HTML & CSS

The easy path.

WILLIAM GHELFI

HTML & CSS

The easy path.

William Ghelfi

This book is for sale at http://leanpub.com/easyhtmlcss

This version was published on 2014-04-06

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Tweet This Book!

Please help William Ghelfi by spreading the word about this book on Twitter!

The suggested hashtag for this book is #easyhtmlcss.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

https://twitter.com/search?q=#easyhtmlcss

To my wife, Daniela. And to our unborn child.

Contents

Baby steps, or: Introduction

Image by Paul Inkles: http://www.visioncreation.co.uk/ used under http://creativecommons.org/licenses/by/2.0/

I started being online, on the Internet, in 1996.
I also started learning HTML, the HyperText Markup Language, in 1996.

Those were the days of *"Wow! How is it done?"* and of *"Let me check what happens if I right click... just... here."* and then a promising `View HTML` menu item would appear to bring you in the realm of the unknown.

As it happens, nothing has changed.

HTML is still there – although now tightly coupled with its *Best Friend Forever* the CSS, Cascading Style Sheets – and its importance has not decreased.

You learn HTML and CSS because:

- you want to know how websites are done

- you want to tame any tool which writes HTML and CSS for you
- you want to learn from the masters, reading their code
- you want **that feel** only a true creator really experiences

In 2014, if you want to learn HTML and CSS you have great resources at your disposal all over the web. Both free and on sale.

What you didn't have, it's a guide which is pragmatic and to-the-point, somewhat funny to go through, and puts together your first steps with HTML and CSS in a single, easy, path.

If that's what you think you need to learn something new, and that something is HTML and CSS, I'm writing this book for you.

To let you learn the basics, being able to create something which is yours and it's a **good** start, and have some fun in the process.

What's ahead

First things first, by Chapter 2 you'll have put together the tools of the trade and then used them to publish online your first simple page, because **that's what this is all about**.

The next couple chapters will focus on what defines a minimal HTML document, but they will also quickly go on showing the first group of useful HTML elements.
Those you'll use well and often, and soon enough they will become like a second language to you.

Meanwhile, throughout the whole book you'll also keep excercising with simple hands-on tasks which will immediately give you the opportunity to put to good use what you just learned.

It will be then time to introduce the CSS, quickly showing the most useful properties in the topics of *typography*, *positioning*, and more.
Then, having learned about positioning, you'll see the common elements of a classic layout, the *not-so-common* elements of a classic layout, and a solid introduction to some – guess what? – *classic types of layout*.
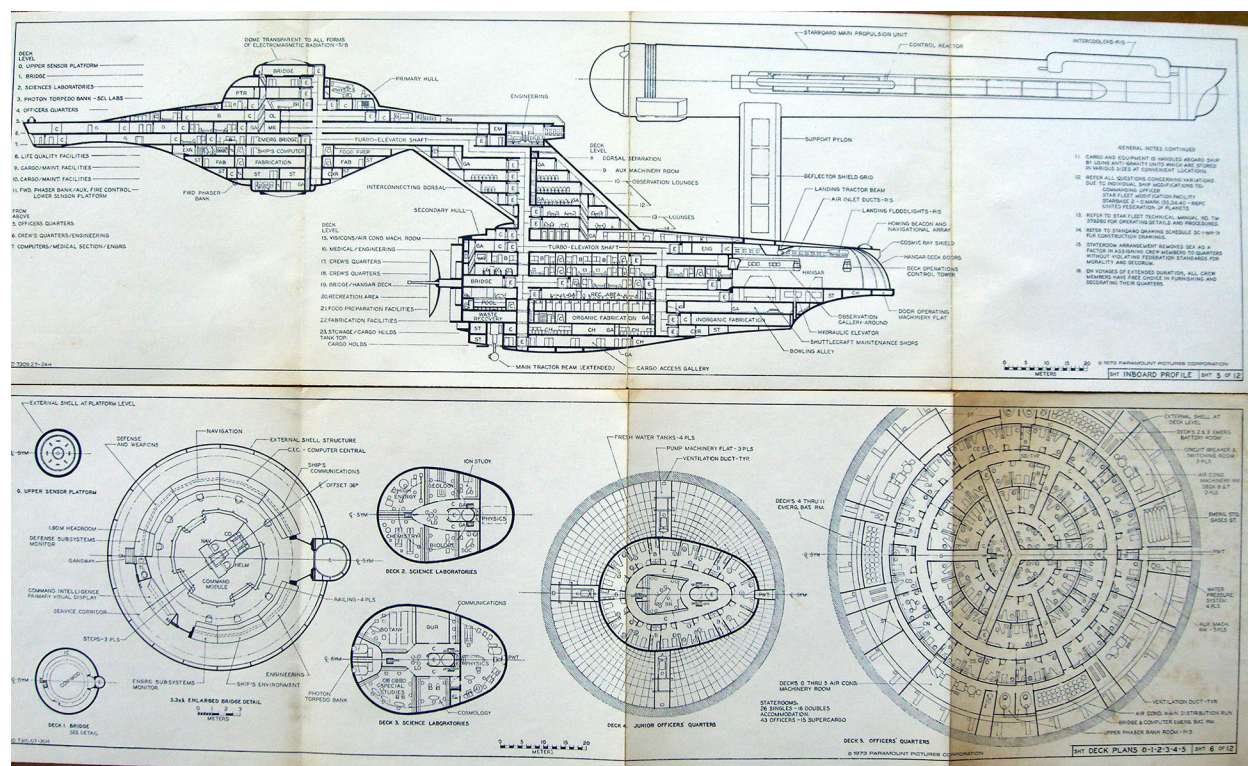
"Understanding CSS layouts is way easier than intergalactic rocket science." – James T. Kirk. Image by http://www.flickr.com/photos/cardoso/ used under http://creativecommons.org/licenses/by/2.0/

