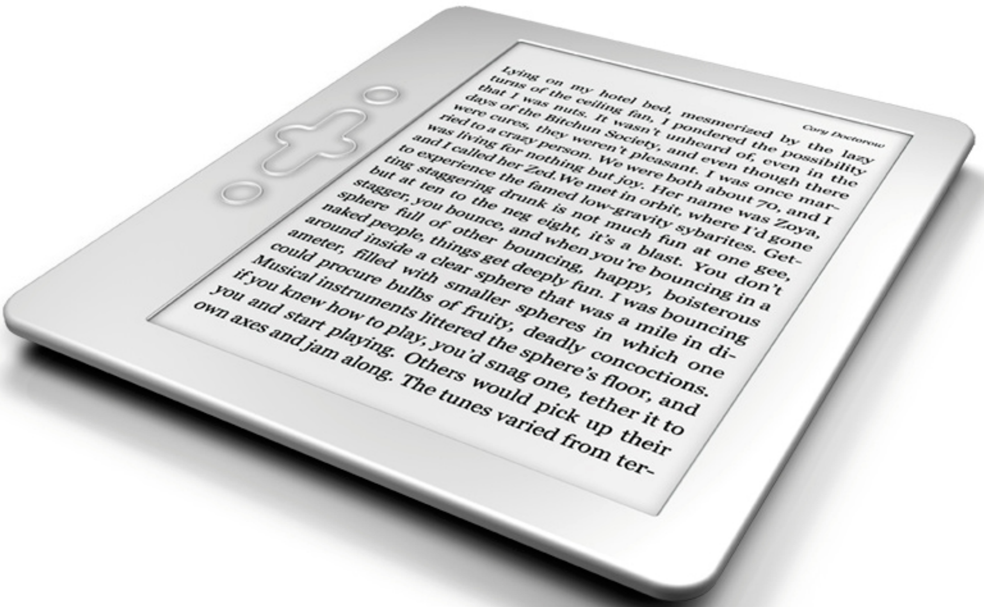
Electric Book Works
electricbookworks.com

EBW



Cory Doctorow

Lying on my hotel bed, mesmerized by the lazy turns of the ceiling fan, I pondered the possibility that I was nuts. It wasn't unheard of, even in the days of the Bitchun Society, and even though there were cures, they weren't pleasant. I was once married to a crazy person. We were both about 70, and I was living for nothing but joy. Her name was Zoya, and I called her Zed. We met in orbit, where I'd gone to experience the famed low-gravity sybarites. Getting staggering drunk is not much fun at one gee, but at ten to the neg eight, it's a blast. You don't stagger, you bounce, and when you're bouncing in a sphere full of other bouncing, happy, boisterous naked people, things get deeply fun. I was bouncing around inside a clear sphere that was a mile in diameter, filled with smaller spheres in which one could procure bulbs of fruity, deadly concoctions. Musical instruments littered the sphere's floor, and if you knew how to play, you'd snag one, tether it to you and start playing. Others would pick up their own axes and jam along. The tunes varied from ter-

Designing for digital

What print-book designers should know about ebooks

Designing for digital

What print-book designers should know about ebooks

Arthur Attwell

Copyright and licensing

Designing for Digital: What print-book designers should know about ebooks

First published by Electric Book Works in 2010, revised in 2014

Print edition ISBN: 978-1-920218-53-9

PDF edition ISBN: 978-1-920218-50-8

EPUB edition ISBN: 978-1-920218-51-5

Contents

The author 9

Note 11

Introduction 13

An overview of digital publishing 15

- What is an ebook? 16
- Ebook formats 17
  - Standardisation 17
  - Major formats 19
  - PDF 19
  - EPUB 20
  - Mobi 21
- Digital Rights Management (DRM) 21

An intro to HTML and CSS code 23

- Content, form and function 23
- What are XML and XHTML? 28

Converting books to ebooks 31

- First-step: decisions 31
- Creating a PDF ebook 35
- Converting to EPUB using Calibre 36
- Creating EPUB from InDesign 37
  - Working in InDesign 37
  - After exporting to EPUB 38

Practical typesetting for print and digital 41

- Text flow 41
- Images 42

- Fonts 42
  - InDesign CS3 quirks 44
- Styles 44
- Local formatting 46
- Pagination and master pages 46
- Spacing 47
- Text alignment 47
- Background colours 48
- Tables of contents 48
- Covers 50
- Metadata 50
- Embedding multimedia 51
- Getting inside the EPUB 51
- Checking ebooks 52

Further reading 53

Feedback 55

The author

Arthur Attwell is co-founder and director of Electric Book Works, where he develops new and better ways to publish in emerging markets. There he started Bettercare, which publishes self-managed course books for nurses and midwives. He's a co-founder of the Digital Minds Network, a group of digital-publishing experts working in developing countries. Arthur is also the author of the EBW Knowledge Base and Encyclopaedia Britannica's 'ebook' entry.

He worked in educational and scholarly publishing for two multinational publishers before starting EBW in 2006. He has presented at conferences in South Africa, India, Europe and the US, and was runner-up for the British Council's International Young Publishing Entrepreneur award in 2009. He won 'Most Entrepreneurial Startup' with Paperight at O'Reilly Tools of Change for Publishing in New York in 2013.

He lives in Cape Town with his wife Michelle Matthews and their son Aidan.

Note

Please remember that this is a fast-changing industry. This material was written in 2010 and revised in early 2014. By the time you read this, some of it could be out of date.

Introduction

I wrote this book as a course at Electric Book Works for book designers, typesetters and production managers, though it's useful for anyone working in book publishing. When I say 'design' I usually mean every aspect of the project design of a book: concept, commissioning, writing, editing, layout, proofing, production and distribution. But in this short book I focus on the things most production staff deal with on a daily basis: the process of typesetting. Then, what you learn here has implications for every stage of publishing books. There are things the commissioning editor can do to help prepare a book for its future life as an ebook, for instance, if they understand how their decisions affect ebook production.

This book's in four parts:

1. Overview: To make sensible decisions, you need an overview of the digital publishing industry, where ebook formats like EPUB fit in, and how traditional book files are converted to ebook files.
2. An intro to code: Just like you once learned how litho printing works, now you need some technical knowledge, especially about major ebook file formats, and how HTML and CSS code works.
3. Converting ebooks: Practically, how are print books converted into ebooks? This is a quick primer of the basics.
4. Practical typesetting: When you get to your office, you should be able to start prepping files (mainly in InDesign) in ebook-friendly ways. This is the crux of what you need to do.

My priority is to equip you to design books in a way that is not print-centric. As books are read more and more on screens, the designer's job – and the job of those briefing and managing designers and typesetters – changes in important ways. The best, most valuable designers will soon be those who future-proof their content, making it easily reusable for distribution to laptops, ereaders, mobile phones, and a range of apps and devices we haven't heard of yet.

In short, you need to be able to work towards much more than good old PDF. In particular, you need to know how HTML works, because it has become the most important computer language in publishing.

I’m going to assume that you like working in something like InDesign. You want eventually to create print books and ebooks at the same time, and where you can, to make those ebooks look good. I don’t try to give you aesthetic design guidance; I simply aim to give you the knowledge you need to prepare your print-oriented work for on-screen reading, particularly as ebooks.

