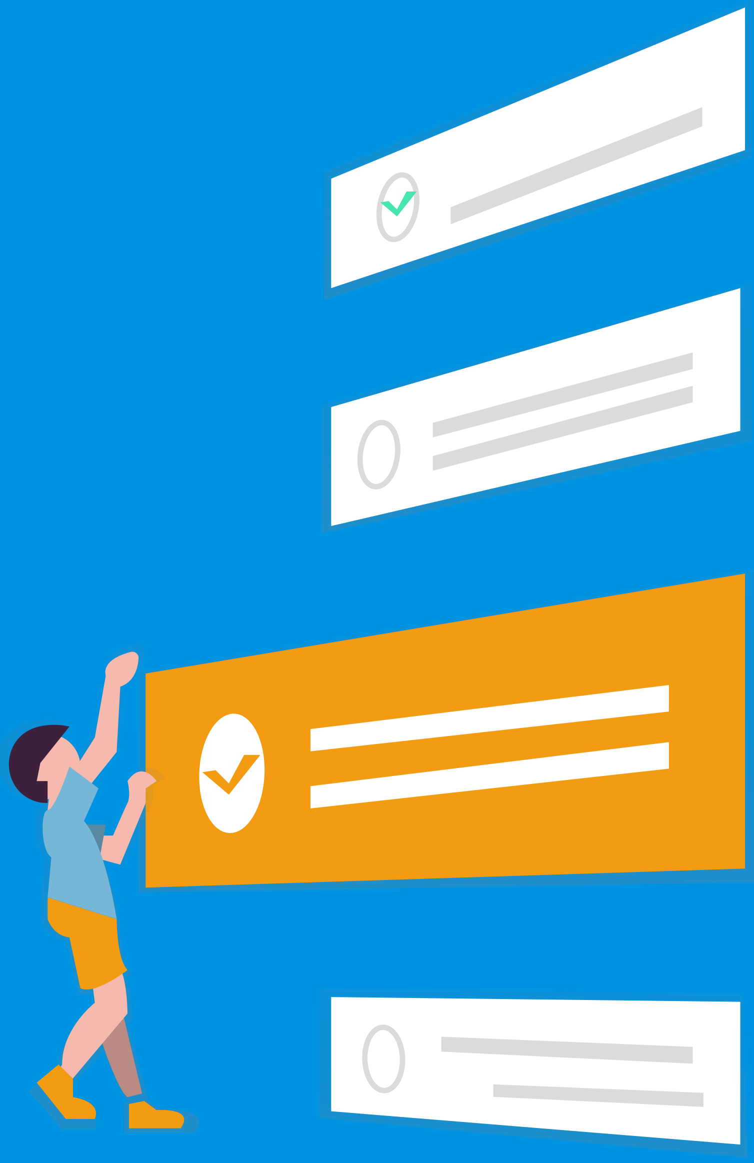


Introduction to Web Development

Absolute beginners guide to
getting started with Web Development

>hackr.io

Preface

“Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.”

- Linus Torvalds
Creator of Linux Operating System

Index

Section 1 - Introduction

Section 2 - Front End Development

Section 3 - Back End Development

Section 4 - Essentials for every developer

Section- 1

Introduction

Introduction to Software Development

What/Who is a Web Developer?

Why Web Development?

Chapter 1

Introduction to Software Development

If you are a complete beginner, junior developer, or are curious about this industry, this ebook is for you. If you are an established developer, you may find some useful links in here as we list the best resources to supercharge your skills.

Using online courses and mostly free tutorials, you can gain a valuable skill that will allow you to be employed in a great industry that is rewarding and challenging.

Why coding?

Before we get into the steps you can take to become a developer, we must first dive into why you would want to go down this path. Every decision that will require significant time of your life should be justified. Time, after all, is the most important resource we have:

A. You want to be working in an industry where there is a high demand for the skill and many possibilities to be in highly important roles at the top of the food chain.

B. You love being location independent. You want a skill that allows you to go anywhere in the world and still be able to find a job easily. If you decide to move to Iceland tomorrow, you want to make sure that you won't have issues finding a job.

C. You've noticed the difference between 2003 and 2018 and how much of a technological progress we have made in those short 15 years. You want to be at the forefront of an industry that is impacting the world.

D. The biggest industry growth in the last couple of years has been in the cryptocurrency (Bitcoin), and artificial intelligence (Machine Learning) space. We interact with technology on the daily basis, and you want to not be left behind in the dust as these things take over our future. You want to understand and be able to pick up these skills.

E. You think change is good, and learning should never stop. So why not do something new?

"But I don't have a computer science degree and I don't even know how the internet works!"

Don't worry, we will use that to your advantage. Keep reading...

Chapter 2

What/Who is a Web Developer?

Here’s the simple answer to this question: “**They build and maintain websites.**”

In other words, they literally create how you experience the web. And for those websites who want users to enjoy their experience, they need great web developers – and they’re usually willing to fork over some serious cash to get these great web developers.

