

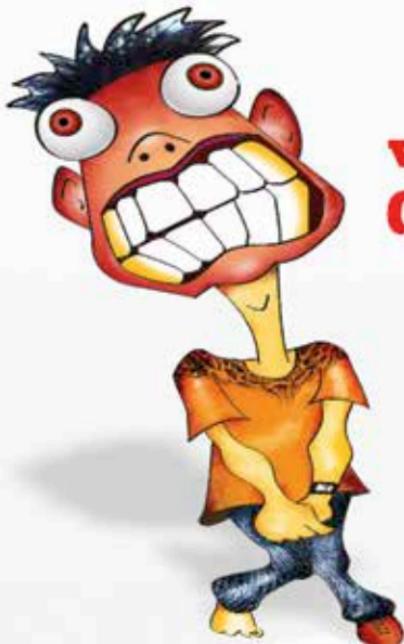
To

# BUILD YOUR OWN WEBSITE



- The Purpose
- Components
- The elements of an ideal website
- Behind the page
- Picking a CMS
- Picking a web host
- The tools of the trade
- One-click setup websites
- DIY - Assisted Setup
- Building a website the hard way
- Secure your website





[www.  
digit.in/forum](http://www.digit.in/forum)

Join the forum to  
express your views  
and resolve your  
differences in a more  
civilised way.

**digit.in  
FORUM**

Post your queries  
and get instant  
answers to all  
your technology  
related questions



One of the most active online technology forums  
not only in India but world-wide

**JOIN  
NOW**

 **digit.in**



# BUILD YOUR OWN WEBSITE

---

powered by

---

**digit**  
YOUR TECHNOLOGY NAVIGATOR

# CHAPTERS

**BUILD YOUR OWN WEBSITE**  
SEPTEMBER 2016

06

PAGE

12

PAGE

21

PAGE

28

PAGE

34

PAGE

## The Purpose

To build or not to build. That is not the question.

## Components

Your cheatsheet to website building!!

## The elements of an ideal website

From content layout to design and SEO, all the right things to get it right!

## Behind the page

The hidden anatomy of websites

## Picking a CMS

A content wizard to do the magic for you

## CREDITS

The people behind this book

### EDITORIAL

#### Executive Editor

Robert Sovereign-Smith

#### Managing Editor

Siddharth Parwatay

#### Technical Editor

Jayesh Shinde

#### Senior Reviewer

Mithun Mohandas

### Writers

Anand N

Dhinoj Dings

Meghana Gupta

Sahil Dawka

Tanmay Patange

### DESIGN

#### Sr. Art Director

Anil VK

#### Visualiser

Baiju NV

39

PAGE

## Picking a web host

Now that you know a fair bit about the front end, you need to delve into the intricacies of web infrastructure. Basically, that part of the www where your site will actually reside.

50

PAGE

## The tools of the trade

Building your own website might seem like a daunting task but if you know the following, then your work just got a lot easier.

59

PAGE

## One-click setup websites

When time is of the essence, a 1-click solution is what you need. Here's how we set up a website in no time.

68

PAGE

## DIY - Assisted Setup

Explore the ridiculously easy world of drag and drop website builders

75

PAGE

## Building a website the hard way

There are very few things as rewarding as building your own website and a completely hands-on approach is all the more worth it.

87

PAGE

## Secure your website

For if you don't, bad things may happen. Better safe than sorry, right?

© 9.9 Mediaworx Pvt. Ltd.

Published by 9.9 Mediaworx

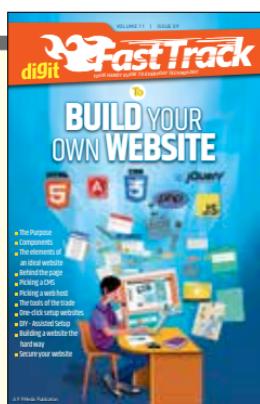
No part of this book may be reproduced, stored, or transmitted in any form or by any means without the prior written permission of the publisher.

**September 2016**

Free with Digit. If you have paid to buy this Fast Track from any source other than 9.9 Mediaworx Pvt. Ltd., please write to [editor@digit.in](mailto:editor@digit.in) with details

**Custom publishing**

If you want us to create a customised Fast Track for you in order to demystify technology for your community, employees or students contact [editor@digit.in](mailto:editor@digit.in)



COVER DESIGN: PETERSON P

# You can build a website...

**W**hether you're a budding entrepreneur, an aspiring photographer, an ambitious writer or even a full fledged unicorn breeder, having a website is gradually becoming more of a necessity than just a means to reach out to more people.

In fact, recently someone got hired because they built their resume in the form of a good website. That being said, it is also true that due to the huge number of websites out there, the competition to actually build a good one is also quite tough, with methods ranging from the ease of a few clicks to full fledged hours of coding for you to choose from.

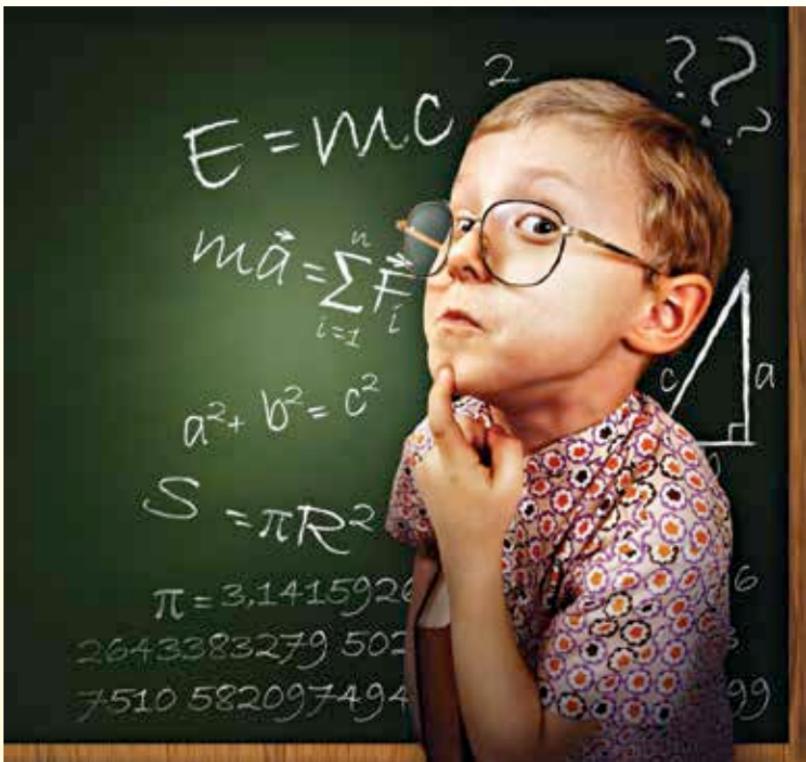
There are literally more than a billion websites out there right now and by the time you blink after reading this line, a few hundred new websites will have come alive. Admittedly it will be a cat-video sharing, click-baiting viral content website but nonetheless, it will be one that is getting the numbers that it needs. The internet, apart from reaching everywhere, has also transformed based on the behaviour of users along with the technology that runs behind it. We recently put out a FastTrack on SEO – essentially a guide on how to optimise your content and website to be search engine friendly. Building a website from ground up with that same objective in mind is even better, and will help you avoid having to put cat-videos on your site for clicks. Unless that's what you had in mind all along!

With that large population of websites, there is obviously a large number of categories they fall into as well. For example, a video sharing website behaves in a significantly different manner as compared to a content discovery platform, yet modern day websites have a set of features that are becoming increasingly common across various categories. Define what

those features will be before you start on your own website. To do that, you have to take into account the domain hosting, the backend servers, as well as which type of building method you will use in the first place. And all of this is even before you've started working on the front-end!

It is fair to be confused with so many things to factor in and take decisions about, especially when the possibilities for the end result are so enormous. While choosing the coding approach gives you almost perfect control on how your website feels and behaves, choosing the automatic website builders will get the basic job done with much less time and effort spent on it. And irrespective of what approach you choose, always know that no website is perfect from the get go and even immensely popular platforms like Facebook, YouTube, Twitter have gone extensive redesigns and feature overhauls to get where they are now, and even they aren't perfect. With that thought in mind, unleash some www magic into your ideas! 

## CHAPTER #01



# THE PURPOSE

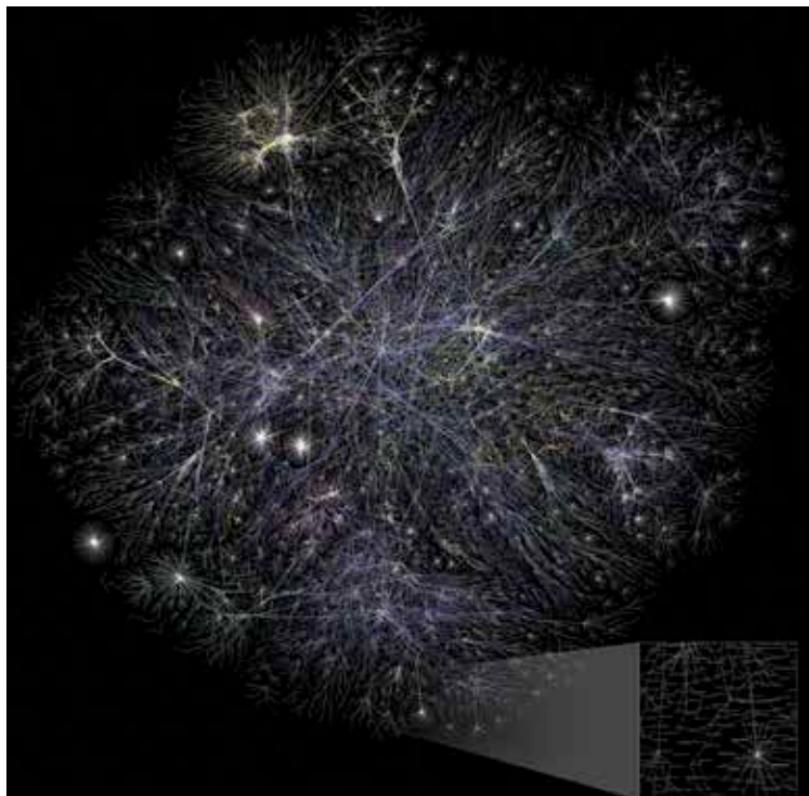
To build or not to build.. That is not the question.

### **The World of Information**

This age is the age of information. Just as early man had no choice but to go out and hunt, today's man has no choice but to go online. It is only logical, seeing as our large and ever increasing population limits and reduces how much is available. There are space constraints when performing functions like storing or communicating information on traditional physical media like paper. Everything is being digitized, and for the better. In the abstract

world of bits and bytes, this limitation is reduced exponentially. Besides, we can always add more servers, which coincidentally may be the only thing an apocalyptic AI technology will need us humans for. Back here in the present, it may not be feasible for one to buy the perfect house or set up shop on the perfect corner street, luckily the web domain market is not as prohibitive. Having a website can often times be a profitable affair, giving your business a strategic advantage and access to the marketplace of the next generation. Furthermore, this marketplace has a targetable audience, which makes it easy for you to take your shop to the customer, rather than wait for the customers to walk into your shop. Not to mention other great advantages of the internet such as real-time communication and an open, almost perfectly flat playing field.

To illustrate what a blessing the last point is, imagine the internet as a dense network of superfast, super thin highways. Websites are all basically



The super-highways of the internet

practically equally valuable properties on the highways of the internet. Contrast the following tasks, given that you are to start with nothing: trying to open a hotel that is even remotely comparable to [insert high end hotel chain here]; and opening a website that is a streamlined and better featured version of [insert popular website here]. If you are determined enough, with this Fast Track you could achieve the second task in a matter of weeks! In less than 24 hours even, if you're that kind of genius who naturally appears to be on NZT-48. Of course, we did say almost perfectly flat. Which was to stress on the fact that it is NOT a level playing ground, unfair advantages do exist. Other than sites that have already made a mark in our minds (like Google), the prime differentiator between websites are the number of highways that link to them. These links come from search engines and other websites, as we saw in detail in last month's Fast Track to Search Engine Optimization, and as we will briefly see in forthcoming chapters. As far as content is concerned, unless you're hosting gigabytes of video (which is still freely stored on YouTube), storage is cheap. Domain registration is cheaper still, unless you want a really common word which is a premium domain (which is why many new companies have names that spell like typos). Overall, other than the effort and expertise, owning a web domain is a highly affordable (and did we mention potentially profitable?) affair. So unless you're an anxious loner by choice, get your website up already!

Of course, you could always pay someone else to do it for you, but that is a much larger investment than it needs to be, until you're dealing with six digits. That's about the number (cost and/or traffic) when hiring a professional specialist can make a significant difference. With high traffic volumes (like Amazon level high volume), even a single word from the content can affect a big chunk of revenue. For the purposes of the above average Digit reader (because all Digit readers are above average), the investment in this month's magazine should be enough. Well, if you want a website with your own domain name (like [www.yourname.com](http://www.yourname.com)), it will cost you some more, say, another magazine. That is still less than quarter the cost of hiring the lowest wage web developer. Now that you hold this Fast Track in your hands, the first step has been taken, your website is in the making.

### **“What to build?”, that is the question!**

To have a clear idea of what you want to build, you first need to decide on why you are building in the first place. Are you expanding the accessibility of your business? Are you showcasing your skills and work to enhance your

employability? Are you creating a community around a certain topic/issue? Even though you may be initiating into the world of web domains only now, the world wide web has been around for a while. In that time, various websites performing various functions evolved in various ways, and patterns emerged from the chaos. Naturally therefore, what you see today is what works, for each type of website. For example - OLX and Quikr, Amazon and Flipkart, Google and Bing, etc. Of course, there are always mavericks who follow no one and put up whatever they want to for its own sake, but if you study the websites who chase the traffic and undergo overhauls based on the data from their users, you will find that the most popular sites in a particular category are more similar than not.

Even if you're the artistic type who wants a completely custom site, you still need to have a look at the most popular sites on the internet that are related to yours, if only to see what's already been done. As long as you think you need traffic, take note of the format of the website, and imagine how you can adapt it to your needs. We have scoured the internet aimlessly for you, so without further ado, here are the most common formats of websites on the internet of today:

## 1. The Blog

This is a very popular format for websites, and accounts for the statistical majority. That is because it also consists of many subtypes. The word 'blog' is derived from 'weblog', which is "a Web site that consists of a series of entries arranged in reverse chronological order, often updated frequently with new information about particular topics." After all, like food is consumed as raw material for repair and regeneration, information is consumed as raw material for action or communication. Most of the time, it's communication. Guess what? Light speed makes the internet very conducive as a medium for communication.

What most people automatically think of when they hear "blog" is the written blog. Then there are image blogs (and platforms) like flickr, EyeEm and instagram, video blogs and platforms (vlogs) like YouTube and Vimeo,



Popular Essential Internet Starter Pack

but mostly just mixed media blogs like Facebook. Yes, every Facebook page is basically a blog. Along with the thousands of people doing this for social sustenance or out of pure passion, there are the corporate blogs (more like blog platforms) like the Times of India, Huffington Post, New York Times, Forbes, Autocar, The Economist, and hundreds more than churn out page after page every few hours.

You should start a blog or a YouTube channel if you want to reach out to a mass of people, gain a following, and share a passion or spread a message.

## **2. Personal/Portfolio**

Some people make a website just for the fun of it, to showcase themselves or their work. These websites are usually one or few pages, containing specific mixed media information. There are even platforms like about.me for readymade 'instant' personal websites with literally a little information about you. If you're any kind of artist, you can host your body of work, and if you're a student, you can put up your resumé.

## **3. Professional/Business**

These are the commercial equivalent of the personal/portfolio style websites, in that they are designed to sell a service, or brand a brand image on your brain. If there are products to sell, there are few. These websites must have a crafty copy with a call to action, testimonials to build trust, videos to demonstrate value, and also contain the most crucial contact information. The name of this game is turn casual visitors into interested inquirers. Build your website according to this format if you have an offline business which you would like to market online, or if you're a startup looking for funding.

## **4. E-commerce**

These are websites that have evolved a design with the specific intent of have users buy as many products as easily as possible. They feature multiple levels of categorization and are usually image heavy, because people like to see what they are buying. User reviews on individual product pages are an important tool for users to make decisions about the purchase. Daily discounts and deals are a common tactic to push products and increase sales. E-commerce websites necessarily have tie ups with merchant transaction handlers such as PayPal, PayU, CCAvenue, etc to enable the customer to purchase instantly. Today, the biggest e-commerce websites have massive amounts of traffic and consequently, usage data, which is

studied to uncover trends and triggers for manipulating consumer behaviour. You should aim for this type of website if your motive is to sell products, especially if you have many products, and even more so if you want save on real world space with only an online outlet.

## 5. Community/Forum

The community/forum format is a design that evolved primarily for intra-site user interaction. They are some form of message boards where users can discuss topics by posting in threads, which functions somewhat

like a non-transient chat room. The first example that comes to mind is Reddit, which is where most of the internet hangs out. Many large websites that may be multi-media blogs, and even business websites, also have their own forum section. For example - IGN forums, Affinity forums. As long as there are a sufficient number of people dedicated to a pursuit, you can be certain there is a forum for that somewhere. These websites are usually a treasure trove of information, with most of it being experiential. Use this format if you want to build your online real estate to be a place where certain people virtually hang out, where they can share information and other resources.

## 6. Cloud Apps

Cloud applications are online software services that can be accessed like any other website. These websites have a lot of complex server-side logic, and amalgamate high levels of functionality with usability. Examples illustrate like nothing else, so here a few types of cloud applications: Search engines, Email clients, Document processors, Storage space, Image editors, Format converters, the list goes on.. Besides highlighting Google's dominance, you can easily see that the possibilities are only limited by imagination and hard work. However, this type of a website is not in the scope of this Fast Track, but if you do want to build something like this, start by heading over to Chapter 11 for some old school basics. d



Don't miss this exclusive offer! Click now!

## CHAPTER #02



# COMPONENTS

Your cheatsheet to website building!!

### Introduction

As you delve into the world of the web developer, the amount of terms and labels you are familiar with will play a crucial role. Too much of what sounds like mumbo-jumbo can throw one off. After all, even the best secret agents are briefed before a mission. This chapter is aimed at making you, the reader, familiar with the terrain. Consider this a survey highlighting the lay of the land, in case you ever get lost. Treat it like a glossary: if ever you read



Everything to do with websites under one roof

something and draw a blank, refer back to this chapter and you should be good to go.

