WEB DEVELOPMENT & BEYOND

THE COMPLETE GUIDE TO BECOMING A WEB DEVELOPER & SUCCESSFUL FREELANCER

KYLE PRINSLOO

THE COMPLETE GUIDE TO BECOMING A WEB DEVELOPER & SUCCESSFUL FREELANCER

About the author:

My name is Kyle, that's me on the left ☺

I am a freelancer specialising in Digital Marketing, Web Development/Design. I have a history in helping companies grow their online businesses and I run a blog at www.studywebdevelopment.com.

My education has come from a mix of qualifications, self-studying and 'real-life-lessons'.

To read more about me, click here.

Why I wrote this eBook

To put it very simply… I am a huge advocate of coding/web development/programming – whatever is the best term these days ☺

I believe it's extremely important that everyone is exposed to web development at some stage in their life. Not only is it an enjoyable career, it is also an exciting challenge to create something that is of value to others and it also develops how you think.

"I think everybody should learn how to program because it teaches you how to think."
Steve Jobs

I've purchased web development and freelancing courses, eBooks and learning materials, but I've found that for the most part there is a lot of 'fluff' in the content. This is also a reason for creating *Web Development & Beyond*.

Table of Contents

This will be slightly different to most other eBooks out there as I am not numbering or structuring the eBook according to the Table of Contents.

The reason for this is that it covers 4 main areas (listed below) and because I feel the whole eBook should be read from start to finish.

After you are done reading this eBook, you will either be:

- Equipped to master web development and become a successful web developer
- Transition from being a web developer to a freelancer
- Take your freelancing career to the next level

This eBook does not have the “fluff” and small narrow paragraphs that takes up unnecessary pages just to make it appear bigger than what it actually is. Each page is intentional and important regardless of what stage you are in on this online journey. This eBook will help you in ‘simple English’ with actionable steps and insight that you can APPLY and not just READ.

I just ask 2 things from you:

1. READ EVERY SINGLE WORD in this eBook and visit the resources that I recommend.
2. Put into ACTION everything which applies to you. Don’t just read and think about it. DO IT.

You might not be interested in all 4 sections, but I urge you to read it. If you are a freelancer already, just browse through the other sections at least. If you are an aspiring web developer, you should definitely read the freelancing section as well because this will prepare you and help you in this journey.

I’m grateful that you have downloaded this eBook.

I’ve decided to give this eBook away for free as I want to help as many people as possible succeed at web development (and beyond). I am no expert by any means, I am just someone who would like to help others where possible.

Most of what is written in this eBook I have applied myself or I personally know of others who have achieved positive results after putting into action everything mentioned below.

LET’S GET STARTED

I love when people succeed… I truly feel happy when I see a positive difference made in the lives of others.

An important question to ask before moving on is: What is success?

Success is different to every person. It could be more time, more freedom, better relationships, a balanced life, happiness, monetary success, physical possessions, etc.

That sounds very “fluff” like, but what does web development/freelancing have to do with SUCCESS anyway?

It actually has EVERYTHING to do with it…

Have you ever thought about your work-life?

I have... Here is my thought process:

You spend most of your life either at work or thinking about work.

Here’s an average daily breakdown:

You work from 9-5 (8 hours)

You travel to and from work (1 hour)

You work an extra 15 minutes each day (15 minutes)

You sleep for 7 ½ hours (maybe more, maybe less)

Maybe you are stressed about a work project you are currently doing? Do you talk to friends and family about work issues or stresses or think about work once you are home? (15 minutes)

Out of a 24-hour day, your average daily ‘work related time’ is 9 ½ hours, you sleep for 7 ½ hours. That means you only have 7 hours each day to spend quality time with your family, to relax, read, eat, watch TV, etc.

The point is this: You spend the MAJORITY of your day (and life) WORKING.

You have to do something that you are happy about waking up to every morning.

You have to do something that you fulfils you and brings you joy. Because you are reading this eBook, it means you are interested in web development and/or freelancing. Both of which is extremely rewarding and fulfilling (if you allow it to be).

GETTING STARTED AS A WEB DEVELOPER/DESIGNER

My definition of a web developer is that you are an 'artist for the internet.' You are creating an "online painting" for the world to see. For a more technical definition, click here.

If you are a 'newbie' and you are completely new to web development, it can be very daunting and confusing at first.

When I started coding, I was so confused… I had no idea where to start, what to study, how to study, what is important, what I need, how to upload a website to the internet, what are things like an FTP, how to apply the code, etc.

To address your common questions and to put things into perspective, I wrote an in-depth article called **Web Development 101: Understanding The Basics** which covers the web development environment, hosting, domains, how to get started with coding, how to upload a website to the internet, text editors, simple explanations and more that you can read here.

The link above is more the 'how-to-steps' that gives you the knowledge in theory and the practical things you need to *know* and *understand* as a beginner.

Now, onto a very important section which deserves the most attention and focus:

How to **APPLY** what you *know* and *understand*.

Think about it…

ANY PERSON ON EARTH CAN LEARN HOW TO CODE. IT IS WHAT YOU **DO** WITH THAT KNOWLEDGE THAT WILL SEPARATE YOU FROM SOMEONE ELSE.

Here are 6 things to consider when you are aspiring to be a web developer and when you are a web developer:

<u>1 – The Right Attitude + Belief</u>

The first hurdle to get through is **<u>yourself</u>**.

Let me explain…

Why is it that some people in difficult situations manage to succeed and those who have 'perfect' upbringings sometimes end up in positions that don't make sense at all?

You need to <u>want</u> to become a web developer – don't aim to be mediocre; aim to be a successful web developer. More than that; aim to be the best web developer that you can be.

The next step after aligning your attitude to becoming a web developer is BELIEVING that you are one.

You may be thinking, "but I don't know enough to be called a web developer" or "but how can I be a web developer if I haven't even started yet?"

These are valid questions… and I'm not saying you should lie about being a web developer or that you should take on work far beyond what you are able to do at this moment.

What I am saying is that you need to "trick" your brain and boost your confidence in BELIEVING that you are a web developer rather than second guessing it. Try it and I can assure you that this method is far better than doubting the fact.

If someone asks you, "so what do you do?" tell them without any doubts, "I'm a web developer/freelancer."

This may lead nowhere, but you'd be surprised how many people will be very interested in the fact that you are a web developer/freelancer. Almost as if you pulled some Star Wars Jedi mind trick on them.

Whether you'd like to become a web developer or freelancer, your mind-set and attitude is crucial to your success.

2 – Determination + Discipline

Failure and doubt are inevitable. The question is how you will deal with it when it happens.

You need to be determined, motivated and driven to really be successful at web development. There are many aspiring web developers out there. Not all of them will be determined and driven and motivated.

Not all of them will be disciplined in learning web development. That might not sound right, but it is true. We are all busy with other things and you may even have a full-time job while you learn web development (that's in fact how I started).

The hard part is not starting… it's whether you keep going. You need to be disciplined in learning.

Set a schedule and stick to it.

I used to work from 8am-6pm, come home to my wife and be exhausted, eat, relax for a bit, and then start learning to code from 8pm to 1am/2am for a few months. That was very challenging for me, but I needed to be highly focused and it worked out well because of that discipline.

Maybe you have children or other important priorities, but don't make excuses. If it's important to you, you will make a plan to learn and you will force yourself to be disciplined.

3 – Stubbornness

You may be thinking, "why stubbornness?"

It's good to be stubborn as you learn.

What I mean by that is if you have an error in your code or if it is not coming out how you'd planned and thought (which happens more often than you might think,) don't just move on if you can't find the solution and don't cut any corners. Be stubborn and figure it out.

Although it may take you longer to figure out, what you learn by being stubborn in this regard is that your knowledge will remain with you and you will be more efficient and effective in your future projects because of this.

4 – Prioritising

If you want to do something great, it comes with sacrifice. This is challenging to accept for most people.

If you want to be a great programmer or freelancer and you are complaining that you don’t have time, then just look at your daily/weekly/monthly routine to see what you can remove to free up more time so that you can learn more, develop your skills and apply what you know.

One of the biggest distractions I’ve eliminated in my life is watching TV. It was hard to do at first, but even if I got given a free TV the size of my wall, with free unlimited movies and series, I wouldn’t even be interested now.

I’m not the only strange one out there who does this. Seth Godin, one of the greatest marketers and thought leaders of our time is also a big advocate for not wasting time watching TV.

Whether you are a student, a full time employee, a husband/wife/father/mother, you need to know how to prioritise.

Every single day there is around 140,000 websites added to the internet. Imagine… that’s almost 2 websites EVERY SECOND!

So if you are interested in creating a website for your own business, it will be good to start asap!

Do what you need to do, don’t neglect your family, cut out the TV, cut out distractions and learn how to code – after all, that is your goal so be serious about it and work towards it.

5 - The Skills

You can’t become an accountant without understanding accountancy. The same principle applies for web development.

In order to be a web developer, you need to know what web development is, how it works, the programming languages and their importance, what to learn, communication skills, problem solving, innovation and more.

It’s important that you know how to learn effectively. We will go into some techniques for learning and other tasks later in the eBook.

6 – Your Goals

You need to define your goals.

WHY do you want to be a web developer/freelancer?

Would you like to learn web development and apply for work and be an employee?

Would you like to be a freelance web developer and work for yourself?

When you create a website, what is your deadline?

Once you have your goals, make sure you are driven by it and focus on working towards achieving them. Don't be like most people and set goals that fade away after a few months…

How to set your goals in 5 simple steps:

Be S.M.A.R.T (Specific, Measurable, Attainable, Realistic, Time-Bound)

Specific: To become a freelance web developer

Time Bound: 12 months

Measurable: I will measure my progress every month and set defined goals to reach my specific goal.

Attainable: Challenging, but yes.

Realistic: Definitely.

Once your main goal has been set, break down the 12 months into medium-term goals, in this case it would be:

(Don't worry if these terms are a bit technical for you, I'll explain in more detail later in this eBook.)

In 3 months, I want to know Frontend Web Development

In 6 months, I want to know Backend Web Development

In 9 months, I want to start working on my portfolio website, build my brand, improve on my business knowledge and perfect my web development skills.

By 12 months, I want a portfolio of 5 websites, I want to be a full-stack web developer and I want to know the fundamentals of freelancing and gaining clients.

Once I have my medium-term goals, I break it up into my monthly goals:

Frontend Web Development

Month 1 – Learn HTML & CSS

Month 2 – Learn JavaScript

Month 3 – Create website using HTML, CSS & JavaScript

Backend Web Development

Month 4 – Learn PHP

Month 5 – Learn more advanced PHP

Month 6 – Make a functional, dynamic website with HTML, CSS, JavaScript & PHP

Portfolio, Personal Brand, Business Knowledge, Web Development Knowledge

Month 7 – Create a personal portfolio website, social media profiles and build a personal brand

Month 8 – Reach out to businesses & organisations to create websites for them

Month 9 – Improve portfolio, web development skills and personal brand

Portfolio of 5 Websites, Full-Stack Web Developer, Freelance Fundamentals, Clients

Month 10 – Must have a minimum of 5 websites on portfolio

Month 11 – Learn freelance fundamentals and business essentials

Month 12 – Reach out to prospective clients, promote your work and get clients

Now that you have your monthly goals, break it down into daily goals:

You don't need to study full-time to make this goal a reality – this would help for sure, but if you are working and you can only study part-time that is also fine. If you can dedicate 4 hours every night to this schedule, I fully believe that you can accomplish this goal if you are dedicated, disciplined and motivated enough.