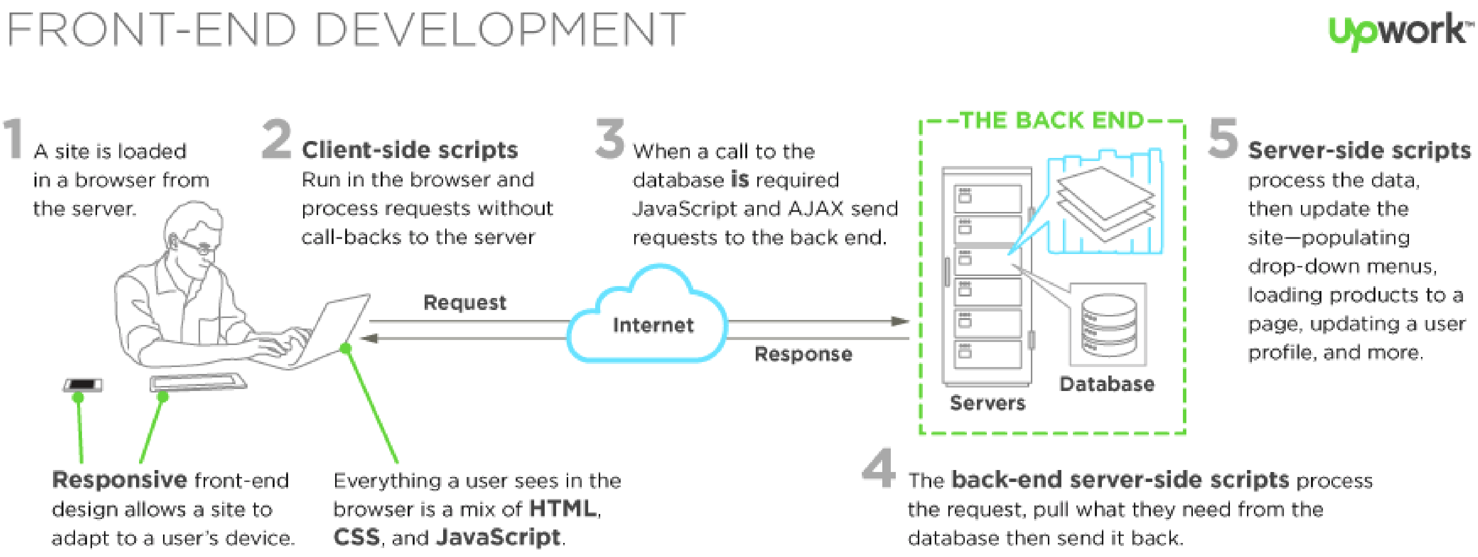
Responsibilities of a web developer:

- Write web pages with a combination of markup languages.
- Create quality mockups and prototypes.
- Build a website from scratch.
- Create quality mockups and prototypes.
- Understand HTML and CMS/Databases, Severs, APIs etc
- Understand UI and UX.
- Develop functional and appealing websites and web-based applications.
- Provide website maintenance and enhancements.

These are the most common, but your responsibilities and tasks will vary depending on the company you work for as well as your web development specialty.

Speaking of web development specialties, there are 3 that you should be aware of:

1- **Frontend development:** The “frontend” refers to “stuff” on a website that you see and interact with, like menus, dropdowns, etc.



FRONT-END DEVELOPMENT

1 A site is loaded in a browser from the server.

2 **Client-side scripts** Run in the browser and process requests without call-backs to the server

3 When a call to the database **is** required JavaScript and AJAX send requests to the back end.

4 The **back-end server-side scripts** process the request, pull what they need from the database then send it back.

5 **Server-side scripts** process the data, then update the site—populating drop-down menus, loading products to a page, updating a user profile, and more.

Responsive front-end design allows a site to adapt to a user's device.

Everything a user sees in the browser is a mix of **HTML**, **CSS**, and **JavaScript**.

2- **Backend development:** The "backend" is like the portion of the iceberg below the surface. Without it, the website can't function. The backend deals with things like servers, applications, and databases.

3- **Full-stack development:** This is the combination of both backend and frontend development.

Chapter 3

Why Web Development?

Web development is an industry that won’t die out anytime soon. In fact, the US Bureau of Labor Statistics has predicted a 27% growth in available web development jobs through 2024.

It was also rated as the #2 hottest job in 2016.

Put simply, if you can develop this skill, you won’t have a hard time finding work for the foreseeable future.

Here are FIVE big advantages/benefits of being a web developer:

- **You can work remotely.** Ever wondered what it would be like to work from anywhere in the world, on your own time? Well as a web developer, you’ll have plenty of opportunities to work remotely. Maybe you can finally fulfil that dream of travelling the world, all while saving money at the same time.
- **You can use your skills to build your own websites and web applications.** Want to build your own profitable website or application? With this skill, you won’t have to pay a ton of money to get it built. You can build it yourself for a fraction of the price.
- **You can work independently.** You no longer have to work under the umbrella of a boss if you don’t want to. You can freelance or start your own business much easier when you know web development.
- **Get into the profitable tech industry.** Web development is basically your ticket into the tech industry. Most tech startups have some need for web developers, so this can be your way to get your foot in the door.

- **You can build awesome stuff!** The fun part is that you can be as creative as you want. Instead of looking at websites and thinking, “Wow, that looks awesome. I wish I could create something like that,” you will now be able to create your very own awesome-looking website.

So, what do you think? Does all of this get you excited about the idea of becoming a web developer? If so, you’re in the right place – because you’re about to learn the steps you need to take to become one!

Section- 2

FRONT END DEVELOPMENT

HTML- The Skeleton of a webpage

CSS- Beauty of a webpage

JAVASCRIPT- Behaviour of a webpage

User Interface Design/User Experience Design

Responsive Web Design

Chapter 4

HTML- The Skeleton of a webpage

Learn the Basics of HTML, CSS, and Javascript

These are the nuts and bolts of all website development that you'll work with daily if you decide to build websites for a living.