By the following chapter it will be about time to cheat a bit, and you'll encounter the magic of CSS grid systems which basically do all the dirty work from the previous chapter and leave only the fun parts for you to enjoy.

After a couple more useful HTML elements like forms, tables, and others, there will come my favorite chapter – with subsequent *hands-on time* – where you'll have some fun with background images, gradients, borders, and shadows.

Next to the end you'll learn about web fonts and icon fonts, which are two of my favorite innovations of the last years and together can greatly improve the beauty and usability of any web page.

I'll bid you farewell with a chapter about the next steps, where to go after you close this book and how to keep on learning and having fun with HTML and CSS.

Oh. One more thing ™ ...

There are also three appendixes:

1. An introduction to Responsive Web Design and Mobile First.
Devised for the first time in 2010 by Ethan Marcotte, the Responsive Web Design approach changed for ever how we think of a web pages and pushed the whole industry forward with

a renewed thirst for a universal web where devices of various sizes and capabilities coexist to form a digital ecosystem.

2. An introduction to Bootstrap, the CSS framework
Bootstrap is the most popular front-end framework for developing responsive, mobile first projects on the web.

3. An introduction to JavaScript
JavaScript is a scripting language every major browser understands. It is primarily used to add a *behavioral* layer on top of the *semantic* and *presentational* layers provided by HTML and CSS.
This separation of concerns, is one of the pillars of *progressive enhancement.*

Illustration by Dave Stewart
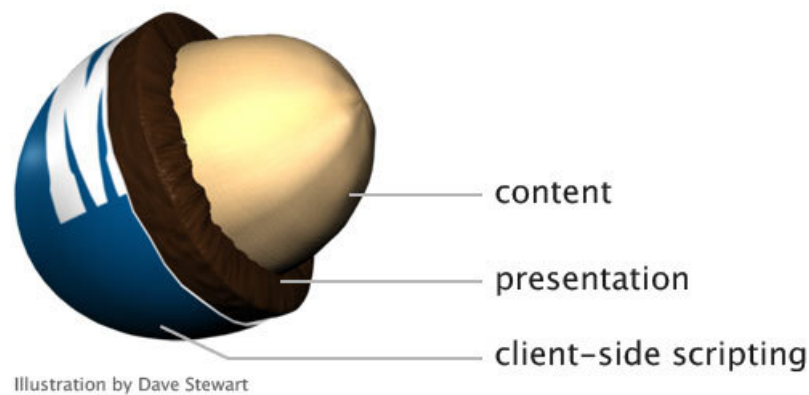
Progressive enhancement as in http://alistapart.com/article/understandingprogressiveenhancement by Aaron Gustafson

You know, just to give you an idea of what's still out there for you to conquer.

And now... action!

1 Tools of the trade

Writing HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) is akin to craftsmanship.