An overview of digital publishing

Why are you reading this? That's not a rhetorical question (you can even tell me right now on Twitter). I suspect it's because you've begun dabbling in ebooks as a designer or production manager, or you've been asked to by your boss or a client. Or you're simply curious about the ever-increasing volume of talk about ebooks, because you know sooner or later they're going to be a part of your job.

We know ebooks are a very important part of the publishing landscape right now, especially as publishers try to recover a share of people's attention from digital devices currently used mostly for music, video and games. But there there is another, much more important process happening: ebooks are a by-product of a great human enterprise, the digitisation of literature. Digitised literature will be more easily shared and sold, and in time it will make education and storytelling cheaper and more abundant.

Now is only the beginning. Despite the incredible size and depth of the Internet as we know it, it's still far, far younger than the world's paper literature. Ebooks are the most apparent, easily monetised evidence of all our efforts to add the value and volume of paper literature to the great database of knowledge that is the Internet. Everything you learn today should be seen in that context. You are part of the greatest knowledge-curation process of all time.

The Internet we all want is one that easily grows into a more and more powerful, increasingly automated way to create and move large amounts of information, and one that helps us make a living in the process. To do that, we have to fill it with information that will be useful for a long time, and that can be easily found and manipulated by machines. (As with any database: rubbish in, rubbish out.) Every ebook is a piece of that database. It's not only a freestanding product.

So, if you're working with ebooks, you need to know how they fit into the Internet, both as consumer products and as an Internet technology.

Today, the ebook industry is most vibrant in the US, followed by the UK and Western Europe. This has been catalysed by more and more people buying dedicated ebook-reading devices, or e-ink ereaders, such as those from Amazon, Barnes & Noble and Kobo. The e-ink has been important because it is profoundly different to read on than on a backlit computer or phone screen. E-ink's text and images are not made of light, like on a backlit screen, but of thousands of microcapsules that rearrange with every page-refresh into the shapes of letters and pictures. There are usually about 150 to 200 capsules (think pixels or dots) in every inch of screen. Reading on an e-ink screen really is almost like reading on paper. This has made ebooks easier for paper-book lovers to adopt.

The most popular ereaders also make it easy to buy ebooks on the device, by connecting automatically to their parent company. Amazon's best at this right now: you can browse the Amazon store, buy an ebook, and be reading it within about a minute, anytime, anywhere.

E-ink readers were just the first big win, though. Far more people are downloading ebooks on their phones and tablets, and reading them in apps from Amazon, Apple, Google and others. The screens may be backlit, but they're easy to read on, and the expense of the device is easier to justify because it can do so much more, like email, visiting websites, playing games, and using a myriad useful apps.

There isn't space or time here to address many of the common questions and arguments about ereading here. ('Will ebooks replace print?', 'I can't read a book on a small screen!', 'What about bookstores?', etc.) I highly recommend an article by John Siracusa called 'The Once and Future Ebook'. It's a few years old now, but as brilliant as ever. Siracusa talks about the history of ebooks, and where and why they're important.

What is an ebook?

What is an ebook? That's easy to answer until you think hard about it. So, first, let's not think hard about it: an ebook is a book you read on a screen. The author and publisher have distributed it as a digital file.

Now, let's complicate things. Here's a list of ebooks features, starting with the most obvious, and then getting into some grey areas:

1. Ebooks are books read on a screen.
2. Ebooks are sometimes converted versions of paper books (created from scans or print PDFs), and sometimes they're designed as ebooks from the start.
3. Ebooks are stored and delivered in carefully chosen file formats (much like word processing is mostly in Word's .doc format, and spreadsheeting mostly in Excel's .xls format). This means they can be read offline – without having to constantly get info from a web server somewhere else.
4. Ebooks depend on software to be opened and read (like a .doc file depends on a good Word-like program). The quality of this software determines the quality of the reading experience.
5. Consumers expect ebooks to cost less than print – usually 50%–80% of the print book price. (Pricing is a whole other issue for a separate discussion.)
6. Ebooks are found and downloaded from the Internet.
7. Ebooks are sometimes read online, much like a live website. (The online/offline distinction is important for all kinds of reasons.)
8. Ebooks sometimes include sound, video, or interactive forms, quizzes, or even games.

Those are some fundamentals, just to get us started.

Ebook formats

A file format is a way of storing information. A set of structures, intentions, computer languages and human languages that have been agreed on by a group of computer engineers and subject experts. Each file format stores information differently.

So, file formats are carefully designed by groups of people to achieve very particular aims. There have been many ebook formats, and over time some have proved more resilient than others.

Standardisation

When we standardise something, we all agree to work with the same set of rules. For example, the rules of soccer have been standardised around the world so that anyone can play against anyone else without having to learn new rules. Standards can be official (written down in a formal document) or de facto (generally accepted conventions).

Only in the last few years has the publishing world settled on standards for ebooks.

There are established and emerging standards for file formats, purchasing processes, security, pricing, availability, and so on. The standards we care about most here are around file formats.

- Standard formats make it easier to open and read files – as a customer you don't want to have to think about which file format you're buying, it must just work on your ereader.
- The more businesses use the same file formats, the easier it is to move large numbers of ebooks around, for instance between publishers, distributors and retailers.
- If everyone's using the same formats, more companies create tools to make and edit them. For instance, in word-processing, almost everyone uses MS Word, even though it's not actually the best file format for storing formatted text. The standard became more important than its technical quality.

Those who come up with new file formats in the early days of a new industry try to use their formats to make loads of money, either by charging a licence fee to software companies for using that format, like MS Word, MP3 and GIF in the early days; or by forcing people to use only their software to open or edit that format. These formats are closed (proprietary), as opposed to being open (free for anyone to use).

Closed formats are those that are still locked down by their owners. No one else is allowed to produce files in those formats without the permission of their owners, who often charge a substantial licence fee. The owners do not publicly publish details (a specification) of how to make files in their formats.

Open formats are free for anyone to create files in, and the specification (how to build the files) is freely available to anyone.

Ultimately the most popular file formats end up being open and free to use. Ironically, the ones that start as open tend to catch on more slowly (e.g. OGG, an alternative to MP3), because they don't have corporate marketing budgets behind them. While ebook file-format standards are settling, we still have to work with a combination of closed and open formats.

Sometimes, the most popular formats start as proprietary and then become open. The best example is PDF. PDF was owned by Adobe, who in 2008 decided to make it an open format, effectively donating it to the world, a move for which they deserve a lot of credit. They've done well to remain the premier PDF software editing company since then.

Major formats

Among major ebook file formats, the main distinguishing features of each format include:

- whether the content is static (e.g. fixed PDF pages) or reflowable (there are no pages, content fills the available screen area, according to font size at the time);
- whether the format is open (anyone can make that format) or proprietary (only the owner of the format can make it or licence others to do so);
- how easily and reliably the format can be converted (by automated systems) into other formats.

There have been dozens of ebook file formats in ebook stores and publishers' repositories over the years. Today, there are only three worth learning about:

- PDF,
- EPUB,
- MOBI (which is the same as PRC).