- HTML dictates the structure
- CSS will make it look pretty
- Javascript will make it function

Let's discuss each one of these and how you can learn them.

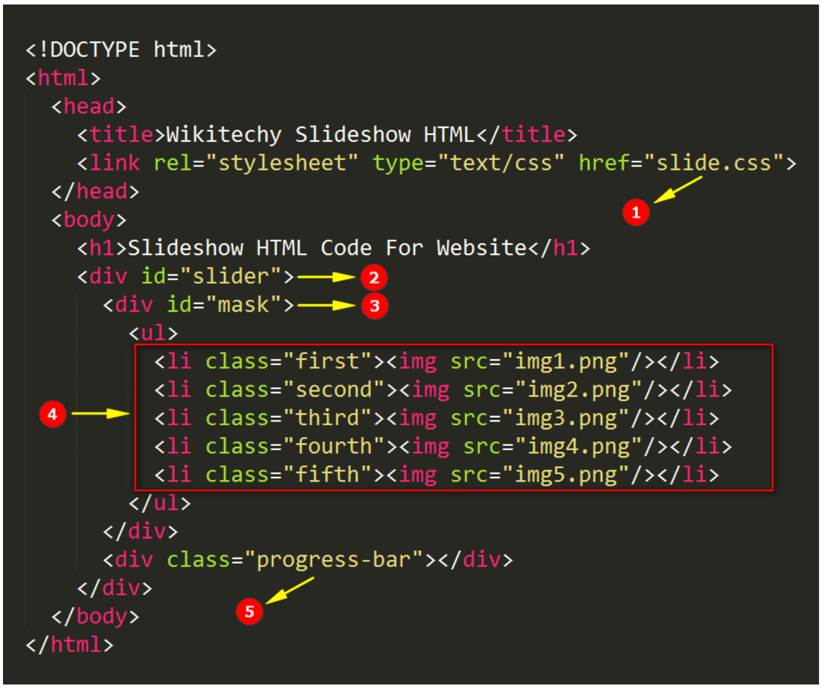
HTML

HTML stands for Hypertext Markup Language. It's one of the main ingredients of any website and one of the so-called frontend languages.

In short, it provides the basic skeleton of a website, and it does so mainly through a series of tags.

<HTML>…</HTML> This tag shows up at the beginning and end of an HTML document. It indicates that the document is written in HTML5.

<title>…</title> The title tag is the title for the page. This is useful both for search engines (when they scan and index pages) and users (it shows up in a browser's title bar) by explicitly stating the primary topic of each page.



```html
<!DOCTYPE html>
<html>
  <head>
    <title>Wikitechy Slideshow HTML</title>
    <link rel="stylesheet" type="text/css" href="slide.css">
  </head>
  <body>
    <h1>Slideshow HTML Code For Website</h1>
    <div id="slider">
      <div id="mask">
        <ul>
          <li class="first"><img src="img1.png"/></li>
          <li class="second"><img src="img2.png"/></li>
          <li class="third"><img src="img3.png"/></li>
          <li class="fourth"><img src="img4.png"/></li>
          <li class="fifth"><img src="img5.png"/></li>
        </ul>
      </div>
      <div class="progress-bar"></div>
    </div>
  </body>
</html>
```

Sample HTML code with various html elements. 1. links 2. Ul elements 3. list items 4. Ul elements
Image source: https://www.wikitechy.com

<head>…</head> This contains information about the specific page, including title tags, meta data, and links to scripts and style sheets.

<body>…</body> This includes all content that will be shown to users, including everything they'll see and read.

Here are some resources for learning HTML:

- Best HTML Tutorials
- HTML Beginner's Guide
- Dive Into HTML5 by Mark Pilgrim

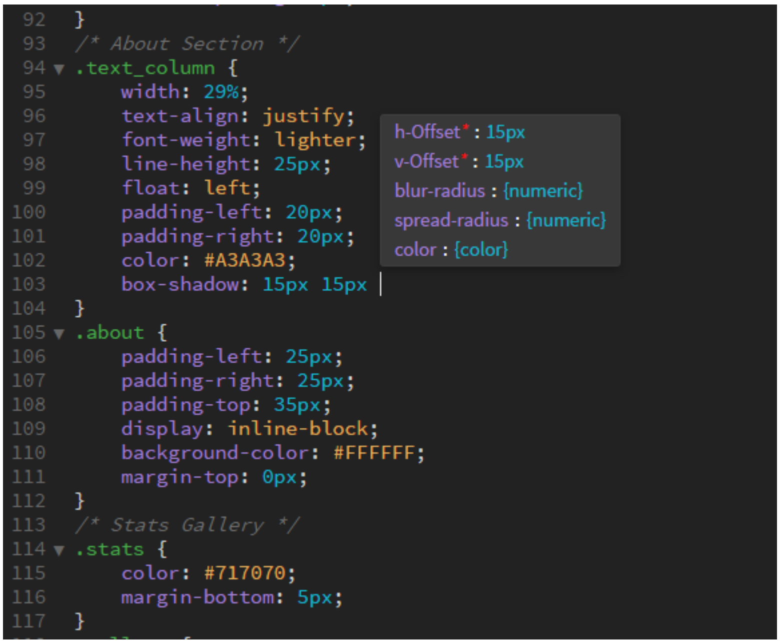
Chapter 5

CSS- Beauty of a Webpage

CSS

CSS stands for Cascading Style Sheets.

