrace the open-source EPUB3 format in conjunction with responsive design practices. However, due to the market share of corporate distributors this is not financially viable for many publishers at this time. As a result, it is recommended that publishers adopt device independent coding for the majority of their designs, by subscribing to the standards and practices set forth by the w3c. With clean, responsive CSS and HTML coding as a foundation for digital publications, publishers can then alter and reformat these files to suit the needs of distributor-specified formats as needed. By making open-source coding for publications a priority, publishers and designers can take the first step toward device independence for the publishing industry.

### *But Is It needed?*

Publishers may already be overburdened with the task of producing print products and converting print publications and raw text to basic ebooks for distribution via corporate retailers. The thought of pursuing responsive publication designs may be low on a long list of pressing priorities. In this regard, publishers may ask what does responsive design matter in relation to simple text-only EPUBS?

With corporate distributors increasingly threatening usurpation of the industry by creating and promoting self-publishing avenues, design is one facet, along with the superior editorial process, that distinguishes publishers from self-publishing platforms in a meaningful way. If publishers of digital editions offer little in terms of aesthetics or functionality over Apple or Amazon, why would an author not opt for self-publication, particularly when self-publishing profit margins are higher?<sup>77</sup>

Many publishers may already comprehend the gravity that design plays on the final product and convert a number of books to fixed-layout EPUBS. The problem with this method is that publishers are hedging their bets on the constraints of one device or one platform that has the possibility of becoming

---

<sup>77</sup> Joe Konrath, "Konrath Ebooks Sales Top 100k," *A Newbie's Guide to Publishing*, September 22, 2010, <http://jakonrath.blogspot.ca/2010/09/konrath-ebooks-sales-top-100k.html>.

## Conclusion

obsolete in a number of years. Just as most traditional publishers would not print on acid-based paper due to its inevitable decline in quality and longevity, so must publishers recognize that each device has a limited shelf life that may expire at any time. Pigeonholing publications into one distribution channel today limits market reach and will mean further — *expensive* — digital conversions in the future, not to mention the possible discord that will emerge with authors when a work is no longer available for purchase in the desired format or platform.

With corporate interests vying to divide the book market, publishers need to be aware of their complicity in relinquishing the digital market. Leaving design out of ebook production, or only creating designs for one distribution channel, allows predatory distributors to whittle away what little is left under publishers' authority. By making functional design a greater priority, publishers can push back and make a name for themselves in the online market. Embracing elastic layouts founded on a relative unit of measure, paired with other open web standards established by the w3c, gives publishers the ability to remove themselves from the burden of proprietary formats and coding to ensure that publications aren't beholden to one format or distribution channel. By designing responsive and device independent publications from the outset, publishers can maximize the value of digitization efforts and take strides to increase the longevity of publications source files for years to come.

## BIBLIOGRAPHY

“Accessibility Principles.” w3c. February 20, 2012. <http://www.w3.org/WAI/intro/people-use-web/principles#distinguishable> (accessed June 2012).

Alexander, Rhoda. “Apple’s Toughest Competition in the Fourth-Quarter Tablet Market Was...Apple.” iSuppli.com. February 16, 2012. <http://www.isuppli.com/Display-Materials-and-Systems/News/Pages/Apples-Toughest-Competition-in-the-Fourth-Quarter-Tablet-Market-Was-Apple.aspx> (accessed June 2012).

Allsopp, John. “A Dao of Web Design.” *A List Apart*. April 7, 2010. <http://www.alistapart.com/articles/dao/> (accessed June 2012).

“Apple iPhone 4S Technical Specifications.” Apple. <http://www.apple.com/iphone/specs.html> (accessed June 2012).

“Apple iPhone 3GS Technical Specifications.” Apple. <http://www.apple.com/iphone/iphone-3gs/specs.html> (accessed June 2012).

Armstrong, Chris. “The Goldilocks Approach to Responsive Design.” Designbyfront.com. April 29, 2011. <http://www.designbyfront.com/workinprogress/article/the-goldilocks-approach-to-responsive-design> (accessed June 2012).