PDF

We all know PDF as a format for sharing static pages. If you embed fonts (the common default), what you create is exactly what your reader will see. PDF is directly analogous to the printed page. This makes it a nice, safe choice for designers. And it's easy to create with existing tools (like InDesign, Acrobat Pro, Word and OpenOffice).

PDF is actually a package for various file format, which gives it its power. It stores both vector and bitmap images, can embed subsets of fonts, contains XML metadata, and can contain flash video and interactive forms that communicate with a remote server. It's far more powerful than most people realise. Produced well (for instance, tagged behind the scenes with structural information about the document), PDF can even be reflowed by some PDF readers (the text broken out of its paged format and flowed into the available screen area). And it can be easily navigated by reading software for the visually impaired. All these reasons should make it perfect for ebooks.

But it has a downside: it's too easy, and therefore common, to produce PDFs that are badly created. From a technical point of view, they can be comprehended by humans, but not by machines, rendering them useless to software that tries to navigate or reflow them. So once something's in PDF, it usually requires a part-manual process to convert it into anything else.

Not all ereading software opens PDF files, and those that can do so badly or only partly (e.g. they don't support PDF navigation).

EPUB

If you haven't already, you really need to take a look at an EPUB ebook to get a feel for what it is. (A simple, free example is EBW's ebook edition of John Siracusa's article, 'The Once and Future Ebook'.) You'll need EPUB-reading software to open it. On whatever device you're using, just search for 'EPUB reader' and you'll find many options; just pick the most popular or highly rated for now.

Over time, you're going to have to use several apps to see how different ebooks look in each one. This will be important for testing your ebooks before releasing them.

EPUB has been the fastest-growing format, and the one I'll talk most about here. It's an open standard developed by volunteer members of the International Digital Publishing Forum (IDPF). It's young (officially published in 2007), so it has its technical teething issues, but on the whole it's simple enough to (eventually) be easy to create, while being sophisticated enough to contain a wide variety of information, including embedded video and SVG (a format for storing vector artwork).

Essentially, EPUB is just a website in a .zip folder. If you took a static website's files (HTML, CSS and images) and put them in a .zip file, you'd have something much like an EPUB file. EPUB has a few extra features (like the file that creates clickable navigation), but in essence that's all an EPUB is: web pages in a .zip file. That makes it fairly easy for developers to create software that reads EPUB (because you just adapt existing web-browsing software) and software that saves documents as EPUB. That software is getting better and better (especially InDesign and Sigil, but more on that later). Now, as publishers, we have to learn how to use these new tools. We're like experienced carpenters learning to work metal for the first time: we have to understand both new tools and new materials.

EPUB also has its inherent failings. As anyone who's designed for the web will know, the downside of having a format that stores its content much like a website is that it has all the downsides of website design and development: that is, massive inconsistency in the quality of the product and the software that reads and renders it.

Mobi

MOBI (also called PRC) is Amazon's main format for ebooks distributed on Amazon Kindle. The format was originally developed by Mobipocket, an ebook retailer that Amazon acquired in 2005. If you're publishing an ebook to the Kindle (e.g. through Amazon's Digital Text Platform), you can upload a MOBI file confident that the Kindle will display it largely as you intended.

MOBI is actually based on the same predecessor formats as EPUB. So MOBI and EPUB are very similar. This makes it very easy to convert between MOBI and EPUB. For converting the best tool is the free, open-source Calibre.

Digital Rights Management (DRM)

Digital Rights Management is the bugbear of the content industries, from film to music to publishing. It makes for great speculative discussions, so find out more if you like a good fight at your next publishing conference.

All you need to know right now is that DRM is any technical measure that restricts what you can do with a file. Usually, DRM stops you from copying or printing an ebook – it's the publisher's or retailer's attempt to slow piracy. It's like putting the ebook in a lockbox. Amazon and Apple have their own DRM schemes, and almost everyone else uses Adobe's. Ebooks locked by an Adobe Content Server can only be opened in software that supports Adobe DRM (such as Adobe Digital Editions), and only when that application has been registered with your Adobe username and password.

Note that while EPUB is an open file format, once it's wrapped in DRM it essentially becomes proprietary (since the DRM component is proprietary). This causes a lot of confusion among publishers and consumers. People tend to associate ‘open’ with free (as in speech, and sometimes as in beer), and companies that use EPUB like to trumpet that they’re using an open standard. But as soon as they apply DRM to an EPUB file, there is nothing open about the file at all. For instance, in order to make software that opened Adobe-DRMed ebooks, you have to pay Adobe a huge licence fee: EPUB or PDF locked with Adobe DRM is effectively proprietary.

Is that a problem? Well, no one knows for sure, there just isn’t enough data comparing the value and effect of DRMed ebooks with DRM-free ebooks. The obvious downsides of proprietary formats are that:

- they are controlled by one organisation, which leaves technical progress at that organisation’s discretion, and
- licence fees to read the format can usually only be paid by bigger companies, slowing the broader development of devices and software that support it.

Some would argue that the money to be made from proprietary formats enables or drives innovation and/or consumer adoption. You decide!

An intro to HTML and CSS code

Content, form and function

All of your work in digital will involve this key principle: in digital documents, we always separate content from its appearance.

- Content is your text and raw images (usually stored in HTML).
- Form is how that content should appear to humans (usually controlled by CSS).

We connect content from its appearance by defining its function. Function is what that content is meant to do: explain, signpost, attract attention, provide peripheral info.

For people new to ebooks, the biggest shock is usually this: you cannot determine or control the appearance of an ebook as you can with print. You don't decide what your reader will see. You just provide your design preferences, and hope that your reader's software and their display choices won't make a mess of it. Beyond that, your job is to structure the content really well, so that no matter how it looks, it can be read easily in the widest possible number of applications.

I can hear the typographers crying. Why and how has this happened? Hundreds of years of typography for nothing? Not quite, there are good reasons for the way things are right now, and they're all really just the Internet's teething problems. But to understand what's going on, you have to know about HTML and CSS.

When you read anything on a computer (or mobile phone or ereader), you're reading your software's human-readable representation of machine-readable information. Here's an example:

- Only-machine-readable information: 0110100001101001
- Human-readable representation: Hi
- Human and machine-readable: `<html>Hi</html>`

This example does make some assumptions:

Information	Assumption
0110100001101001	The computer knows you're using the ascii standard for encoding characters.
Hi	The human knows you're speaking English.
`<html>Hi</html>`	The computer has HTML-capable software installed, and the human knows what HTML looks like.

Those assumptions are very, very important. They define the standards we're using. With our assumptions/standards in place, the trick to making ebooks is to combine machine-readable information and human-readable information in one, well-crafted package.

Print-centric book design involves working with very little machine-readable information. Or rather, the tools we use hide the machine-only stuff from us. For instance, when you apply an InDesign style to a paragraph, all that really matters for print is that, on your screen, you get an appearance that is consistent and appealing to human beings. But behind the scenes, in the depths of your computer, InDesign is marking that paragraph with technical information that will give the right instructions about appearance to any future machine that opens your file.

Ebooks-making tools don't hide as much of the machine stuff from us (the latest versions of InDesign do try). This is because the standards for ebooks are still settling, and the software can't hide what it can't completely standardise.