It puts the style behind the HTML structure. Basically without CSS, HTML would be boring and as a result the web page would be boring.



```
}
/* About Section */
.text_column {
    width: 29%;
    text-align: justify;
    font-weight: lighter;
    line-height: 25px;
    float: left;
    padding-left: 20px;
    padding-right: 20px;
    color: #A3A3A3;
    box-shadow: 15px 15px 
}
.about {
    padding-left: 25px;
    padding-right: 25px;
    padding-top: 35px;
    display: inline-block;
    background-color: #FFFFFF;
    margin-top: 0px;
}
/* Stats Gallery */
.stats {
    color: #717070;
    margin-bottom: 5px;
}
```

h-Offset* : 15px
v-Offset* : 15px
blur-radius : {numeric}
spread-radius : {numeric}
color : {color}

As you can see in the image above that with CSS you can change font-size, background-color, padding, margin, height or width etc of the webpage to make it look the way you want.

Here are some great resources for learning CSS

- Learn to Code HTML & CSS - Shay Howe
- Getting started with CSS- MDN
- Visual guide to the most popular CSS properties

Chapter 6

JAVASCRIPT- Behaviour of a Webpage

Javascript

Javascript is a programming language that allows you to interact with web pages. It allows you to add interactivity to your web pages by implementing click events, hover events or events like drag and drop. Any action you take on a web page has most probably some Javascript functionality written behind it.

Javascript really is everywhere today be it on the front end or the server. Anything you can imagine can be made with javascript today. There are awesome front end and back end javascript frameworks today that can help you build any digital product you can think of.

Here are some great Javascript tutorials for you to with:

- Eloquent JavaScript
- Mozilla Developer Network Javascript Guide
- Build 30 things in 30 days with 30 tutorials

Chapter 7

User Interface Design/User Experience Design

UI (User Interface) and UX (User Experience) are the basics of user experience design.

Most developers aren’t design experts – these are two different realms. So, you don’t need to be a design rockstar.

However, by learning the fundamentals of user experience design, you can better understand how a website is supposed to work in order to keep more users on the site, help them find what they’re looking for, and ultimately spend more money on that site.

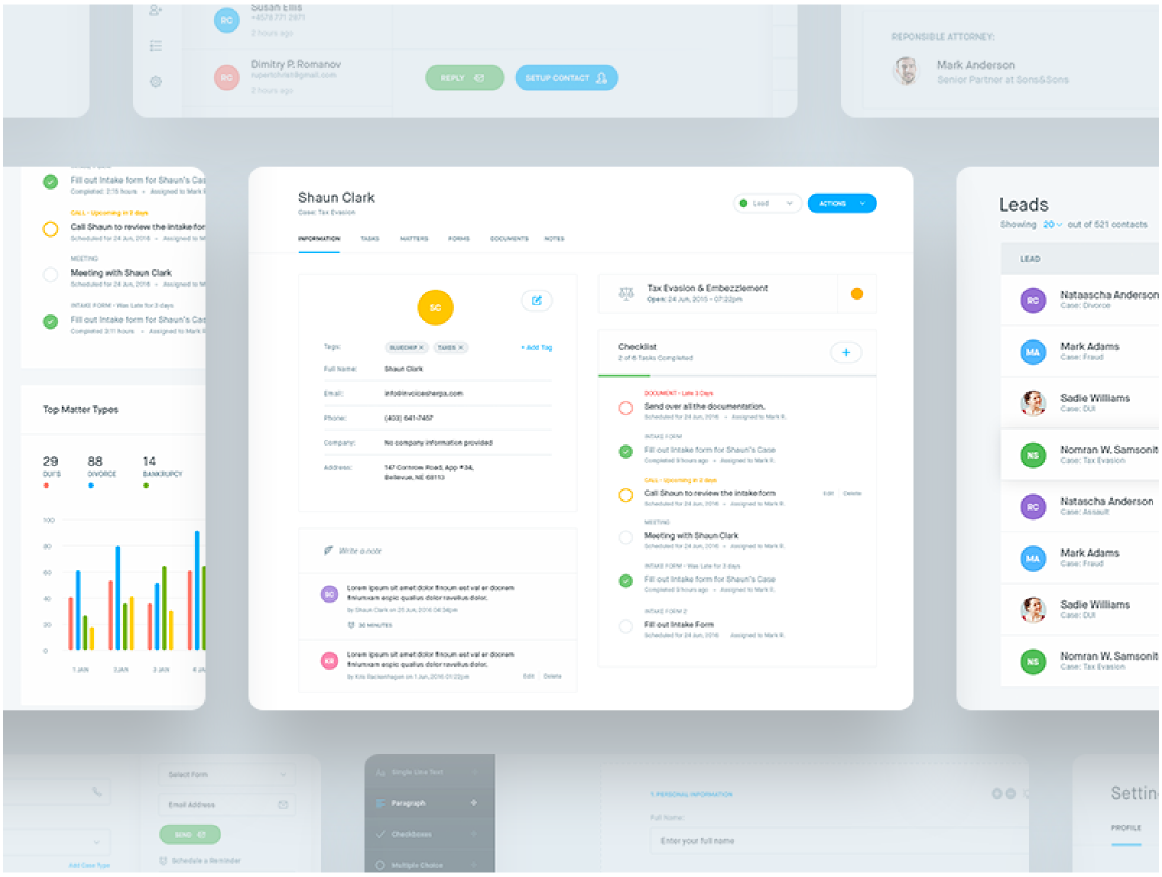


Image Source : https://dribbble.com/shots/2850451-CRM-Dashboard

To learn the building blocks of design skills, I recommend learning the Adobe Creative Suite. Photoshop should be the first thing you dive into, as it's the go-to for most serious designers. If you don't like Adobe, you can also dive into Sketch, which is a rising star among designers.