## The more you know..

Consider the following a compendium of beginners' essential web terminology, literally anything that has anything to do with websites, in alphabetical order.

**Accessibility** - This term refers to how easy it is to use a website, across different situations, say, on the various screen sizes of different devices, and for visually or otherwise technologically challenged users. This is an important component of a website emerging from the design choices.

**ASCII** - American Standard Code for Information Interchange. It is a code that assigns values to symbols and actions (of the keyboard) so that it can be mapped to what is displayed on the screen.

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	NULL	32	20	SPACE	64	40	!	96	60	~
1	1	START OF HEADING	33	21	!	65	41	K	97	62	#
2	2	START OF TEXT	34	22	“	66	42	L	98	63	%
3	3	END OF TEXT	35	23	”	67	43	M	99	63	<
4	4	ZERO OF TRANSMISSION	36	24	5	68	44	D	100	64	=
5	5	ZERO-NINE	37	25	6	69	45	E	101	65	_
6	6	ACKNOWLEDGEMENT	38	26	7	70	46	F	102	66	:
7	7	SYNCHRONIZATION	39	27	8	71	47	G	103	67	@
8	8	QUESTION MARK	40	28	9	72	48	H	104	68	^
9	9	INFORMATION DATA	41	29	10	73	49	I	105	69	~
10	A	ALPHA-TELETYPE	42	2A	11	74	4A	J	106	6A	`
11	B	GRAPHICAL TAB	43	2B	12	75	4B	K	107	6B	;
12	C	SYNCHRONIZED	44	2C	13	76	4C	L	108	6C	,
13	D	CHARACTER RETURN	45	2D	14	77	4D	M	109	6D	!
14	E	CHARACTER OUT	46	2E	15	78	4E	N	110	6E	“
15	F	CHAR-OUT	47	2F	16	79	4F	O	111	6F	”
16	10	DATA-LINK ESCAPE	48	30	0	80	50	P	112	70	@
17	11	DEVICE CONTROL	49	31	1	81	51	Q	113	71	4
18	12	DEVICE CONTROL	50	32	2	82	52	R	114	72	5
19	13	DEVICE CONTROL	51	33	3	83	53	S	115	73	6
20	14	DEVICE CONTROL	52	34	4	84	54	T	116	74	7
21	15	UNPRINTABLE	53	35	5	85	55	U	117	75	8
22	16	SYNCHRONOUS	54	36	6	86	56	V	118	76	9
23	17	SYNCHRONOUS BLOCK	55	37	7	87	57	W	119	77	0
24	18	SYNCHRONOUS	56	38	8	88	58	X	120	78	1
25	19	ZEND OF AVERAGE	57	39	9	89	59	Y	121	79	2
26	1A	UNSTRUCTURED	58	3A	10	90	5A	Z	122	7A	3
27	1B	SEGMENT	59	3B	11	91	5B	J	123	7B	4
28	1C	FILE-SEPARATOR	60	3C	12	92	5C	“	124	7C	5
29	1D	ZERO-SEPARATOR	61	3D	13	93	5D	1	125	7D	6
30	1E	RECORD-SEPARATOR	62	3E	14	94	5E	“	126	7E	7
31	1F	UNIT-SEPARATOR	63	3F	15	95	5F	”	127	7F	8

The ASCII table

**Attribute** - Attributes are additional properties that can be specified inside the html tags of an element. They specify information about the element that can be crucial to its functioning, which are given by values following an '=' sign. For example, the href attribute specifies the target address when using the `<a>` tag, as in `<a href="http://www.digit.in"> Digit Home </a>`.

**Backend** - The backend refers to everything that is server-side, i.e- the database and server-side code that integrates the data from the database into the web pages. For example, everybody's Facebook profile page will

have the same code making up the page, however the backend code returns different content from the database depending on who is logged in.

**Backlinks** - Backlinks refer to hyperlinks that point back to your website. Increasing the number of backlinks from other websites gives you more traffic, and may also increase your ranking in the search algorithms.

**Bandwidth** - Building on the metaphor of the internet as interlinked highways, bandwidth is equivalent to the width of the roads. It is the amount of data that can be exchanged in unit time with the web server. Low bandwidth limits the amount of traffic that your website can handle.

**Banner** - Banners are wide graphics usually found in the top half of the home page, and also sometimes at the bottom, in the footer. They often contain artistically emphasized text, and are meant to be eye-catching. At times, a banner may also be a hyperlinked advertisement.

**Binary** - Binary is a number system. It has base 2, and two digits, 0 and 1, which are also called bits (binary digits). This is the most basic language which computers understand on a hardware level, represented by on and off states, or high and low states in transistors.

**Bounce Rate** - The bounce rate is defined as the percentage of visitors to a website who leave the site (by navigating to another domain) after viewing only one page. In other words, this is the percentage of people whose attention your website probably didn't grab.

**Browser** - A browser is the application software that you use to view and interact with websites and web pages. It is your metaphorical surfboard for surfing the internet. When talking about communication with the server, the browser may also be referred to as the 'client'.

**CSS** - Cascading Style Sheets are the files that control how the content of a page looks. This can be accomplished by specifying the values of appropriate element attributes in the CSS files. CSS can even be used for transitions, to create some cool simple animations.

**CMS** - CMS stands for Content Management Software. They allow not-so-tech-savvy individuals who want to build their own website to do so easily, and with as less to do with coding as possible. Many CMSs are in fact meta-websites, websites that build websites. For details, see Chapter 6.

**Conversion Rate** - If your website or landing page has a specific purpose, like getting users to subscribe to your mailing list, the conversion rate refers to the number of users that subscribe divided by the total number of visitors to the page (multiplied by 100% if you like percentages).

**Cookies** - Cookies are files stored by a website locally on your computer or phone, in the cache of your browser. They record information about your usage of the website and can be very convenient at times. At other times, they can be a breach of privacy.

**Domain** - A domain name is what you enter into the address bar of your browser. It is the name of your website, like [www.digit.in](http://www.digit.in), [www.google.com](http://www.google.com) or any other website that you can think of. Domain names have to be purchased, and the registration usually lasts for a year or two, after which it has to be renewed.

**DNS** - This stands for Domain Name System, a critical part of a functioning world wide web. If you've ever wondered how a domain name URL written in letters corresponds to a numeric IP address, the answer is DNS. The DNS maintains a list of registered domains and their corresponding server's IP addresses. DNS is the translator between these two address spaces.

**Element** - An element can be considered a 'basic unit' of HTML files. Everything on a web page is defined by / within an element. Each element is represented by tags like <html> and </html>. In fact, everything on a web page (except the doctype) resides within the html element, and either in the <head> element </head>, or <body> element </body>. Elements can have attributes specifying properties like color, in the opening tag.

**Favicon** - The favicon is the small image icon that is displayed in the address bar, next to the URL.

**Fonts** - Font families are the different typographic styles (or typefaces) for displaying the symbols that represent the letters of the alphabet, like Sans-Serif, Roboto, etc. They also have the properties of size and weight (boldness), which are together referred to as the font.

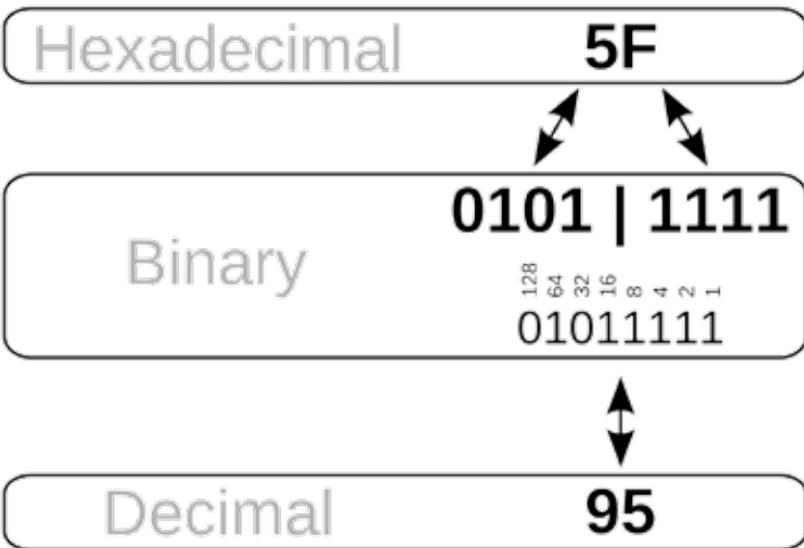
**Frontend** - The frontend of a website refers to the code that mainly makes the parts of the web page that users see. The front end includes the user interface, navigation structure, fonts, formatting, etc. It is also called the presentation layer.



The Static Frontend Force - All you need for a great static website!

**FTP** - File Transfer Protocol is a communication protocol used on to transfer files on the internet, such as to upload the files for your website to the host server.

**Hexadecimal** - Hexadecimal (often shortened to hex) is a numbering system with base 16. Our everyday numbering system is decimal, with base 10. The decimal system has 10 single digit numbers and correspondingly the hexadecimal system has 16 'single digit' symbols. The symbols 0-9 represent the values zero to nine and the symbols a-f represent the values ten to fifteen. So 10 in hex is the same numerical value as 16 in decimal.



Practice problem: Figure out the question

**Hit** - While many manipulative marketers utilize this statistic in a way that implies number of hits is equal to the number of visitors, this is simply not true. The website registers a hit every time a request is made to the server, and a request is made every time your browser wants to access a resource, even when you submit forms. Thus a single visitor opening a single page can generate multiple hits.

**Host** - Your website and its pages reside on a server that needs to have adequate storage as well as bandwidth for the site to function properly. The provider of such servers are called web hosts. See chapter 7 for more.

**HTML** - HyperText Markup Language is the language that all web pages are written in, one way or the other. Pure HTML will result in a static

page, and thus pages are usually designed with a combination of HTML, CSS and JavaScript.

**HTTP** - HTTP stands for HyperText Transfer Protocol, and is the communication protocol for the exchange of information on the internet, between browsers and servers. It is written in the address bar as part of the URL, preceding the address of the resource being accessed.

**HTTPS** - [HTTPS](https://en.wikipedia.org/wiki/HTTP_Secure) is HTTP Secure, or HTTP over SSL (Secure Socket Layer). This is also a protocol for communication between servers and browsers, the difference being that unlike plain [HTTP](https://en.wikipedia.org/wiki/HTTP), it is encrypted.

**Hyperlink** - Commonly called a 'link', it can be an image or text or even blank space on a web page, which can be clicked on. Clicking on a hyperlink takes you to the address that the hyperlink links to, enabling a network of interconnected web pages. The linked address can be on the same website or a different one.

**Iframe** - Inline frames are an element which enables the viewing of other web pages within that web page.

**Impression** - An impression is a view on a page or any element thereof, say, a link. An impression is not a click, but a click definitely implies an impression.

**IP Address** - An IP address is an Internet Protocol address. It is a string of numbers in the format x.x.x.x, where  $0 < x < 255$ . When someone enters your domain URL into their browser, the DNS parses it to the corresponding IP address, and they are directed to the server hosting your website.

**ISP** - Internet Service Providers are the businesses who give you internet connectivity. There are around 3 different classes of ISPs that constitute a global hierarchy. Only the top level ISPs control the internet, while all the other classes purchase bandwidth from a provider in the class above them to sell it to consumers or ISPs in lower classes.

**JS** - JavaScript is a front end scripting language that enables the use of logical structures in webpages. There are also JavaScript programming environments like Node.js, Angular.js and many libraries such as jQuery.

**JPG** - jpg (or jpeg) is a highly compressed image file format developed by the Joint Photographic Experts Group. It offers compression ratios of 10:1 with very little loss in quality, and thus is the most widely used image format on the web.

**Keywords** - Keywords or key phrases are what you expect your potential users to enter into a search engine when you want them to find your website. For example, if your website offers professional Chartered

Accountant services for individuals, the keywords (key phrases) you would target on AdWords may be something like “how to file income tax” and related terms.

**Landing Page** - This is a specialized home page on a website, the page that the user ‘lands’ on when he/she visits your URL. Landing pages are usually specifically designed to elicit certain behaviour from the user, like signing up for a newsletter or donating to a cause.

**Libraries** - Libraries are a bunch of code written by someone else that you can incorporate into your own code to save time and effort. A lot of awesome libraries are freely available so it’s good practice to credit the authors when you use their work. The most popular example for an aspiring web developer would be jQuery, which is a JavaScript library.

**Liquid Layout** - A liquid layout (as opposed to a fixed-width layout) is a design choice that defines element widths in terms of percentages of the total width. This way, content is rearranged in real-time according to the width of the browser, and the presentation can be optimised according to the size of the maximum width. An example is the box model in Bootstrap.

**Meta data** - Meta data is information about the web page itself. It is contained in the `<meta>` tag in the header. This information is not displayed on the website itself, only in the source code. Some search engines use this data in their ranking algorithms.

**Navigation** - Navigation pertains to the system on your website that allows user to visit the various pages while having an understanding of where they are. The most common navigation systems used in websites are menus.

**Optimization** - Optimization may refer to Search Engine Optimization, or structural and content optimization of your webpages. In the second case, images need to be compressed as much as possible and the scripts and other resources need to be in order so that the page loads as fast as possible.

**Parked Domains** - Parked domains are those domain names which point to the same hosting server as another domain name. That is, if you enter either domain url into your browser, you will land up on the same website.

**PHP** - PHP is a server side scripting language that can be used to create dynamic websites. Some CMSs are written in PHP.

**Plugin** - A plugin is third-party code that you can add to extend the functionality of your CMS of choice. There are free and premium plugins, and they offer a range of easy enhancements like e-commerce integration, SEO, slideshow banners, etc.

**PPC** - Pay Per Click is a payment method when using advertising services like Google AdWords to generate traffic to your site. You specify the details of the ad and the contexts in which it should be shown, (and many other details,) and you pay a certain amount whenever a user clicks on the ad and you get a visitor.

**Ranking** - Your website contains content, and people who haven't heard of your website can find it if they are searching for that (or similar) content on a search engine. Search engines take a query to return a relevant list, and ranking refers to how high your website is on that list.

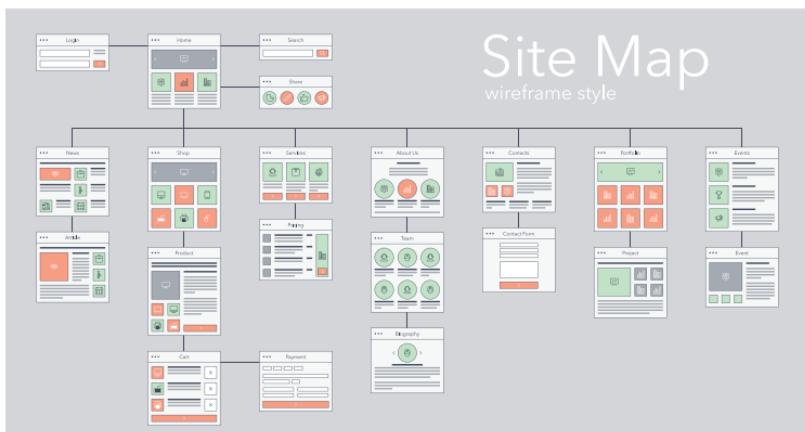
**Root** - The highest level folder which contains all the files and folders that make up your website is called root, a term that Linux users will be familiar with.

**Script** - A script is a collection of code that enables some kind of interactive/logical functionality on a web page. The most common language for scripting on websites is JavaScript.

**SEO** - Search Engine Optimization is a strategy to increase the visibility of your website by having it rank higher in search results for related content. For in depth knowledge about SEO, grab a copy of the Fast Track to SEO, covered last month.

**Server** - A server is where the files that make up your website are stored. Whenever someone enters the url of your website from their browser, it sends a request to the server, which in turn serves back the required pages.

**Sitemap** - A sitemap is literally like the map of your website. It is usually a chart or an index which conveys the page structure of your website.



How a hard working designer would make a sitemap.

Sitemaps are useful for users to navigate your site, as well as for search engines to properly crawl and index your site.

**Subdomain** - Subdomains are a way to split up your single domain name for multiple websites. For example, if your domain name is [www.funkynname.com](http://www.funkynname.com), you could have almost unlimited subdomains of the type [www.subdomain1.funkynname.com](http://www.subdomain1.funkynname.com), [www.sd2.funkynname.com](http://www.sd2.funkynname.com), etc.

**Tag** - A tag is used to specify the starting and end of an HTML element, in the format `<tag>element content</tag>`. A few tags do not require the end `</tag>`, like the (outdated) `<br>`.

**Themes** - Themes are pre-packaged designs for websites, especially those made using popular CMSs or website builders. Themes ensure that all pages have a consistent look across your website, while providing a professional look. Some themes are free, some are premium and cost money.

**Traffic** - Just like the vehicles that cause physical traffic on the roads, every packet on every digital route contributes to the traffic on the highways of the internet. For a website, the 'traffic' generated refers to the number of visitors who have viewed any web page. This statistic is often used to judge the market value of a website for monetization. A high traffic also requires the proportionate amount of bandwidth, which is determined by the hosting service (see Chapter 7).

**URL** - Acronym for Uniform Resource Locator. This is the address of any web page (containing the domain name), which you enter into the address bar of your browser.

**Web page** - A webpage, also referred to simply as a 'page', is a single .html or .htm file on a website (that may also be associated with a .css and .js file). Although all the content of a website can be fit on a single web page, websites are divided into multiple pages for ease of access and comprehension. A web page may be static (remain the same unless modified) or dynamic (constantly appears different, as determined by the backend code).

**Website** - A website is a collection of webpages that are linked to each other according to the sitemap. Most often, all the pages of a website reside under the same domain. A website may also organize different sections under different subdomains. **d**



# THE ELEMENTS OF AN IDEAL WEBSITE

From content layout to design  
and SEO, all the right things to get  
it right!

**S**ome argue that website design should be considered as an art form in its own right. While most of us have far more modest ambitions, it's still worth noting that the intricacies of website development and the attention to detail it demands could well be at par with what an artist invests on his work, if not more so.

The fact that designing and launching a website has become very easy in the 21st century shouldn't lull you into thinking that it's all a breeze. Not if you want to create a website that will truly cater to the needs of your audience.

Design and interactivity play crucial roles in the success of a site. You will be surprised by how much these things depend not a highly evolved visual aesthetic (though that undoubtedly helps) but largely on careful consideration of different elements.