If you have passion, in time you will master it. But you could also be content with just a basic knowledge of the trade, do your job, and move on to the next challenge.
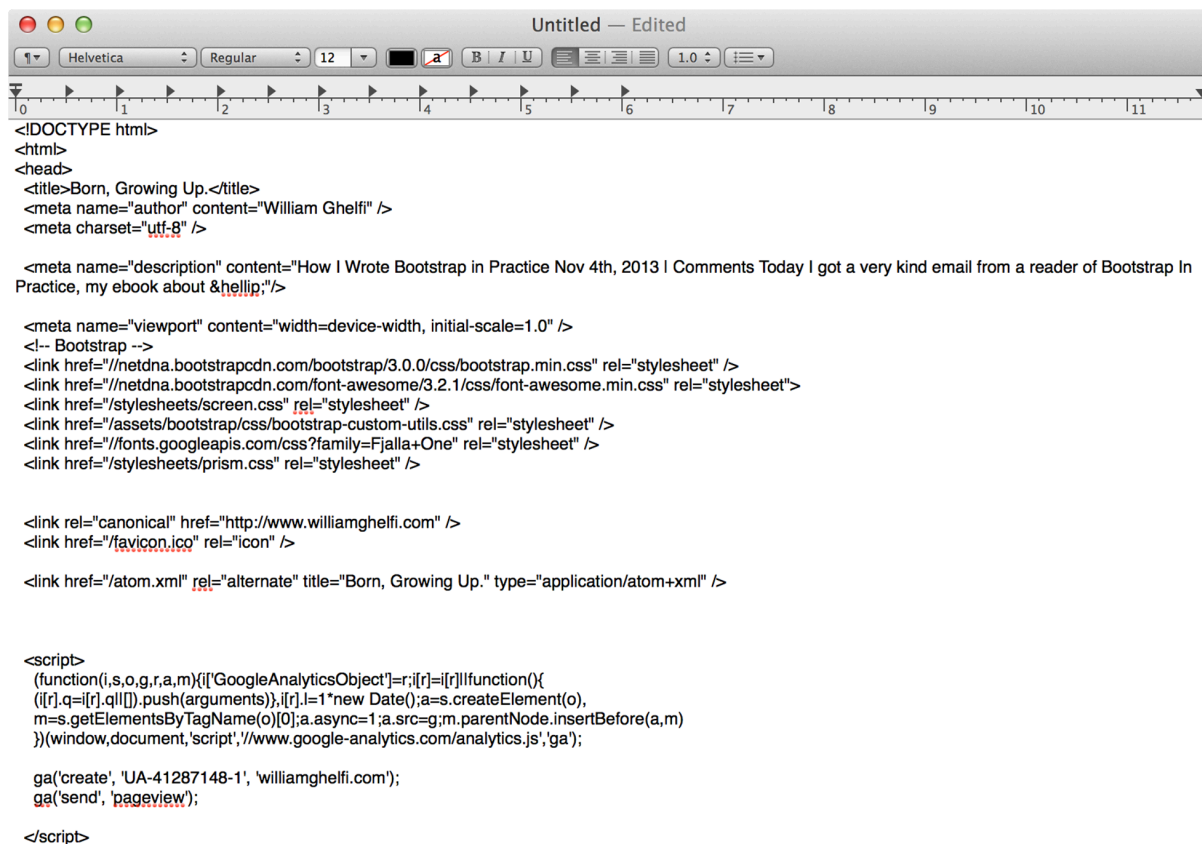
Either way, you'll need a working Internet connection, and some tools.

A text editor, to write your web pages

First of all, I'm going to assume you are an average computer user and know how to write a document in Microsoft Word.

As you may already know, Word is a word processor[1]. What you need instead is a text editor.