Here are some resources for understanding and learning UI and UX:

- The Difference Between UX and UI Design-A Layman's Guide
- Learn UI Design
- Fundamentals of UI Design- Invision

Chapter 8

Make Sure Your Site is Responsive

When you’re creating your own website (which will be a good idea if you want to get more clients) – you should make sure your site is responsive.

Responsive means that the elements of the website adjust according to the screen size. So, your website will look good whether the visitor is on a laptop or mobile device.

Today responsive design is very crucial. All of your clients/employers will expect it– So lead by example and make sure that your website is responsive.

Here are some resources for understanding and learning Responsive design:

- Responsive Web Design W3Schools
- Responsive Web Design - Introduction
- Responsive Web Design – What It Is And How To Use It

Section 3

BACK END DEVELOPMENT

Programming Languages

Databases

Servers

API- Application Programming Interface

Introduction

To understand the back end, or the "server side," you have to know the front end very well and how the two interact. The front end, also called "client-side" programming, is what happens in the browser—everything the end users see and interact with. The back end, on the other hand, happens on the server (on site, or in the cloud) and databases. It's the machinery that works behind the scenes—everything the end user doesn't see or directly interact with, but that powers what's happening.

The "traditional" back end is a mix of the server, databases, APIs, and operating systems that power an app's front end.

Here's some more information about each of these components of the backend.

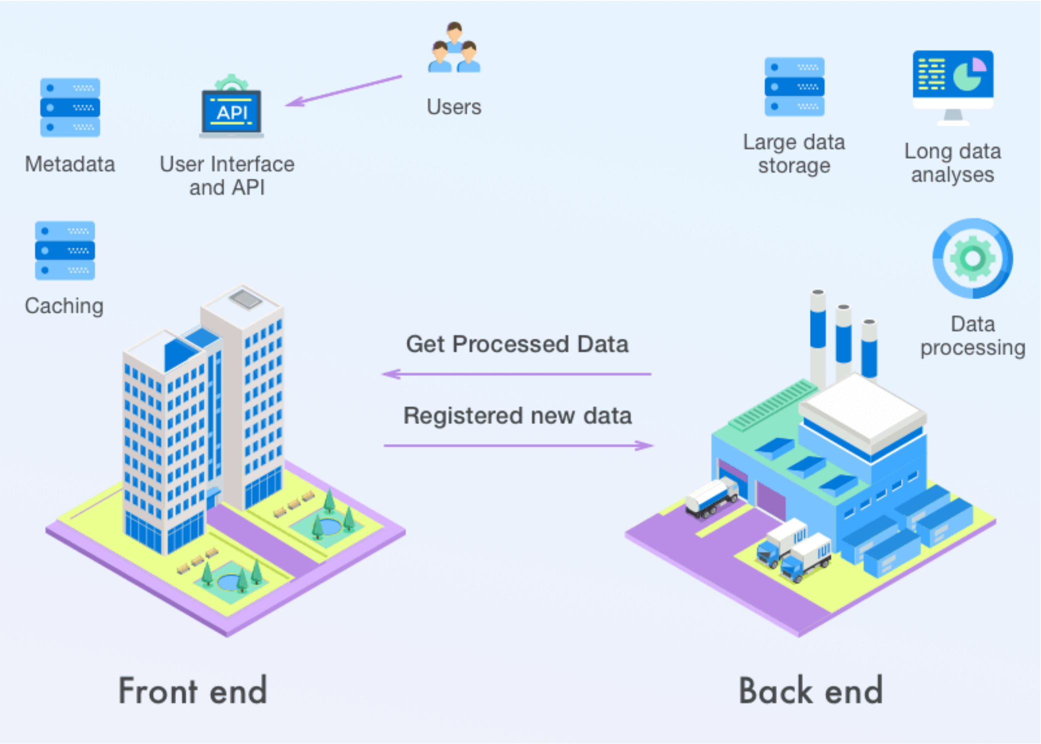


Image source: https://www.cleveroad.com

Chapter 9

Programming Languages & Frameworks: The Nuts & Bolts

Back-end developers can choose from a variety of languages and frameworks depending on the type of application they're building, its specific processing requirements, and what other components already exist on the back end.

Languages will differ in file size, performance, compatibility, how many lines of code required, and the style of programming. Some back-end scripting languages are object-oriented programming languages, a style of programming that bundles attributes and functions within objects. Other languages may be compiled rather than interpreted, something that affects load time, readability, and processing power required to run the application.