Let's take a look at them right here:

## 1. Use icons only where appropriate

You've heard the saying, "Too many cooks spoil the broth." Now, here's something you haven't heard before: "Too many icons ruin the site." But just because it's gone largely unarticulated in popular culture doesn't mean that it's not true. On the contrary, stuffing a webpage with icons can make browsing a clutter-filled prospect, one that could confuse the audience more than guide them.



This icon can mean "click" or it can have something to do with changing your input. Which is it?

The ideal then lies in reductionism- eliminating as many icons as possible, leaving just the necessary ones. Sometimes, the icons could replace text-bursts entirely, bringing more space to the page. A couple of things to keep in mind about icons are mentioned here:

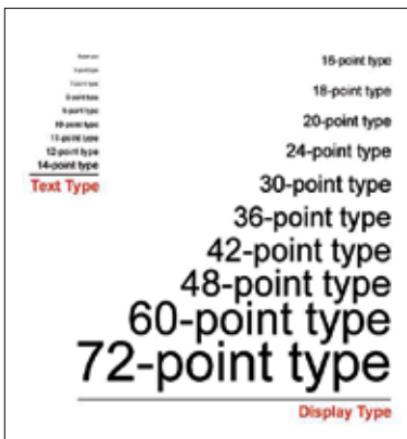
- You can deliver the icons in the SVG (Scalable Vector Graphics) format. The format is great for creating websites that will be viewed on multiple devices-the icons can be resized according to your requirements, and that too without 'fattening' the file size.
- Trying simple animated icons could also be a great idea (provided you don't go overboard with it). You can animate SVG files using CSS. This way, you needn't wallow in a lot of extra scripts to bring in some decent effects.

## 2. Never under-estimate the significance of the font size

For one thing, the font should go well with the overall design of the website. If the design template falls on the serious side of things but the font choice is... erm... Comic Sans, it may not be a good fit. Perhaps more important is the element of readability. There are plenty of websites out there with great content but you don't feel like reading just because of the poor font selection.

Here are a few pointers for avoiding mistakes with fonts:

- There are certain typefaces that have thin letterforms. These work well enough if the screen size is large but when scaled down to the smaller screens of a handheld device, the font is the equivalent of ants on the lawn of your home when viewed from the first floor balcony. Okay, maybe not that bad but bad enough.
- Make sure that you add to the height of a line of text in direct proportion to the width of the text column. This will make it easy for the reader to find the beginning of the subsequent line.
- While the default font size on most browsers is 16 pt body type (and it affords decent readability on all screen sizes), feel free to experiment with the font size.
- The heading should be at least 1.6 times bigger than the text that comes under it.
- The colour of the text shouldn't be overlooked either. Make sure that it contrasts with the background colour of the webpage. Also, ensure a different colour for the links to make them stand out from the rest of the text.
- Visit websites like <http://typographica.org/> to make the best font selections



Don't make your website visitors reach for their magnifying glasses

## 3. Images should be awesome, so should the download speed

It's imperative that the images used in a website are kick-ass. Sometimes, there's nothing people like to see more than a well-made image. And as a mode of capturing and holding the audience's attention, high quality images are indeed effective. But you should be mindful not to sacrifice

the page performance on the altar of visual aesthetics. Here are certain considerations:

- If you're designing a responsive web page, keep a hawk's eye on each and every single visual element on the page. Assess the significance of each piece, reassess, and make sure that only the necessary ones make the final cut.
- Things like homepage carousels should be included only if necessary. Carousels imply multiple numbers of large images. Not just that, they also call for the JavaScript to put these displays in motion. The page size can be considerably reduced by foregoing a carousel with four images for a static image.
- Once you have taken away all unnecessary images, optimise the remaining ones so that they are apt for web delivery. Make sure to use the right image formats like SVG or PNG.



Says a thousand words, doesn't it?

#### 4. Ensure scalable navigation

Smooth navigation is one of the trickiest things to pull off in a website—particularly if the menu is rather large or deep. The key issue is not that you cannot devise a good navigation model—it's that what works on larger screens could be utter chaos on smaller devices.

What this essentially means is that instead of searching for a navigation experience consistent across devices, you are better off creating user-friendly navigation on a step by step basis: you would have to tweak the steps according to different devices.

While breaking away from the UX convention of being consistent across devices, you should know about the 'method to the madness' in this. To that end, here are two pointers:

- You should indeed be consistent with certain elements



Users have to find it easy to get around

in the navigation-for instance, with such things as the font, text colour and the labels on the buttons

- The break away from consistency could happen with the way in which the navigation works. For example, whereas a fly-out menu which is accessed when the mouse cursor hovers over it may work brilliantly for desktop visitors, for touch screens and smaller screens, it needn't be the case. In such instances, alternative functionality may need be applied; perhaps a navigation drawer.

## 5. Prioritising the content

One thing that's inevitable is that the user won't have time to peruse all the content that you put in a website. The user, more often than not is looking for some specific information when they visit a website. This should be the first parameter in arranging the content in your website-with the most relevant or significant information coming at the top and content tapering down in the descending order of significance.

But that's not all that you should keep a tab on:

- For larger screens, you can include an editorial image and maybe even the author's photo along with the core content. Other elements could include the title of the content piece, the author's name and the date on which the piece was published.
- But then, that's for bigger screens. For smaller devices, the layout need to be altered. The editorial image, in such a context may even be avoided since it's not crucial to convey the content. Space-saving is the name of the game here. The author's photo and the heading need be the only elements listed one below the other on the lead page in the smaller screens.

## 6. Make the context central to the design

Context in this case can simply be considered as the wealth of information that's available about the product/service/establishment which the website stands for.

Usually, a responsive web design implies nothing more than scaling a single column layout on a small screen to multiple columns as the size of the screen progresses. While that does make sense from a sheer scalability perspective, the context in which the content is accessed is not taken into account.

For example, if it's a website for a restaurant, there would be such content as the menu, reviews, the location and the map. Now, as far as mobile users

go the common assumption is that they are either looking for reviews or maybe the map to reach the restaurant. But that assumption will be proved wrong if the user is already at the restaurant. In this case, they might be looking for some other information-like the nutritional facts about each entry on the menu.

The upshot is that even before starting the design process, you should amass as much info as possible about the project-adding depth to the context. Then, ensure that it's easy for the user to collect whatever information s/he is looking for.

## 7. Remember that responsive design doesn't mean "for small sizes" but "for all sizes"

For some reason, successful responsive design has come to mean that designs which suit the smaller screens well. While it's true that more number of people now access the net through mobile and other smaller devices, as even a cursory glance through workplaces show, bigger screens are far from obsolete.

So, in terms of website design, the best option is to adopt the "Mobile First" approach. *UX Design Trends 2015 & 2016* – a free ebook you

should read – gives some great advice for responsive design. You will do good to rank the content categories listed there according to your priority and lay these out in the smallest viewport first. Elements can be added as you require when moving to larger devices.



Your pages need to adapt

As suggested in the heading, do not overlook larger screen sizes. Not just that, you should also look at sizes between popular breakpoints. Since new devices enter the market almost every day (one estimate puts the number of distinct Android devices available at 24,093), breakpoints that are popular today needn't remain so even in the near future. This also means that sizes which needn't be considered at this point could turn crucial further on. So, to reiterate, to achieve something with true scalability, you should focus on the popular breakpoints as well as the gaps between them.

## 8. Make sure the content is SEO-optimized

It's hardly enough to have a super-looking website with great interactivity. You should also make your website discoverable by having great content. And certain specifics are to be mentioned about what is meant by 'great content' here:

- **Quality of the content:** Don't just provide content that will be available on a bazillion other websites/brochures. Put across a unique value proposition or information that will make your content stand out from the rest.
- **Use keywords:** Make sure that your content contains keywords through which you want visitors to come to your website. Think of what those keywords will be and add them. For instance, if you are a vegetable and fruit vendor, both vegetable and fruit could be keywords. But if you have a special orange supplier who brings you some awesome varieties, you can consider orange too as a significant keyword. You can also try the different keyword planners available online (like the Google keyword planner) to assist you.
- **Content freshness:** Search engines generally are biased towards fresh content. But that doesn't mean that you can just pile up content after content on a daily basis in the hope that it will give your site a "freshness quotient." What Google does is, when there's a sudden surge in search for a particular topic, it would apply the tag "Query Deserved Freshness" (QDF) to that topic. It will then scout for fresh content related to it. The fresher content, in such cases will be given priority in the search results. For more depth on SEO be sure to read our last month's FastTrack.



This is what will get your site eyeballs

That about covers what you need to know about the components that make a rocking website. Hoping to see a savvy website online soon! 

## CHAPTER #04



# BEHIND THE PAGE

The hidden anatomy of websites

### Introduction

Before getting proper into building a website, it is necessary to understand how websites and the internet work. To understand how things work, we dissect the congregate down to its bare essentials. A website, like everything else in the information world, is made out of code, which is appropriately displayed in a browser. Sounds simple, right? That's because 99% of what is happening wasn't taken into account. In this chapter, we take you behind the scenes, into the inner workings of the Internet.

### The Internet Protocol Suite

Consider the scale of the internet. Billions of devices with exponentially more connections between them communicating at the same time. How

does one device (say, a browser) know which computer is the one it is looking for? How does it let the other computer know what it needs? How do technologically different devices communicate with each other? How do they coordinate the exchange seamlessly every time?

Just as the rules functionally define a game, the Internet is functionally defined by a set of protocols, collectively called the Internet Protocol Suite. Today's Internet Protocol Suite began as one of the first research projects for the development of computer networking models, funded by DARPA of the United States Department of Defence. That effort led to the creation of the Transmission Control Protocol/Internet Protocol (TCP/IP) by Robert E. Kahn and Vinton Cerf, which was later into split into further layers. TCP/IP provides end-to-end data communication specifying how data should be packetized, addressed, transmitted, routed and received; it tells the Internet how to work, and it is still in use today.

For formulating the TCP/IP, Cerf and Kahn had the intention of enabling the communication of multiple different networks with their own local protocols. They wanted to make a network model that handled the transfer of information as efficiently as possible, with all other intelligence handled at the nodes. Today's Internet Protocol Suite is composed of four layers of abstraction between the information on the computer and the information that is physically transmitted. Within the abstraction layers there are many individual protocols for specific purposes which are as different from each other as sugar and salt. The central two layers are the Internet Layer and Transport Layer, of which the Internet Protocol (IP v4, v6) and Transmission Control Protocol (TCPv4) are major components. The peripheral layers are the Application Layer on the software side, and Link Layer on the hardware side.

What does any of this have to do with websites, you ask? Well, to play a game well, one needs to understand the rules. Similarly, understanding the protocols of the IP Suite will definitely give you a leg-up in your dealings on the Internet. We will discuss the pertinent protocols of the Application Layer and refrain from proceeding down the funnel of abstraction, as they don't pertain specifically to websites.

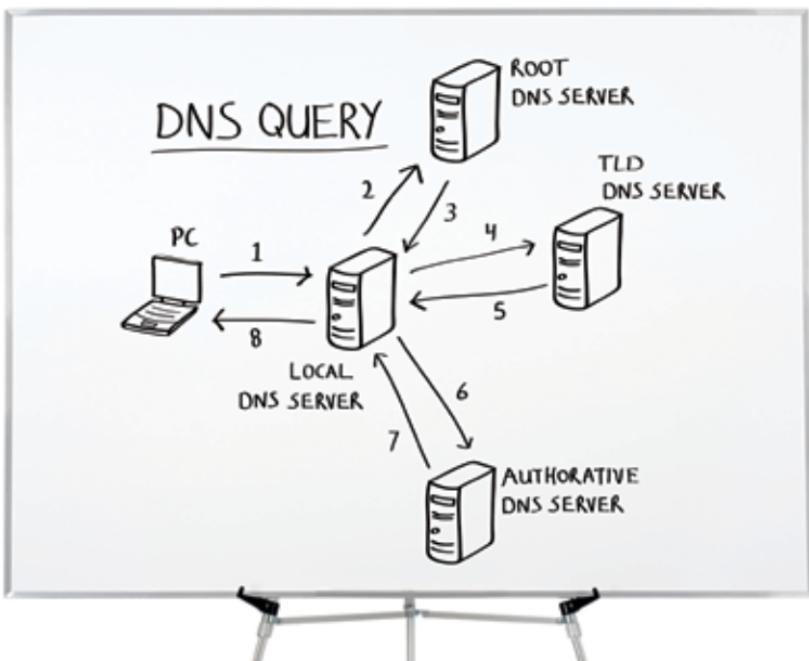
## **Application Layer - Protocols for functioning websites**

The application layer handles the interface between the message data from host and the message that is communicated. It is not responsible for the communication itself, and has no specifications for how the data is to

be communicated (mapping, routing, etc). The various protocols that we are about to go over briefly have one overarching theme - they specify the standard for how communication is packaged. The transmission is left to the higher abstraction levels of the TCP/IP.

## DNS

The Domain Name System is a protocol for mapping resources between the two major namespaces - domain names and IP addresses. IP addresses are the equivalent of physical locations of websites (technically, of the servers that host the websites) on the internet, while domain names are easily recognizable labels that refer to those locations. This reference is what is taken care of automatically and efficiently by DNS. It essentially provides a worldwide, distributed, decentralized and scalable directory system that makes it possible for websites to have human-readable URLs. When you enter a url into your browser, the DNS Name Server directs the request to the appropriate IP address, according to the domain name entered by the user. However, the DNS Name Server also resides on a computer connected to the internet, so the IP address of the name server



DNS is a hierarchical system.

must be entered manually into the configuration settings of the internet connection of the client computer for it to be able to access the internet. Some ISPs have their own DNS Name Servers, else you can always use servers provided by OpenDNS or the Google Public DNS servers. For your own website, if you purchase your domain name and hosting from the same place, the DNS settings and registration will be taken care of automatically for you. However, if you bought a domain name and want to use free hosting provided by someone else, you will need to manually specify the Name Servers of your host.

### Web pages and hyperlinks (HTTP)

If there was a popularity contest for protocols, the HyperText Transfer Protocol would surely be one of the top contenders. Consider the process to open a website:

You enter the URL of the website you want to visit, because you only know the domain name. The DNS Name Server parses the domain name and maps it to the corresponding IP address. Your request for the website reaches the Web Server, and it serves you back the page you requested. Of course, this



Isn't it just everywhere!

is one of the simplest case scenarios, but almost any interaction of a user and a website is handled according to [HTTP](#). That is why almost all URLs start with "<http://>" or "<https://>". [HTTPS](#) is HTTP using Secure Socket Layer (SSL) for encryption.

HTTP was designed as a 'request-response' protocol, meaning that is modelled after how humans communicate in the real world. In general, a client sends a request to a server, which appropriately sends back a response. The response can also contain resources, if requested. The latest version of HTTP is HTTP 2.0 which was standardized in 2015. Prior to that we had HTTP 1.1, which is very much in use even today, because older technologies take time to phase out. HTTP (along with HTML) was invented by Tim Berners-Lee and his team at CERN, the original development starting in 1989.

HTTP is essentially a format for packaging data, and only hardcore web developers need to know more than the four parts of a request packet. The second entry field in that format is the header, containing the request method, which deserves some attention. The first version of HTTP had only one method, GET, which is sent by the client to request some resource (like a webpage) from the server. Some progress later, HTTP 1.1 has eight methods, and the second most popular request method is the POST method which is how elements like forms return user data to a web page. There are also the PUT and DELETE methods, which can be used for telling a database engine in the backend what to do (for example appending details in the database for creating a new user account).

## **File Transfer Protocol (FTP)**

The File Transfer Protocol was first developed by Abhay Bhushan in 1971. It is the predecessor of [HTTP](#), which was originally developed as an enhancement of [FTP](#), optimised for transfer of information on the scale of websites. FTP is still in use today for transferring web content, many hosting vendors' interfaces for uploading your website's files are driven by FTP transfers. There are many dedicated client softwares for FTP data transfer with a server, such as the open source FileZilla. Some WYSIWYG web page building software also include direct FTP upload functionality.

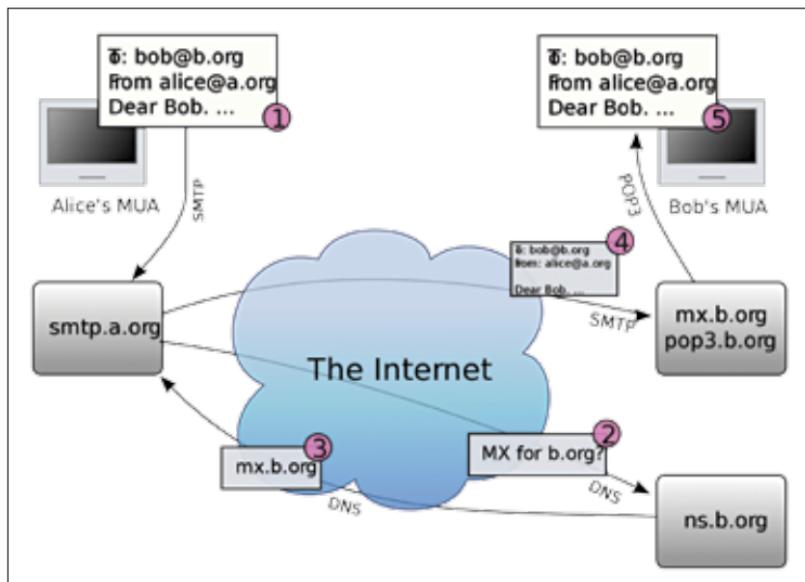
However, FTP was not designed to be secure, and has many vulnerabilities and holes as data is not encrypted. In fact, all transmissions are cleartext, so mere packet sniffing can easily land the username and password into malicious hands. To overcome this, FTP over Secure Shell (SSH) and FTP secured by SSL ([FTPS](#)) were developed.

## **SMTP**

SMTP stands for Simple Mail Transfer Protocol, which defines the how electronic mail is sent and received amongst the various mail servers on the networks of the internet. Mail servers are gateways between the client computer/application and the internet, that specialize in sending and receiving email. You, as a user, use an email client software (ex - Microsoft Outlook, Apple Mail, Google Inbox) to read and write your emails. The client software sends emails to the mail server using SMTP, however, it retrieves emails from the mail server using POP3 (Post Office Protocol version 3) or IMAP4 (Internet Message Access Protocol version 4). When you purchase the domain and hosting space for your website, you may also

get a few email accounts. This means you can have an email address like [you@yourdomain.com](mailto:you@yourdomain.com) for official purposes. To enable usage on your favourite email client, you will need the IMAP/POP settings enabled in your mail server provider, which you then copy while entering your authorization details into the email client.