---

[1] http://en.wikipedia.org/wiki/Microsoft_Word

```
<!DOCTYPE html>
<html>
<head>
 <title>Born, Growing Up.</title>
 <meta name="author" content="William Ghelfi" />
 <meta charset="utf-8" />

 <meta name="description" content="How I Wrote Bootstrap in Practice Nov 4th, 2013 | Comments Today I got a very kind email from a reader of Bootstrap In Practice, my ebook about &hellip;"/>

 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
 <!-- Bootstrap -->
 <link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet" />
 <link href="//netdna.bootstrapcdn.com/font-awesome/3.2.1/css/font-awesome.min.css" rel="stylesheet">
 <link href="/stylesheets/screen.css" rel="stylesheet" />
 <link href="/assets/bootstrap/css/bootstrap-custom-utils.css" rel="stylesheet" />
 <link href="//fonts.googleapis.com/css?family=Fjalla+One" rel="stylesheet" />
 <link href="/stylesheets/prism.css" rel="stylesheet" />

 <link rel="canonical" href="http://www.williamghelfi.com" />
 <link href="/favicon.ico" rel="icon" />

 <link href="/atom.xml" rel="alternate" title="Born, Growing Up." type="application/atom+xml" />


 <script>
  (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
  })(window,document,'script','//www.google-analytics.com/analytics.js','ga');

  ga('create', 'UA-41287148-1', 'williamghelfi.com');
  ga('send', 'pageview');

 </script>
```

A web page as seen inside a text editor

Here's some commonly used text editors:

- **Windows**: Notepad
- **Mac**: TextEdit
- **Linux**: Gedit / Kate / Nano / ViM

Notepad and TextEdit come pre-installed with your PC or Mac. If you are using Linux, chances are you know how to install a text editor using a package manager – if needed.
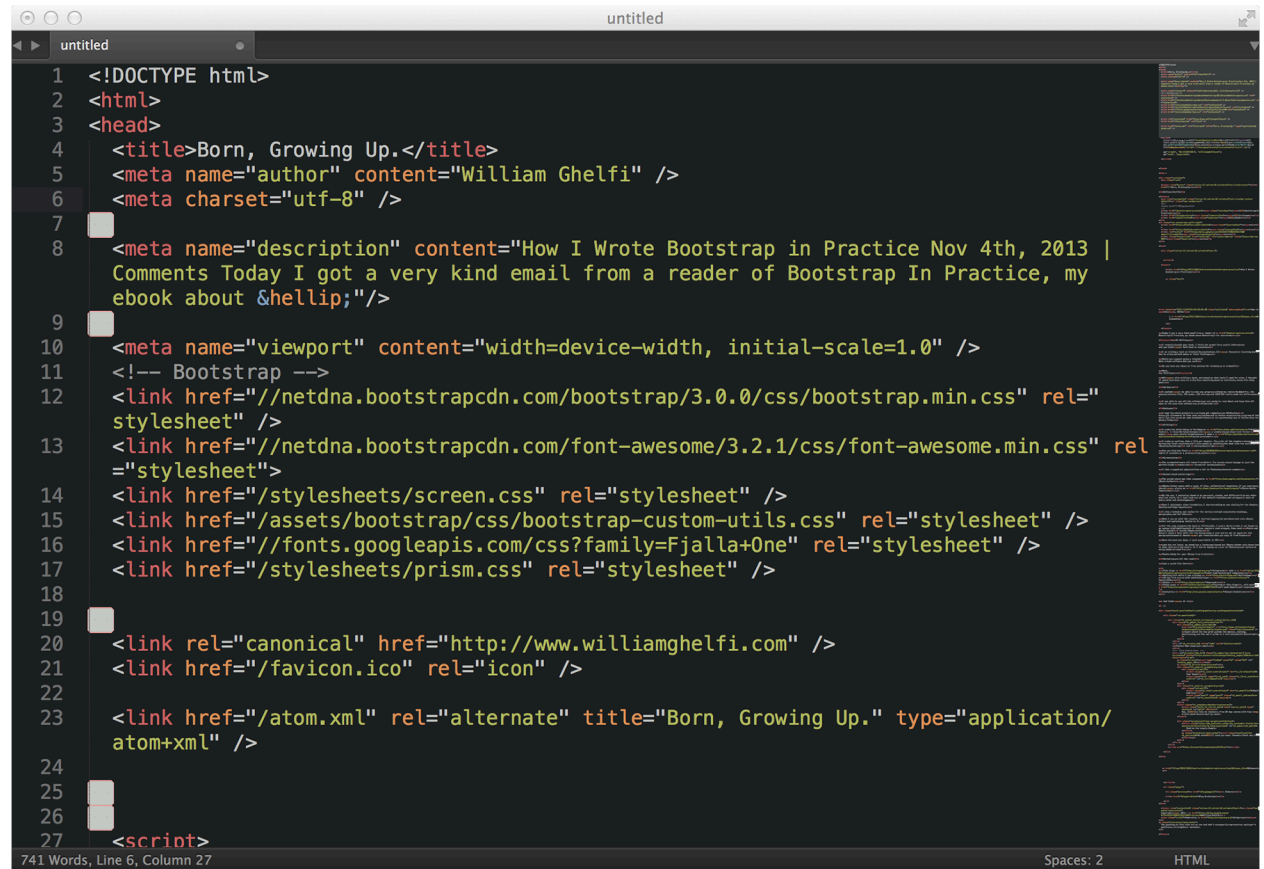
For example, on Ubuntu Linux you can install Gedit opening a terminal – `ctrl + alt + t` – and executing the following command (ignore the `$>` part, which is the command prompt):

```
1  $> sudo apt-get install gedit
```