So, if you're going to be ebook-aware in your work, you need to know what's happening to your machine-readable information behind the scenes. That starts with learning a little HTML.

HTML is human sentences marked with tags about those sentences. There is a standard list of possible tags, and the computer needs those tags to know what's going on in the document. Adding a tag to content is called 'marking up'. So HTML is a 'markup' language.

Many publishers have been using tagging for human typesetters for years. For instance, they might have this in an edited manuscript:

```
 [A head]Carving wooden animals

[box]Here you’ll learn how to carve wooden
animals.[end box]
```

The tags are in square brackets, and the typesetter is meant to use them as a guide when applying the design spec, and remove them from the text as they work through it.

That is exactly what a computer does with HTML.

Let’s do that for our wooden animals example:

```
 <html>

  <h1>

    Carving wooden animals

  </h1>

  <div class="note">

    <p>

        You’ll need a chisel for this chapter.

    </p>

  </div>

</html>
```

That may suddenly look a lot more complicated. But just take it step by step:

The tag `<html>` tells a machine that this is an HTML document. So the computer knows to follow the rules of HTML when reading it. The tag with the slash `</html>` says where the HTML document ends. If a web browser sees that document, it knows what to do with it: display the content as a web page.

I've now got a heading tagged `<h1>` (for heading-level 1).

My note text I've put in a `<div>`, for 'division'. Since a div can be used to mark off any section of text, I have to say what kind of div this is. So I've put this div in a class I've called `note`. I can make up my own class names, and I think 'note' describes this feature's function well. A class is an attribute that I use to describe to an element for the computer, like a `<p>` or a `<h1>` element, much like an adjective describes an attribute of a noun.

The sentence inside the `<div>` is a paragraph. So I mark it up with a `<p>` tag.

Note how I've indented the lines to show how some elements are nested inside others (like the paragraph is nested inside the note div). This makes the structure easier for humans to grasp.

Structure is also important for computers. In addition to its standard list of tags, HTML has rules about structure. For instance, you can't put a paragraph inside a heading, like this:

```
<h1>

  <p>

    This paragraph shouldn't be inside heading
tags. It's invalid HTML.

  </p>

</h1>
```

A heading like `<h1>` is completely different from a paragraph `<p>`. This invalid HTML would just break an ebook.

Now, with the markup in place, the machine can finds its way around the document. It knows the document is written in HTML. It knows where the heading, note, and paragraph go. The information about a document is called metadata – data about data. Metadata has always been a key part of publishing. For instance, the title of book is metadata about the book; library catalogue cards are good examples of old-school metadata.

Computers need metadata to automate things, and digital publishing is all about automation. In digital publishing, in addition to creating catalogue metadata about titles, author names and prices, we're now creating metadata – in the form of markup tags – about every paragraph or even specific words.

In a very small, very simplified nutshell, that's HTML. Yes, sure it gets more complicated, but the point is this: work systematically, and anyone can recognise and pick their way through HTML, and make changes to a document if necessary.

You may have noticed, though, that while the computer knows what each element is, we've provided no information about how it must look. What font size? What colours? Where on the page? It needs a design spec, a stylesheet. To be specific, cascading style sheets, or CSS.

So what's CSS? CSS stands for 'cascading style sheet'.

We all know what a stylesheet is in publishing: it's a list of elements or features (headings, paragraph types, page elements, and so on) describing how each one should look. For instance, first-level headings: green, 20 point, Helvetica. Body text: black, 12 point, Caslon. And so on. CSS is a formal way of writing style information for a computer to read.

CSS code looks different to HTML, since it's a separate language with its own syntax. Here's an example of a CSS file that styles our HTML. As you will see, it's pretty intuitive. After the tag name, you put the style info in curly brackets:

```
h1 {

  font-family: Helvetica, Arial, sans-serif;

  font-size: 20px;

  color: grey;

}
```

```
p {

  font-family: Georgia, Sylfaen, serif;

  font-size: 12px;

  color: black;

  line-height: 130%;

}
```

Note that I've included some font choices for the machine to choose from, in order of preference. And, in the case of h1 for example, if the computer doesn't have my first two choices installed, any sans-serif font will have to do.

On the computer, the HTML (content) and the CSS (design) are separate. To change content, you only edit the HTML files. To change design, you only edit the CSS. Most websites and ebooks will have one CSS file, and many HTML files, usually one for each chapter. So to change the design of a feature, you just change the CSS file, and the change automatically applies through all the book's chapters.

There are many things you can do with CSS. There are some great examples of creative uses of CSS for online ebooks at ePub Zen Garden. At ePub Zen Garden, the same content is made to look radically different just by referring the computer to a different CSS file.

(I'm not going to go into why cascading style sheets, 'cascade'. But you can look it up if you're curious.)

What are XML and XHTML?

You will have heard the term XML before – most people are confused about what it is. XML is the family of computer languages that includes HTML. XML stands for 'extensible markup language', and HTML stands for 'hypertext markup language'. You'll see why in a moment.

Let’s take a step back first. You know now that, in HTML, content is marked up with tags in angle brackets. And there is a specific list of tags we can use in HTML, like <h1> for a first-level heading and <p> for a paragraph. HTML was designed specifically for the web, and ‘hypertext’ is just a geeky word for text on the web, much like ‘hyperlink’. Hyperlinks are part of hypertext.

There are many other kinds of markup languages, too, in addition to HTML. For instance, MathML is specifically for marking up mathematics. MARCXML is specifically for marking up bibliographic information about books, mostly for library catalogues. They almost all use specific tags in angle brackets to mark up information.

All of these markup languages are kinds of XML – extensions of it. That’s why XML is ‘extensible’, it can easily be extended for different uses.

You may also hear the term ‘XHTML’. Many ebooks contain XHTML rather than HTML. XHTML, which stands for ‘extensible hypertext markup language’, is essentially just a stricter variant of HTML.* By stricter I mean that if the XHTML code contains a mistake, it can just break the whole document. In other words, you can bend the rules of HTML and your ebook will still open in most ereaders, albeit with glitches. But you can’t break rules in XHTML. Then why even use it? Well, its strictness ensures that your ebooks will work better on more devices, because it forces you to follow the standards properly.

* A note for purists: I’ve simplified the story of XML, HTML and XHTML here; the full story is a bit more complicated.

Converting books to ebooks

First-step: decisions

In an ideal world, turning books into ebooks would be as simple as clicking "Save as…" in any good text-layout software. It will be one day, but right now it's not (not even in InDesign, though it gets better with each version). This is partly because the industry is young, so the tools haven't been developed fully yet. But more than that, it's because you (and your clients or bosses) have to make some decisions about the nature of the ebooks you're creating. Each ebook can be a little different.

Let's go through some of those decisions.

Is design and layout (including typography, colours, and the exact placement of images) important for understanding the content?

- Yes: Just create a PDF ebook. You'll lose the ability to sell it on Kindle and iBooks, but using any other format will be a real headache. PDF files are much easier to make in-house than EPUB or MOBI files.
- No: Create an EPUB file. You'll have more flexibility in the long run, and can distribute easily through more important stores (like Kindle) now.

Do you want DRM?