A small burst of dopamine for guessing what SMTPS is.



What happens after you click Send.

## TL;DR

The Internet runs according to a set of rules (protocols) called the IP Suite. The IP Suite manages the packaging and transmission of data across the entire World Wide Web. Some of those protocols are DNS, for accessing websites using URLs made of common words, [HTTP](http://), which is how websites (servers) and browsers (clients) communicate, [FTP](ftp://), for clients and servers to transfer data, and SMTP, which is how email is sent across the Internet. d

## CHAPTER #05



# PICKING A CMS

A content wizard to do the magic for you

### **Introduction**

Now that you know about all the moving parts of a fully functioning website on the internet, how soon do you think you could put one up? In other words, how soon do you think you can learn to code in multiple languages? Did we mention that it has to be done while coming up with content at the same time?

Thankfully, there are tools for easily constructing your online real estate in a short amount of time (a few hours to a few minutes), known as Content Management Systems. CMSs, for brevity. As the name suggests, a CMS helps you build, maintain and update a site with a focus on the main component - the Content. The CMS handles the rest, with almost all of the functionality configured and deployed behind the scenes. As we will see, CMSs need to evolve constantly with the internet, and therefore necessarily possess modularity, in the form of plugins. Plugins exist for all sorts of

extra automation like auto-SEO or auto-responsiveness, or even a whole e-commerce module. You can view the range once you pick your tools. Other than the innards (code creation), another important facet taken care of by the CMS is the outer appearance. The ability to use templates and themes greatly streamlines the amount of time and choices to a great design. It is not possible to give too much general information about CMSs because each is unique in its own way (except the blatant rip-offs of course), but that's what this chapter is for! Read on so we can help you choose the CMS that will be right for your needs!

## Hunt for the Holy CMS

While there are many CMSs, some paid and some free like Wild Apricot, eXo Platform, and Smartimage, there are three very popular completely free options namely, Wordpress, Joomla and Drupal. Of course, they are only free because there are some in-app purchases, but all costs can be avoided with adequate effort and ingenuity. Other advantages include the fact that they are very powerful in terms of functionality. Their popularity ensures that documentation will be user-friendly and the product will continue to be developed to stay relevant. Free help is available for any kind of trouble after a quick search query, and there's no dearth of paid support. Moreover, the number of themes available will also be proportional to the popularity, a sure-fire customization bonus!

### Drupal

The Drupal Project is completely Open Source, so anyone can download and share it. It is the most powerful and developer friendly CMS out there, and also one of the most secure. No wonder it is used by the official websites of the government of Trinidad and Tobago, the White House and the University of Chicago to name a few. The price of this awesomeness is a steep learning curve and a good amount of technical knowledge and concepts to be grasped. Additionally, as part of the price, this CMS suffers the drawback of having a long installation process, with multiple important intermediate steps from zero to launch. That being said, Drupal is a great choice of CMS for creating and maintaining a top-



Drupal: The cool CMS with the cool logo

notch website with a lot of content, as it is known for its highly structured and efficient content representations. Drupal is additionally known for its capacity to support multiple websites easily within the same interface. So in case you own/manage a brand or a large company with multiple products and projects, or need a complex website with the ability to easily author and publish massive amounts of content, Drupal is a great choice for you. Drupal is functionally extensible with the help of additional pre-written code called modules, which are the equivalent of plugins in Drupal.

Older versions of Drupal were criticized for being a nightmare to customize, because the code is written in the procedural style of PHP4. Instead, the official recommendation for custom applications is the use of hooks, which is considered a bad practice in general because it leads to very messy code that is hard to look into or debug. The latest version of Drupal, Drupal 8, boasts of object oriented programming, a simpler, more intuitive UI for authoring, and RESTful practices which ensure working integrated experiences. If it weren't for Drupal 8, Drupal would even be in this chapter.

## Joomla

Joomla is also an open source content management software that specializes in most areas that are similar to Drupal. While providing most of the advanced features for website development like Drupal, Joomla does so in a more user friendly manner. Though not as steep as that for Drupal, Joomla also has a considerable learning curve for the complete novice. Joomla is

made with PHP and MySQL but in an object oriented architectural design, and using the MVC framework. Its functionality can be extended with 'Extensions', the equivalent of 'Modules' in Drupal. Out of the big three CMSs, Joomla has the least number of extensions, templates and themes, but that doesn't mean it's not enough to choose from. Joomla is easily the most customizable powerful CMS, so with a little know-how you can work wonders!



Joomla: An awesome CMS with an awesome name!

An added bonus with Joomla is the integrated support for multi-user sites like social networks, and back-end heavy sites like e-commerce sites. Drupal and Wordpress require plugins to

properly develop an ecommerce website. Some more nifty features include out-of-the-box Bootstrap integration in the templates for readymade mobile-friendly web pages. Joomla is used by major organizations like Linux (need we say more?) and Harvard University. Joomla also supports free hosting with a free sub-domain as in ‘\_\_.joomla.com’

## WordPress

For the aam aadmi who has heard a thing or two about websites, ‘blogs’ can be synonymous with ‘wordpress site’. Just goes to show how large of the market for easy content publishing they control. WordPress is mainly a CMS for blogging, and has good inbuilt SEO support that can even be enhanced with certain plugins. Although it too can be customized to just about any function with the use of plugins, excess third party code tends to make your site slow. Moreover, without the requirement of a certain degree of code comprehension, the customization isn’t as fine tuned as that on Drupal or Joomla. The admin panel is extremely intuitive (as non-technical as web development can get) and WordPress even offers paid hosting and custom domains. A great all in one solution for people who don’t want to get their hands dirty with code yet need extremely affordable online publishing!

WordPress easily has the largest collections of themes to choose from, but many of the good ones are premium. However it is a small price to pay compared to hiring a good designer! If you are not an organization with highly complex marketing and user experiences (like Nike’s custom UI billboards) or you really don’t want to learn to code, WordPress is easily your best bet! Better yet, read on...

## Bonus: Special Mentions

In today’s world, programming languages are coming close to being as need-to-know as our mother tongue. That said, it is definitely a good idea to learn a bit, if only to stretch your brain. Consider the fact that the big three CMSs are just websites that configure websites. If you take a web developer video course and plough through it like you binge watch your favourite show, in a month you could be building your own custom CMS, streamlined to your needs! Nonetheless, here are some more free options that are especially optimized for certain types of content management.

**Jekyll** - If you purely want to publish content, like a blog, but without comments and databases and all that jazz, Jekyll is the solution for you! If you’ve never seen the terminal, this free software may look daunting at first,

but once you get the hang of it, content publishing will be a breeze! There's good documentation available, as well as a free course on codeacademy for those who need a little more guidance. A good concept to have before using Jekyll is that of a JSON object. Jekyll is literally a gem, because it is written in Ruby. Also because it is blog aware (supports categories, posts, etc) so you can set a custom layout and then begin posting your static pages. What's more, Jekyll also integrates with the free hosting on GitHub Pages!

**Contentful** - Contentful describe themselves as a 'content management developer platform, with an API at its core.' This is a developer friendly CMS (as in, not user friendly), worth your notice if you preference is still leaning towards Drupal or Joomla. It is a great tool for literally managing content and then distributing it, as opposed to creating, formatting and presenting it as well. Editors can enter content into a special web interface for authoring, and the content gets returned as a JSON which can be requested

as and when needed with an API call. They have an SDK for all popular languages, including JS, Node.JS, Ruby, PHP, Swift, Java, cURL. This way you can push content easily to wherever you need to, be it mobile or website, while leaving the designers to do their job. They have different pricing plans for different requirements, and a 30 day trial. They also have a free pricing plan for



Octojekyll: Which came first, Octopress or Jekyll?

enthusiasts, but if you're a professional, maybe you wouldn't mind paying for an automated hassle-free 'one size fits all' automated backend that scales with your needs. Nike installed interactive wall-touchscreens in its flagship showrooms, for users to see social data from previous buyers, and product information. These systems use the Contentful API to deliver their custom content. If you have any unconventional use in mind, head to [www.contentful.com!](http://www.contentful.com/) 

## CHAPTER #06



# PICKING A WEB HOST

Now that you know a fair bit about the front end, you need to delve into the intricacies of web infrastructure. Basically, that part of the www where your site will actually reside.

**T**o run a successful website or blog, you need to ensure that it's served to your audience properly. This largely depends on how good your web hosting provider is. Although not impossible, it is a little difficult to migrate to another service provider after your website is established. Hence, it is important for you to get it right the first

time around. In this chapter, we will examine all the basic concepts which you should know about while selecting a web hosting provider including the steps required to establish online presence.

## What is Web Hosting?

At its simplest level Web Hosting is a process by which your website files are made available to a computer located in a remote destination. Your content will be visible to the outside world only if you host your website properly.

In web parlance, the end-user (your site viewer) is referred to as a client and the computer in which your files are stored is popularly termed as a server. The server will be functioning all the time. You need to transfer your files, databases, images, videos and other materials from your computer to the server. It could be located either in India or in any other part of the world.

First, you have to register a domain name and then purchase server space from a web hosting provider of your choice. The server will run either in Windows or Linux platform which depends upon your requirements.



Establish your online presence today by self hosting your website

If you want to simply create a blog you can buy Linux hosting package. If you want to run ASP.NET files or a site with Microsoft SQL Server, you should choose Windows hosting.

The web hosting packages are available based on disk space and bandwidth. It can be purchased from various providers for a monthly or yearly fee. Some providers offer unlimited resources which you can make use of as long as you host your site with them.

## Windows vs Linux

When it comes to web hosting, you have two choices - Windows or Linux. If you would like to host WordPress, Drupal, Joomla or any other PHP scripts, Linux is the best choice. In fact, Linux servers are fully compatible with PHP based software scripts and databases like MySQL. The Windows hosting will be beneficial if you would like to develop your website on Microsoft technologies with SQL Server or Microsoft Access database. The learning curve related to the management of emails, databases, backups will be easy in Linux than Windows hosting.

You should be careful not to register fully qualified domain name (.com/.net/.org) for free since it will have hidden conditions. You will not only have to shell out extra money for its renewal but also will find it difficult to regain the control of the domain name in future.

## When to use Windows Hosting

If you want to host ASP.NET scripts and software (both readymade and custom) you can choose Windows based hosting platform. The server will be running Windows Server 2008/2012 with Microsoft SQL Server and MySQL database. Some providers will ask you to pay extra for SQL Server databases. Windows hosting can be managed with control panels such as HELM & WebsitePanel.

If you have a blog with free hosted URL like [yourname.wordpress.com](http://yourname.wordpress.com), you will not be able to create your own custom e-mail ids.

Moreover, you should opt for Windows hosting only if you have an intention to run scripts specific to Microsoft technologies. For example, you can install DotNetBlogEngine instead of WordPress without requiring



Host your website in secured servers

database connectivity. If you want to host a small business/portfolio site with WordPress, you can opt for Windows.

### Pros

- Ability to run ASP, ASP.NET, ASP.NET MVC and PHP files/scripts
- Microsoft SQL Server compatibility
- Less downtime

### Cons

- Expensive
- Difficulty to create SEO Friendly URLs
- Inability to use PHP mail
- Shortage of reliable providers
- Lack of unlimited space and bandwidth
- Runs slow during backup time
- Difficult to migrate to Linux

To create SEO friendly URLs for WordPress in Windows hosting, you need to provide specific code in .htaccess file located in the root (wwwroot folder) of the server.

## When to use Linux Hosting

You should purchase Linux hosting package if you want to run a blog or CMS based site, which fetches high traffic. It could be WordPress, Drupal, Joomla or any other PHP scripts. Moreover, if you host your site in Linux, your site will load faster depending upon the server configuration and location.

Nowadays, providers include plenty of resources such as unlimited disk space, bandwidth, e-mail accounts, and MySQL databases along with their hosting plans. However, you should buy a plan based on your requirements.

### Pros

- Simplicity and low cost
- Plethora of service providers
- Can be easily managed with cPanel control panel
- Easy creation of SEO Friendly URL's in WordPress
- Easy application installation via Softaculous
- Full compatibility with WordPress, Drupal, Joomla
- Easy migration

### Cons

- Downtime during updates/reboots

- High risk of DDos attacks
- Overloading of servers

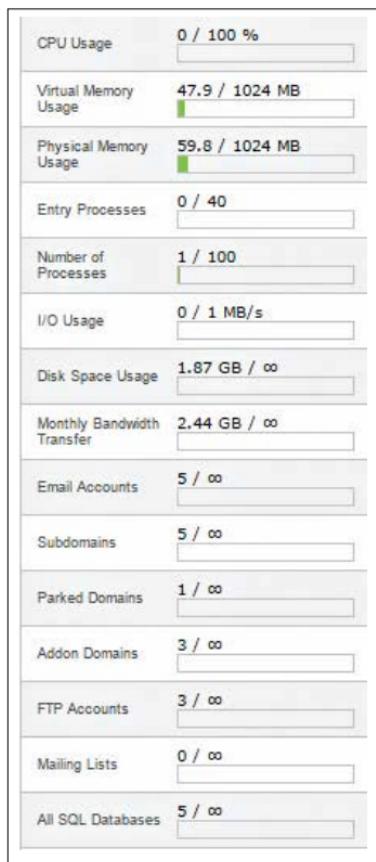
## Types of Web Hosting

Before proceeding to purchase a web hosting package, you should identify the type of hosting mode such as shared, reseller, VPS and dedicated. Let us examine the each hosting mode in greater detail.

### Shared Hosting

In shared hosting, all your sites will be hosted along with other web sites in a single web server. There are also chances that the server in which

your site is hosted will become slow over the period of time. Moreover, you will not be able to install any software by executing an EXE file. The chances of downtime are very high in shared hosting. The shared hosting will be suitable for students, beginners and those who want to start their own blogs because of the low cost.



Shared hosting with unlimited resources in action

### Virtual Private Server (VPS)

If you blog becomes slow due to huge traffic, a reliable Virtual Private Server (VPS) will be useful for you. In VPS, will be able to login to the server remotely and execute commands including the ability to install a control panel of your choice. VPS will be suitable if you are experienced with Linux. If you are not familiar with

FFMPEG is required for hosting video portals with Clipbucket. Be sure to ask your provider about its availability before purchase.

Linux, you should consider a managed VPS hosting package so that your provider will assist you. VPS packages are expensive than shared hosting.

## Reseller Hosting

If you want to provide web hosting service to other people/businesses, you should opt for reseller hosting. You will have rights to create customized hosting plans for other customers with full control over all accounts.

Windows reseller hosting is comparatively expensive than Linux because of licensing costs.



Take advantage of shared/VPS hosting to work with a wide range of scripts

plenty of disk space, memory and dedicated bandwidth. You should note that dedicated servers are expensive per month for both Linux and Windows. You will have to buy softwares like MySQL, SQL Server, and control panel separately.

## Cloud hosting

If you want excellent reliability and stable servers, you should opt for cloud hosting services like Microsoft Azure and Amazon Web Services. The servers will be scattered all over the world and you will be able to host your website at the nearest possible destination, which will provide enhanced performance for your site. However, the cloud hosting will be beneficial for advanced users who have experience in web hosting due to the cost and technicalities involved.

## Importance of bandwidth

When each visitor navigates to your blog/site, the server consumes few resources in the form of bandwidth. The amount of bandwidth consumed

will depend upon the content, number of images and other factors. Once the allocated bandwidth is fully finished, your site will show a message stating that the site has exceeded the limits. Normally, bandwidth is calculated per month basis and is visible inside your control panel. If you purchase a hosting plan with unlimited bandwidth, you need not have to worry about this at all.

### Relevance of server location

The location of your servers plays a crucial role in the performance of your website. If your server is located in India or Europe, your websites will load faster than US-based servers.

However, the performance of your site will depend upon the server configuration. Hence, it is advisable to contact the provider before purchase and ask them questions about the server location and configuration.



Host your website on the cloud without any worries

### How to find a correct web hosting provider?

Before proceeding to purchase a web hosting plan, you should find a reliable provider. The industry has grown considerably over the past 10 years and providers are scattered all over the world. You should search for reviews on Google and social media platforms like Twitter and Facebook. Moreover, [webhostingtalk.com](http://webhostingtalk.com) is a dedicated forum where people discuss about hosting providers in addition to news, tutorials, directory and employment opportunities. You can submit a post with your requirements to get recommendations from other experienced people.

Companies based in the US like Hostgator, Bluehost and GoDaddy have already established their presence in India. If you would like to test drive their services, you can purchase a package for one month to start with. You will find plenty of reviews about all major hosting providers on the web.

While picking a web hosting provider, you should also verify whether the company has provided the complete contact details. Ideally, you should ask several questions before purchasing a hosting package. You should also pick a provider who offers support via live chat to resolve issues quickly. The presence of a comprehensive documentation in the form of knowledge base, FAQs will be ideal to easily find answers to common problems.

## Steps to purchase a web hosting package

You have learned the various aspects related to web hosting. Let us now explore the steps required to purchase a hosting package.

Don't fall into the trap of bogus claims from providers like unlimited disk space, bandwidth and other resources for low cost per year.

### Domain name registration

Firstly, you should register a domain name of your choice directly from registrars like GoDaddy. Nowadays, hosting providers provide free domain registration along with the packages provided you pay one year payment in advance. You should be aware that few hosting providers lock your domain, which prohibits you from transferring the domain name in future. Hence, we would suggest you to buy domain name directly from registrars to avoid disputes.

The screenshot shows a search interface for domain names. At the top, there is a search bar with placeholder text 'Type the one you want here' and a 'Search' button. Below the search bar are two buttons: 'Bulk domain search' and 'Transfer domains'. The main area displays four domain options in cards:

- .com**: Get the domain that will never go out of style. **Rs 199.00\*** (was Rs 469.00). [Register](#)
- .in**: Connect with millions and share the latest trends. **Rs 99.00** (was Rs 469.00). [Register](#)
- .club**: Get organized with .club. **Rs 99.00\*** (was Rs 469.00). [Register](#)
- .store**: Keep your .store open 24/7. **Rs 299.00\*** (was Rs 5,549.00). [Register](#)

Register and host your domain name on the go

### Domain control panel

If you register a domain name, you will get access to a separate control panel where you can modify your contact details and server names. You can

also conceal your contact information from the whois database by enabling privacy protection.