Bergel, Marguerite, Ann Chadwick-Dias, Lori LeDoux, and Tom Tullis. “Web Accessibility for the Low Vision Use.” Fidelity Investments. 2005. [http://www.eastonmass.net/tullis/presentations/Web\\_Accessibility\\_for\\_Low\\_Vision.pdf](http://www.eastonmass.net/tullis/presentations/Web_Accessibility_for_Low_Vision.pdf) (accessed June 2012).

Bjarnason, Baldur. “The iBooks 2.0 textbook format.” baldurbjarnason.com. January 19, 2012, <http://www.baldurbjarnason.com/notes/the-ibooks-textbook-format/> (accessed June 2012).

Boulton, Mark. “A Richer Canvas.” Markboulton.co.uk. March 24, 2011. <http://www.markboulton.co.uk/journal/comments/a-richer-canvas> (accessed June 2012).

## Bibliography

- Bringhurst, Robert. *The Elements of Typographical Style*. Vancouver: Hartley & Marks, 2008.
- Butler, Mark, Fabio Giannetti, Roger Gimson, and Tony Wiley. "Device Independence and the Web." HP Laboratories Bristol. October 2, 2002. [http://www.hpl.hp.com/research/papers/2003/device\\_independence.pdf](http://www.hpl.hp.com/research/papers/2003/device_independence.pdf) (accessed June 2012).
- Carvalho, Lawrence and Christian Heilmann. "Text-Resize Detection." *A List Apart*. September 12, 2006. <http://www.alistapart.com/articles/fontresizing> (accessed June 2012).
- "Cascading Style Sheets Level 2 Revision 1 (css 2.1) Specification." w3c. June 7, 2011. <http://www.w3.org/TR/CSS2/syndata.html#length-units> (accessed June 2012).
- Castro, Liz. "Creating pop up footnotes in EPUB3 (and thus in iBooks)." *Pigs, Gourds, and Wikis*. May 22, 2012. <http://www.pigsgourdsandwikis.com/2012/05/creating-pop-up-footnotes-in-epub-3-and.html> (accessed June 2012).
- Castro, Liz. "EPUB3 has finally arrived." *Pigs, Gourds, and Wikis*. July 3, 2012. <http://www.pigsgourdsandwikis.com/2012/07/epub-3-has-finally-arrived.html> (accessed August 2012).
- "Changing Demographics of Tablet and eReader Owners in the US." NielsenWire. August 25, 2011. [http://blog.nielsen.com/nielsenwire/online\\_mobile/changing-demographics-of-tablet-and-ereader-owners-in-the-us/](http://blog.nielsen.com/nielsenwire/online_mobile/changing-demographics-of-tablet-and-ereader-owners-in-the-us/) (accessed June 2012).
- Chen, Brian X. "iPhone 4s 'Retina' Display Claims Are False Marketing." *Wired.com*. June 9, 2010. <http://www.wired.com/gadgetlab/2010/06/iphone-4-retina/> (accessed June 2012).
- Clark, Joe. "Web Standards for E-books." *A List Apart*. March 9, 2010. <http://www.alistapart.com/articles/ebookstandards/> (accessed June 2012).