This is just a very short example and a simplified guideline, and it is not intended to be a comprehensive goal-breakdown, I just want to illustrate how it could be done. The programming languages mentioned above are just examples too.

Write out your goals, print it out and stick to it.

One of THE MOST important aspects of achieving your goal is:

ACCOUNTABILITY

If you are accountable to someone and you have weekly/monthly calls to discuss the progress and he/she holds you accountable and motivates you when you feel like giving up or have any doubts, it will push you through and help you stay focused. The more honest and 'hard-core' your accountability partner is; the better!

Last, but not least… **DO IT** and stop making excuses.

I came across a guy named Justin on social media and I'd like to share his quote:

"One Summer I decided I wanted to finally start doing something with my life. So I went to the store and bought a book that taught C++ in 30 days. I then went to the bank and took out all of the money I had at the time (I was 14 years old) and I went to my dad and I said, 'Dad, if I don't learn this whole book in 30 days you can keep all of this money, every cent.'

So that Summer I started learning to code, and I slacked off at first but when I realized I put my life savings on the line I studied for hours and hours until I knew the book inside and out.

If you really want something, take the time to do it. If you don't want to take the time, then obviously you don't want it bad enough. After I learned that book I didn't start making billions like everyone says. I needed practice if I truly wanted to become good at coding. So for the following 4 years after that Summer I coded for free, making websites, apps, designs and much more. I would contact charities or family friends and see if they needed work done. My school constantly needed websites for teachers or applications to do certain things.

Almost 2,000 hours of my time went into building websites and I didn't see a single dime, because I didn't care about the money – I cared more about gaining knowledge.

That's how I learnt how to code."

Justin Ryan, 18-year-old entrepreneur and freelancer

Does this motivate you and encourage you, or does it convict you with your delayed excuses in getting started?

Either way, Justin shared some wise advice and I am excited to see his progress and future successes.

BECOMING A WEB DEVELOPER

Engineers have fancy terms they need to know, and so do web developers ☺ Here are some fancy terms you need to know

What is server-side and backend development?

(Just to keep it simple for now, think of them as the same thing.)

This is everything you can't see on a website. Think of a server (a big hard drive with all the site's information) in a location anywhere in the world, processing all the website data and then sending it to the client's browser.

I'll go into more detail on the next pages, but here are some backend development programming languages:

Java, Ruby, PHP, .NET, Python, Perl, JavaScript (NodeJS), C++ and SQL

What is client-side and frontend development?

(Just to keep it simple for now, think of them as the same thing.)

This is everything you see, click and interact with on a website.

I'll go into more detail on the next pages, but here are some frontend development programming languages:

HTML, CSS and JavaScript

What is a framework?

A framework is basically a package of files, folders, code, concepts and practices that make coding with the fundamental programming languages much easier and quicker.

What is OOP?

Object oriented programming stores all of its data and programming logic in objects. Objects are defined as data and programming logic bundled together in a nice neat package.

If you're new to web development, you are probably thinking:

What do I need to study to become a web developer?

That's a great question. Below is a short and understandable breakdown of common programming languages and some other things you should learn.

As a great foundation to **start**, know HTML, CSS, JavaScript and Bootstrap (in that order).

Don't worry about the guys out there who say 'HTML and CSS are not programming languages'. Start with this and learn more from there.

Side note: I've updated the courses section on the site. If you haven't done so already, go to https://studywebdevelopment.com/web-development-courses.html and get the right courses sent to you.

HTML:

Also known as Hyper Text Markup Language, is a set of defined tags used to structure a website. Think of building a wooden house… HTML will be the foundation and the wooden framework of the house. It doesn't look pretty (that's where CSS comes in) this is just the 'raw shell' and layout of a website.

W3schools has a great beginner's intro to HTML here.

HTML has an updated version called HTML5.

This is the one you need to watch out for! It's one of the fastest growing job trends on indeed.com which shows its popularity. HTML5 is probably *one of the best* long term languages to study within the next 3 years. Some sites that make use of HTML5 are Ford, Peugeot and Lacoste – they are really cool (they just take long to load which is my main criticism).

This is a good (free) HTML5 course

The good thing about HTML5 is that it's more dynamic (you can create awesome sites using less code and it does more).

CSS:

Also known as Cascading Style Sheets, is a collection of style tags used to make the website styled. In the wooden house example, CSS would be the paint and how the house looks like.

Most websites make use of CSS. To get an idea of the practical aspect, check out this free course by Udacity here. For short tutorials, click here.

CSS3 is the latest version of CSS. It's more dynamic and works hand-in-hand with HTML5. Just like salt and pepper goes together, so it is with HTML5 and CSS3.

Take this very helpful (free) HTML5 and CSS3 course here.

JavaScript:

JavaScript is the most popular programming language in the world. A simple example of what JavaScript can do is: *If this, then that*. So JavaScript is the interactive aspect of a website/app. If used correctly, it can really make a website stand out from the rest.

Did you know that JavaScript has more frameworks than universe has stars?

(OK that was a bad joke, but it has a lot and the more you master, the more your skills will grow which results in higher fees for your work).

As you can see below, JavaScript is very enjoyable:

Learn JavaScript here, here, here and here.

Websites using JavaScript are: Lost World Fairs and Cascade Brewery.

jQuery:

jQuery is just a simpler, quicker and easier way of using JavaScript. Think of it as a bundle of different plugins to add to your code which saves you a lot of time and makes it much easier for you to add a feature that you'd like.

Before you get started with jQuery, it's advisable to have knowledge of HTML, CSS and JavaScript. jQuery is used by some big names like Google, Microsoft and IBM.

To learn this 'art form' click here, here and here.

Bootstrap:

Bootstrap is the most popular framework for building responsive, mobile-first websites. I personally love Bootstrap and I use it on at least 90% of my own websites and client websites.

It's extremely easy to learn and it's amazing how responsive and clean it can make websites look. Read this article on some juicy Bootstrap templates.

Sites that make use of Bootstrap are: Newsweek and Lyft.

Learn Bootstrap here and here.

PS – if you like Bootstrap, sign up to GetBootstrapWeekly.com

Ruby:

Ruby is a great object-orientated programming language. What separates Ruby from languages like PHP, Java, C# and Python is simply because it is faster and more productive. The websites that you can create using Ruby (and the Ruby on Rails framework) are incredible. If you want to create sites like Airbnb, Fiverr, GitHub, Kickstarter and Zendesk for example, then make sure you know Ruby.

Learn Ruby here, here, here, here and here.

Ruby on Rails:

Think of Ruby on Rails as 'jQuery for JavaScript'. Rails is the framework, and Ruby is the fundamental language. I highly recommend Rails once you have a grasp of Ruby.

Make sure you learn JavaScript along with Ruby on Rails as you will need to use JavaScript when you advance in Rails. Also make sure you know Git and GitHub (recommended).

Learn Ruby on Rails here, here, here and here.

Get Michael Hartl's Ruby on Rails Guide here.

Java:

Java is a big player in the programming world. *Some* argue it's slowly being replaced with Python. I would personally say that Java should definitely be a learning priority after HTML, CSS and JavaScript. Java is used to create Android Apps, software, games and website content. Websites like LinkedIn, Amazon, Twitter and Netflix make use of Java.

Learn Java here, here, here, here, here, here, here and here.

Python:

Python is *similar* to PHP, Ruby and JavaScript in the sense that it is an object-orientated language. The good news about Python is that it's considered to be a very easy programming language to learn and it's also a good career choice. Sites like Pinterest, Instagram, YouTube, DropBox, Google, Reddit and NASA make use of Python through Django – Python's web framework (more info below).

Learn Python here, here, here, here, here, here and here.

Django:

Django is the most notable framework for Python. The main benefit of Django is that it's built for speed and structured data. It's mostly used for CMS's, database/news sites like Disqus, The Guardian, The Washington Post and social networks like Pinterest and Instagram – so in other words; big apps/websites.

Learn Django here, here and here.

PHP:

Also known as Hypertext Pre-processor, is the most popular server-side programming language. PHP is often used as the foundation of CMS's (Content Management Systems) like WordPress and big websites like Facebook and Wikipedia.

Learn PHP here, here, here, here and here.

Laravel:

Laravel is the most popular PHP framework out there. I won't touch on all of them - read more here, but this is most noteworthy. The main Laravel benefit is that it is made for speed. You will be able to create clean apps and websites much easier and more effectively.

Sites that use Laravel are One Plus and True Lancer

This is a good Laravel course

SQL:

SQL (Structured Query Language) is a standard language for database-centred websites and accessing databases. If you need a login system or if you'd be dealing with data/accounts, then you definitely need SQL. By using SQL, you'd need to use a server-side language like PHP for example.

There are a few databases to learn, but to start, I'd recommend MySQL which you can learn here and here.

As your knowledge of general SQL and MySQL improves, I'd recommend learning MongoDB. Read more here.

C++:

C++ is a general purpose object orientated programming language. It is considered to be difficult to master, but once you know it, chances are you will love it. C++ runs well with almost every other programming language. Most systems out there can run the C++ code.

C++ is a powerful language to use in Microsoft's .NET Framework.

When/where is C++ used?

Large-scale applications, PC games, game development, video games, application software, operating systems and databases.

Companies like Intel, IBM, Microsoft, Adobe and Firefox use C++.

Learn C++ here, here, here, here, here and here.

C#:

C# ('see-sharp') is the programming language for Microsoft's .NET Framework.

C# is used for Windows applications, Android Apps & iOS Apps with the technology from Xamarin.

Learn C# here, here, here, here and here.

Node.js:

Node.js is a server-side framework for JavaScript. Node.js basically makes JavaScript (which is a front-end script) into a server-side technology. That sounds fancy, but all it means is that Node.js makes the overall website faster and it is perfect for real-time apps/websites that require a news feed, chat rooms, and other data-heavy applications.

Some sites that run on the Node.js framework are: LinkedIn, PayPal, Dow Jones and Groupon. If you'd like to use Node.js to build server-side applications (or big sites mentioned above), you can use Express.js (this is the framework for Node.js).

Learn Node.js here, here and here.

ASP.NET:

Made by Microsoft, ASP.NET is an object-orientated, server side framework which is written using the C# and Visual Basic programming language.

So what can .NET do and where is it used?

Dynamic websites and web apps with HTML5, CSS3 and JavaScript, Enterprise-level software development and web APIs.

Websites that use ASP.NET are Dell, ASOS, TacoBell, W3Schools and MSN.

Learn ASP.NET here and here.

AngularJS:

AngularJS is the most robust of the JavaScript frameworks. It is perfect for data-heavy websites like iStockPhoto, Udemy, Virgin America, Ford, Lego, and GoPro for example.