### Hosting sign up

Once you registered a domain name, you need to purchase a web hosting package as per your requirements from a provider of your choice.

To transfer a domain name, generate a secret key and provide correct admin email id from the domain control panel of your existing domain, which needs to be transferred.

### Account information email

As soon as you complete the signup process, you will receive account information e-mail with all the required data such as hosting control panel credentials, name servers, e-mail and database information.

### Modify name servers

The next step is to point your domain to the name servers provided by your provider. It will look like ns1.digitalindia.com and ns2.digitalindia.com. You need to login to domain control panel to modify name servers. You can skip this step if you had purchased the domain name along with the hosting package.

### Uploading files

You have to upload the files via FTP by using the credentials supplied by your provider. You can modify the password by logging into your hosting control panel. If you want to upload files immediately after purchase of hosting, you can do so by using FTP IP address supplied by your provider. If you are using Linux, you should upload all files inside www folder. In case of Windows hosting, you need to upload all files inside *wwwroot* folder.

You can upload files using FileZilla or CoreFTP. You can also make use of web based file manager included with cPanel control panel.

### Installing content management software scripts

If you decide to use CMS like WordPress, Drupal, Joomla, you can make use of Softaculous located inside cPanel. In case of Windows hosting, you can install from the Web app gallery located inside WebsitePanel control panel. Alternatively, you can install scripts like WordPress manually by following the steps given below.

- Download the latest version from [wordpress.org](http://wordpress.org)
- Create MySQL database and user
- Provide the database name, user and password by opening `wp-config.php`
- Upload files to the root folder (`www`) of your hosting account using FTP
- Navigate to your website and complete the setup process

You can upload the ZIP file (after editing `wp-config` file) to your FTP account using FileZilla and extract the contents from the web based file manager.



Install scripts easily from within your control panel

## Resources

- cPanel Demo - <http://dgit.in/cpanelcpdemo>
- cPanel Tutorials - <http://dgit.in/cpaneltutorials>
- WebsitePanel Tutorials - <http://dgit.in/websitepaneltut>
- Webhostingtalk - <http://dgit.in/webhosting-talk>
- DigitalPoint Forum - <http://dgit.in/digitalpointweb>
- Web Hosting Search & Reviews - <http://dgit.in/hostsearch>
- Microsoft official Windows hosting directory - <http://dgit.in/winhostingdir>
- Web Hosting deals - <http://dgit.in/webhostingdeals>
- Hosting Discussion - <http://dgit.in/bostingdis>
- Siteuptime - <http://dgit.in/siteuptime-check>
- Domainwhitepages - <http://dgit.in/domainwhitepages>
- Alertra - <http://dgit.in/alertra>
- WhatismyIP - <http://dgit.in/knowip>

- FileZilla - <http://dgit.in/filezilla-soft>
- CoreFTP - <http://dgit.in/coreftp-soft>

## Conclusion

Finding a correct web hosting provider is always a challenge. If you pick a provider that offers services at a low budget with undeliverable promises, you are always at risk. This is because the provider will likely oversell the server and your site will be affected at one point of time. If you want premium web hosting, you have to shell out extra cash, which will provide ultimate peace of mind. However, there are providers who offer great services at a medium budget. You should keep an eye on the web hosting related forums for deals and offers. You can also ask for recommendations by stating your requirements. We hope that this chapter clears the various concepts related to web hosting including the very first steps required to establish your online presence. Happy hosting. **d**

## CHAPTER #07



# THE TOOLS OF THE TRADE

Building your own website might seem like a daunting task but if you know the following, then your work just got a lot easier.

For those new to web development, the idea of creating your own web page can be daunting. But if you break down your requirements to the bare minimum, it's easy to see that to create a bare bones static webpage with minimum styling, all you need to know

is HTML and CSS. As you start building up on your website and decide to incorporate a few frills, starting with forms and pop-ups and ending with complex web application-like behaviour, you'll most likely use Javascript and its dependent frameworks. In addition, if your website would be handling large amounts of data and will need to store, update/modify and delete said data, you'll need to incorporate custom functionality for your database. This is where server-side languages such as PHP and MySQL and frameworks such as ASP.NET come in.

## **HTML:**

HTML, an abbreviation for Hypertext Markup Language, forms the backbone of any webpage. If you open any website, right click on any point and hit "View Source", you're going to see a complex jumble of HTML code, which is responsible for defining the use and designation of each element on the page, and quite often, their positioning as well. This "designation" of sorts is done with the help of commands, or tags, that tell your browser how you want your document displayed. This is pretty much what marking up your document means. It's also a basic reminder for beginners that HTML-formatted documents aren't that far removed from documents created by a word processing program, which are also basically text. So instead of manually typesetting certain parts of text in paragraphs, and dictating to your processor which elements you'd like in bold and which in italics, you're telling your browser to do the same using a markup language called HTML. Since the early days of web design, HTML has been the standard for professional websites. It stands for Hypertext Markup Language. HTML is the language, or code, used to edit and position the text, images, frames and other web page elements. If you go to your web browser and select View and then Source – the code used to design that website is available for anyone to see. For example, a very basic "Hello World" website might look like this:



A basic "Hello World" webpage

Viewing its source will present you with something like this:



```

view sourcefile:///C:/Users/meghanag/Desktop/Hello.html
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
<title>My Small Website</title>
</head>
<body>
<h1>My first</h1>
<p>This basic document.</p>
</body>
</html>

```

The source code for the "Hello World" webpage

You can see there isn't much to the code. HTML syntax is nested, i.e. your content must be enclosed in a tag that indicates what its function is. Headings are enclosed in `<hx></hx>` tags, where the x can range from 1 to 6, depending on the size of the text, with 1 being the largest. The head and body of the document are enclosed within corresponding brackets, as you can see.

And this is just the beginning. A few more tags, and a designer can create an extremely simple webpage. The more complex the HTML, the more you can do with a site. As we'll see in the next article, there are a ton of online engines that allow people with no prior knowledge of HTML to create their own website. They allow you to create a webpage without using code and the editor assumes what HTML you need and does the work for you

## CSS:

If HTML is in charge of layout and positioning, CSS is what gives elements their styling and look. Seeing as how a lot of computer languages aim to bring modularity and clear demarcation in functioning, the main aim of CSS was to separate the positioning from the presentation, including colours, fonts and layout. This separation improves content accessibility, provides more flexibility and control in the specification of presentation characteristics and most importantly, enables multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file. This allows developers to specify rules in one place, and makes modification in larger codebases even easier. The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

In the example provided above, the earlier rule that was followed was the usage of inline CSS - all the presentational attributes of the HTML

```
File Edit Selection Find Go to Tools Project Preferences Help
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a simple <b>Hello World</b> document.</p>
    <p>To align="center" just to illustrate the convenience of CSS.</p>
  </body>
</html>
```

## An example of inline CSS

were specified within the HTML markup. More often than not, it lead to repeated description of font colours, backgrounds, borders, sizes and element alignments for similar elements. CSS allows authors to move much of that information to another file, the stylesheet, resulting in considerably simpler HTML.

```
File Edit Selection Find View Goto Tools Project Preferences Help
4 <html>
5   <head>
6     <title>Hello World</title>
7     <link rel="stylesheet" type="text/css" href="example.css">
8   </head>
9   <body>
10    <h1>Hello, World!</h1>
11    <p>Here comes the content of the page</p>
12  </body>
13</html>
```

## HTML with an external stylesheet

```
File Edit Selection Find View Go To Tools Project Preferences Help
File Edit Selection Find View Go To Tools Project Preferences Help
 1 body { font-family: sans-serif; font-size: 14px; }
 2 h1 { text-align: center; }
 3 background-color: #f0f0f0;
 4 height: 100px;
 5 width: 100px;
 6 margin: auto;
 7 }
 8
 9 h2 { text-align: center; }
```

## Externally embedded stylesheet

We can see that, though the end effect is similar, it's much easier to apply a constant effect across multiple tags, rather than doing the same in inline CSS, which leads to a tremendous amount of repetition and extreme inconvenience with a larger codebase.

## Javascript:

Though in the last several years, JavaScript is been increasingly used in server side scripting and software applications, going as far as iOS and Android apps, its primary use is that of a client-side scripting language designed for use in web browsers. Initially, its uses were mostly limited to interacting with forms, giving feedback to the user and detecting when they do certain things. `alert()` was used to notify the user of something, `confirm()` to ask if something is OK to do or not and either `prompt()` or a form field to get user input. But it was only after browsers started supporting the Document Object Model (DOM) that Javascript began to take shape into its present form.

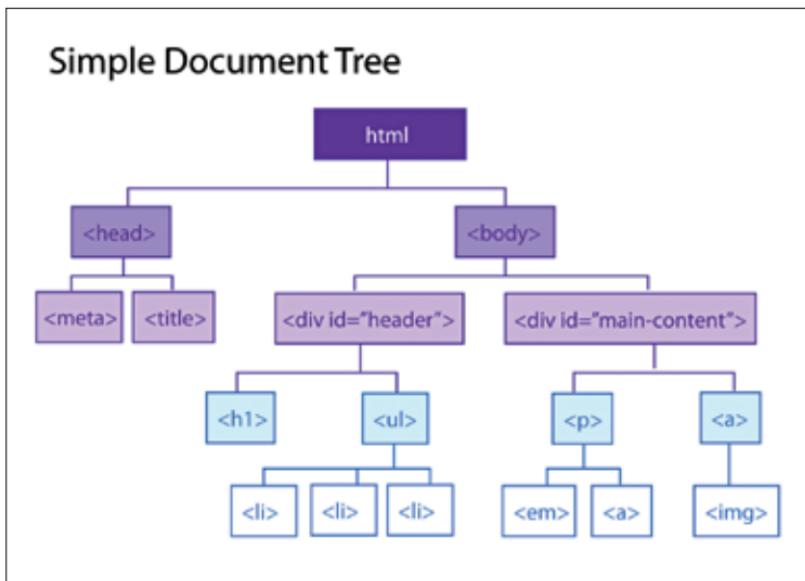
The Mozilla Foundation documents define the DOM as “a programming interface for HTML and XML documents. It provides a representation of the document as a structured group of nodes and objects that have properties and method”. In a very abstract definition, this model was created to provide a convenient structure for programs to change document structure, style and content, with the program being a script in most cases.



A Javascript `alert()` box

Javascript is slightly tougher to master, compared to HTML and CSS. While using frameworks like JQuery allows you to hide behind a layer of abstraction and achieve certain effects in just a few lines of code, you'll most likely end up with a very limited knowledge of Javascript. You'll be able to perform simple rollovers and slideshows, but run into big problems once real DOM manipulation comes into play. But if your primary aim is not to get into webpage/webapp development, but to simply get your page up and running with minimal external effects, you can understand the basics of JQuery and get away with it. But it

is best to understand what tool you're using, so as to avoid getting entangled in a massive mess of callbacks and slow page load speed.



code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere. PHP is used in conjunction with a relational database management system (RDBMS). This has led to a LAMP architecture or its variants. PHP is commonly used as the 'P' in LAMP alongside Linux, Apache and MySQL.



Most web hosting providers support PHP for use by their clients. You can get a basic idea of PHP's popularity by noting that, as of February 2014, almost 82% of the world's websites used PHP as their server side language.

With companies such as Yahoo!

On its side, PHP is a pretty strong winner. But due to security, scalability and open-source development concerns, people have started turning to alternatives like Django and Ruby on Rails.

### **Django (Python):**

Django is a high-level Python Web framework, that according to its website, "encourages rapid development and clean, pragmatic design". So if you think your website will need to handle large amounts of data, Django would be an ideal choice, considering it's meant for complex, database-driven websites. Considering the point of a web framework is to avoid having to reinvent

the wheel, Django takes care of that by providing user authentication, content administration, site-maps, RSS feeds, and other features right out of the box. And the framework has been created in such a way that updating or

replacing provided code with your own custom codebase is pretty simple. If you're still unsure of how scalable or popular Django is, it would be helpful to know Google uses a variation of Django (closer to Python than Django) for its search engine. Other websites like Pinterest, Instagram, Mozilla and The Washington Times make use of it as well.

### **Ruby on Rails:**

Ruby on Rails (RoR) is a framework for building websites. As such, Rails

establishes conventions for easier collaboration and maintenance. Considering that one of key principles of RoR development is convention over configuration (which basically means that the programmer does not have to spend a lot of time configuring files in order to get setup since Rails comes with a set of conventions which help speed up development), it's a pretty great framework to get started with when you have lesser time to chalk out decisions methodically and need a prototype up and running in a short span of time. And while there have been concerns regarding the scalability and speed of RoR applications, there are plenty of high profile websites such as AirBnB, Channel 5 and Groupon which use RoR to power their websites, proving that Rails is a pretty good option for anyone looking for a beginning application.



### **ASP.NET:**

ASP.NET is a Microsoft developed open-source server-side web application framework designed for web development to produce dynamic web pages. ASP.NET Web pages, known officially as Web Forms, are the main building blocks for application development in *ASP.NET*. *.NET* is very big and very powerful. It is used to build large corporate applications. This power comes with an overhead though. For most websites, this is the equivalent of a supercomputer being used to perform simple arithmetic on two numbers. Moreover, ASP.NET was developed in early 2002, meaning it isn't particularly suited to modern web development, which is about agile and iterative development *cycles*. *.NET* is particularly rigid and makes rapid development much harder, which is code for longer development cycles, overblown budgets and overdue projects.

There are two basic methodologies for Web Forms, a web application format and a web site format. Web applications need to be compiled before deployment, while web sites structures allows the user to copy the files directly to the server without prior compilation. Web forms are contained

in files with a “.aspx” extension; these files typically contain static (X) HTML markup or component markup. The component markup can include server-side Web Controls and User Controls that have been defined in the framework or the web page.

### **Databases:**

Any dynamic website that needs to manipulate data would require integration with a database software. The most popular databases in the world operate on a relational model, and are hence called relational databases. Virtually all relational database systems use SQL (Structured Query Language) as the language for querying and maintaining the database. Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is an object-relational database management system, which is also the most widely used RDBMS in the world, followed by MySQL. 



# ONE-CLICK SETUP WEBSITES

When time is of the essence, a 1-click solution is what you need. Here's how we set up a website in no time.

If you're not down with the idea of learning a whole new bunch of technologies just to be able set up a website for your business, there are plenty of alternatives that will let you do the same without any intervention from your side. All you'll be need is a Facebook page for your business, and these automated builders will use the information provided in your business page to create a customised website, without any hassle or haggling.

We'll demonstrate how anyone with a Facebook business page can create a customized website using a new but popular website builder, [www.impress.ly](http://www.impress.ly). An alternative to impress.ly would be [www.dudamobile.com](http://www.dudamobile.com). However, Impress.ly has stood out in this field, despite being a newcomer, because of its unique focus on creating websites functionally equivalent to native mobile apps. Considering how important responsive websites are in today's time, this approach helps impress.ly stand out in the crowd.

After clicking on the "Getting Started" button on impress.ly's homescreen, you'll be presented with the choice of entering your business' Facebook page name, or going ahead with a custom build in case your business doesn't have a Facebook page. Since we're trying to demonstrate a one-click setup in this article, we'll go with the former option. However, the latter is just as simple, and needs you to give out only a few lines of basic information regarding your business.



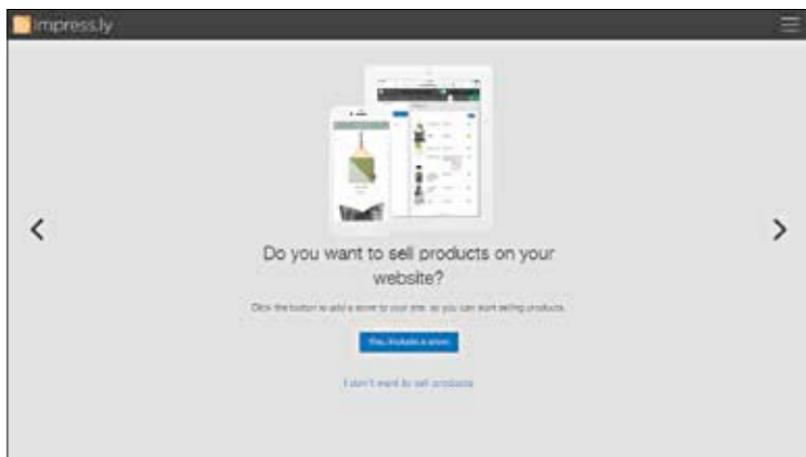
Getting Started

On entering your Facebook page's name, you'll see a drop down menu with a list of business' having pages on Facebook. Choose yours, following which you'll see a screen like this.



Choosing your business' Facebook page

After finding and choosing your business from the drop-down menu, you'll be asked if you are interested in selling products through your website or not.



To store or not to store?

The only point of choosing either option is to allow the builder to create additional pages for a store. If you don't intend to sell merchandise through your webpage and need it purely for promotional purposes, impress.ly works just as well.

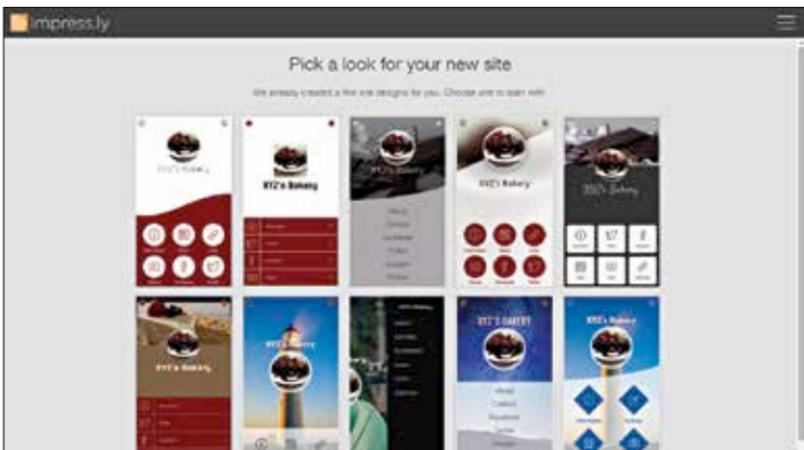
I decided to go ahead with the option of having a store on my website. You'll now come across this page, which basically indicates all the information impress.ly is pulling off of your Facebook page.

Photos, videos, ratings, location and timing related information, impress.ly takes care of it all.