- Yes: Are you sure? Okay. Building your own DM software is too expensive, and you don't want your customers to have to install special software to open your books, because most won't and those that do will hate you for it. So use the industry standard Adobe Content Server (ACS). You could install your own ACS for about $5000 a year and a lot of technical heartache, but it's easier to just use a distributor that includes Adobe DRM in their offering.
- No: Super, your life will be much simpler, and your customers happier. If you positively want not to have DRM, you must make that clear to your distributors who mostly use DRM by default.

How big can the ebook file be, considering your market? (Think about bandwidth and the devices they'll use to read it.)

- Small: If you need customers to read the ebook offline on mobile phones and older ereaders, you need to keep the ebook under, say, 2MB. A novel in EPUB is often around 1MB, about half of which is the cover image. Keep it this small if your users are in low-bandwidth areas (most of Africa, especially outside major cities). The device-restrictions on size sometimes apply to specific components of the ebook, such as images or video inside the ebook.
- Medium: If you have colour images in your book, you might get away with 3 to 10MB, which customers will download in several minutes, depending on their bandwidth and their location relative to the download server. Smartphones (not basic phones) might handle this. Some ereaders and all computers will.
- Large: Anything over 10MB (lots of images or video and audio) may start to push customers' patience or slow their software down. But if you're offering a great product, they'll wait for this. Anything over 20 to 25MB is only going to work on computers and very powerful smartphones.

How much linking and interactivity does your market expect, and how much time can you spend on providing that?

- Value: If your product is a reference book, it's a good idea to spend time making sure links are clickable. And perhaps not just links but references to or pictures of things like famous people (e.g. to their websites or Wikipedia pages) or places (e.g. to a location on Google Maps). Assume you'll spend about three minutes creating every link; a hundred links then takes about five hours.
- Waste: Ebooks can suck a lot of your time and money for little return. Keep it simple. Some publishers produce 'enhanced editions' of some their books, where words in the text link to related websites. These link are often just boring and distracting!

Must this ebook be read offline, or is it only online content?

- Online: Here, users must be connected to the Internet to access the content. Making some or all of an ebook available only while online is a great way to restrict illegal copying and gather user data. But content that requires an internet connection (e.g. Sesame Workshops' content or any DRM that requires an internet connection every time you open the ebook) is not, in my opinion, an ebook: it's a website behind a paywall, filled with book-like content. Which is fine, if your customers always have a good Internet connection.

- Offline: Well, that's what ebooks are, really: downloadable (i.e. save-me-to-read-offline-later) packages of book-like content.

Is accessibility for the visually impaired important?

- Yes: If you're making a PDF ebook, make sure it's a 'tagged PDF'. That will help users with speech-emulating software to navigate the book. If you make a well-structured EPUB, that's already pretty accessible, though DRM can muck it up.
- No: Seriously? See 'Yes' above.

What systems or services will you use to distribute this ebook? (Different ones can handle different kinds of files.) Some examples:

- Amazon Kindle: By far the biggest ebook retailer around. You can upload a wide variety of formats (even MS Word) and Kindle will automatically convert them to Amazon's PRC or MOBI formats. But check the conversions carefully, because automated conversions are often buggy.
- iBooks: Apple's ebook store for iPad and iPhone only sells EPUBs.
- Scribd: Known as the "YouTube of books", you can upload PDFs to Scribd for people to see for free. You can make parts of the book visible to the public (e.g. free excerpts), and embed a Scribd book viewer on your own website. Depending where you're based, you can sometimes sell your ebook in PDF or EPUB format.
- Issuu: Similar to Scribd, but aimed more at the magazine market. Only for free content (e.g. excerpts), you can't sell here.
- Wattpad: Aimed at authors and self-publishers sharing their work for free. Fine for excerpts, for example. It's main advantage is its usability on mobile phones, and large following among younger people. You can upload plain text, MS Word or EPUB files.
- Smashwords: Intended mainly for self and small-scale publishers, Smashwords has one main advantage: upload a Word file, and Smashwords automatically converts into several other formats, and gets your ebook listed on "Barnes & Noble, Sony, Apple iPad iBookstore, Kobo and the Diesel eBook Store, and to all major smartphone platforms via app providers such as Aldiko, Page Foundry, Kobo and Word-Player." Note, not Kindle. This is

perhaps the easiest and most comprehensive free distribution service available. Smashwords is entirely DRM-free.

- Lightning Source: If you want a free way to sell through many retailers and restrict your books with DRM, Lightning Source is an excellent option. (Plus, if you want print-on-demand distribution, Lightning Source's true claim to fame, you can manage that through the same interface.) Lightning Source is part of Ingram, a major international book and ebook distributor. Retailers working with Lightning Source (they pay a setup fee to carry the Lightning Source catalogue) automatically sell all the ebooks stored with Lightning Source on their own sites. Lightning Source provides industry-standard Adobe DRM to restrict copying and printing. (Very small publishers are directed to IngramSpark for similar services.)
- Overdrive: Similar to Lightning Source in operation, Overdrive is an aggregator that makes your ebooks available to a large number of retailers. Overdrive provides the same DRM restrictions as Lightning Source.
- E-Junkie: E-Junkie is an example of a paid-downloads service. For a small monthly subscription, it's an easy way to sell any digital (and therefore downloadable) file without having to set up your own website or downloads server, and keep the full retail-price revenue for yourself. You do need to use a major payment-processing service like PayPal or Google Checkout to receive your money.
- Paperight: Paperight takes your ebook files (PDF or epub) and sells them printed out on-demand in copy shops. The copy shops pay you each time they print a book out for a customer.
- Facebook: You can sell directly from a Facebook page using a service like Shopify or Shop Tab.
- A data-asset-management-system (DAMS): Large companies pay for complex, powerful systems for managing their ebooks, and each one has different requirements. Most of these include automated output to various formats, and can even send those formats straight to distributors and retailers for you. Examples are LibreDigital, North Plains' Telescope Publishing Platform, and codeMantra's Collection Point, or Publishing Technology's iPublishCentral and pub2web. DAMS like this range from, say, R30 000 a year to millions a year. They are difficult to research in advance. For details you need to have a conversation with a sales consultant that usually includes an excruciating process of trying to extract prices from someone who first wants to know

how deep your pockets are. For big businesses, this process is a necessary evil.

- Your own website and server: This gives you maximum control, but is the most expensive option to set up, since you'll incur software-development and design costs. This is fine if you're a big company and want full control. But you lose out on the good traffic, network effects and search-engine results that existing large platforms generate by their sheer size and popularity.

Creating a PDF ebook

This is the simplest ebook to make. It is just a PDF, but with settings applied that make it great for reading on screen. This is most easily done in Adobe Acrobat Pro (you may be able to get by with lesser, cheaper equivalents, like Foxit Phantom), and is more a case of knowing what steps to follow and which settings to choose than having any particular technical skills.