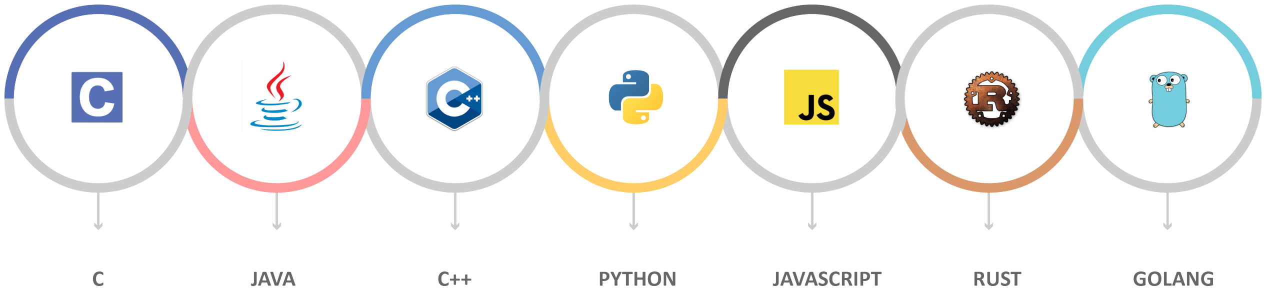


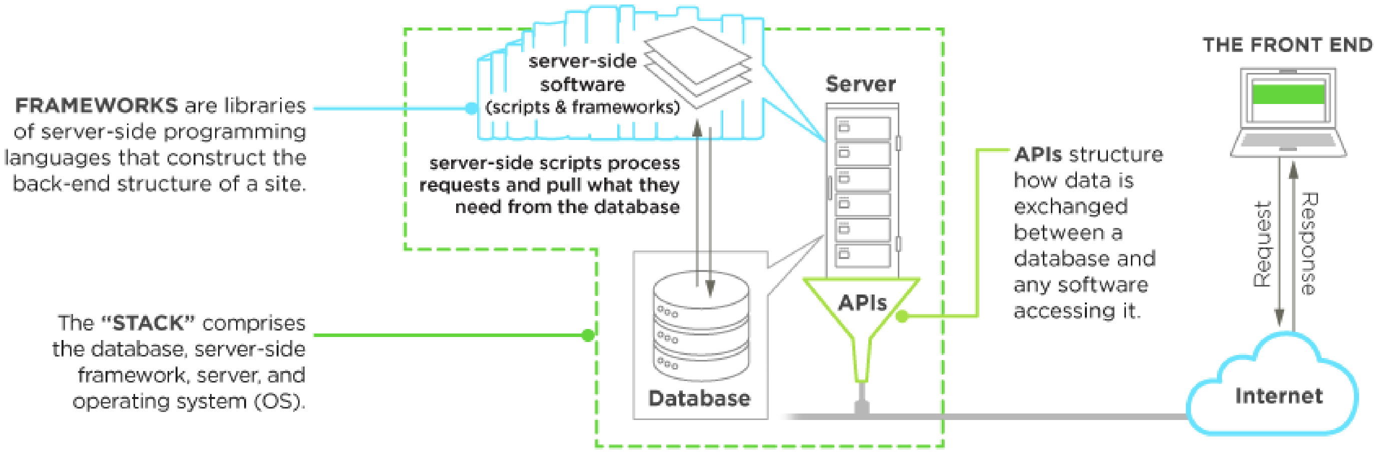
Image Source: http://www.cuelogic.com

A majority of sites are built on PHP, making it one of the most popular back-end scripting languages. But there are plenty of others to choose from. For instance, you may want to use the processing power of Python for a data-driven site, or leverage the speed of lightweight languages like Ruby for faster prototyping. It's up to you and your developers, and every language has its pros and ideal applications.

Take a look at a few of the other big hitters in back-end programming, like:

- Java
- C# and C++
- .NET
- Perl
- Scala
- Node.js (and JavaScript)

BACK-END DEVELOPMENT & FRAMEWORKS IN SERVER SIDE SOFTWARE


FRAMEWORKS are libraries of server-side programming languages that construct the back-end structure of a site.
The "STACK" comprises the database, server-side framework, server, and operating system (OS).
server-side software (scripts & frameworks)
server-side scripts process requests and pull what they need from the database
Server
Database
APIs
APIs structure how data is exchanged between a database and any software accessing it.
THE FRONT END
Request
Response
Internet

- Django (for Python)
- Spring framework (for Java)
- Node.js Frameworks like MeteorJS and ExpressJS
- Ruby on Rails
- Symfony (for PHP)
- JSF (Java Server Faces)