## Bibliography

- “Device Independence Principles: w3c Working Group Note 01 September 2003.” w3c. September 1, 2003, <http://www.w3.org/TR/di-princ/> (accessed June 2012).
- Diener, Matthew. “Media Queries in ePUBs.” *ePUB secrets*. February 9, 2012. <http://epubsecrets.com/media-queries-in-epubs.php> (accessed June 2012).
- “Diversity of Web Users,” w3c, February 20, 2012, <http://www.w3.org/WAI/intro/people-use-web/diversity.html#visual> (accessed June 2012).
- “EPUB Content Documents 3.0: Recommended Specification 11 October 2011.” IDPF. October 11, 2011. <http://idpf.org/epub/30/spec/epub30-contentdocs-20111011.html> (accessed June 2012).
- “eReaders for the elderly – A useful gadget for older people.” eBookAnoid. March 3, 2010. <http://www.ebookanoid.com/2010/03/03/ereaders-for-the-elderly-a-useful-gadget-for-older-people/> (accessed June 2012).
- Felici, James. *The Complete Manual of Typography*. Berkeley: Peachpit, 2012.
- Gillenwater, Zoey Mickley. *Stunning CSS3: A project-based guide to the latest in CSS*, (Berkeley: New Riders, 2011).
- “Global Online Consumers and Multi-Screen Media: Today and Tomorrow.” Nielsen.com. May 2012. <http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2012-Reports/Nielsen-Multi-Screen-Media-Report-May-2012.pdf> (accessed June 2012).
- Griffiths, Patrick. “Elastic Design.” *A List Apart*. January 9, 2004. <http://www.alistapart.com/articles/elastic/> (accessed June 2012).
- Hjerde, Morten. “Mobile screen size trends.” *Sender 11*. April 15, 2008. <http://sender11.typepad.com/sender11/2008/04/mobile-screen-s.html> (accessed June 2012).

## Bibliography

- Kasdorf, Bill. "EPUB3 (Not Your Father's EPUB): Opening Pandora's Box in the World of E-Books." *Information Standards Quarterly* 23, no. 2 (2011). <http://www.niso.org/publications/isq/2011/v23no2/kasdorf> (accessed June 2012).
- "Kindle Format 8." Amazon. 2012. <http://www.amazon.com/gp/feature.html?docId=1000729511> (accessed June 2012).
- Knight, Kayla. "Responsive Web Design: What It Is and How To Use It." *Smashing Magazine*. January 12, 2011. <http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/> (accessed June 2012).
- Konrath, Joe. "Konrath Ebooks Sales Top 100k." *A Newbie's Guide to Publishing*. September 22, 2010. <http://jakonrath.blogspot.ca/2010/09/konrath-ebooks-sales-top-100k.html> (accessed June 2012).
- Koch, Peter-Paul. "A pixel is not a pixel is not a pixel." *Quirksmode*. April 20, 2010. [http://www.quirksmode.org/blog/archives/2010/04/a\\_pixel\\_is\\_not.html](http://www.quirksmode.org/blog/archives/2010/04/a_pixel_is_not.html) (accessed June 2012).
- "Combining meta viewport and media queries." *Quirksmode*. September 14, 2010. <http://www.quirksmode.org/blog/archives/2010/09/combining-meta.html> (accessed June 2012).
- Krug, Steve. *Don't Make Me Think!* Indianapolis: New Riders, 2000.
- "List of displays by pixel density." *Wikipedia*. [http://en.wikipedia.org/wiki/List\\_of\\_displays\\_by\\_pixel\\_density](http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density) (accessed June 2012).
- Marcotte, Ethan. "Fluid Grids." *A List Apart*. March 3, 2009. <http://www.alistapart.com/articles/fluidgrids/> (accessed June 2012).
- "Fluid Images." *A List Apart*. June 7, 2010. <http://www.alistapart.com/articles/fluid-images/> (accessed June 2012).
- "Responsive Web Design." *A List Apart*. May 25, 2010. <http://www.alistapart.com/articles/responsive-web-design/> (accessed June 2012).