If you want to create websites that are much simpler than the website examples above, chances are you will be fine with a JavaScript framework like Backbone.js.

Learn AngularJS here, here, here, here and here.

Backbone.js:

Backbone.js is a 'light-weight' JavaScript framework with the sole purpose of making it easy to connect with a site's server-side application. It loads very quickly and it is ideal for creating awesome single-page web applications that don't always require a full-page refresh (think Gmail as an example).

Websites that make use of Backbone.js are: Reddit, Pandora and Airbnb.

Learn Backbone.js here and here.

Meteor.js:

Meteor.js is an awesome framework to learn. In fact, I would go as far as to say that it is one of the easiest frameworks for beginners to learn. It's advisable that you know JavaScript before studying this framework.

This is the ultimate list of Meteor.js resources to learn more.

CMS:

A CMS (Content Management System) is basically a platform that is the simplest way to create a website and make quick changes that (mostly) doesn't require any coding knowledge – this is a VERY simple overview. The 2 main CMS's are: WordPress and Drupal.

WordPress doesn't really need an introduction, but for the complete newbies, WordPress is responsible for around 25% of the internet!

Drupal is considered to be difficult compared to WordPress, however, once you get used to it, it's pretty powerful in terms of the features, scalability and speed.

Generally speaking, Drupal is favoured more by developers due to its features and the fact that we (developers) can customise it further. Saying that may get some WordPress developers offended and they'd want to hack my PC ☺ so to be safe, choose for yourself and read this comparison article.

Swift:

Swift is the fastest growing programming language in history! It's built by Apple (not the granny smith apple) and they have big plans for it so it would be good to take note of it as the popularity grows. Swift is what you learn if you'd like to become an iOS App Developer.

For the best Swift courses out there, learn here and here.

Git:

Git is a version control system that is used for software/web development. It allows you to revert files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more.

Many large companies and development teams make use of GIT to work and manage projects. Whether you are a web developer in a company or a freelancer, this would be a good move.

Learn more about Git for free here.

GitHub:

GitHub is a code hosting platform for version control and collaboration. Just like you get a website hosting platform like BlueHost, HostGator or NameCheap for example, GitHub is hosting your code (how cool is that?). To really understand GitHub, you need to understand Git.

You will be able to 'show off' all your projects and contribute to other projects you find. It's almost like a social network for developers.

Learn more about GitHub for free here and here.

Note that this is NOT an exhaustive list of programming languages and things to learn about… This is just a list of what I have seen to be popular and noteworthy through my own experience and speaking with many web development professionals.

Still not sure what to learn?

Here's a very simple breakdown:

If you want to create sites that will use a database and data analysis:

Learn SQL and Python

If you want to make your own WordPress themes and work with WordPress:

Learn PHP

If you want to make modern, interactive Websites and Web Apps:

Learn Ruby (on Rails), JavaScript, JQuery, HTML5, CSS3, Python, Meteor.js, Angular.js and Java

If you want to create games:

Learn C++, Java and Unity

If you want to create Windows Applications:

Learn C# and Visual Basic

If you want to make iOS Apps:

Learn Swift, Xcode and C#

If you want to make Android Apps:

Learn Java and Meteor.js

Still not sure about what to learn?

Just type in, “Junior Web Developer Jobs in [your City]” on Google – even if you are not in the job market at the moment it gives you an idea of what the recruiters and employers are looking for so you know what is in demand.

Being a web developer doesn’t mean you have to master every programming language available. Sometimes it’s about just knowing enough and being able to know where to get information from and to best make use of the tools and resources available online.

It’s like most CEO’s for example. They don’t need to be able to create a Financial Statement from scratch or to create an Employment Contract from scratch and so on… They just need to know enough to understand it and in this case they can speak to their Finance Department and Human Resources Department for the information.

I generally prefer learning through online courses, but you may prefer printed books or articles. Whatever it is, just do what is best for you. Learn at your own pace and have the dedication to complete it and stay committed!

Did you know that the average student does not even complete the first hour of a three-hour course? – That’s after paying $57 for it… That says it all.

Source: Huffingtonpost

PS – I’ve studied a few courses through **Thinkful** and they are offering an exclusive discount to SWD readers. This is not a ‘fake’ exclusivity deal – it’s the best deal you will get. **Click here** to view their courses and get between $50-$150 off.

I also highly recommend Firehose Project. You can get 2 weeks free with them here.

MYTH:

“I know more programming languages than you, so I will be more successful as a developer.”

Not true. You can know the most advanced programming languages, but if you can’t apply it properly and create a website that meets the client’s needs then what is the point with all that knowledge?

If ‘Bob’ knows HTML, CSS, JavaScript, Bootstrap, SQL, Java and PHP, technically he *should* (by conventional norm) be able to create way better websites/apps than someone like ‘Fred’ who only knows HTML, CSS, Bootstrap and basic JavaScript.

Of course this is not always true. ‘Fred’ could just APPLY his knowledge and skills better than ‘Bob’ and create a far better site that functions well and is visually awesome.

Just because you have knowledge of all the programming languages, it does not mean that you can create a better website than someone who knows half as much as you do. That’s why it’s important to keep reading and learning more about the industry and trends.

I’d also like to add that just because you know how to program in every programming language, it does not mean that you should use aspects of every programming language for every website or landing page that you create. Sometimes keeping it simple is often best.

When you are a web developer, you have 3 choices:

1 - You can work for someone (a company)

2 - You can work for yourself (freelancing)

3 - You can work for someone AND yourself (do freelancing on the side)

If you don’t code your dream, someone else will hire you to code theirs.

Not everyone wants to be their own boss, and if that’s not for you, that’s no problem.

If you’d like to work for yourself as a freelancer, then this next section won’t apply to you, although I’d recommend just browsing through it. If you’d like to work for a company, then carry on reading below.

We will go through each of these choices in more detail and I will provide practical advice and actionable steps you can take to excel in the decision you make.

If you’d like to work for someone, unless you are headhunted by the company itself, you will need to go in for a job interview.

More on the job interview process and advice on the next page.

JOB INTERVIEW PREPARATION

Here are a few things to consider for the interview:

Stand out from the crowd.

The chances are you are not the first person going in for the interview, so make a lasting impression. Do your best to research the company and speak about what you've read or found out and then also see what type of solution you can provide in the interview as well.

When I went in for a job interview as a Digital Marketer at a large company, I went through all of their websites before the interview and noticed a few broken links and some poor reviews on Hello Peter and some overall suggestions to improve sales. When I mentioned these findings, it helped me stand out from the crowd and showed initiative from my part (and I got the job). I urge you to follow this same principle.

Make sure you dress appropriately.

Don't go in for the interview wearing shorts and a casual shirt if you should be wearing jeans and a button up shirt for example – the same applies for the other way around.

Make a stand-out CV.

If you have a CV, make sure you update it and think creatively about how you present it. Make sure it has no spelling mistakes (ask someone to proof read it or use Grammarly). In one of my previous jobs, I had to interview and hire staff and I can't stress enough how important a good CV is. it is fascinating how many poorly written CV's are out there and people wonder why they don't get the job. Make sure the layout is done very well and make sure it looks clean and it portrays you in the correct manner.

Click here to get an idea of some uniquely designed CV's. This one is insane…

Be yourself.

Stay true to who you are. If you are an introvert (like I am), you will most likely struggle to 'sell yourself'. Try be more confident in your interview and really put yourself out there. Leave nothing left to be said. Make sure your interviewers have no doubt about who you are, what you've done, what you can do, and why you will be an asset to them.

Make it count.

You have one chance. Believe in yourself. There may be more than one interview, but approach each interview as the first and final interview – don't have any regrets when you walk out.

Some general interview questions to prepare for:

Tell me a bit about yourself?

What are your weaknesses?

Why should we hire you?

What is your greatest accomplishment?

What is your biggest failure?

Do you prefer to work alone or in a team?

Do you have any questions?

"Can you show us your portfolio?"

This is one of the most important stages of the interview. If you may have been nervous in answering the questions earlier or if you feel you didn't answer the questions well, then this time right here is vital.

Although this is showing off what you've done in the past (and possibly what you are currently doing), this part of the interview is when you should 'shine.'

If you don't have a portfolio to show off, I highly recommend that you build your portfolio BEFORE going in for the interview.

Take a look at these 3 helpful articles here, here and here on how to create a portfolio website.

Your web portfolio

Think about it in more detail… say you go for an interview and you sit next to 3 other candidates for the interview. All of you have the same college/university qualification which states that you've studied programming and you know how to code. What will set you apart from each other then? Maybe how you get along with the interviewer in this case, but on paper at least; you are all the same…

Now, imagine this same scenario above, but YOU have a portfolio of 3-5 websites that you've created by freelancing on the side or by creating websites for businesses and you bring this 'proof' along to the interview. You will instantly stand out from the rest of the candidates and your chance of getting the job has significantly increased.

Your portfolio is what sets you apart from everyone else and it will eventually make your qualification a *piece of paper* – seriously.

Bottom line is… you need a brilliant portfolio to show off what you've done.

If you are still unsure about building your portfolio, then please email me and I'd love to talk to you because it is vital that you understand the importance of this.

If you are thinking, "but I don't have a formal programming qualification." I'd like to address your thoughts on this very simply: I dropped out of school, I taught myself how to code and it worked out fine BECAUSE I built up a portfolio of websites and businesses and that was my 'piece of paper.' There are MANY self-taught programmers out there who have followed a similar path and if your portfolio is good enough, you really don't *need* a qualification (although it can definitely help).

Actionable tips to grow your portfolio:

Contact a local charity shop/restaurant/NGO/church and tell them you will create a website for them (at no charge) and just mention that this is to build your web development portfolio and to gain experience.

In exchange for the website, you'd like their permission to add it to your portfolio and you'd like a testimonial. It's a win-win for you and for them. They get a good website, and you learn more by creating more projects and you gain more experience on your portfolio which is an asset to you.

Think of this strategy as the 'liability into asset strategy', it is a liability for you because you are not getting paid and it's taking your time to create, but at the end of this process you will have an asset to your name that no one can take away from you.

SAim for at least 5 websites for your portfolio before applying for jobs. This is a good foundation to work from and it shows off your variety and skillset.

Big companies will often give you a test in the interview like creating a simple 3-page website that makes use of [insert whatever programming languages are necessary] or to create something relevant to your job position like a simple database for SQL developers as an example.

Another test you will be presented with is a ‘bug’ (code with errors) and you’d have to fix it and make the code work – this is quite a common one because it tests how you deal with problem solving and how quickly you find the solution. If you can find the errors, it means you know your code well so it is a perfect test.

It’s important to distinguish the different *types* of interviews. What I mean here is that the interview process for a Junior Web Developer will be completely different to the interview process a Senior Web Developer will experience.

(This depends on the company size and various other variables, but it should help you understand the types better).

So for example, Senior Developers will be asked advanced technical questions and possibly questions related to management. They will also be presented with, “How would you solve this problem?” questions and very specific problems to see how they would handle it and come up with a solution. The point of hiring Senior Developers is making sure they know how to handle every scenario and execute it with excellence.