🔑 Protip!

A better tool for writing HTML and CSS is a code editor: a code-oriented text editor
with added awesomeness.

A web page as seen inside a code editor

My code editor of choice, Sublime Text², is available for Windows, Mac, or Linux.

Two other well appreciated code editors are LightTable³ and Brackets⁴.

If you are feeling confident enough, I definitely suggest using a code editor instead
of a text editor.

Or, you could just start with a text editor and switch to a code editor when you are
ready.

---

²http://www.sublimetext.com/

³http://www.lighttable.com/

⁴http://brackets.io/

A web browser, to see your work in action

Ok, this was easy. After all, a web browser is what you used to get this book!

There are quite a few top-class web browsers out there. In alphabetical order:

- Apple Safari[5]
- Google Chrome[6]
- Microsoft Internet Explorer[7]
- Mozilla Firefox[8]
- Opera Software Opera[9]

All of them are available for free, but there are some differences in availability.

Browser	Windows	Mac	Linux
Safari	x	x	
Chrome	x	x	x
Internet Explorer	x		
Firefox	x	x	x
Opera	x	x	x

Throughout this book – and just to ensure a common baseline – we are going to use Google Chrome.

I'm expecting all the examples and code snippets to work on every other major browser – this is called *writing cross-browser code*, and it's a fundamental best practice – but you know… just in case.

A versioning system, to save your work

Using a versioning system, means that all the subsequent versions of what you are writing – or at least, many of them – are always available to get restored if you need them.
Think of it as a backup system.

The simplest versioning system I can think of at the time of writing, is one that lets you just work while it does its thing.

- You save, it saves.
- You delete a file, it lets you restore it if that was an error.

---

[5] https://www.apple.com/safari/

[6] https://www.google.com/intl/en/chrome/browser/

[7] http://windows.microsoft.com/en-us/internet-explorer/download-ie

[8] http://www.mozilla.org/en-US/firefox/new/

[9] http://www.opera.com/

- You create a file, it starts monitoring and updating it while you keep working on it.
- You switch workstation, you have all the files at your disposal in the new one.

Ladies and gentlemen, Dropbox[10].

So, what are you waiting for?

1. Go create a Dropbox account, if you don't already have one
2. Create a new folder under the Dropbox one, and name it something like `easyhtmlcss-work` That is your work folder, where you are going to store all the examples, and where you will experiment on your own when you are feeling brave
3. Create a new blank file, `justatest.txt`, and see if Dropbox synchronizes it upon save with its remote server

🔑 Protip!

A more code-oriented alternative to Dropbox, would be a proper Source Control Management system.

At the time of writing, Git[11] is the winner, the most used and loved. GitHub[12] is a hosting service for Git repositories... and much more.

The point is: using Dropbox just works and it's perfect for scope of this book. But if you already use an SCM, or feel *tech-savvy* enough to try, then it's perfectly fine to put Dropbox aside and go with an SCM.

A web hosting service, to share your work

Web hosting is the act of *hosting* one or more *web pages* on a computer connected to the Internet, in a way so that other people can visit a web address – or URL – and see those web pages.

Facebook, the Guardian, Twitter, are all *doing* web hosting for you to be able to visit their websites.

So, what if somewhere through this book you happen to become particularly proud of what you have achieved and want to let a distant friend admire it?

Ladies and gentlemen, Dropbox[13]. Again.

But fret not, my dear, because further explanation will come in the next Chapter, along with a proper section featuring some alternatives!

---

[10]https://www.dropbox.com
[11]http://git-scm.com/
[12]https://github.com
[13]https://www.dropbox.com

Lessons learned

The tools of the trade are not too many, and they are also easy to get and use.