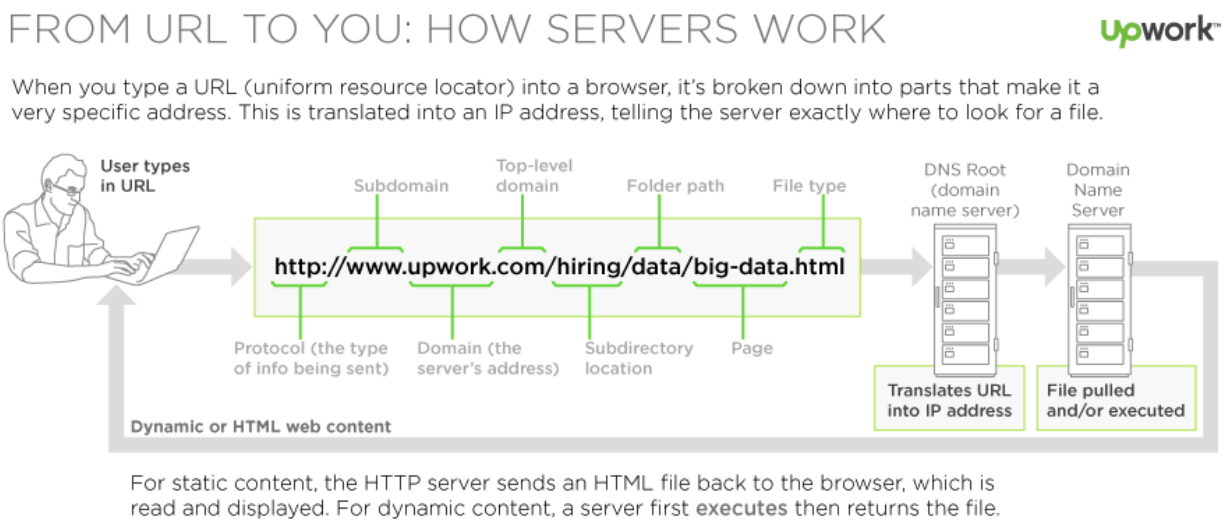
That covers the most basic concepts and tools of back-end development, but there's still more to learn.

Chapter 10

Servers: The Machinery

Of your back-end stack's four components, whether it's on-site or in the cloud, the server acts as the lifeblood of the network. These high-powered computers provide shared resources that networks need to run, including file storage, security and encryption, databases, email, and web services. Learn all about types of servers, what they do, and how they play into the back-end stack with our Guide to Server Technology.

Once you've gotten the basics down about on-site servers, expand your knowledge with a look at virtualization, how servers get provisioned to house multiple apps, and containerization, another way servers provision their operating systems out to house compartmentalized applications.



FROM URL TO YOU: HOW SERVERS WORK

When you type a URL (uniform resource locator) into a browser, it's broken down into parts that make it a very specific address. This is translated into an IP address, telling the server exactly where to look for a file.

upwork
User types in URL
Subdomain
Top-level domain
Folder path
File type
http://www.upwork.com/hiring/data/big-data.html
Protocol (the type of info being sent)
Domain (the server's address)
Subdirectory location
Page
DNS Root (domain name server)
Domain Name Server
Translates URL into IP address
File pulled and/or executed
Dynamic or HTML web content

For static content, the HTTP server sends an HTML file back to the browser, which is read and displayed. For dynamic content, a server first executes then returns the file.

Here are some resources for you to read more about servers:

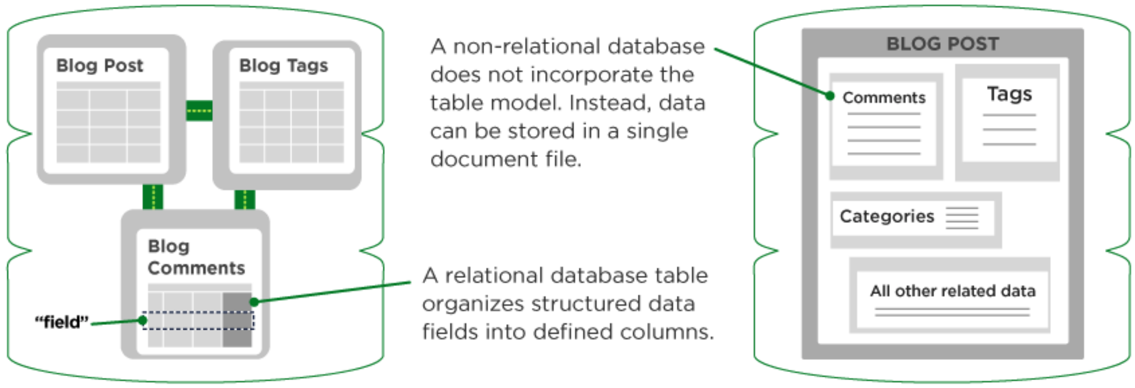
- Windows Server Administration Fundamentals
- Ubuntu Server Fundamentals

Chapter 11

Databases: The Storage

Databases, in the context of a website, are the brains that make websites dynamic. Any time you request something from a website—whether you're searching for a product in on online store or searching for hotel locations within a specific state—the database is responsible for accepting that query, fetching the data, and returning it to the website. Databases can also accept new and edited data when users of a website or application interact with them. The client can change information in a database from the browser, whether a user is posting articles to a CMS, uploading photos to a social media profile, or updating their customer information.

RELATIONAL VS. NON-RELATIONAL DATABASES



A non-relational database does not incorporate the table model. Instead, data can be stored in a single document file.

A relational database table organizes structured data fields into defined columns.