Junior Developers will be asked questions related to their programming knowledge and skills, explaining definitions and general questions. The point of hiring Junior Developers is making sure they have the knowledge and can do the work given to them. If you work in a big team, Junior Developers don’t often receive the best work, but it’s a good place to start.

If for whatever reason you are unable to find a job as a new web developer, you need to improve on your portfolio of websites, your development knowledge, your social skills, your strategic approach in finding potential employers and your attitude – nothing else.

Should web developers have web design skills?

I ran a poll amongst the SWD community and out of 500+ responses, over 80% mentioned that having design skills or at least thinking like a designer will be helpful as a web developer – so there you have it; have some design skills ☺

I'd take this one step further and personally say that I believe it is very important to know at least the basics of web design even if you are a Backend Developer. What's interesting is that MOST backend development is driven by what the user experiences. Think about it… if users want more dynamic videos with a social community database, you (as the backend developer) need to create this.

If you can think like a designer, it can only benefit you. You don't have to be an expert at user design, user experience or web architecture, but make it a priority to know at least the basic concepts of design.

Learn more about web design here, here, here, and here.

BECOMING A FREELANCER

Before moving over to freelancing, it's important to ask yourself:

Why would I like to become a freelancer?

Once you have your "WHY" you need to structure everything around it because this is your motivation behind what you will need to do to accomplish it.

You don't always need to move over to freelancing without some financial security or projects lined up. Some have succeeded by just doing it and figure it out during the hard struggles.

I personally won't advise doing that, so I will be taking the 'safer' approach in this section.

If you are transitioning over to becoming a full time freelancer, you need to ask yourself if you are receiving constant work coming in on the side which will justify you leaving your full-time job and you need to make sure the income that you are generating will pay for your expenses.

Generally, if you can make around 75% of your full time salary income from your side freelancing, you *should* be able to make a lot more once you work full time on it as that will be your main focus.

When going the approach of side freelancing in the hope of moving over to full-time freelancing, you need to really be dedicated and goal orientated. Are you willing to wake up early in the morning and work on your side business, come home after work and spend time with your family and then work until late at night and also sacrifice most of your weekend working until you are able to move over to full-time freelancing?

These are tough questions to ask, but you need to answer them.

It’s important to understand the main benefits of working for a company, and that is the fixed income every month which allows you to budget, medical or retirement benefits and you also have the potential to grow in your position and increase your income over time.

One of the most difficult truths to accept in freelancing is that you don’t have a fixed and consistent income. Some months you may have very good months and others may be very bad months. In Brad Hussey’s words, “you get mountains and valleys.”

I mentioned the ‘bad’ things about freelancing, but you can think of the ‘good’ things. This will often be determined by your “WHY” – whether it be working from home and spending more time with family or the freedom that comes with it too.

Either way, whether you are freelancing or working for a company, there are pros and cons for both sides and only you can make this important decision.

Here are some things to consider for freelancing:

Cash reserves:

Save as much as possible – preferably up to 6 months’ worth of expenses BEFORE you transition into your freelancing job.
Once you get clients, you will also need to have an emergency cash fund to last you a few months should you not have any client work within that period. Be wise about this and save for the ‘rainy days’. This is important – especially if you have a family.

Personal portfolio:

One of the most important things that a client looks for when hiring someone is your portfolio. This shows proof of your talent and what you can offer. Your portfolio doesn’t only include the actual websites; it should also include the successful case studies.

Online and social media presence:

Your website is essential and it should be viewed as your ‘online CV’ so make sure you stand out from the crowd in how you create it. Take a look at this awesome example.
Here are some website examples:
Brad Hussey, Laurence Bradford, Ryan Robinson, Martin Wolf, Brennan Dunn and David Walsh.

You need to think creatively about how you will differentiate yourself on the internet in how you visually offer your business solutions to prospective clients and how you showcase your knowledge.

You should always have a blog. Whether it talks about your knowledge and experience or whether it’s just useful to others, you need to be talking about it. Social media is important. Make sure you are on Twitter, Facebook and LinkedIn and start engaging with a community and add value in whatever you do.

A contract and invoicing template:

It's important to have contracts with your clients before doing any work. Smashing Magazine wrote an awesome article on contracts and what is needed, I highly recommend that you read it.

Click here to read the article. This article is also quite helpful.

PayPal is fine for simple invoicing as well. Also check out FreshBooks for more advanced invoicing.

An invoice is a document which is a bill-of-sale applied to services rendered or skill-based work. It includes a list of products and services and the charges associated with them as well as any tax information and discounts offered.

Take a look at these invoicing templates to get an idea of how to create an invoice.
Also read this post on some more details on invoicing.
More finance related tools later in this eBook.

How much should you charge for your services?

This is a very important question and a popular question asked by freelancers.

You have two options:
Charge by the project, or charge per hour.

Most freelancers charge by the project (I do as well), while others charge an hourly rate. It's important that you understand how much you need to earn to provide for the lifestyle and expenses that you have or would like to have.

To figure out how much you need to earn or would like to earn, do this below advanced algorithmic formula:

Define your (desired) annual salary, let's say $75k per year.
Divide the annual salary by 12 months = $6250 per month.
Divide the monthly salary by 4 weeks = $1562 per week.
Divide the weekly salary by 5 = $312 per day.
Divide your daily salary by 8 hours = $39 per hour.

Side note: Read this detailed article - **https://studywebdevelopment.com/how-to-charge-for-a-website.html**

Now, of course you'd like to take a 2 week/one-month vacation so you need to factor that in as well.

To do this, take your vacation expense, let's say $3k and divide it by 50 (for a 2-week holiday) or 48 (for a 4-week holiday) and add it to your weekly salary.

For some, this example may be a lot of money. For others, it might be way too little. It should just be seen as a general formula and not 100% accurate.

I'd advise that you speak to someone with financial experience to help you in your decision making and expenses as this is an area where most freelancers don't really excel. You also can't expect to be booked for every hour of every day.

OK, so practically… how much should you charge?

It all depends on what value and solution you are offering your client and also where your client is based.

Just take a look at the price of a Big Mac Burger from McDonalds around the world and you will notice there is a big difference in the pricing because of this very reason. Yes, there are many factors involved in this pricing, but I hope you are understanding what I'm referring to here.

I've had clients where I'd charge a premium rate because I knew it was worth the price based on what they'd receive in sales. If you had a business making $200k per month and I told you I can increase your revenue by 10-20% within 3 months, but it will cost you $8k once off, you would definitely make the investment.

You could say, "I charge $80 per hour" or "I'll charge you $1k for the website" and then just be like most developers out there, but I'd really encourage you to think like a business consultant and charge based on the potential return on investment and not just the project time.

Be realistic… don't expect to make $'x' when you are starting out. As your portfolio grows and experience improves, you can really be picky about what type of clients to work with. Eventually you can even work less and earn more!

An important question to ask prospective clients before you go into so much effort in analysing their goals, their competition, and putting together an awesome highly detailed portfolio on how you will achieve their desired goals is this:

Do you have a budget set aside for this project, and is it more than $'x'?

This is an important question to ask as you may be thinking the client would be willing to pay $5k for the project and they could be budgeting no more than $1k.

A FREELANCER SHOULD:

Be a good communicator.

It’s crucial that you know how to speak appropriately to clients and to other sub-contractors. This point deserves a separate chapter all together, but to summarise it, you need to ensure that there are no “grey” areas in your communication as this could lead to time wasted in the future due to “oh, I thought you meant *that*.”

On a practical note, when you send emails, make sure you send the next steps thereafter. So for example, when you send your proposal email to a prospective client, don’t say, “here’s your quote, looking forward to hearing from you.” Explain at the end what the next steps will be. Add: “Once I receive your confirmation email and sign-off for the project, we can setup a Skype meeting for ‘x’ time to discuss the process of updating and progress of the project etc.”

By doing this, it gives the client an idea of what to expect next and it shows you are professional and know what you are doing.

Understand exactly what the client wants to achieve.

In order to understand what the client wants to achieve, you need to ask the RIGHT questions.
For most businesses, they want two things: more customers and more sales – the rest (leads, phone calls, traffic, more conversions etc.) all point to more customers and more sales.

You are responsible for creating a solution which achieves the client’s goals and objectives. Imagine creating a website and business solution for $10k and the client doesn’t generate income from it. Yes, maybe the client’s service or product sucks, but a lot of the reason for failure will point to you because you were responsible for taking on this goal to generate more sales and you failed – let’s not ‘sugar coat’ it.

I have not faced this scenario myself, but I can imagine how it must be like. Just make sure this is not you. This is added incentive for you to work hard and smart and dedicate everything you have to every project that you take on.

Be black and white – not grey.

You should always be clear about what you can provide, what the client can expect and about your terms and the process of how you work.

Update the client.

The worst thing that can happen is when a client says, “Hey man, what’s happening with the site, I haven’t heard any feedback on it this week?”

If you don’t update a client, you are not doing your job properly – simple.

Depending on the project size, you need to be updating the client AT LEAST every week or so. When I say “update” I don’t only mean, “I finished this feature and it works,” I am also referring to: “I have had some issues with implementing this feature, but I am working on a solution and I will update you the moment it works.”

Do you know how many freelancers wait for the client to follow up and then give them an update on the issue? Many… Please make sure you don’t do this if it applies to you.

You need to think like a business. If your hosting provider had issues with their servers and you send a support request to [whoever your hosting provider is] and you don’t hear back from their customer care department you will be upset.

But if they sent you an email to say, “Hey Kyle, we are experiencing some issues with our server and we are working on it. We will update you as soon as possible once we’ve fixed the problem.” You will then know what’s happening and know that they are working on it.

Which option would you prefer? It’s a similar principle with how you update your client.

Another reason why you need to update your client regularly is to minimise revisions as the project nears completion.

Know what they are doing.

It’s important that you are good at what you do.

Would you listen to a 35-year-old business coach who has just finished his MBA with distinction or would you listen to a 35-year-old business coach who dropped out of high school with no certifications but has experience in running 4 highly successful businesses and has sold 1 for a profit?

When you offer a solution, it needs to be backed up by your expertise (or solid data at least). So for example, I mention to some clients that I can help generate traffic to their site through SEO and AdWords because I have experience in managing very big accounts and I have achieved success with it.

The same goes for creating a website. You need to know how a visitor thinks at every single step of the sales funnel and what questions they will ask themselves.

You need to know WHY using a specific colour theme is important and what it implies to the visitor. You need to know what aspects of a websites work well in converting and why, etc.

All these things are important. It's **EASY** to make a website, but where you should come in is saying, "I can make a website and offer a business solution that **WORKS** and achieves the goals you'd like to achieve."

Another reason why it is important to know what you are doing is a scenario when a client says, "We'd like to remove the video from our sales page."

You should be in a position to know that by adding a well-designed, brief video to a sales page could potentially increase conversions by up to 80% (see source). So you should then present the client with some stats and data and just mention that you don't think it's a wise decision to make and you'd advise keeping it there or improving it.

It doesn't mean that the client will always listen to your advice, but it's your responsibility as a freelancer to voice your viewpoint on these situations that come up.