- Since web pages are written, you'll need a **text editor** to write them.
- A good **web browser**, like Google Chrome, will let you see what you are working on.
- **Backups** are always a good idea, and being able to restore a previous version of a web page can save you from a bad headache. Dropbox comes in handy.
- One of the best parts of writing your own web pages by yourself, is to brag about them with your friends! Dropbox is the quickest and easiest way to **publish something on the web**.

2 Your first page

Now that you have a basic working environment at your disposal, let's start already with your first HTML document!

Not a big deal, mind you, just a barebones page.

An *incipit* to put you on the run, even if it lacks the fundamental parts you are going to read about in the next chapter.

Hello, I'm …

Here's what I'd like you to do:

1. Fire up your editor
2. Create a blank file (most editors will start with a ready blank file)
3. Type in `Hello, I'm William!`. Well, don't just type `William` – use your real name
4. Save the file into your `easyhtmlcss-work` folder from Chapter 1, and name it `index.html`
5. Fire up your browser
6. Assuming it's Google Chrome, choose `File --> Open File...` from the top menu and open your `index.html` file

And behold that beautiful blank page!
(I'm lying, it's just frankly horrible – but that's going to change in the next chapters).
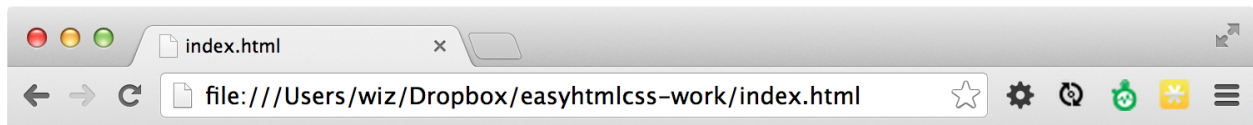
Horrible, light, and unequivocal. You first, barebones, HTML document.

That's it. Your first web page, with a few caveats I'm going to disclose in the Lessons learned section of this chapter and not a second sooner because it's now time to go online!

Your first page goes online

So you have now a Dropbox account, and are using it so far as a simple but effective versioning and backup system.

But that's just half of the awesomeness! When a file is on your Dropbox account, it goes on the Dropbox cloud meaning that you can always get it back wherever you are or in whatever device you installed Dropbox with your Dropbox account.

And now that your `index.html` is in fact *already* online, you can share it with whoever you like to. Go on, try it now:

- Go to your Dropbox folder
- Descend into the `easyhtmlcss-work` folder
- Click with the right mouse button on the `index.html` file, and choose `Share Dropbox Link`

An URL pointing to that file in the Dropbox cloud has been copied into your system clipboard, meaning that you can now open a new browser tab, paste it into the address bar, and be extremely disappointed by what you'll get.