First, though, you want to do as much of the work in InDesign as possible, before tweaking the ebook in Adobe Acrobat Pro. Most of these things you should do as part of your design and typesetting process. In InDesign, you'll create or see to:

- Bookmarks (mainly using the automatic Table of Contents feature)
- Hyperlinks (to places on the web and for email addresses)
- Format and font (if you're not using the print-edition layout, you might use a squarish page format to better fit a screen, and a larger-than-normal font)
- Symmetrical margins and centred page numbers (same margins left and right, since an ebook is usually read page by page, not in spreads; unequal margins make the text jump from side to side when scrolling; same for page numbers)
- Colour (for headings, if you like, or put off-white frames behind your page features in your master pages to avoid a glaring white page)
- Book metadata (add metadata in InDesign's File Info dialogue box)
- PDF export settings (see below).

Then, export to PDF, and refine the PDF ebook in Acrobat Pro:

- Check the PDF bookmarks (the clickable Table of Contents)
- Add the cover image (as the first page, and add it to the Table of Contents/Bookmarks)

- Add metadata (in the PDF's Properties add at least the document title and author)
- Set the Initial View ("Bookmarks Panel and Page", "Single page", "Fit page", "Show: Document Title")

Finally, check the PDF on various apps and devices to make sure everything is there and works. Check the hyperlinks open the right web pages and email addresses, and that the bookmarks (what Acrobat calls the clickable table of contents) go to the right pages. Check the metadata, too. For instance, many people forget to change or add the PDF ebook's ISBN, which should be different from the print book's ISBN.

Converting to EPUB using Calibre

Much of the time as a publisher, you want to be able to create lots of ebooks quickly. They don't need to be works of art, you just need to get the things up for sale. They need to work and to look fine, not spectacular. Importantly, you need to make them cheaply. You could outsource this (e.g. to one of many companies in India that do this), but you'd rather do it in-house.

Calibre is a great open-source ebook-management application for converting ebooks from one format to another. Since you can work from a variety of file formats, your editors can do this, not just your designers. If you have a clean, final MS Word file, creating a sellable basic EPUB file can take about 20 minutes, including some testing.

In short, to create an ebook file (say, EPUB) in Calibre, you can start with an MS Word document. In Calibre, add it to your Library. Choose the output format (EPUB in this case), and press Convert. You can bulk convert, too, converting lots of different books to ebooks at once.

The Calibre user manual includes a thorough explanation of the conversion process and how to tweak it for best results. When you glance through this, you'll see why you needed to learn about HTML and CSS earlier on.

Another open-source ebook program is Sigil. Sigil is a bit more technical, since it requires that you know a bit of HTML and are familiar with the way EPUB files are structured. It's a WYSIWYG EPUB editor (as opposed to a converter): so you can design your EPUB with simple, click-and-drag type tools. You have to start with an EPUB file, or HTML or plain text. So you could start with Calibre or InDesign to create a basic EPUB, and then refine it in Sigil.

Creating EPUB from InDesign

This is probably the best way to produce an ebook. InDesign has been able to export InDesign documents to EPUB since CS3, but only in CS6 has it become fairly reliable. From CS3 and CS4, you have to do a lot of work after export to finish the EPUB (much of this extra work is covered in detail on the Electric Book Works Knowledge Base). In CS5, a few hours work, too. In CS6, you may have to put in an hour or two to polish up the EPUB before it's ready for proofreading and sale.

Here, I'll just provide an overview. For really detailed guidance, check out Elizabeth Castro's excellent book EPUB: Straight to the Point, or visit the EBW Knowledge Base.

Working in InDesign

Before you start, look through the book you're about to work on. You are probably going to be changing or making decisions about:

- how the book's InDesign files are structured (for instance, you'll break up long documents into shorter ones, preferably chapters, and gather them in a single InDesign Book)
- the fonts you use and their colours
- the spacing before and after paragraph styles
- how text in frames is threaded
- master pages
- images that need anchoring.

I'll cover these issues in more detail later.

If you're definitely only using InDesign to create an EPUB ebook (not a print book), you do NOT need:

- running headers/footers and page numbers on master pages
- to specify margins or page size.

These two things will be discarded in the export-to-EPUB process. But why limit yourself to ebook only? You might as well set things up for a possible print edition, too.

Once you've set up your InDesign document properly, you can export to EPUB. Note that in InDesign CS3 and 4 you export to EPUB using "Export to Digital Editions…" (Adobe used to refer to EPUB as "Digital Editions", conflating the EPUB file format with their own EPUB-reading software Adobe Digital Editions.)

Note: I'm assuming for the most part here that you're using InDesign CS4 or later. If you're using CS3, the EPUB export is very buggy, and you'll need to do a lot more to the EPUB files after export in order to create working EPUBs, let alone good-looking ones.

After exporting to EPUB

After you've created a basic EPUB from InDesign, there's more to do to make a polished, sellable EPUB. Depending on the version of InDesign you're using and how you've created your files, you may need to:

- extract the contents of the EPUB file
- add a cover
- add all fonts to the manifest
- make images resizable
- check the order of book parts
- add page breaks, remove span-related spaces, fix non-breaking spaces
- correct the titles of the chapters (especially capitalization)
- check that the Table of Contents works
- add metadata
- edit CSS
- embed video/audio files if necessary
- optionally (especially CS3): replace head tags with structured HTML head tags
- zip the contents of the EPUB up again (in the right order).

And you'll definitely need to test the EPUB thoroughly during and after this process. In the end, you must get a human being to at least glance at every page: just because it's digital doesn't mean it's quick and easy!

To do all this you'll almost always have to work with the EPUB's code at some stage.

Practical typesetting for print and digital

By now, you've worked through a lot of information, much of it new and technical. Well done even getting this far. Now we're really getting to the nitty gritty of making books that work for print and digital.

It makes a huge difference if your existing way of designing and typesetting follows rules that make for easy ebook conversion. So I'm going to work through some practical design and typesetting issues that make conversion to ebook formats easier and more reliable, particularly for EPUB. For instance, you don't want to discover, when it's too late, that a text box you create on page 150 doesn't appear in your ebook, or that your ebook won't open on a Sony Reader because of the way you've set up your InDesign documents.

So, the practical suggestions here are meant as best practice whether you're designing and typesetting for print or for digital. Just make them part of how you work.

And one more thing: the key here is discipline. The guides here are worth very little if, from time to time, you let your guard down and get sloppy. Maybe you forget to add metadata to the File Info, or unconsciously use local formatting in a section of a book – you'll understand soon why those things can let you down. Now that you're creating books that computers must understand, being disciplined about how you structure and process content is critically important. Computers, alas, are unforgiving automatons.

Text flow

There are two rules of thumb for managing text flow:

1. Set each chapter in a separate document, and gather them in an InDesign Book. Inside an EPUB, each chapter will then be in a separate HTML file. This means the software only has to open one chapter at a time, making your EPUB open more quickly and read more smoothly. (Old Sony Readers can't even open an HTML file that's bigger than 300K.) This also creates a nice

page-break effect in the final EPUB, without having to add a page-break tag to the code.

2. Thread (join together) all your text frames in one story flow. If you have any text boxes floating in your document, not threaded into the main story flow, after export that text will appear right at the end of your EPUB. (If you must have floating text boxes, anchor them, and give them their own paragraph style. This can make it easier to fix the threading problem after export later on. InDesign CS6 tries to get around this by detecting where boxes fall on the page, but it can be unreliable.)