You should not get hired to 'do work', you should get hired to solve a problem.

Not only that, but you should be proactive in recommending solutions which can potentially yield high returns.

You need to have a 'split-personality'.

One of the most important things to do when creating a project is to **not** think like a freelancer.

You need to have a 'split-personality' when looking at every project that you create.

What do I mean by this strange statement?

You need to think like a site visitor and as the business owner.

This is important. As the site visitor, you need to be thinking: What is going through my mind at every stage I am on this website? Am I getting the answers I came here for? Is the site fulfilling what I wanted?

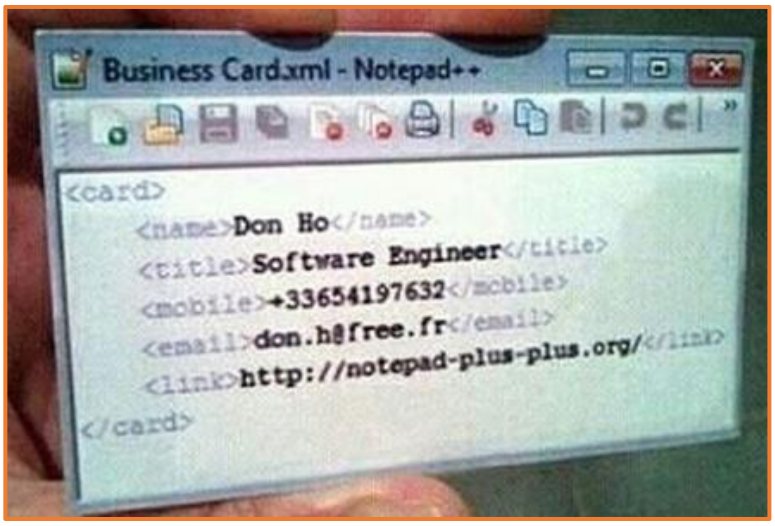
As the 'owner' of the website, you need to be thinking: Is my website answering the questions and doubts my site visitor is thinking at this moment? What can I do to answer their questions? What can I do on my website to convince them to perform the action I'd like them to take?

So for example, if I am a site visitor on the checkout page, I will be thinking: Is this website secure enough for me to put my credit card details in? What will happen after I make the payment? What will happen if I'd like a refund? How do I know if I selected the right product option? etc.

If you don't think like that when you create a website, you aren't thinking correctly. I've worked on checkout pages for large ecommerce sites and just by tweaking a few words and mentioning '14-day money back guarantee' and adding a live chat software with an automatic popup trigger after 60 seconds, increased conversions by a large number and significantly increased sales.

If you think like a freelancer and not like the site visitor and business owner, I'd encourage you to start thinking differently.

Added tip: think creatively and create a business card something like this one below:



```
<card>
    <name>Don Ho</name>
    <title>Software Engineer</title>
    <mobile>+33654197632</mobile>
    <email>don.h@free.fr</email>
    <link>http://notepad-plus-plus.org/</link>
</card>
```

Common Mistakes Freelancers Make:

No one is immune to making mistakes. Here are a few things which freelancers should look at improving (I also speak to myself on some of these points):

Wrong mind-set.

Once you become a freelancer, it's important to shift your mind-set from being an employee to being a freelancer.

Wrong mind-set: I build dynamic websites for my clients.
Correct mind-set: I run a business that provides solutions for my clients that gets positive results.

Lack of knowledge.

Don't be deceived with 'freelancing'. It falls into the same divisions of a business:

Legal – Contracts
Finance – Invoicing, receiving payments, how to deal with clients that don't pay, pricing
Admin – Emails, phone calls, running the business
Marketing – How to get clients, pitching yourself, selling

Just because you are a good programmer or designer, it doesn't mean you should become a freelancer – especially if you are not aware that you will be dabbling in the above business divisions as well.

In order to stand out from every other freelance web developer, you need to understand how psychology and digital marketing works. I say this because you won't just offer a quote for a website then, you will rather be offering a proposal on 3 different packages like:

1 – website
2 – website + SEO + designs
3 – website + SEO + designs + copywriting + setting up AdWords + email marketing + social media marketing

So in a way, you are essentially offering digital marketing solutions + development solutions + conversion solutions.

More info on learning later in this eBook.

Poor use of time management.

It's important to spend time on the important things and to find ways in being more efficient.

Don't spend so much time creating proposals or going to meetings or doing things which aren't worth your time.

You are trading your time for an income so use it wisely.

Create templates and automation systems to help with the common tasks which come up and also make sure to make use of tools and software which can help you work smarter.

Also remember to balance out your work-life and spend time with your family or reading or the important things in life.

Going from $5k per month to $10k per month may sound awesome, but if that means working 30% longer, spending 30% less time with your family and having 30% more stress – is it *really* worth it?

You can always make more money; you just can't make more time so use it wisely.

Bad workstation setups.

It's important to have no distractions where you work. I generally work in multiple places throughout the day, but some prefer to work in their home office or by the dinner table.

Wherever you work, make sure your posture is straight otherwise it will catch up to you – trust me!

It's important to have an ergonomic chair or incline wedge if you lie down and work.

A workstation cannot be called a workstation if you don't have a customised coffee mug as well 8)

Not charging correctly (I don't mean not charging your iPhone properly).

I touched on this in more detail previously in this eBook, but here are a few more notes:

You should not be the majority. You should aim to be the go-to person for getting results. Businesses hire you because they have a need. You need to solve this need, and to solve it properly you need to charge a premium price.

If I were selling you a bottle of wine and I told you that I have 2 bottles, one priced at $5 and the other at $55 you would probably think something is wrong with the $5 bottle wouldn't you?

The same applies for seeing a website theme at $10 or hiring a full-stack developer at $10 an hour.

To read more about pricing, bundling and the reasons behind it, click here, here and here.

Accepting every client.

I list this point in the 'mistakes' section because it's important and often overlooked.

If you are completely new at freelancing, I would actually recommend that you take on almost any client that you can get. This will help you deal with many other requests that come your way and it helps you get an idea of how to deal with clients in the future.

If you are too busy and don't have the resources and the time at that moment, it's ok to turn down clients. I've had to turn down a client wanting to pay me a decent amount for an urgent website that was around 2 days' work, but I couldn't do it at the time. I knew if I took on the project I would've had to find a sub-contractor I haven't worked with or run the risk of not delivering and wasting my time (and theirs) all together.

Another important thing to consider is if a business comes to you and would like you to improve on their website or to increase their sales. It's ok to say, "I actually don't have a solution for you. I can't fault anything you're doing wrong."

David Ogilvy was famous for turning down the New York Times because he didn't think he could produce better advertising campaigns than they were currently running.

This comes from the most renowned advertising agency owner in history.

Be true to yourself. If a business would like you to create a website for them and it's something which goes against your morals and what you stand for, then politely decline their offer.

If you have a client who is not willing to pay what the project is worth, don't just drop your rates by half to gain the business – I can almost guarantee it won't be a good move.

If you have a difficult client (they do come around) how do/would you handle them?

Would you continue working with them for the remainder of the project or will you part ways before the project ends?

Both has its pros and cons of course, and I am not going to influence your decision in this, but just really think this one through.

I'd recommend reading what Seth Godin did in a similar scenario here.

TOOLS TO USE AS A FREELANCER

Tools are essential for productivity and effectiveness. Here are a few tools and resources you should take note of:

Gmail: This email service is underrated and it's amazing how many people don't make use of the features and filtering settings. Take a look at these tips for ways you can master it.
Google Calendar: A cloud based calendar that syncs with your Gmail account and all your devices. Easy to understand and highly recommended.
Skype: A great video chat software to make calls without in-person meetings.
Hootsuite: A social media posting platform that automates your social posts at certain times on predetermined dates. I use Hootsuite on my own businesses and client businesses (it's free).
Google Drive and **Dropbox:** These are cloud based storage solutions which are a necessity to backup information or to share large files. You can also use this storage locally on your PC and sync it to the cloud.
Evernote: A popular organisational cloud based tool with features like notes, article capturing, chats, intuitive designs, task lists and more. It syncs with your phone and PC. I use it all the time and it's an awesome tool.
Grammarly: This free tool corrects your grammar and spelling – basically it just makes you more professional in your communication.
Planscope, **Harvest** and **BallPark**: Advanced project management applications/software that allows you to track your time, invoicing and to make sure you are within budget (and more).
Mint: Manage your finances easily with this app.
Freshbooks and **QuickBooks**: More advanced business finance management software.
Trello and **Remember The Milk:** The best to-do-list tools to help manage your day and projects.
Egg.timer, **RescueTime** and **Focus Booster:** If you easily get distracted (like me) then these are brilliant tools to help you stay focused on your current project without going to distracting sites like social media and news sites. You set a defined time period to work on a task and it will also remind you when to take a break so you don't 'burn-out'.

When you work, work in highly focused sessions. A technique I've started using recently is the **Pomodoro Technique**. Read about it here and use this free app.

One of the best productivity tools to use is ***yourself***. You need to keep asking yourself if what you are doing is bringing you closer to finishing your project and/or your goals.

If not; stop it, delegate it or outsource it – simple.

WHERE TO FIND WORK AS A FREELANCER

Finding work can sometimes be quite challenging. Here are a few recommendations:

The best type of work is recurring clients.

The reason for this is that they are used to how you work and you both know what to expect from each other. They've seen that you can deliver, and you know their business and that they pay on time – it's a win for both sides.

If and when possible, try and get clients on a retainer (a fixed monthly agreement that they pay you in exchange for priority work). I personally have a 2k per month retainer for 2.5 hours every week and it's a great way to get guaranteed income each month – just make sure you deliver on what you agreed upon.

I will be honest with you… this is a difficult stage in the freelancing journey. You have the same concern as every business does which is HOW DO I GET PAYING CLIENTS?

It would be nice if there were paying customers immediately but it doesn't work like that.

This part requires ACTION and it may even require you to step out of your comfort zone, but if you really want your "WHY" you WILL do it.

If you don't have any client database, don't worry. Here are some more platforms to find work:

Upwork - link
LinkedIn - link
Ziprecruiter - link
Indeed - link
Craigslist - link
Freelancer.com - link
Letsworkshop - link
EnvatoStudio - link
Fiverr - link

Dice - link
Smashing Jobs - link
The Muse - link
Guru - link
Flexjobs - link
99Designs - link
Gun.io - link
Toptal - link
Authentic Jobs - link
Idealist.org - link
Coroflot - link
Mashable Job Board - link
Behance Job List - link
PeoplePerHour - link
RentaCoder – link
Truelancer - link
Stackoverflow - link
Hired - link
Hirable – link
Crew – link
CSS-Tricks - link

An often-neglected action step is to speak with friends, family, neighbours, co-workers, acquaintances, friends of friends or any other group and just mention that you can do work for them and if they know of someone who needs an online business solution you can do the job.

If you are **really desperate**, just walk into a web development agency and ask them if they would give you 2 of their biggest clients.