index.html

1 min ago · 19 bytes
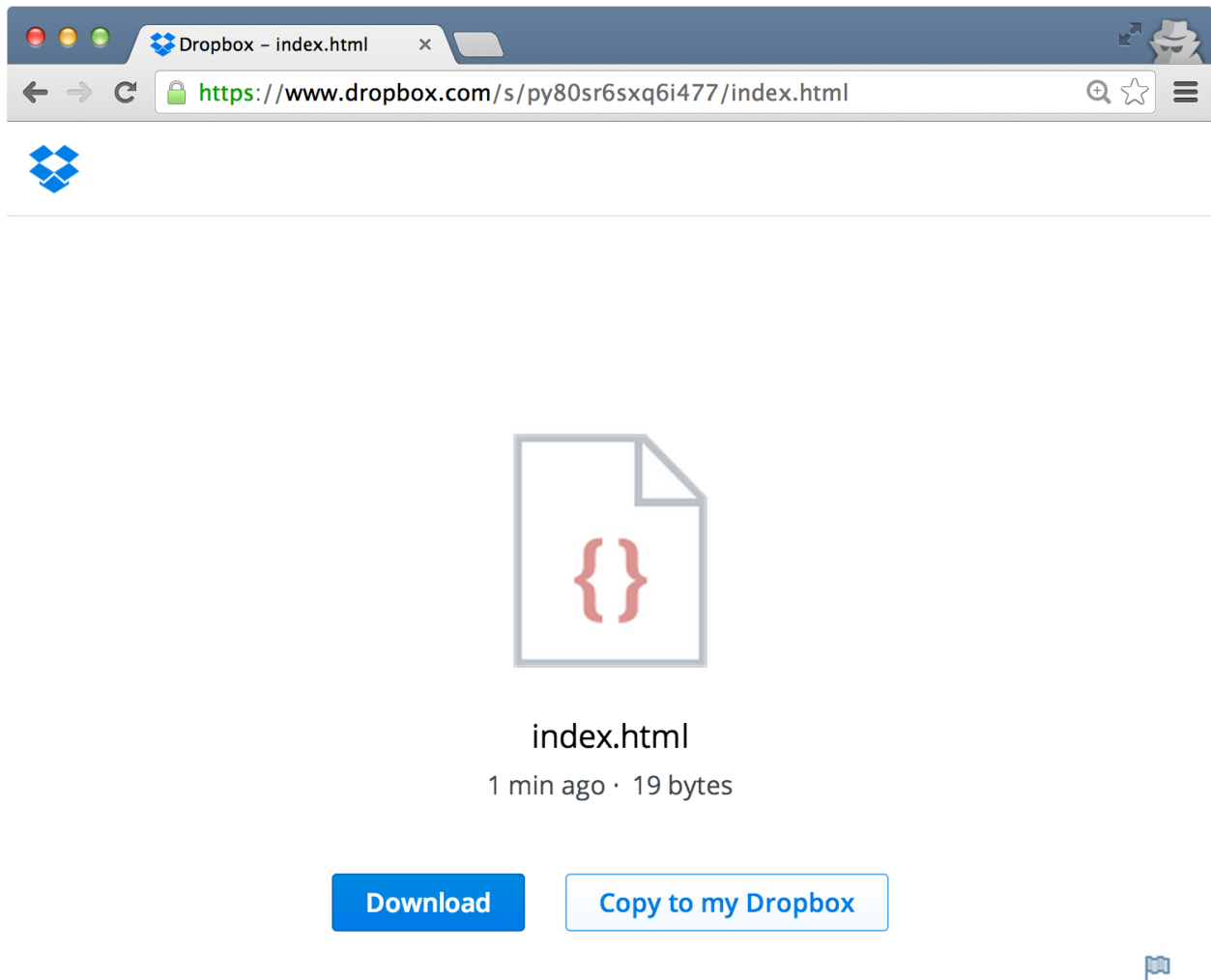
Not yet what we were expecting it to be

Don't panic! This is not what you want your audience to see when you share your work with them. What you want them to see is the actual web page, just like you normally see them on the web.

To achieve what we want, let's now tweak the situation a bit:

- Go back to your `easyhtmlcss-work` folder under the Dropbox one

- Move the `easyhtmlcss-work` folder down into the `Public` folder you'll also find sitting there under the Dropbox one
- Descend into the freshly relocated `easyhtmlcss-work` folder
- Click with the right mouse button on the `index.html` file, and this time you'll have to choose `Copy Public Link`

Now open a new browser tab, paste it into the address bar, and rejoyce!
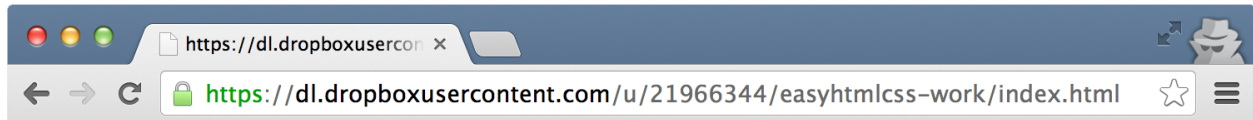
Hello, I'm William!

Now that's what I was talking about

Now that your work folder is under `Public`, you can keep on working on your files and the friend or coworker you sent the public link to will always see the most updated version as if it were on a real professional hosting service.

Just always remind that such a setup is suitable for experiments only, and it is not a full-fledged hosting solution capable of serving your web pages to even tens of concurrent visitors.

That's a totally different problem, with totally different easy solutions.

Easy hosting services

Altervista

[Altervista](1) is the easiest of the whole lot.
You'll just have to sign up, choose a name for your website, and you'll be then given the choice among three different approaches to building a website.

Given right now you are reading a manual about how to write HTML and CSS, let's just assume your choice will be to directly mess with the files and folders of your website.