Images

Like floating text boxes, any image that isn't anchored won't appear next to the relevant text after exporting to EPUB. (Again, CS6 will try to detect their position.) So, anchor all your images.

Unfortunately, an anchored frame can't contain other frames, so you can't anchor an image and a caption in the same anchored object. To add a caption, you're going to need to work on the actual the HTML code (for instance, using Sigil).

Finally, if you have very large images in your book, you may need to resize them, because some devices can't display images larger than 10MB. (From InDesign CS4 you can choose to have InDesign automatically resize your images when you export to EPUB, which saves you the hassle. Just make sure all images export as JPG, and not GIF or on auto – the GIFs lose too much image quality.)

Fonts

Fonts in ebooks are tricky. To display text in a given font, the device that opens it must have access to the font files. So if you want a reader to see your text in a specific font, you have to actually include the font files in the ebook. This is called embedding the font.

When you make PDFs, most PDF software (like InDesign by default) will embed the font files inside the PDF. That's why fonts usually look fine when you open PDFs from someone else, even though you don't have their fonts installed on your computer – the font files you need are inside the PDF. But importantly, by default the PDF only includes a subset of the font's characters; just those characters the file needs. So if the letter 'q' is not used anywhere in the document, the font file for 'q' might not be included in the PDF.

This is very useful for two reasons: one, it makes the PDF smaller than if the entire font was embedded. Two, it means that when you send a PDF to someone, you're not sending them the full font files, which could be a violation of the font's owner's copyright.

So PDF is easy, but EPUB is harder: you cannot embed a subset of a font in an EPUB. You either include the whole font or nothing. If you include the whole font in the EPUB file, you may be violating the font owner's copyright. If you include nothing, the reader won't see the fonts you intended. Their device will just pick another font they do have installed already.

There is another problem: even if you do embed a font, many ebook readers will simply ignore it anyway, and display the text in its own font instead. Why? First, its software developers might not have had the time to include code that reads fonts. Second, it's likely that software developers (e.g. Amazon and Apple) know that many publishers create ebooks badly, including using ugly fonts, and they don't want their customers having a bad, inconsistent experience. Companies like Amazon and Apple need their customers to have a consistently elegant experience, even if it means overriding some good designs. In 2014 Amazon Kindle started allowing Kindle users to manually switch to seeing the publisher's font, rather than the Kindle default.

So, in the end, I recommend not embedding fonts in EPUBs. The defaults in good ereading apps like Kindle and iBooks are fine.

However, if you have really good reasons for embedding a font, here are some guidelines.

Try to use font files in OpenType format. Many TrueType fonts can work, but OpenType is usually more reliable.

As I explained earlier, when you include a font in an EPUB file, you're including the entire font file. So, technically, you're redistributing the font.

For most fonts, this is illegal. InDesign offers to encrypt fonts (also called font mangling) to get around this, but that encryption can cause technical problems with your EPUB, and is best avoided.

So, use open-licensed fonts, such as those from the Open Fonts Library. Or use Google Fonts, which are all open-licensed. You may redistribute open-licensed fonts, so embedding them in an EPUB is not illegal. Good options for text fonts are Linux Libertine and Gentium.

InDesign CS3 quirks

In InDesign CS3, a specific font weight or style (bold, italic, light, etc.) only survives the export to EPUB if you create a special character style for it that specifies both the weight/style and the font family. So you must create a separate character style (i.e. bold, italic, underlined) for each font family you apply these styles to. You can't use a generic 'bold' style and apply it to different fonts. CS4 and later are much more forgiving, but actually it's good practice anyway to do this, because it makes for better, more flexible CSS code in your ebook in the end.

Styles

Good designers always use paragraph and character styles to format text, and never local formatting. That is, they never highlight a word and click the 'Italic' button on the toolbar. They'd create a character style, perhaps called 'emphasis', highlight the word, and apply that style to it. This is best practice for all formatting, because you're then controlling all formatting in a book from one place: your styles list.

But even good designers name their styles badly. What is a bad style name? A bad style name is any name that describes the style's appearance. A good style name described the style's purpose, or function. For instance, it's bad to name a heading style 'blue-head'. 'Blue' is how it looks, now its function. It's function might be to start all the quiz sections. In that case, a better style name might be 'quiz-head'.

Why? Well in your print book the heading might be green, but on a grayscale ereader screen it will appear dark grey. And on a computer, blue is usually used only for clickable links, and you don't want to confuse people with something that looks like a link. So you might use CSS to make the quiz-heads pink, for instance.

Remember how we said you must always separate content and appearance? This is an instance of applying that principle:

- Function: 'this paragraph is an item in an unordered list'.
- Appearance: font, size, colour, line-height, margins, etc.

So, always name your styles by their function not their appearance. For example, use 'chapter-title' for your main level of heading, not 'large-bold'.

When you export your InDesign book to EPUB, your style names will become part of the tags in the HTML code. Remember how we used a class attribute to define a <div> as a 'note' in our HTML example earlier? Your style names in InDesign will be the class names in your EPUB's HTML. That is very important to remember!

Another important implication of this is that you should always make your style names lower-case, and never use spaces or special characters.

So, a paragraph styled with a style called 'note' in InDesign will look like this in your EPUB's HTML:

```
<p class="note">

  Your note text goes here.

</p>
```

Technically, InDesign is categorising that paragraph with a class that uses the style name you chose: 'note'. In the CSS InDesign creates for you, it will define the appearance of that class of paragraph much like this:

```
p.note {

  font-family: “Helvetica”;

  font-size: 10pt;

}
```

(When creating CSS, InDesign tries to create CSS styles that follow the appearance of your InDesign styles, but the result won't be exactly the same as the text on your printed page.)

Local formatting

In InDesign CS3, only defined styles (paragraph styles, character styles and object styles) are exported to EPUB; all local formatting is discarded when you export to EPUB.

This means that if you want a single word or phrase in bold type, you need to create a character style (e.g. 'strong-text'), and apply this character style to all of the text that you need to be bold. If you then decide that you want some of those words bold and italic, you must create a second style to apply to those words to be turned bold and italic (e.g. 'strong-emphasis-text').

In InDesign CS4, you get to choose when exporting to EPUB whether to keep your local formatting or use your defined styles instead. However, you shouldn't use local formatting for your styles anyway. Local formatting in the resulting EPUB makes a mess of your HTML and CSS code, which then makes it far harder to change and manage text and formatting in the EPUB later on. So, avoid any local formatting, and only use styles to control the appearance of text.

Pagination and master pages

PDF ebooks have pages like print books. But reflowable formats, like EPUB and MOBI, have no fixed pages as such. So there is no pagination. So when you export an InDesign file to EPUB, it will ignore:

- text on master pages, including headers, footers, page numbers,
- margins or page size.

So don't put any important content, necessary for understanding the book, on a master page.

Spacing

Never use empty lines or empty paragraphs to create space in your books. They will get ignored in the HTML of your ebook, and paragraphs you've separated with space will lose that space. Rather, use your styles to add spacing before and after paragraph styles, especially headings and blockquotes.