Sorry, bad joke again – don't do that 8)

Another thing you should do is go to business/entrepreneurship meet-ups (an organised social event) and connect with business owners. Notice I didn't say a developer meetup? That's because you are a consultant to businesses now.

Who pays your income; businesses or developers? Of course the businesses, so make sure you are where they are and just engage and communicate with them. You will eventually get some clients by doing this.

I'm not saying you shouldn't go to developer meetups… what I am saying is that your main focus should be on your target market. Your main target market is business owners (or the niche you choose) and not developers.

If you do a good job for a client, they will often hire you for more work in the future or they will refer you to other friends/businesses so make sure you treat every client as if it is your last one.

Design websites for niche industries.

I've created my own website themes for a few niches and I sell it for $30 (only the theme) or I will create a website for the client around $500-$800+.

Remember, this is for small businesses and it would take less than one day to do the website because you are basically just changing the content, images, hosting and SEO – all the *hard* coding work has already been done.

Step by step instructions:

1. Design a responsive, well designed website theme for a niche of your choice. Make sure this is designed VERY WELL.
2. Go through your local yellow pages, business listing sites, Google and any other way of gathering emails for a niche industry, then compile an email list for that niche that you will be sending an email to.
3. After you have created the website theme by coding it yourself (from scratch), save the files as a JPEG version.
4. Send the JPEG version in an email to your email list that you have compiled using the above methods.

Why JPEG? Because you don't want to send the 'live' website to your prospect in case they steal your code if they are savvy enough.

Ultimate Tip: Rather than coding the full website from scratch and to save time, just create a JPEG format of the home page and then send the design to your email list. If you receive order enquiries AFTER you have sent the JPEG, you know that the theme works and your design looks good.

Email template example (on 2 pages)

Hey there,

My name is Kyle, I am a freelance web developer and I have created a website theme for your business.

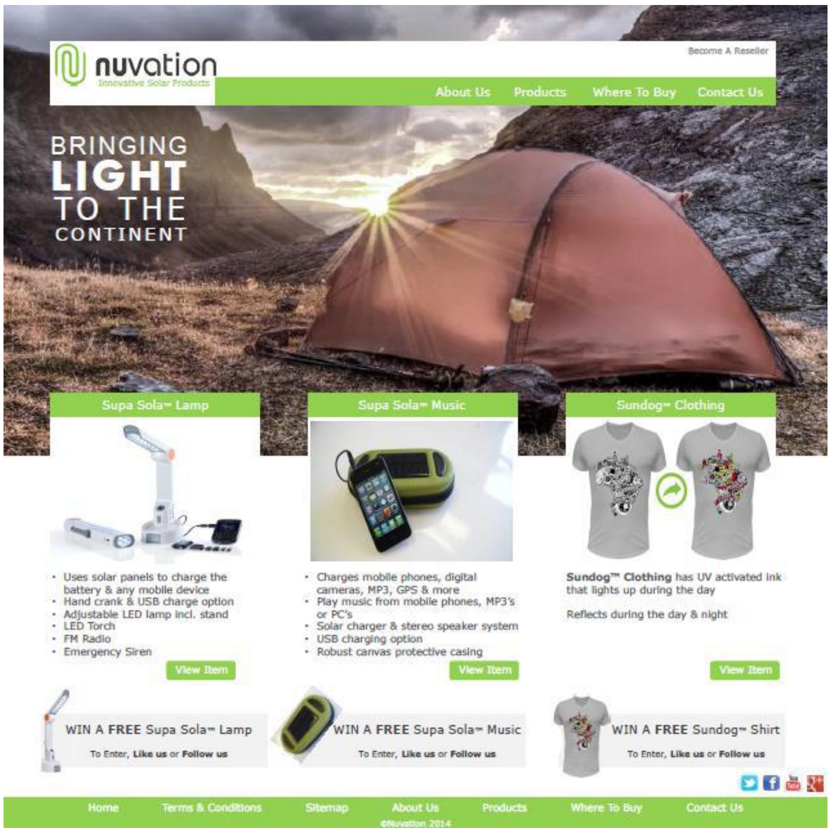
I've noticed that your website is not mobile responsive and it can definitely be improved so I created a new website for your business.

OR (choose option above or below based on which option is correct)

I've noticed that you don't have any website so I created a new website for your business.

Please take a look at the image below for reference of the home page. I've also attached it to this email.

(this is just a home page example that I designed for a niche about 2 years back)



Supa Sola™ Lamp

- Uses solar panels to charge the battery & any mobile device
- Hand crank & USB charge option
- Adjustable LED lamp incl. stand
- LED Torch
- FM Radio
- Emergency Siren

View Item

Supa Sola™ Music

- Charges mobile phones, digital cameras, MP3, GPS & more
- Play music from mobile phones, MP3's or PC's
- Solar charger & stereo speaker system
- USB charging option
- Robust canvas protective casing

View Item

Sundog™ Clothing

Sundog™ Clothing has UV activated ink that lights up during the day

Reflects during the day & night

View Item

WIN A **FREE** Supa Sola™ Lamp

To Enter, **Like us** or **Follow us**

WIN A **FREE** Supa Sola™ Music

To Enter, **Like us** or **Follow us**

WIN A **FREE** Sundog™ Shirt

To Enter, **Like us** or **Follow us**

Home Terms & Conditions Sitemap About Us Products Where To Buy Contact Us

©Nuvation 2014

You can buy this custom theme for only $30 (once off).

If you'd like me to create the website for you and add all the relevant details and images within 2-3 working days, I can do the changes for you at an additional fee and I can guarantee that you will be happy with the results.

I'm sure you have many questions and I look forward to hearing from you.

Thank you in advance for your reply.

Regards,

Kyle

(yourwebsiteaddress)

---

*The image/s NEED to be inserted in the body of the email AND attached. This is very important otherwise most prospects will think it is spam. If you insert/embed the images in your email, they will receive it on their side without having to download the image (most of the time) and if they would like to see the theme in more detail they will then open the attachment from there.

Remember that most people don’t know ‘programming talk’ so it’s important to communicate on the knowledge level of the recipient and to speak very simply even if you might not be technically correct.

TIP: Make sure you send from a REPUTABLE email address. Would you reply to someone who has an email like kyle-b-p-27@gmailscousinsaunty.com or kyle@studywebdevelopment.com?

Of course you would choose the latter as you can see it is legit and they can go directly to the website and see if it’s not one of those spammy websites.

How to get noticed in a crowded tech space

It helps if you have a well-designed website, a social profile on Facebook, Twitter and LinkedIn and a portfolio before you do anything here.

Once you’ve sorted out those basics, it’s important to get out there and engage with the community on StackOverflow, Reddit, Quora, Social Media and Related Blogs in your industry.

Once you’ve created a bit of authority, then reach out to some bloggers/podcasters in your industry and ask them if you could write something valuable to their audience or be interviewed. This will help you stand out a bit more in this crowded space.

The most important thing you can do is to write a blog that adds value to others and you will eventually get noticed. Please make sure that you are consistent in your blog posts. Try and post every week, if you can’t do that then at least aim for one blog post every 2 weeks.

“You will get all you want in life if you help enough other people get what they want.”

Zig Ziglar

‘What if I get no clients?’

1 – Design more/different themes in JPG, change/improve your designs until you get a positive response.

2 – Send out more emails.

3 – If this still does not work then change your niche and design and your approach of emailing your prospects. Try phoning them if needed.

4 – Still no luck? Repeat the process…

Added TIP: Have you ever heard some people say, “It’s all in the follow up”?

It's true. In this case, if you do not get any reply after 3 working days, send a follow up email to the company saying (with the forwarded message below):

Hey there,

I'd just like to find out if you have received my email sent to you below?

Thanks, looking forward to hearing from you.

Regards,

Kyle

You may be thinking, why not just list my website theme on WordPress, Creative Market, Themeforest or Colorlib?

You can do this *eventually*, but this strategy is a good place to start and make some cash from it. It also builds your portfolio and you gain more experience.

Slightly off topic from the themes, but on topic on the niche idea is someone who has done well in a niche. His name Chris Coyier, from CSS-Tricks.com – he has positioned himself as an expert in CSS and in fact, he blogs and talks about a lot of things besides CSS. He's also one of the creators of CodePen.io

If you really have skills, and you are confident, then aim to create a course for Udemy – they have a huge platform and you could potentially make some decent sales as well. This isn't easy and it may not be for you, but at least you know of the possibilities.

HOW TO SPEND YOUR MONEY AS A FREELANCER

One thing I personally didn't enjoy about school is the fact that the education system does not prepare you for 'real-life finance management'. That's a broad statement, I know, and maybe it's not true for you, but this is a very important topic to discuss.

I will mention 3 examples of people I know/knew very well:

1. Someone who retired with well over $2,000,000 and spent it all within 7 years and now literally has to borrow money for food every month.
2. Someone who earns a big salary (well over $20k per month) and is so deep in monthly debt he can't afford to pay the essential expenses.
3. Someone who literally has a blank pay-check every month and lives very humbly and simply.

What is one common denominator in these 3 examples?

How they handle their finances…

I'm no financial expert by any means, but I will give you the best financial advice out there (for free):

Make sure you manage your finances, don't let it manage you.

On a practical note, this is what you need to keep in mind for when you get paid and when you start earning a good income:

Pay the bills

Very simple, but make sure you know the exact 'break-even point' on what you need to cover all your necessary expenses like rent/mortgage, food, medical, car, etc. and also the additional services like hosting, email marketing, productivity apps, internet, etc.

Added tip I learned from a wise man: Take your annual expenses like domain renewals, banking fees, accounting fees, vehicle fees, birthday gifts, home repairs, etc. and divide it by 12 months and add it to your monthly expenses so you are well prepared for the annual fees.

Save for repairs and upgrades

Make sure you save for laptop/PC repairs, new upgrades in software, new laptops/PC's, new hard drives, new tools, new apps, etc.

Education

Why's this here?

Because you need to keep learning. It doesn't matter whether you are a grandmaster at programming, you need to keep learning. Save for online/offline education and invest in yourself.

Networking

Pay for conferences and social events to meet businesses and make connections. This is important to do and often neglected when things are going well.

Treat yourself

Take a break and go for a weekend away or aim for frequent mini-holidays throughout the year. I struggle with this because I am very goal and project orientated, but I am working on it ☺

So let me be hypocritical here and say you need to save for these times away to travel and take your family with you. Life should be full of memories to cherish, not full of memories behind a laptop screen.

Save for a rainy day

I mentioned it before, but this is important. What would happen if you had to go into hospital for 3 months? What would happen if you lost your 2 big monthly clients? Try save as much as you can. By this I mean cash in the bank… don't think that because you are earning a lot you can upgrade your lifestyle – refer to my 3 examples above to illustrate this point.

JOB PROPOSALS

When you get job proposals as a freelancer and it may be out of your skillset, just say, "Yes, I can do the job."

If you only know how to code HTML, CSS, JavaScript, Bootstrap, PHP and Rails, but your client asks you to add a SQL database.

What would you say?

This is what I would (generally) say:

“Yes, sure. I will get back to you as soon as possible on the fee for this extra add on.”