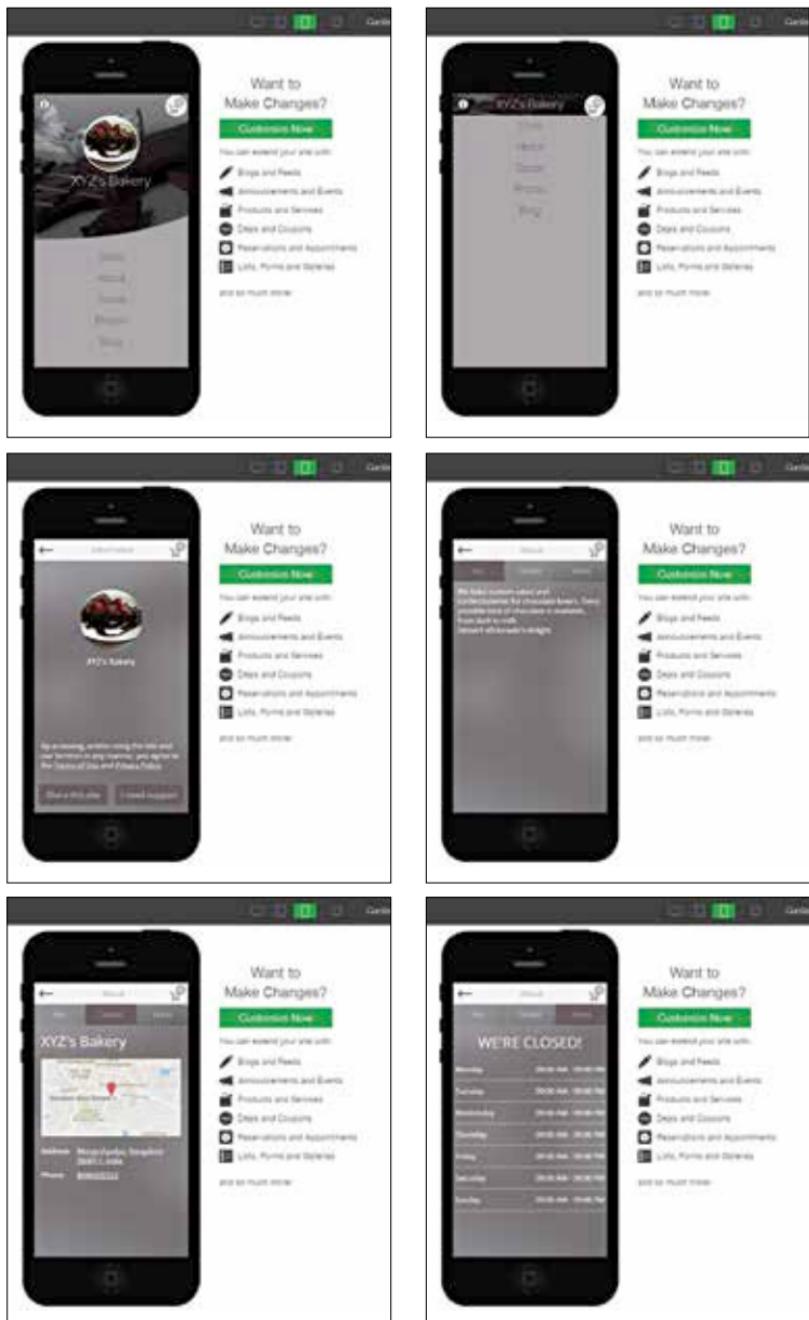
Getting there...

Your website is now ready.



Choose a skin

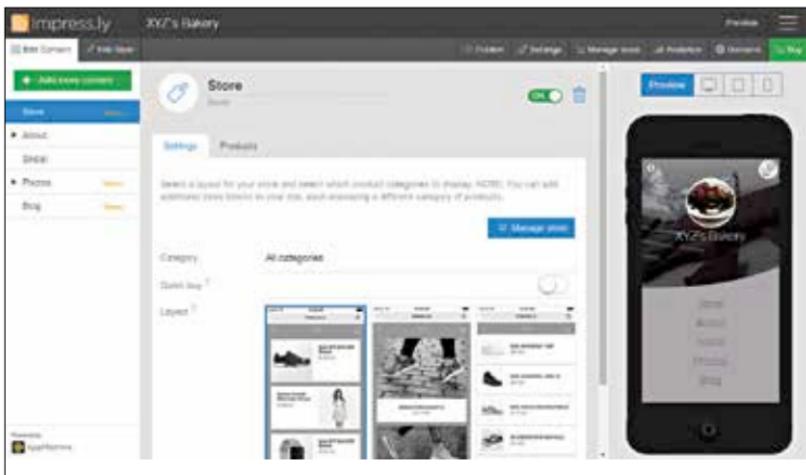
After being able to choose from a bunch of 10 skins, you'll be presented with a pre-publishing preview of what your website will turn out to be



Mobile preview of your website

on desktop, tablet and mobile devices. Toggle between the three views by clicking on the top three device buttons in the top right corner of the page. You can even check how the website will respond to different gestures. But be warned in advance that the skins presented to you aren't very innovative nor very different from one another. But then again, Impress.ly is new, and we're hoping they'll add a few more to the list of skins, or atleast allow users to pay for better options.

Click on the continue button and you'll move on to the screen where you can customise all your content and presentation styles.



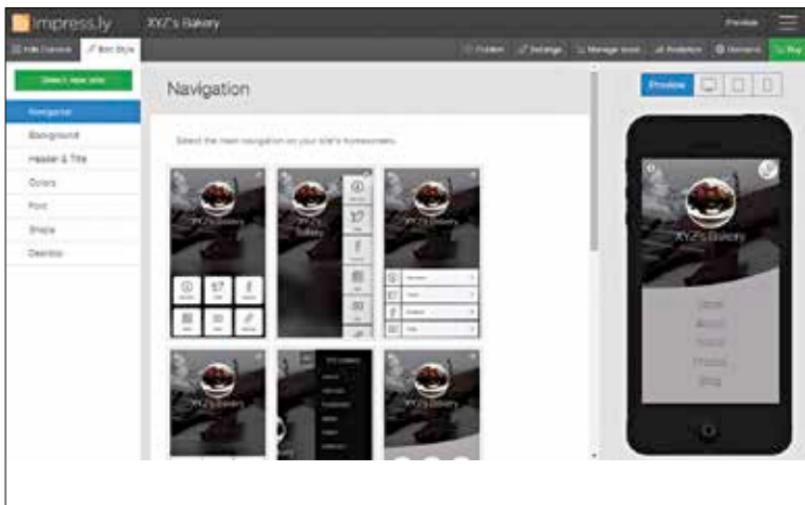
Content editor

On this screen, you can see the variety of options available to customise your content. Change your content, add and remove details - you can pretty much make a whole host of changes to your website, allowing it to feel distinct from your Facebook business page. And if you feel no compulsion to set your website apart from its Facebook counterpart, you can go that way as well.

The “Edit Content” pane will allow you to make changes to each section that will be available on your webpage - the “About” section, your blog, and it allows you to manage links to other social media avenues such as your Instagram, Twitter, Pinterest page and even provides an RSS reader. You can add and remove pictures on your website, making it distinct from your FB Business page. Customize your store - right from the styling to the products you’re selling. Impress.ly really manages to

make things a breeze, and its easy and intuitive user interface sets it apart from its competitors.

The “Edit Style” button directs you to the style customiser, that will let you change your navigation layout, font styles and colour and a whole other host of stylistic aspects.

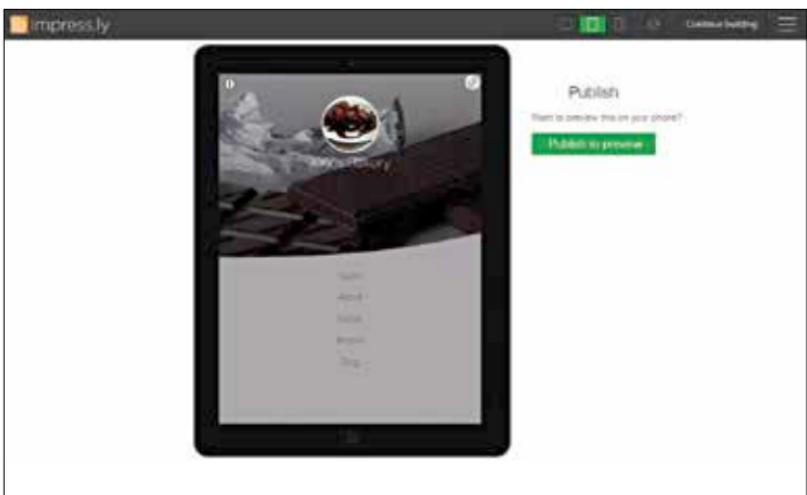


Style editor

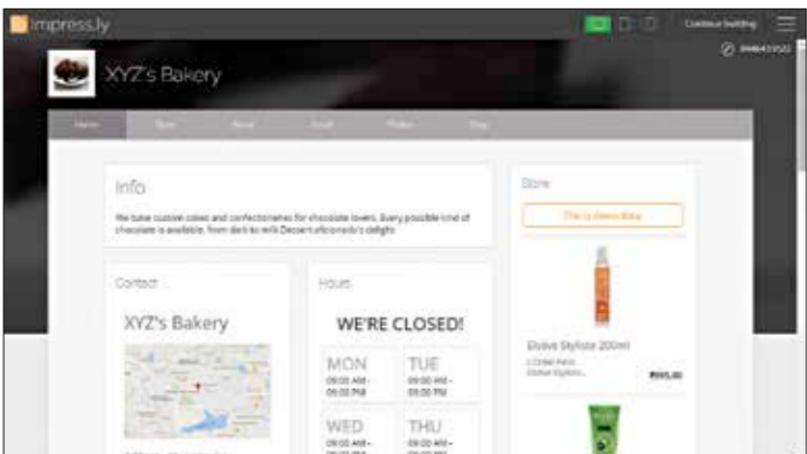
You can start by changing your navigation pane to suit your tastes. Again, all the options provided to you are pretty standard, so don't go looking for something out of the world. Then there's the standard options for changing the background picture, which is by default, your business' Facebook page's cover picture, the colour scheme your website uses, the font colours, the header and title of your page.

If you go to your settings, you'll be able to change the currency using which your store operates, add more members to your website's maintenance team and even use SEO tracking, which is inbuilt into your page during the time of creation itself.

The “Manage Store” option presents you with a slew of options to create a fully-functional store. You can keep track of orders placed, choose to offer discounts and customise your payment provider. While this might not be a complete management solution, it's surely a great alternative for a first-timer, and reduces the headache that would come with setting up a website from scratch, with a store to boot as well.



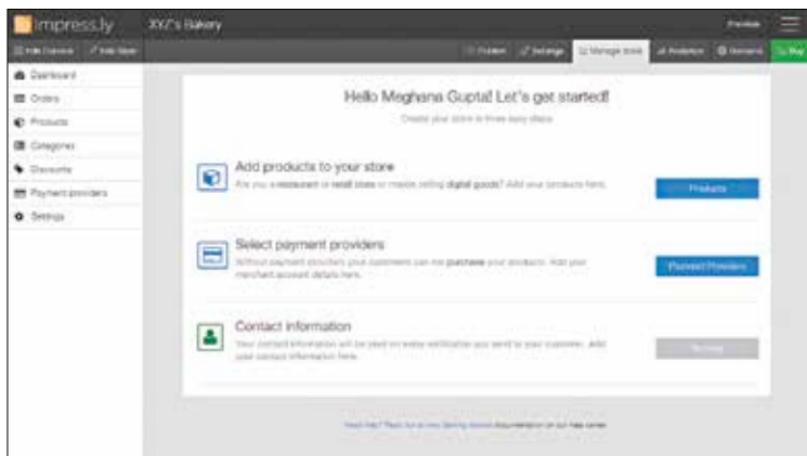
Tablet view



Desktop view

All in all, you get the whole package with impress.ly, though the fact that this website builder hasn't been on the market for very long means it hasn't rolled out a whole list of features yet.

What does help Impress.ly stand out in the crowd is that they provide an app that gives you full mobile editing ability. It's as easy to use as the desktop editor, if not easier. It's a step forward for the industry. Moreover, the fact that the generated website follows mobile-first principles means that they're in



Store manager

it for the long haul, especially after Google has started pushing up websites with responsive and mobile websites higher up in their search results.

One downside to Impress.ly is its “distinct” pricing structure. You can build your site while exploring the options and features and not worry about payment. Only when you go to publish your site will you be required to sign up for a paid plan. While this means Impress.ly offers no true free plan (as many other site builders do) you’ll at least be able to test all the style options and features without entering your credit card info.

There are other alternatives to impress.ly, and for anyone looking for a one-click setup with minimal hassles, this is the perfect gateway to your own functional website. **d**

## CHAPTER #09



# DIY - ASSISTED SETUP

Explore the ridiculously easy world of  
drag and drop website builders

While you can learn to code and build a website on your own with the help of what we are going to cover in Chapter 12, let's face it, getting your hands dirty with code isn't everyone's cup of tea. Luckily, there are a few cloud-based web development platforms like Wix.com, Weebly, Squarespace, Jimdo and uKid that don't require one to have any prior knowledge of web programming and development. In this chapter, we will demonstrate building a website using Wix. The other drag-and-drop website builders we've listed above are pretty similar in terms of functionality but we've chosen Wix just for the sake of illustrative purposes.

## Getting started with Wix

Wix offers free and easy-to-use website builder that allows one to create remarkable websites without spending much time. As we have mentioned already, there are no coding or technical skills required when it comes to building websites using Wix website builder. There are over a hundred templates you can choose from, or you can build and design your website from scratch using a blank template. We will talk more about these templates in our subsequent sections. Wix also gives the option to opt for a premium plan, which unlocks many features and benefits that are normally inaccessible with the free version. Some of the advantages include removing Wix ads, connecting a domain name, additional bandwidth/storage, etc.

## Signing up

As we have already figured out some of the basic things about using Wix, let's proceed further to create an account on the platform and deep dive into the ocean of a stress-free website building. The procedure to sign up with Wix is pretty simple and straightforward. All you need to do is follow on-screen instructions.

In order to sign up with Wix:

- Go to <http://www.wix.com>.
- Hit "Sign In" at the top right corner.
- Click "Sign Up" right beside 'New to Wix?'
- Enter your Email ID.
- Choose your password.
- Hit "Sign Up."

## Editing an existing template/website

Wix has many templates to choose from different categories. Some of the popular categories include Business, E-Commerce, Blog, Music, Photography, etc. Working on these existing templates mean Wix has already done all or at least most of the essential work on your behalf. Now all you need to do is make some final changes and add things you find necessary. Meaning, all these templates come preloaded with the content inside. For example, a business template may contain the information that may not be suitable for your company website. Hence, you can simply replace the existing information with text, pictures, and logos of your business without worrying about site's fonts, columns, width, etc. Soon, we will talk about all the fundamental controls that make building or editing site content easier.



There are many features available to add to your site pages

For the sake of a demonstration, we built a sample website based on an existing template. Make sure that you are already signed-in to a Wix accounts.

**Step 1:** As soon you open Wix homepage, you will be redirected to <https://www.wix.com/my-account/sites/>. Meaning, all your published and unpublished websites can be found under this URL to manage. First timers will see a blue button – ‘Create Your Website.’ Upon clicking it, you will be given several options to choose from with a message ‘What kind of website do you want to create?’ Some of the options are Business, Designer, Blog, Beauty & Wellness, Portfolio & CV, Accommodation, and more.

**Step 2:** How about creating an e-commerce website? Upon clicking on the ‘Online Store’ website category, you will come across hundreds of e-commerce site templates. Before finalizing or editing any template, you also get to explore its looks and working as a preview. In the event that you

want to know more, you can find the option 'Info' by hovering your mouse cursor over the template.

**Step 3:** Let's go with a template called 'Home Tech Store.' After hitting the 'Edit' option, it took us some 10-15 seconds before we could get inside Wix Website Editor. It may vary depending on your Internet connection. The best part is it will not make you feel like meeting some stranger. As a newbie, if you have worked with WordPress, Joomla, or Drupal, you must be knowing how exhausted it feels even while making some simple changes to menu bar position, text size, or fonts, etc. It is not the case with Wix. Why? Because you will find everything in front of your eyes way before your website goes live. As a result, you will always know how your site looks in the real-time and how will it look like if some changes are made. Isn't it cool?



Playing with text and fonts in the real-time is as simple as you are editing some MS Word document.

**Step 4:** We are going to create an online Digit store where visitors will get to know about last month's Digit issue. Not just that, they will also be able to buy the magazine directly on Digit.in. As planned, we replaced the featured product image on the homepage with Digit magazine. All we had to do was open Wix Website Editor > click on the existing image > select 'Change Image' > choose 'Upload Image.' You can follow this procedure for any picture or logo.

**Step 5:** Replacing the image was not enough as we also had to put in an appropriate product description. Hence, our next job was to change the text to something we could easily relate to Digit's August 2016 edition. All we had to do was click on the existing text > choose 'Edit Text' > Enter the new text. We also had to hyperlink the 'Preview' and 'Buy Now' button. We followed this process that you need repeat: Click on the text or button



Editing existing content and replacing the same with your content is no big deal

you want to hyperlink > click on the 'Change Link' icon, which is next to the 'Change Text' option > insert the URL where you want to send your audience.

We followed the similar processes for all other objects and content we edited. We will look at what more we can do in the next section where we will design a website from scratch with the help of a blank template. If you want to visit the website we created, go to <http://techtsp.wixsite.com/buy-digit>. You can, of course, connect your sites to your domain names. However, that option unlocks if you prefer to go with a premium plan.

## Making a website from scratch

In this section, we are going to build a website from scratch by making use of a blank template. We must remind you, a blank template on Wix resembles a lot to an empty haunted house, and you just cannot let yourself settle down before filling it out with all exciting and essential controls that we are going to talk about.

**Step 1:** Let's go to <http://www.wix.com>. As we already have a Digit store e-commerce website in our account, we will longer see a message 'Create Your Website' on display. Rather, as we had mentioned earlier, we get to see our previous website listed over here. At the top right, there is an icon that lets one create a new website. Upon clicking the option, we landed on a template selection page. This time, however, we don't need to worry about any category as we are building a website from scratch with no ready-made content present inside.

**Step 2:** In order to start designing a website from scratch, go to Blank Templates. Despite choosing this option, some of the templates present inside are not completely blank. Wix has done it a lot easier with pre-determined

layouts and other essential things. However, as we are aimed at making a website without any help offered by Wix, we are going to choose the template called 'Start from Scratch.' Do remember that it's darn scary!