## Bibliography

- McLuhan, Marshall. *Understanding Media*. New York: New American Library, 1964.
- Miner, Wilson. "Setting Type on the Web to a Baseline Grid." *A List Apart*. April 9, 2007. <http://www.alistapart.com/articles/settingtypeontheweb> (accessed June 2012).
- Mod, Craig. "A Simpler Page." *A List Apart*. January 11, 2011. <http://www.alistapart.com/articles/a-simpler-page/> (accessed June 2012).
- "Post-Artifact Books and Publishing." *Craigmod.com*. October 2010. [http://craigmod.com/journal/post\\_artifact/](http://craigmod.com/journal/post_artifact/) (accessed June 2012).
- Pilgrim, Mark. "Let's Take This Offline." *Dive Into HTML5*. 2011. <http://fortuito.us/diveintohtml5/offline.html>. (accessed January 9, 2012).
- Palmer, Jonathan W. "Web Site Usability, Design, and Performance Metrics." *Information Systems Research* 13, no. 2 (2002): 151-167.
- "Designing for Web Site Usability." *IT Systems Perspective* (2002): 102-3.
- Powers, Shelley. "Using svg For Flexible, Scalable, and Fun Backgrounds, Part I." *A List Apart*. January 26, 2010. <http://www.alistapart.com/articles/dao/> (accessed June 2012).
- "PubIt! Epub Formatting Guide." *Barnes and Noble*. 2010. [https://simg1.imagesbn.com/pimages/pubit/support/pubit\\_epub\\_formatting\\_guide.pdf](https://simg1.imagesbn.com/pimages/pubit/support/pubit_epub_formatting_guide.pdf). (accessed October 2011).
- Simon, Herbert A. *The Sciences of the Artificial*. Cambridge: MIT Press, 1996.
- Rainie, Lee, Kathryn Zickuhr, Kristen Purcell, Mary Madden, and Joanna Brenner. "The rise of e-reading." *PewInternet.org*. April 4, 2012. <http://libraries.pewinternet.org/2012/04/04/the-rise-of-e-reading/> (accessed June 2012).
- Rutter, Richard. "Compose to a Vertical Rhythm." *24ways*. December 12, 2006. <http://24ways.org/2006/compose-to-a-vertical-rhythm> (accessed June 2012).

## Bibliography

- Schwartz, Barry. "Google Finally Takes A Clear Stance On Mobile SEO Practices." SearchEngineLand.com. June 6, 2012. <http://searchengineland.com/google-finally-takes-a-clear-stance-on-mobile-seo-practices-123543> (accessed June 2012).
- Sieling, Todd. "The Curation of Future Moments." February 21, 2012. Slide 48. (accessed June 2012).
- "Study reveals many Canadians unnecessarily living with vision loss." CNIB. August 5, 2012. [http://www.cnib.ca/en/Pages/Study\\_reveals\\_many\\_Canadians\\_unnecessarily\\_living\\_with\\_vision\\_loss.aspx](http://www.cnib.ca/en/Pages/Study_reveals_many_Canadians_unnecessarily_living_with_vision_loss.aspx) (accessed August 2012).
- Tan, Jon. "The Incredible Em & Elastic Layouts with css." Jontangerine.com. September 25, 2007. <http://jontangerine.com/log/2007/09/the-incredible-em-and-elastic-layouts-with-css> (accessed June 2012).
- "Top 10 High Resolution Smartphones." *Tech & Trend*. November 17, 2011. <http://www.ibtimes.com/articles/251596/20111117/iphone-4s-10-high-resolution-smartphones.htm#page1> (accessed June 2012).
- "Using the viewport meta tag to control layout on mobile browsers." Developer. mozilla.org. May 26, 2010. [https://developer.mozilla.org/en/Mobile/Viewport\\_meta\\_tag](https://developer.mozilla.org/en/Mobile/Viewport_meta_tag) (accessed June 2012).
- Verou, Lea. "Every Time You Call a Proprietary Feature 'css3,' a Kitten Dies." *A List Apart*. February 14, 2012. <http://www.alistapart.com/articles/every-time-you-call-a-proprietary-feature-css3-a-kitten-dies/> (accessed June 2012).
- "Web Accessibility Initiative (WAI)." w3c. 2011. <http://www.w3.org/WAI/> (accessed June 2012).
- "World Wide Web Consortium (w3c) Launches International Web Accessibility Initiative." w3c. April 7, 1997. <http://www.w3.org/Press/WAI-Launch.html> (accessed June 2012).