You may need to search your document for empty paragraphs by searching for two 'End of paragraph' marks, and solve each instance using the paragraph style's before-and-after spacing. You may need to create a few new paragraph styles for this to work (e.g. a block quote of three paragraphs would need three separate paragraph styles: a 'blockquote-first' with space above, 'blockquote-next' with an indented first line, and a 'blockquote-last' style with space after.)

A tip: If you want to risk a global search-and-replace, you can't just remove all paragraph marks, or you'd end up with one long paragraph! So first change all instances of two 'End of paragraph' marks with a totally unusual sequence of special characters, like %\^@. Then replace all instance of %\^@ with one paragraph mark. This can be a bad idea, though. For instance, in many novels the author may have typed an extra line space to create a break between sections, and you don't want to lose those. They should be individually managed by applying a style to the first paragraph in the new section that has extra Space Above.

Text alignment

It's best not to use justified text in ebooks. On very small screens this causes big spaces between words, especially if text ever wraps around images. Rather use Left (ragged right), Right (ragged left), or Centered.

Background colours

I once tried to create an off-white ebook page, much like a paper book might be printed on creme paper. I don't recommend spending time on this, it's almost always wasted. Page colour is now often determined by the ebook-reading device software for different contexts (e.g. dim or reversed out for low-light situations). Sometimes users (like myself) even choose to read their ebooks as white text on a black background, completely subverting the designer's intentions!

Tables of contents

Most setters create tables of contents manually, typing in page numbers by hand, and updating them manually when they change. Many don't know that InDesign lets you create a table of contents automatically. There are a couple of good reasons why it's worth learning how, even though setting up styles for an automated table of contents can be a lot of work.

The two good reasons are:

- an automatic table of contents tell InDesign which paragraphs in your book are headings
- it helps InDesign create your EPUB's clickable navigation.

InDesign needs to know which of your paragraph styles are headings, and which are text paragraphs, because when exporting to EPUB, in the code your paragraphs need <p> tags and headings each need one of `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` or `<h6>` tags. There's no real way for InDesign to tell the difference between `<p>` and the six `h` s from your styles alone – the table of contents does that.

InDesign needs you to use its 'Insert Table of Contents' feature to signal which paragraph styles are headings, and which level of heading they represent. In the EPUB code, InDesign will then tag those paragraphs you include in your table of contents as HTML headings (`<h1>` etc.), not as `<p>` paragraphs.

Why is this important? Well, some ereading software will not read your CSS file at all, or only in part, and will instead impose its own CSS based on standard HTML tags. That means it ignores your classes, and won't show the fonts, colours, weights and so on that you've so carefully defined. So if in your HTML both your headings and your regular paragraphs are tagged as `<p>`, such software will not distinguish between them, no matter what classes they're in, and your headings will look like regular text.

Note: If you export to EPUB from CS3, you have to change the tags in the EPUB code manually. If you use CS4 or later, create an automatic Table of Contents style in your InDesign document, where you include your heading levels in the Table of Contents.

Importantly, creating an automatic table of contents in your InDesign document also gets you a table of contents in your exported EPUB file, which ereaders display as a clickable list. So always create one, even if you don't include it on any page in the for-print document. You can just put it on the pasteboard, off page. It doesn't even need page numbers. It just needs to list all the headings you want in your EPUB's clickable table of contents.

Some tips:

- When creating the Table of Contents in InDesign, create the TOC style and save it. You can then import the same TOC style into other InDesign documents with similar style names, saving you time later on.
- If you're working in an InDesign Book, create the Table of Contents in your style-source document. This is not necessarily the first document in your Book. When exporting to EPUB, InDesign will ignore any tables of contents created in other documents in the Book.
- Avoid special characters in headings. For technical reasons (related to character-encoding), they can occasionally upset the EPUB code (the .ncx file) and break your table of contents and your ebook. (If your table of contents seems to get cut off early in your EPUB, check for special characters in the code.)

Covers

Most setters create separate files for the cover of a book and its interior. There are good reasons for this: a cover document that includes a spine and back cover is a completely different size from interior pages, and is often created by a separate designer anyway.

However, if there isn't a front cover in your InDesign file, InDesign can't add it to the EPUB, right? You have to add it manually to the EPUB after exporting from InDesign, using an EPUB editing program like Sigil. If you do this, keep the image almost, but just below, 1000 pixels on its longest side – that's best for most ereaders.

There is another option: in your InDesign Book, include a one-page document that contains only the front cover. You might call it 'front-cover.indd', for instance. Make it the first document in the Book. When you export to EPUB, include it in the export. When you export to PDF for printing, don't include it. Remember that adding a single page to the start of your Book will by default make the first interior page of your book a left-hand page, which you don't want. You'll have to set up your Book to keep the first interior page a right-hand page.

Metadata

Some basic information about the book must be included in the EPUB file itself. Some information is mandatory:

- the title
- the creator (usually the author)
- a unique identifier of some sort (such as an ISBN or UUID; InDesign generates a UUID for you when you export to EPUB).
- a date of publication or creation.

InDesign can add this metadata for you if you enter it in the InDesign document's File Info (Info > File information). In InDesign CS3, this has been known to cause problems with the EPUB code, but it's still a good idea to include at least the title and author in the File Information, since that's the minimum that an EPUB needs in order to be a valid EPUB.

InDesign CS4 and 5 don't let you add the publication date to the EPUB, which is required by the EPUB specification. You may need to add the pub date metadata by hand in the EPUB code, or using an editor like Sigil or Calibre's metadata-editing tools.

Embedding multimedia

Here's an irony: in the early days (around 2006 to 2010) embedding multimedia like soundtracks and video was surprisingly easy to do. Usually, it involved including a Flash video in your EPUB. Then Apple brought out the iPad and decided it would not support Flash. This effectively made putting Flash in EPUBs a waste of time. And as a result, there's currently no easy way to embed video or music in an EPUB. No matter how you do it, only a handful of applications will be able to open and play it, and that's just not good enough.

So I suggest placing video or audio on a media-sharing site (like YouTube, Vimeo, or Soundcloud), embed it on your own branded site if you like, and then simply link to it from the ebook. This does mean readers will have to be online to get the video or audio, and be on a device that opens web pages (i.e. not an e-ink reader), but it's the simplest, cheapest way to link to video or audio.

Getting inside the EPUB

You can edit almost anything in an EPUB using the free, open-source editor Sigil, though you do need to know (or be willing to learn) a bit about EPUB structure, HTML and CSS.

If you want or need to really get your hands dirty and open up an EPUB yourself, just change the .epub file extension to .zip, and unzip the file. Easy as that. Zipping it up again is not entirely straightforward. Most importantly, the order in which files go back into the zip archive is important (mimetype file first!). See the EBW Knowledge Base for details.

Checking ebooks

Many people think that ebook conversions are quick and reliable. Sadly, creating ebooks presents many opportunities for gremlins to creep in. Check ebooks thoroughly before releasing them. A human being must at least glance over every page, click on most or all of the links, and open it on a variety of devices.

Further reading

Check out the EBW Knowledge Base for detailed info on creating EPUB files from InDesign, as well as lots of other digital-publishing guidance.

Feedback

I need to know what's good, bad and ugly in this book so that I can make it better.

What were the most useful parts for you? Please let me know. You can send me feedback at arthur@electricbookworks.com.