**Step 3:** As soon as you get inside, the Wix Website Editor will prompt you to set a site name as well as the URL. Whatever name you select in a free version, it will appear at the end of the URL. In our case, we decided to go with the name 'techtsp.' Hence, the website URL becomes this: [`http://techtsp.wixsite.com\(techtsp\)`](http://techtsp.wixsite.com(techtsp)). You can anytime change the website URL. All you have to do is go to [`http://wix.com`](http://wix.com) > select your Website > click on 'Manage Site' > Go to 'Manage' in the 'Site Address' > makes the changes.

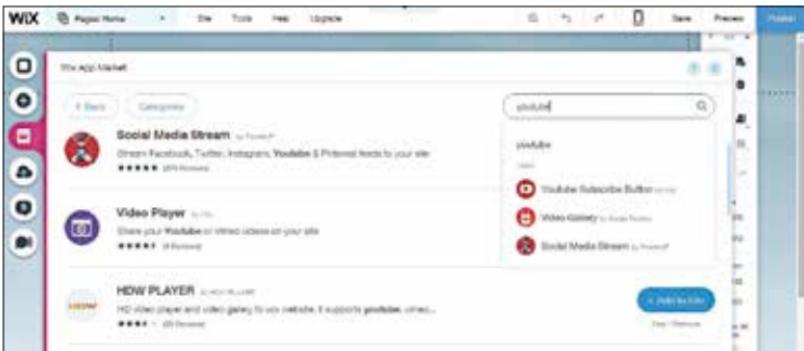
**Step 4:** If you pay close attention to this blank template, you will figure out that the template is basically separated into three parts - Header, Page, and Footer. It makes it a lot easier to prioritize where your content is going to be positioned comfortably. The header will be comprised of site logo and menu bar. The page will act as a body. In the end, the footer will contain a short bio and logo with a link to your about or contact page.

**Step 5:** Enough said! Now it's time to explore what we can add to our website, which is literally blank at this moment. There are different controls located on the left. Our first job was to add a decent background to our site so that we could get rid of that scary white background. In order to change the site background, click on the 'Background' icon at the left > select 'Image' > choose 'Upload Images' > select the uploaded image > hit 'Change Background.'

**Step 6:** Later on, we added a site title in the header. As we have already seen earlier, all you have to do is go to 'Add' option on the left > select 'Text' > choose 'Site Title' > position the text in the header. However, playing with the site background and title is not going to be enough. Hence, we added a couple more pages. If you want to add a new page, click on the 'Pages' at the top left > select 'Add Page' from the drop-down list > rename the page to be added to a menu bar.

### NOTE:

Changing the site background will not modify the background of site pages. The change will only be reflected to site header and footer. By default, your page will remain as it is. However, in the event that you want the newly uploaded background to be reflected on your site's homepage or other created pages, go to that particular page > choose the 'Change Page Design' option > select 'No Color.'



You can add any of your favorite videos and enable autoplay as soon as the page loads

**Step 7:** Similarly, you can make background and design changes to your newly created pages. We created two pages – ‘About’ as well as ‘Contact’. When it comes to adding content to the contact page, the most important thing is to have a contact form. However, it’s not something tedious when it comes to adding a contact form in Wix. Go to ‘Add’ option on the left > select ‘Contact’ > Choose your contact layout. All you have to do is simply input your email address where you want to receive messages sent through this contact form.

In addition, there are many more exciting things that you can add to your website. For example, we added a YouTube video to our homepage by going to ‘App Market’ > searching YouTube in the search box > add ‘Video Player’ > Insert a video URL. There are so many things that you can explore. However, as the matter of fact, whatever we have mentioned in this Fast-Track is just never going to be enough. **d**



# BUILDING A WEBSITE THE HARD WAY

There are very few things as rewarding as building your own website and a completely hands-on approach is all the more worth it.

**N**ow before you start listening to the Rocky theme song on loop getting yourself pumped up for what's to come, here's a little disclaimer. We aren't going to be completely coding a website from scratch. If it wasn't obvious, then building a CMS along with a framework and coding Javascript libraries is way out of the scope of this FastTrack. In fact, we've written individual FastTracks for building a CMS, exploring Javascript libraries and frameworks, so squeezing them all together within one little book is quite, you guessed it, ridiculous!

So what are we looking at? WordPress, of course! What's better than getting your hands dirty with the most popular CMS on the planet. WordPress is inherently made simple so as to help anyone and everyone get a website running in absolutely no time. In fact, many hosting services have a 1-click installer / script for WordPress. Which isn't what we will be doing in this chapter. We're going all the way in order to get a better understanding of the entire process. So let's begin with the bare essentials.

### Get a web server up and running

There's an easy way and a hard way of doing most things. We're going to stick to the easy way here. Setting up a web server is no ordinary task, the amount of configuration required to ensure a smooth and easy functioning web server is enough to ward many away from Linux for good. Yes, you read that right, Linux. For the sake of stability we're going to need a web server running on Linux and that would mean getting to know how Linux



XAMPP is an easy-to-install web server stack

works and knowing your way around a terminal. But the web server we're looking at, i.e. Apache, runs just fine on Windows and there's absolutely no hiccups when you move your web site's source files from a Windows Apache installation to a Linux Apache installation.

If you really do want to get your hands dirty with the web server installation as well, then here's what you need:

- Operating System - Linux or Windows
- Web server - Apache or NGINX
- PHP version - 5.2.4 or greater
- MySQL version - 5.0.15 or greater or any version of MariaDB

(Optional) Apache mod\_rewrite module (for clean URIs known as Permalinks)(Required for WordPress Multisite)

If possible, get PHP version 7 as WordPress performance nearly doubles with the latest PHP release.

If you're content with a slightly easier approach then get an easy to install web server stack like XAMPP. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). The (M) used to stand for MySQL but as of late 2015, XAMPP moved on to MariaDB since it's a lot more open when it comes to open source contributions. Moreover, MariaDB performs significantly better than MySQL, however, with WordPress you're only going to see a performance increment of 3-5%. This scales to slightly higher numbers as your website grows.

Once you've installed XAMPP, you should know a few simple things about working with the software. For starters, XAMPP needs to be installed in a root directory of any partition. Inside the installation directory, you'll find a folder titled 'htdocs'. This is where all the files that need to be uploaded are stored. Once XAMPP is installed, you will find that the 'htdocs' folder is not empty. There exist a few pages that'll make it easier for you to configure the web server.

In a browser, type "127.0.0.1" or "localhost" and you should see a welcome page for phpMyAdmin. There are a few things you'll need to modify to get XAMPP working properly on your machine if your OS isn't a fresh install. This doesn't mean that you should format your drive just to install XAMPP, troubleshooting XAMPP errors are quite easy thanks to extensive documentation. However, there are two errors which are very common with XAMPP installs on older OS installations. This is because certain resources like open ports are limited. If you do come across an error message that mentions port 80 being unavailable then open [http.conf](http://conf) and change port

80 to 8080 and the error message should go away. Find the following line and make the modifications:

***Original***

Listen 80

ServerName localhost:80

***Modified***

Listen 8080

ServerName localhost:8080

You can change the port to any valid port number as long as it is open. And to check if your modification is working fine, head over to <http://localhost:8080> (replace 8080 with whatever port you chose). If the url doesn't work then you might have to change the SSL ports as well, open \xampp\apache\conf\extra and make the following changes:

***Original***

Listen 443

<VirtualHost \_default\_:443>

ServerName [www.example.com:443](http://www.example.com:443) or ServerName localhost:433

***Modified***

Listen 8014

<VirtualHost \_default\_:8014>

ServerName [www.example.com:8014](http://www.example.com:8014) or ServerName localhost:8014

Check the url again. If XAMPP continues to act up then in the XAMPP Control Panel go to Config>Service and Port Settings>Apache. Input the Main Port and SSL Port as per your earlier modifications, i.e. 8080 and 8014 in this case. Everything should be working fine.

Also, XAMPP will mention which service is currently latched on to port 80 and shutting that service (only if it's an unnecessary service) should allow you to bypass the error, if not simply click on the "Netstat" option to figure out which service is the culprit.

The other error is related to PHP memory size and upload limits. Certain plugins are quite large in size so your default installation might not allow those plugins to be installed. Simply open up "php.ini" or "php5.ini" in your XAMPP directory and look for the following values and modify them accordingly:



Input the Main Port and SSL Port as per your earlier modifications

Upload\_max\_filesize = 1000M

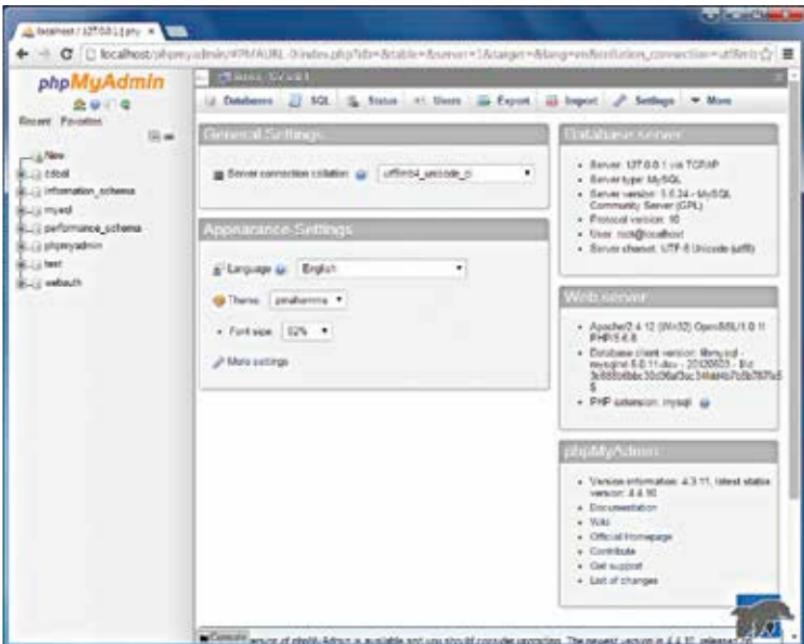
- post\_max\_size = 2000M
- memory\_limit = 3000M
- file\_uploads = On
- max\_execution\_time = 180

These are large sizes and should be sufficient for development purposes but when deploying your site, tone these numbers down to whatever limits your hosting currently has.

With Apache and MySQL/MariaDB set up, you are ready to start developing.

## Setting up a Database and a User

Before you proceed, you need to create a database and a user for your WordPress installation. Navigate to your XAMPP installation via a browser by typing <http://localhost> and open phpMyAdmin. Under “Databases” create a new database under any name and ensure that the collation is anything within the “utf8\_” series and, if you don’t find your language, then choose “utf8mb4\_general\_ci”. Similarly, under “Users” create a user with access to just the newly created database. For the sake of simplicity, you may provide “All Privileges” to the new user. Do note the following for your WordPress installation:



Access the phpMyAdmin page via your browser

- Database Name
- Database Username
- Database Username Password

## Installing WordPress

Head over to <https://wordpress.org/download> and get the latest .zip file for WordPress. Extract the contents of the file into the “htdocs” directory within XAMPP. You may keep the existing files within the htdocs folder or if you prefer, move them to a separate folder before unzipping the WordPress installation files into the directory.

Now you can either edit the wp-config.php file manually or navigate to <http://localhost> in your browser to proceed with the installation. If you are editing wp-config.php then find the following fields and modify them according to the previous steps.

- DB\_NAME - The name of the database you created for WordPress
- DB\_USER - The username you created for WordPress

- DB \_ PASSWORD – The password you chose for the WordPress username
- DB \_ HOST – We used localhost, if necessary append the port name after a colon
- DB \_ CHARSET – No need to modify this
- DB \_ COLLATE – Leave this blank

## If you have a URL

To ensure a smooth transition from your local web server to your deployment, we'll be setting up the WordPress installation with the URL you have in mind. Let's assume that to be <http://www.yourname.com> for now. You now have to modify your hosts file so that all requests to the URL <http://www.yourname.com> are re-directed towards <http://localhost>, this kind of re-direction is handled by the hosts file which can be located at \Windows\System32\Drivers\etc\hosts. The hosts file will have no extension and you'll have to add the following line:

127.0.0.1 <http://www.yourname.com>

127.0.0.1 is the IP address for <http://localhost> which is how the re-direction takes place.

## ... Installing WordPress

Before you proceed, ensure that you have the database username and password. All you have to do now is simply type in 127.0.0.1 or your url which you entered in the hosts file and you'll be taken to the WordPress install screen. Fill in all the fields and check the option that says “Disable websites from indexing this page” and click on submit. Your WordPress website is now up... in the most barebones manner possible.

## Customising your WordPress website

Now that you're a WordPress guy/gal, you need to realise one thing – there's a plugin for everything! You'll get this once we dive deeper into customisation. For now, let's start with customising the appearance of your WordPress website. And for that you need to login to <http://www.yourname.com/wp-admin> with the user account name and password you generated during the installation phase (not the database username and password). You'll land up on the admin control panel with which you can customise every single aspect of your website. If there is any option that you do not understand, feel free to toggle it and figure out what it results in. Or you can head over to <https://codex.wordpress.org/> for a more in-depth explanation of each default option.



Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="wordprz4_wp"/>	The name of the database you want to run WP in.
User Name	<input type="text" value="wordprz4_wpuser"/>	Your MySQL username
Password	<input type="text" value="yourpassword"/>	...and your MySQL password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost does not work.
Table Prefix	<input type="text" value="wpxm_"/>	If you want to run multiple WordPress installations in a single database, change this.

Entering your database details into WordPress

Do note that as you add plugins and themes, you might end up seeing a lot more options than before, these will be described in the plugin/theme documentation and not the WordPress Codex.

## Customise looks

There are numerous websites out there offering WordPress themes. You can even browse the free ones available within the WordPress themes menu option in the admin panel. You can either take the easy way and install a theme and get this out of the way or you can customise portions of the theme to your liking.

We aren't going to tell you which theme to get for your website but you should ensure that it is responsive and the central style is prominent, be it a portfolio, a blog, an ecommerce shop or any of the other common styles.

Once you've installed the theme, you can head over to Admin-panel > Appearance > Customise to modify some of the readily available elements of the website. These options may be severely limited when it comes to free themes. So in case you get a premium theme for anywhere between

USD40-60, then you can customise them even further with ready-made templates and colour palettes to choose from. But what if you want an extra level of customisation?

This is where you should make use of the Child Theme functionality. When you modify theme options, the options are written to a database so when you upgrade your theme to a newer version, those options are preserved. However, code modifications will not be present. This is where a child theme comes in. A child theme inherits everything from a parent theme and you can make whatever modifications you want to the child theme and when you upgrade to a newer version of the same theme, your code modifications are preserved since child themes are a separate entity.

Making a child theme is very easy, create a new folder and you can name it parentthemename-child for the sake of simplicity or you can give it whatever name you want and then create two files within the folder.

style.css

functions.php

Open style.css and add the following lines to let WordPress know which theme will be inherited by the new child theme.

- `/*`
- `Theme Name: Twenty Fifteen Child`
- `Theme URI: http://example.com/twenty-fifteen-child/`
- `Description: Twenty Fifteen Child Theme`
- `Author: Jane Doe`
- `Author URI: http://example.com`
- `Template: twentyfifteen`
- `Version: 1.0.0`
- `License: GNU General Public License v2 or later`
- `License URI: http://www.gnu.org/licenses/gpl-2.0.html`
- `Tags: light, dark, two-columns, right-sidebar, responsive-layout, accessibility-ready`
- `Text Domain: twenty-fifteen-child`
- `*/`

You should replace the field with anything that you wish to but do ensure that the template line points to the directory name of the theme you wish to inherit. We repeat, this has to point to the directory name and not the theme name. Also having more than one child theme for the same parent theme might confuse WordPress so it's best to stick to one.

The functions.php file is next. We use it to enqueue the parent and child theme stylesheets. So always start with the following.

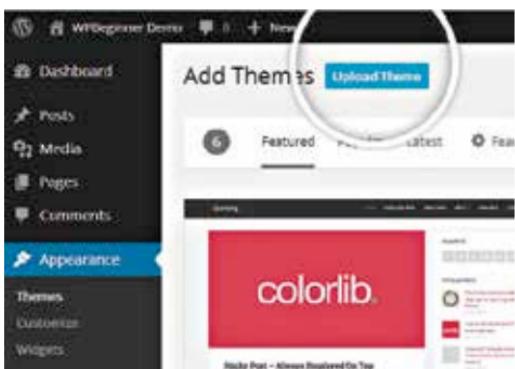
```

• <?php
• function my_theme_enqueue_styles() {
•
•     $parent_style = 'parent-style'; // This is 'twentyfif-
teen-style' for the Twenty Fifteen theme.
•
•     wp_enqueue_style( $parent_style, get_template_directory_uri() . '/style.css' );
•     wp_enqueue_style( 'child-style',
•         get_stylesheet_directory_uri() . '/style.css',
•         array( $parent_style ),
•         wp_get_theme()->get('Version')
•     );
• }
• add_action( 'wp_enqueue_scripts', 'my_theme_enqueue_styles' );
• ?>

```

Once you've done this, zip the child theme folder and upload it via the "Upload Theme" option within the Add Theme menu item under 'Appearance'. It should pop up in the theme options installed on your WordPress blog, all that's left to do is to activate it.

Now if you wish to modify any CSS element in your main theme, just add the selector in the style.css file of your child theme. Any modifications made here will be enqueued and will be reflected in your WordPress installation. Even if the file isn't part of the CSS or is some sort of dependency then you can still override it with the child theme. Simply drop the modified file into the child theme directory while maintaining the same folder hierarchical structure. If for some reason you can't figure out why your new modification isn't showing up then you should refer to this image to figure out how WordPress decides which file to prioritise over another.



Upload customised themes onto WordPress



WordPress template hierarchy looks something like this

## Moving it all to production

Once you are done creating a masterpiece of a web site you might want to show the entire world your creation. And this is where migration comes into the picture. While you might not have added a lot of posts, your database will be quite lean. However, we're going the extra mile and letting you know what all needs to be done to migrate from your local development machine to an actual web host.