The interface you'll land into after having chosen to directly manage your files – as opposed to use a *Content Management System* or their web page builder – should be pretty easy for your to learn.

The basic concepts are:

- You can create / rename / edit / delete files and folders
- You can upload new files vie the web interface
- You can do all of the above via FTP, if you know what it is and how it works

And your website will always be available at http://*the-name-you-chose*.altervista.org

Mine, for instance, is at [http://easyhtmlcsstest.altervista.org/](2) and, as you can see, I deleted the default `index.html`, created a new one, and copy-pasted the text from the very first section of this chapter into the brand new `index.html`.

GitHub Pages

[Github Pages](3) is a service by GitHub.
GitHub is a code repository service, with an amazing collaborative and social twist. It is based upon [the distribuited versioning control system Git](4), but if you want you can start using it as an hosting platform like many others do.

The guide at the Github Pages website is well written, easy to follow, and pretty straightforward so I won't try to rewrite it here.

Nonetheless, here's some advice:

---

[1] http://www.altervista.org

[2] http://easyhtmlcsstest.altervista.org/

[3] http://pages.github.com/

[4] http://git-scm.com/

- Of course you'll first need a GitHub account, so head over to http://github.com/[5] and sign up if you don't have one.
- If you don't know Git, the second thing to do before starting to follow the guide is to download and install the GitHub desktop app. It's available both for Mac[6] and for Windows[7].
- While on the guide, follow the path for a *"User or organization site"* (yours will be a *user site*) and remember to use the desktop app from the previous point.

And you're done!

Divshot

Divshot[8] recently switched from simple and intuitive prototyping of websites, to fully featured static hosting.

The prototyping tool formerly known as Divshot is now called *Architect.*

Architect is quite powerful and awesome, and after having completed this book you'll be able to approach it with a new confidence in both your and its capabilities.

The *new user* experience in Divshot is blazing fast and streamlined. The second after you sign up, you are ready to create your first *app* within the *Nano* service tier – the starter, free one.

Since we are talking about HTML, CSS, and static websites, to us a Divshot app is just a website and nothing more. You'll have three app environments at your disposal:

1. **Development**: for when you are working on a new website, or to an addition to an existing website and don't want anyone else to see it, yet
2. **Staging**: for when you have something which you think is finished and just needs some testing in the wild before going on and updating the real website
3. **Production**: the real website

There are two ways of uploading your files on a Divshot app. We are going to use the simplest one, leaving the more advanced one to your curiosity or lack thereof.

Once you have created your first app – which you should have done when you signed up:

- Stay within the *Upload* tab
- Compress **the contents – not the entire folder** – of your `easyhtmlcss-work` folder into a `.zip` or a `.tgz`
- Hover with the mouse on the *Development* environment mini-panel

---

[5]http://github.com/

[6]http://mac.github.com/

[7]http://windows.github.com/

[8]http://www.divshot.com/

- Choose *Upload* and upload the compressed folder

Divshot will handle it and will produce a revision of your app.

At the time of writing, there's what I think is a minor bug in the interface so in order to force the list of the available versions of your app to be updated, click on the *Development* mini-panel to close it then click again to reopen it and you'll see the first revision – v1 – of your app listed.

Here's a complete sample line:

```
1  v1 william@williamghelfi.com: Pushed to Development 21e516 ... less than a \
2  minute ago
```

To see it in action, click on Development. The URL will be something like:

```
1  http://v1.development.easyhtmlcss.divshot.io/
```

You can then upload another revision, promote a revision from development to staging to production, or decide to directly upload on production.

Either way, you have three different environments to streamline your workflow as it best suits your needs.

Lessons learned

In this chapter you went from zero to online in the minimum possible steps, and I hope you liked it as much as I still do after 14 years of professional web development and a couple more as a hobbyist.

Starting to feel slightly proud and amused?

Well, it will be like this for years. And it only gets better over time, as you learn more and your skill level skyrockets.

Here's a quick recap of what you learned:

- Web pages are not that much of a deal, **just some text interpreted by a web browser** (well, ok, I'm cheating right now... but you'll learn the truth in the next chapter).
- Hosting a few web pages to a few people can be **damn easy with Dropbox**.
- Hosting a fair amount of web pages to a fair amount of people can be **fairly easy with GitHub Pages or Divshot.
- Writing a web page and putting it online is **fun**.