Then based on the client’s expectations, I would speak with a friend who is a specialist in the areas I am weak in and give them the briefing, ask them how much they would charge to create the SQL database for me and then I will get back to the client with the fee.

This can go 2 ways:

1. If I get quoted say $500 by a friend, I would charge the client $700. This may seem like a lot, but remember, I will be the ‘middle-man’ (which requires a lot of work) and you need to account for changes and edits. So I would keep the client happy and make a small profit on this by outsourcing it.
2. I would refer the client to a friend directly and say to the client, “I’d love to help out, but in all honesty, I don’t have the advanced skillset for this yet. I highly recommend my friend, Bob, he is a SQL master. He said he will do the SQL database for $500 and you can contact him directly here: [Bob’s email address] or I will liaise directly with him for $100 of my time.”

If I go option 2, I would either just help my friend out, OR I would just say, “Hey man, I have referred a client your way. Can you give me 10% of on-going payments with them in exchange for this referral?”

Bottom line is… don’t turn clients away unless you are REALLY unable to help them. Rather improvise and learn from there. If you can’t deliver, apologise and refund them.

Of course you should use discretion on this by assessing everything.

Some may say, “but that’s wrong.” But others may say, “hmmm… not a bad idea.” I prefer the latter.

This principle is true for design work or coding. The best part is that if you do a good job, there is a good chance that the client will offer you more work in the future. Recurring clients are the best clients because they know how you work and they are familiar with the processes and you know what they want.

Another pro tip is if you refer a client to a friend. Because you’ve agreed on 10% **on-going** commission for each payment, you make some cash even though you don’t do any work.

What I am trying to get at here is that it’s important to **THINK like a business person** and **not like a programmer**. I don’t mean that in an offensive way at all, but

I am implying that you should think more creatively about every situation and try to problem solve in a different 'non-programmer' way.

No one gets this right every time, and I continue to struggle with things I could've handled better or explained better or if I just thought it through better. This is a continual process and we need to all train our minds to THINK in more creative and solution orientated ways.

Whether it's to land your dream job at 'Company X', or to become a freelancer, these tips will apply to you:

ALWAYS IMPROVE

Don't just accept the task… **think about ways that you can improve the task**. In other words, don't just react and accept it, show that you are concerned about the success of what you've been asked to do.

This will show your boss/client that you are concerned about the wellbeing of the company. Stand out from the crowd and prove what an asset you can be in these situations.

If you feel "but how can I add more value to this project?" or "but I only know how to code" etc. To that I say: STOP MAKING EXCUSES.

If your boss/client asked you to create a website in the retail sector for example, and you have the following (simple) brief:

"Create an ecommerce site with 20 pages of content listing around 50 products and it must look like [example site that looks average]."

What you should do before even starting is this:

Go through the design, compare it to other successful websites in that industry by researching on Google. Test their site by clicking on links and seeing what stands out to you and what you like.

Is their site usability (UX and UI) good? Are they missing something that you can take advantage of on your site that you need to create?

Present your boss/client a simple one-page proposal on what you recommend which would make the site even better than what you've been asked to do.

If it works out, you will be seen as more of an asset in the company and if you are a freelancer, you will be seen as irreplaceable – either way it's a win-win.

If your boss/client says, "no" after seeing your proposal, don't feel discouraged. I'd recommend that you keep doing this on every occasion. Although your boss says, "no" it still shows that you are different to the majority of the 'robots' out there.

ALWAYS KEEP LEARNING

Possibly one of the most important and less mentioned points is that you **always need to keep learning web development and improving your knowledge**.

I'm sure you are familiar with the huge mobile phone brands like Nokia and Blackberry? I won't get into stats, but basically they were the global monopoly of the mobile phone market a few years back and now their market share is nowhere near what it used to be.

LESSON: Don't get comfortable. ALWAYS KEEP LEARNING. Even if it's 30 minutes each day, but if you would like to keep your competitive edge and keep improving yourself this is so important.

These next tips cross over into marketing which may not be "web development" related, however it is related to the success of a website. If you don't use all 3 of these tools, you should ALWAYS use at least one of them:

Zopim

Zopim is a live chat software that is **extremely** important for most websites that sell products/services.

I've personally implemented this on a few of my clients' sites and it's been extremely successful. The annual investment has the potential to pay for itself within 1 week.

I'd encourage you to convince your boss/client to try the 14-day risk free trial.

SumoMe

SumoMe is an awesome set of tools to get more results. This ranges from dynamic email sign ups forms/drop-downs/pop-ups, social shares, site analytics and more. The great part is that even the FREE option is perfectly fine.

I have SumoMe on StudyWebDevelopment.com and it's really been beneficial from day 1.

Whether you create blogs, ecommerce sites or article sites this is a MUST on every website that you create.

Mailerlite

Mailerlite is one of the best and most popular email marketing platforms available.

Collecting emails is one of the most underrated 'tools' out there.

I can discuss the importance of email marketing in a separate eBook, but in short: your boss/client NEEDS to grow their email database.

I use Mailerlite for pretty much every website. There's no excuse to use them – it's free for under 1,000 subscribers. It's one of the best investments I could make.

Ask your client how they are measuring their analytics? Are they using tools like Google Analytics, Kissmetrics, Crazy Egg, SumoMe or Clicky? If not – why not?

It's like when you buy a car. The car salesman would be completely incompetent if he just sold you a car and did not recommend any other add-ons or warranties.

Obviously you aren't selling this, but I'm saying you need to be proactive in your advice and recommendations to your clients. You will stand out. Make sure you recommend a mobile site as well… provide some stats and show the data backing up the importance of it.

By recommending these tools, you will stand out from the crowd and it also gives you another advantage over other 'robot web developer' or 'robot freelancer' out there.

If you have created websites for clients previously, contact them and suggest these tools so you can implement it and get paid for this.

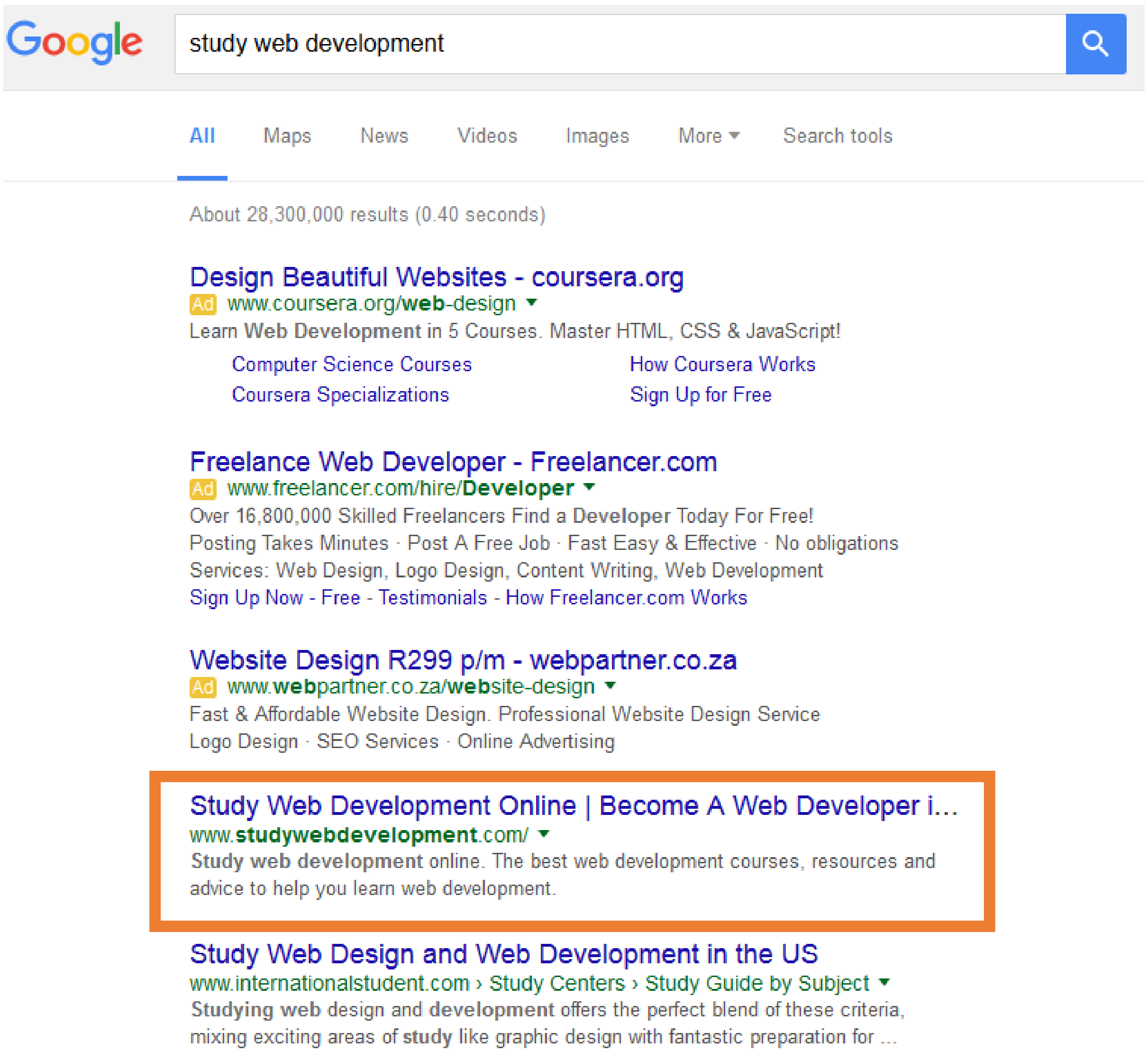
THE BASICS OF SEO

It's so important to create websites that are SEO friendly.

What is SEO? SEO stands for Search Engine Optimisation.

In "simple English" it is creating or editing a website that Google can 'read' easily and your ultimate goal is to rank #1 for your search term.

Example:



About 28,300,000 results (0.40 seconds)

Design Beautiful Websites - coursera.org
Ad www.coursera.org/web-design
Learn Web Development in 5 Courses. Master HTML, CSS & JavaScript!
Computer Science Courses — How Coursera Works
Coursera Specializations — Sign Up for Free

Freelance Web Developer - Freelancer.com
Ad www.freelancer.com/hire/Developer
Over 16,800,000 Skilled Freelancers Find a Developer Today For Free!
Posting Takes Minutes · Post A Free Job · Fast Easy & Effective · No obligations
Services: Web Design, Logo Design, Content Writing, Web Development
Sign Up Now - Free - Testimonials - How Freelancer.com Works

Website Design R299 p/m - webpartner.co.za
Ad www.webpartner.co.za/website-design
Fast & Affordable Website Design. Professional Website Design Service
Logo Design · SEO Services · Online Advertising

Study Web Development Online | Become A Web Developer i...
www.studywebdevelopment.com/
Study web development online. The best web development courses, resources and advice to help you learn web development.

Study Web Design and Web Development in the US
www.internationalstudent.com › Study Centers › Study Guide by Subject
Studying web design and development offers the perfect blend of these criteria, mixing exciting areas of study like graphic design with fantastic preparation for ...