## Create a database dump

Start up by firing up phpMyAdmin in your XAMPP control panel. Locate the main database that was created for your installation. The default prefix

Table	Action	Rows	Type	Collation
wp_commentmeta		0	InnoDB	utf8_general_ci
wp_comments		0	InnoDB	utf8_general_ci
wp_links		0	InnoDB	utf8_general_ci
wp_options		139	InnoDB	utf8_general_ci
wp_postmeta		0	InnoDB	utf8_general_ci
wp_posts		0	InnoDB	utf8_general_ci
wp_terms		0	InnoDB	utf8_general_ci
wp_term_relationships		0	InnoDB	utf8_general_ci
wp_term_taxonomy		0	InnoDB	utf8_general_ci
wp_usermeta		23	InnoDB	utf8_general_ci
wp_users		1	InnoDB	utf8_general_ci
11 tables		133	InnoDB	latin1_swedish_ci
			Sum	
	With selected			

Export your entire Database

is “wp\_”, however, if you can’t seem to locate it, then open up the database and look through the tables. If you find any of the following table names after the prefix, then you’re in the right place.

Now click on the Export tab in the top menu bar and select the database on the left. Ensure that all tables within the database are selected and then export. You should get a .sql file named after the database.

Open the admin panel for your host and follow the steps to import a new database. There are a lot of variations between different hosts, so if you get to choose phpMyAdmin, then click on the import tab and select the file from your PC and upload. If not refer to the instructions provided by your host.

## Move all files

Now open up the “htdocs” folder we’ve been working on for so long and move them to your web host. You can either use a client like FileZilla since it checks the integrity of each file before being transferred or you can simply upload an entire directory. Again, there are variations between individual web hosts so no one method applies to all. Which is why FileZilla is the easier option.

Download FileZilla and run it, now check the email you’d received when you’d purchased your web hosting space. That email will contain the FTP url, username and password. Simply use these to login and you’ll end up in the root directory of your web hosting space. This is akin to the htdocs folder and every single file from the htdocs folder needs to be transferred here.

## Point your URL

Now, if you’ve purchased your domain from the same provider as the web host, it’ll do all the job of linking the domain to the hosting. However, if you’re looking at two different providers then refer your hosting email again. You will find a couple of nameservers mentioned. Open up the admin panel of your domain registrar and replace the existing nameservers with the ones pointing to your hosting. Allow a couple of minutes of a few days for DNS tables to reflect this new change. Oh and lastly, open up your hosts file and remove the redirection entry. Otherwise, you’d still end up at your local installation.

That’s about it. You now have your very own installation that you’ve tirelessly put up. We guess now would be the best time to mention that all of this could have been done with just one click. But hey, the learning experience will help you out with a lot of the troubleshooting. 

## CHAPTER #11



# SECURE YOUR WEBSITE

For if you don't, bad things may happen.  
Better safe than sorry, right?

**H**aving gone through the trouble of setting up your own website, it's now imperative that you don't let all that hard work go to waste by not implementing some sense of security into it. Whether it's a Wordpress blog setup on a whim or business portal through which financial transactions take place and user information is aggregated, you need to take steps to protect your website from nefarious elements on the Web. Maybe your obscure blog -- started on a whim -- has a couple of blockbuster posts, goes viral, attracts lot of traffic and gets hacked

or defaced by punks posing as hackers. Worse, what if sensitive user data and financial transactions taking place on your online business gets leaked or falls into the wrong hands? Our increased online presence means digital dangers have a far-reaching impact that could potentially hurt you in the real world. Therefore, in the 21st century, building any sort of online presence from scratch and not protecting it in any way is just madness. You're just inviting trouble.

Don't be that guy (or gal). Learn to protect your digital universe, however inconsequential it may be. While nothing can guarantee your online assets 100 per cent security, you can try to make life as difficult as possible for potential hackers or online thieves trying to deface, bring down, or break into your website. Here are some essential steps:

### **Hide your footprint**

One of the easiest ways to secure your website is to make it invisible to prying eyes. Imagine a thief trying to break into a house guarded by a lock that he can't recognize. Only a genius, badass master thief (one in a million!) could successfully break in, most "thieves" would just turn around and bolt away, cursing their luck. Similarly, by making it difficult for people to guess your website's CMS or server details, you'd make your website that much more difficult a target to break into.



Some details are best hidden, under a 'big rock' if possible!

Similarly, while configuring your website from scratch (as we covered in some of the earlier chapters of this FT), change default settings. For example, if you're running a Wordpress.org installation on your web server, change the default username from "admin" to something else... that's more obscure. Change the default URL for admin login, as well. If you're making a LOT of changes to a CMS's default installation, don't leave sensitive stuff in the comments section of the code – as anyone can read it easily while viewing the source of your web page through a web browser. For WordPress users (which is what a large number of self-hosted independent blogs and websites consists of), check out the Hide My WP plugin from Code Canyon (<http://dgit.in/HideMyWP>) which is an absolute must have for WordPress site owners, as it provides a whole host of features to make your website's CMS hard to guess – it's totally worth the money. And that's a great start to minimizing your website's recognizable footprint.

### One site, one server

Real the temptation is, resist it you must. If Yoda wasn't a Jedi Master

**Unlimited Web Hosting**  
Web Hosting you can rely on!

- Unlimited Web Space**
- Unlimited Email Accounts**
- Unlimited MySQL Database**
- Host Unlimited Websites** HOT

100Webhosting.com

### Unlimited Hosting

Host your site with Unlimited possibilities

- Unlimited Web Space
- Unlimited Bandwidth
- Unlimited Emails
- Unlimited FTP Accounts
- Unlimited Sub-Domains



It's a trap!

but a Web Master (see what we did there, eh?), that would be his advice when it comes to inexperienced webmasters getting swayed by the spell of “unlimited” hosting space. What this means is that you should never (and we mean never ever) host multiple websites on the same web server, as it’s a virtual kiss of death when it comes to protecting anyone or all of the multiple websites in question.

Let’s just say it’s easier to manage an infected host server when there’s only one website hosted on it. If there are multiple websites, based on different content management systems, all hosted on the same server, you’re basically leaving the door wide open for intruders to come in and wreak havoc on your web server. There are multiple fail points, in this case, where there are multiple websites hosted on the same server, and anyone being compromised ensures the infection spreads to other CMS installed and hosted in the same space. This is a webmaster’s worst nightmare, and should be avoided at all cost.

The worst part of hosting multiple websites on the same server comes when you’re left to deal with changing every single password associated with every single account that has access to the web server. This includes the CMS installation, MySQL (or any other) database, [FTP](#), and more. If you managed to painstakingly get rid off the infection on your server but didn’t spend time changing all the passwords, it’s a job half done and you’ve left another gaping hole in your websites’ defence waiting to be exploited in due time. Duh!

## Deny file permissions

Simply put, file permissions on a web server decide what an entity can do to a certain file hosted online. And no, we aren’t talking about preventing image files being hotlinked here! This is way more important than that, so pay close attention.

In web server terminology, each file hosted online has three levels of permissions attached to it. Read (4), Write (2), Execute (1). The first permission allows for anyone to merely read the file contents, the second permission allows anyone to edit the file contents, and the third and final permission allows anyone to effectively run the program file on the server.

The types of users, who have access to these three types of file permissions, are also divided into three. Owner, Group, Public. An Owner is someone who creates the file for which a permission needs to be set – the values are given by default. Group is a cluster of users with common or

## Permissions: an octal representation

### `chmod permissions file`



■ Read:	4	100
■ Read + Write:	6	110
■ Read + Execute:	5	101
■ Read + Write + Execute:	7	111
■ Write:	2	010
■ Write + Execute:	3	011
■ Execute:	1	001

A quick list of octal representation of different type of file permissions

shared permissions – any file in a group will give the same level of permissions to each and every user in that group. The final user type is Public, which is defined as not only any random user on the Internet, but anyone who's not an Owner or part of a Group policy for web server files.

If you're absolutely paranoid about your website's security, and you should be (why not?!), you can go ahead and deny permission to anyone but yourself (as the Owner of all files) access to your web server files. For anyone else that's not you, giving Read (4) level access should be fine, just to ensure that they don't make changes to it or execute it at all.

### Server configuration lockdown

You can't call yourself a security-conscious webmaster if you haven't gotten your hands dirty at configuring critical web server files for your online website. For most of you who've tried your hand at hosting your own blog before on an easy-to-install Apache web server, that .htaccess file is the critical server configuration file. What's a .htaccess file, we hear you ask? Well, simply put, it's a plain text file that contains the directory level arrangement of all web server files and its associated access permissions. By rewriting or editing the .htaccess file, which usually sits in the root folder of the directory, you can essentially dictate access to different web directories under the tree, enforce password policies, 404 redirects, and more. Similar to Apache web server's



Controlling access via configuration files is not only good, but necessary too!

.htaccess, Microsoft IIS servers have useweb.config and Nginx servers use nginx.conf as their respective server configuration files.

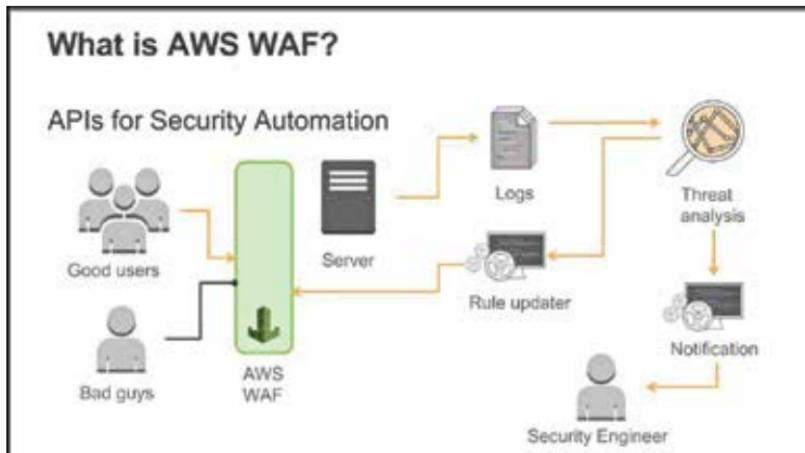
Some essential rules that you should enforce on your web server through the configuration files are as follows. Prevent directory browsing – prevents malicious users from browsing and cataloguing every directory on the website or server; Protecting sensitive files – especially the CMS configuration file, since it's one of the most sensitive files on the web server, holding the database login details in plain text.

## Install a web application firewall (WAF)

Just like you are keen to protect all your digital endpoints – PC, laptop, smartphone, etc. – with a security software (antivirus) of some kind, similarly you need to secure your website, too. Enter WAF or a web application firewall, which is a server plugin or a separate software appliance that sits between your web server and incoming / outgoing communication lines, monitoring the traffic that wants to establish a connection to your website.

Essentially what a WAF does is that it defines a strict set of rules when it comes to allowing who gets to connect to your website and who doesn't. It's typically sensitive to sifting through HTTP traffic and keeping an eye out for XSS (cross-site scripting) and SQL injection attacks – some of the most common hack attack vectors – but it can be configured to stay alert and vigilant about more sophisticated attacks as well. Most modern WAF solutions are capable of dealing with DDoS and other advanced potential web threats, too.

WAF can be deployed both at a hardware and a software level. Unlike very few popular and reliable PC security vendors, the market for WAF is inundated with a whole bevy of products offering different levels of



The antivirus for your website

security. Take a look at Amazon Web Services' WAF offering (<http://digit.in/AmznWAF>) which only debuted in October 2015, and offers web security at an attractive price.

### TLS, not SSL

If you've ever delved into online network security then you'd have definitely come across acronyms like TLS and SSL – hey, you even see them advertised on domain registering and web hosting providers online! But for the uninitiated, TLS (Transport Layer Security) and SSL (Secure Socket Layer) are two protocols used for establishing secure, encrypted network connections between web browsers, apps, websites and web servers. So why are we saying choose TLS over SSL?

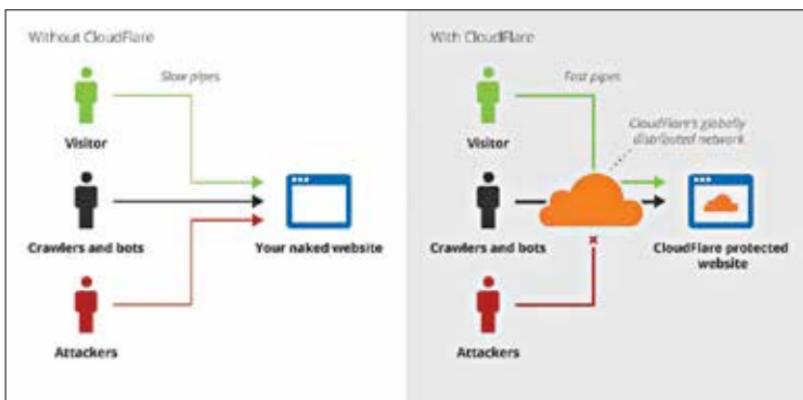
In many ways, SSL is considered as a precursor to TLS, and is therefore more widespread across the Internet. What's concerning about SSL – especially SSL 3.0 – is the fact that recently POODLE (a vulnerability that lets attackers gain access to private data of users on websites like passwords, cookies, and other information) has successfully circumvented SSL 3.0 with ease, effectively ringing its death knell. Most online business websites, which collect user data, are quickly shifting over to TLS as a result, which is more robust and at present one of the most widely available secure encryption protocol for public communication over the Internet. TLS 1.2 is the most recent version of TLS, offering a lot of advanced security and encryption measures over the now outdated SSL.

When you're configuring your website, make sure you definitely configure a secure web server – especially if your website's going to contain a username, password, and other user information. Definitely choose to go for TLS 1.2 even if it comes at an added premium (usually it doesn't, it's just a case of clicking one radio button over another). By doing this, you'll ensure the security and integrity of communication between your website's server and different users (or just yourself, be that as it may), not letting anything slip into the prying ears of online eavesdroppers.

## CDN

Content delivery networks (CDN) are also worth evaluating in your quest to secure and safeguard your website online. Not only do they ensure super fast delivery of your web pages or online service all across the globe, through a vast array of servers all over the world, it can also be a line of defence in your security strategy when you're website's encountering DoS attacks -- and especially if your website happens to go down for some reason.

A security-focused content delivery network (CDN), CloudFlare (among others) is worth looking into. It likes to describe itself as an online community watch, keeping the internet's bad guys from disrupting your online business, and in a nutshell that's true. By adding CloudFlare to your website, you essentially allow it to filter content requests flowing into your website, blocking out all the spam and DDoS elements in the process. It also caches all your website's static and dynamic content across its network of web servers, so even if your website happens to encounter some down time (even for maintenance) a copy of it can always be online and accessible to users



This is how CloudFlare adds some flair to your website's security

from around the world. What's more, CloudFlare is free to use, and it also has paid plans for more serious businesses who don't want to compromise with their website's security and reliability.

While CloudFlare is the most popular free CDN out there (even the Pirate Bay uses it!), there are alternatives like InCapsula, Myra Cloud and CDNify that are also worth looking into.

## **Hack your website (before someone else does)**

OK, so you're not a hacker. We get that. And learning how to hack in meticulous detail will take time. We get that, too. But think how a hacker would – he or she wouldn't really go out trying to hack a website from scratch, without gathering some early vulnerability report, would they? That would be madness, if they did that! You need to do something similar to your website, find out all the chinks in its armour before the bad guys do.

Just like you can download a piece of software onto your PC or smartphone and test it for viruses and vulnerabilities, similarly you can run malware and vulnerability scanners on your web host to readily find a gaping hole or loose brick in its defensive wall. This step's important, because without this you're essentially in the dark, and won't know what steps to take to reinforce your website's defence.

Wapiti (<http://dgit.in/WapitiFzr>) is an open-source fuzzer, and a good place to start testing your website's security. Although a couple of years old, Wapiti tries to inject code into a website (through pages or forms) without thoroughly detecting its CMS. Another neat online tool used to audit a website's security is ScanMyServer (<http://dgit.in/ScnMySrvr>). It offers a vast and comprehensive report on a variety of parameters of security testing – SQL Injection, Cross Site Scripting, PHP Code and HTTP Header Injection, and lots more. Detectify (<http://dgit.in/detectify>) and SUCURI (<http://dgit.in/SucuriSite>) are other great website vulnerability testing tools available for your benefit, performing hundreds of automated tests to give you a robust report on your website's security status. Do act on the findings of these online vulnerability scanning tools!

## **Backup without fail**

An oft-overlooked aspect of running or managing your own blog or website is taking regular off-server backup of your CMS' directory files. Why? Because when push comes to shove, and your site gets hacked against all odds, at least you can restore it back to some immediate point in the past, and not

start from scratch (which is Day 0). Make sure you use some FTP app like FileZilla to copy and backup your entire website's directory files (with all the subfolders included, obviously) and copy it onto a different online location (Google Drive or Dropbox or something similar). Along with all the CMS files, don't forget to regularly backup the database associated with your website's CMS, too. All of this over and above regular server-side backups that are done by your web host, obviously. If you're in doubt, backup before doing anything else. And when you're done doing that, backup some more. The value of backups cannot be neglected!



When in doubt, hit the backup button!

We know these are nothing but beginner level steps to pique your interest into securing and protecting your website. We are confident that after reading these tips, you'd want to go online and hunt for more robust, in-depth tips to protect your website. We hope that everyone who takes the pain of starting or running their own website doesn't do themselves a disservice by ignoring its security and protection. Websites or any other online extension of our life needs to be guarded with the same level of seriousness and security as we safeguard our physical assets. Until we do that, the bad guys will always be a step ahead of us in this eternal game of cat and mouse. **d**



All this and more in the  
world of Technology

**VISIT  
NOW**

 **digit.in**

# Join 800K+ members of the digit community



<http://www.facebook.com/thinkdigit>

**facebook**

**digit.in**

**Digit** Like

Your favourite magazine on your social network. Interact with thousands of fellow Digit readers.

Wall | 1214 | 18 Photos | 133 Friends | 12 Groups

<http://www.facebook.com/IThinkGadgets>

**facebook**

**I Think Gadgets** Like

An active community for those of you who love mobiles, laptops, cameras and other gadgets. Learn and share more on technology.

Wall | 1214 | 18 Photos | 133 Friends | 12 Groups

<http://www.facebook.com/GladtoBeaProgrammer>

**facebook**

**Glad to be a Programmer** Like

If you enjoy writing code, this community is for you. Be a part and find your way through development.

Wall | 1214 | 18 Photos | 133 Friends | 12 Groups

<http://www.facebook.com/devworx.in>

**facebook**

**devworx** Like

devworx, a niche community for software developers in India, is supported by 9.9 Media, publishers of Digit

Wall | 1214 | 18 Photos | 133 Friends | 12 Groups