“study web development” is one of the terms I aimed to rank for. I created my site in a way that I am now able to rank #1 for that term and other strategic terms.

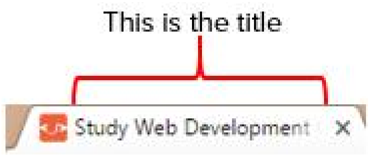
I’d like to share some simple, practical SEO tips that has helped me rank my own websites and my clients’ websites. I can actually make a separate eBook on this topic so I will just cover the basics and get to the point ☺

The good thing about knowing SEO is that you are able to charge for this service. Depending on how large the website is and the work required, I would charge anywhere between $300 - $800 for this basic SEO service (if it’s an ecommerce or a very large site, I’d charge much more) – **you** can charge whatever you feel this service is worth and whatever you feel the client would be willing to pay for it.

Here's a list of simple changes you need to do to make a website more SEO friendly:

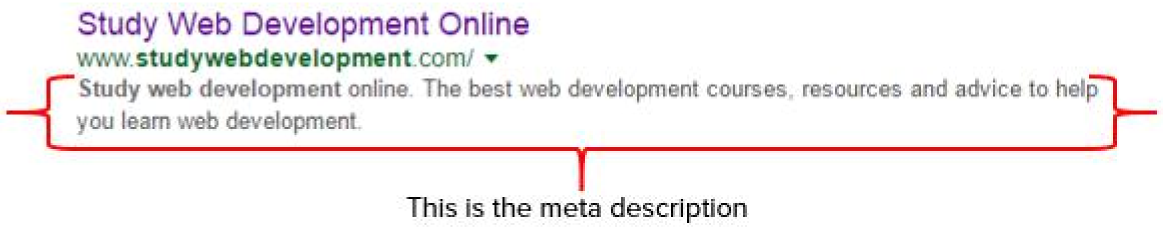
PS: View an updated article about SEO here.

Title



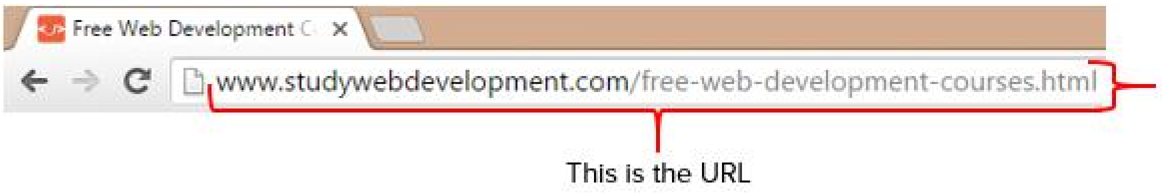
The title must be under 70 characters in length. This needs to change on EVERY page. Make it relevant to the search term you want to rank for.

Meta Description


Study Web Development Online
www.studywebdevelopment.com/
Study web development online. The best web development courses, resources and advice to help you learn web development.
This is the meta description

This can only be seen on the SERPs (Search Engine Results Pages) – as in the image above. It must be under 156 characters in length. This needs to change on EVERY page. Make it relevant to the search term you want to rank for.

URLs


Free Web Development C
www.studywebdevelopment.com/free-web-development-courses.html
This is the URL

This is a 'SEO-friendly' URL (website link). Keep the URL as short as possible, but use the search term you want to rank for. Again, the URL needs to change on EVERY page.

Images

I can't tell you how many developers take the 'easy-shortcut' on this one. You need to give EVERY SINGLE IMAGE an ALT Tag. Let me rephrase that better: You need to give EVERY SINGLE IMAGE a 'SEO-friendly' ALT Tag.

ALT Tags need to be structured like this: alt="web-development-and-beyond-ebook" (notice the hyphens – this is important as it indicates a 'space' in between the words).

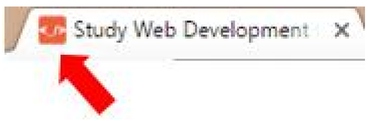
Not only that, but when you SAVE the image in your folders, save it like you would save it as an ALT Tag. So the name of the image and the ALT Tag must basically be the same.

H Tags

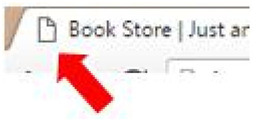
You cannot use H Tags for paragraphs… It might not apply to you, but I've seen this happen very often. H Tags show Google, "Hey Google, this 'header-section' of the site is important and the content on this page is related to my H Tag. I am giving it a H1/H2 tag so take note of it." Make sure you don't overuse H Tags – use them sparingly and use your search term you want to rank for or at least a close synonym where possible.

Favicon

A small, but underrated and neglected image for a website.



Make sure you add a favicon to the site. If you're like me (I'm a bit O-C-D) and you really don't like seeing websites like this:



Then we will get along very well ☺

But on a serious note, it doesn't look professional and it ruins the whole site (maybe not, but just add it).

XML Sitemap

A sitemap is an XML document on your website's server that basically lists each page on your website. It tells search engines when new pages have been added and how often to check back for changes on specific pages – all this helps Google read and understand your website better.

To create a sitemap is quite easy. Go to: www.XML-sitemaps.com and follow the straight forward steps.

Once you've created the XML document, submit it to Google Search Console (webmaster tools).

The next steps would be to make sure the website loads quickly. Make sure it's nice and 'clean' – so it must be user friendly. Make sure the text and copy on the website is relevant. Make sure you link to authority sites like Wikipedia for example as well as strategic internal linking.

Generally, just doing the above would be a good boost for a website. Of course there are MANY other factors and variables to consider (such as backlinks, competitor analysis, site architecture, user experience, no-follows, content, bounce rate etc.) This is in no way a full guide on advanced SEO techniques. This is enough to get you started and you at least know enough to get paid for it.

SOME GENERAL TIPS:

Have you backed up the website?

Make sure to make constant backups of the site before making new changes. This will help if your clients ask you to roll back a new feature to how it was 3 weeks ago and it also helps in case you make an error on the code and you need to get the old site back again.

Just use GitHub, Dropbox, Google Drive or a hard drive for this.

Have you created a 404 page?

A big thing that many developers neglect is the 404 error page.

A 404 error (page not found) is basically a page that pops up due to a page not being valid on the website.

Refer to this 404 page on SWD and 10QuestionPoll as an example.

If you want some 404 page ideas, click here.

Learn how to make a 404-page step by step in under 6 minutes, click here.

Use code snippets where possible

Make use of code snippets like Bootsnipp and CodePen.

A code snippet is basically a selection of code that you can copy for your own website. So if someone creates a modal image carousel and is willing to give you the code, you just copy it and there you have your modal image carousel. It saves a lot of time and if you search hard enough you can find some quality snippets.

You HAVE TO make a site that is mobile responsive

I have seen so many Web Developer “PROs” create an incredible desktop website, but the moment it is viewed on a mobile device, the site doesn’t look good at all.

According to Statista, around 38% of website traffic in 2016 will be on mobile phones (this excludes tablets).

This stat alone shows the importance of thinking what others call ‘mobile-first’. Google has also openly expressed that it will penalise websites that are not fully mobile responsive.

I personally love working with Bootstrap for this very reason as it is ‘auto-responsive’.

Remember to test your work

Have you tested mobile responsiveness? Have you tested it on multiple browsers? A sad truth is that just because something works on Chrome, it won’t always work on Explorer or Safari so make sure you test everything. Don’t send a client the final check before you do some tests. It looks bad if you’ve neglected something very basic.

Don't forget to workout

You need to remember to stay active and do some stretches and general workouts. I'll be honest with you… I prefer to work out my mind than my body, but it's important to keep fit and not be bound to a chair all day - it's really not good for you (and me).

Check out these sites for inspiration and exercises:

Nerd Fitness

12 Minute Athlete

Recommended Blogs:

Learntocodewith.me – link

Study Web Development – link – Don't know whose blog this is 8)

Upwork – link

Smashing Magazine – link

Speckyboy – link

David Walsh – link

Treehouse – link

A list apart – link

Brad Hussey – link

CSS-Tricks – link

Double Your Freelancing – link

CFH – link

Viperchill – link

SmartPassiveIncome – link

Recommended Podcasts:

Learntocodewith.me

GoTreehouse – not really a podcast, but it's good.

ViperChill

TheWebAhead

SmartPassiveIncome

DoubleYourFreelancing

ShopTalkShow

EntrepreneurOnFire

TheDigitalEntrepreneur

CodePen

Recommended Resources:

Bluehost – My top hosting recommendation

Typing.io – Improve your typing speed and accuracy (compulsory for everyone)

Caniuse – Helpful answers to many developer queries

Larger.io – See which programming languages are used on any site

Call to Idea – Cool site to get ideas on designs

Subtle Patterns – Background images

Scroll Watch – Add infinite scrolling

Responsinator – See how responsive your site is on all devices

Pixabay – Free images

TinyPNG – If you don't compress your images, we will not be good friends

Blokkfont – Wireframing

Pixlr – Free photo editor if you don't have PhotoShop

Trianglify – Awesome triangle backgrounds

CodePen.io – Cool place to list your work and get code snippets

Divi – A must-have if you create Wordpress websites

Workroll – Get high-paying customer leads for web development, marketing & more

For more resources, click here

Follow these Twitter profiles:

SWD (me)

Laurence Bradford

Smashing Magazine

Creative Bloq

ViperChill

Brennan Dunn

UpWork

Follow these Facebook Pages:

SWD (me)

Learntocodewith.me

Smashing Magazine

Creative Bloq

ViperChill

UpWork

I’d like to end off on these questions:

Why is it that someone born in poverty, with no formal education, no ‘connections’ and everything in opposition to him/her end up being successful in whatever job he/she chooses or company he/she creates?

Why is it that someone born in wealth, with a formal education, with ‘connections’ and everything in his/her favour ends up being unsuccessful in whatever job he/she chooses or company he/she creates?

I have my own personal reasons for an answer to this, but to relate it to this eBook, I’d like to come to these final takeaways and conclusion:

- Tech careers are booming and *eventually* it will become a lot more challenging to find tech related careers due to more saturation in the market and each programming language.
- You need to find a way to stand out from the crowd in your own unique way. Whether you are an employee, or you work for yourself, you need to think creatively and be an asset in whatever you do.
- Be the best that you can be as a person. Whether this is improving knowledge, social skills, communication skills, productivity, attitude or mind-set.
- You need to always-be-learning, APPLYING and mastering your craft.
- We cannot change our past, but we can change our future based on what we do in the present.
- Don’t neglect what is truly important in life. All that is tangible will perish. Live your life with that in mind.

(ending remarks on next page)

I truly hope that you gained some knowledge and tips from this eBook. Please APPLY what you've read and I'd really appreciate your feedback, whether it's good or bad.

If you learned something from this eBook I'd love to hear about it! I read every single reply. It would mean a lot if you shared and recommended it to others – thank you.

Thank you for reading this eBook. Keep moving forward.

Kyle

Connect with myself and the 70,000+ SWD community:

www.studywebdevelopment.com

Thank you once again and I will see you online!

PS – if you are a web designer, sign up to WebDesignerWeekly.com