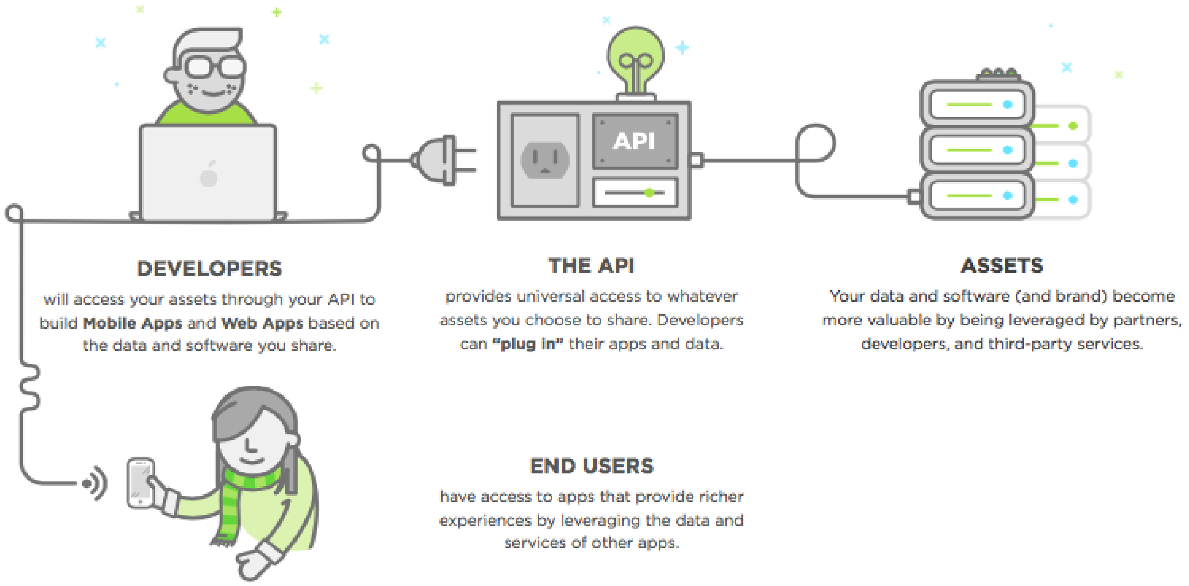
Here are some resources for you to read more about databases:

- Try SQL
- SQL basics by Khan Academy
- SQL Tutorial by Tutorials Point

Chapter 12

APIs: A Crucial tech in Back End Programming

You can't talk about the back-end portion of an application these days without touching on APIs (application programming interfaces) and how they connect software, applications, databases, and services together seamlessly. APIs play an integral role in how most server-side software architectures are built, oftentimes replacing more complicated programming to allow software to communicate and data to be transferred.



DEVELOPERS

will access your assets through your API to build **Mobile Apps** and **Web Apps** based on the data and software you share.

THE API

provides universal access to whatever assets you choose to share. Developers can **"plug in"** their apps and data.

ASSETS

Your data and software (and brand) become more valuable by being leveraged by partners, developers, and third-party services.

END USERS

have access to apps that provide richer experiences by leveraging the data and services of other apps.

Here are some resources for you to read more about API's:

- What is an API?
- Public APIs vs. Private APIs: What's the Difference?
- SOAP vs. REST: Comparing Two API Architectures

Section 4

DEVELOPER ESSENTIALS

Data Structures and Alogrithms

GIT

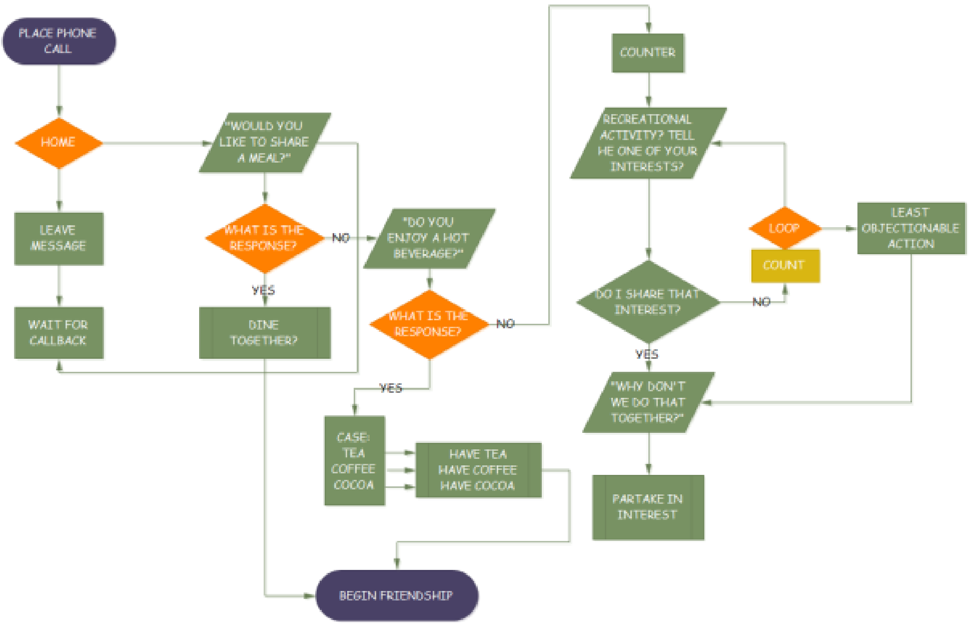
SEO

Chapter 13

Data Structures and Algorithms

Data Structures are the programmatic way of storing data so that data can be used efficiently. Almost every enterprise application uses various types of data structures in one or the other way.

Having a solid understanding of data structures will help you crack complex logics behind problems as well as it will increase your problem solving capabilities. Below is a sample flow diagram of an algorithm:


PLACE PHONE CALL
HOME
LEAVE MESSAGE
WAIT FOR CALLBACK
"WOULD YOU LIKE TO SHARE A MEAL?"
WHAT IS THE RESPONSE?
NO
YES
DINE TOGETHER?
"DO YOU ENJOY A HOT BEVERAGE?"
WHAT IS THE RESPONSE?
NO
YES
CASE: TEA COFFEE COCOA
HAVE TEA HAVE COFFEE HAVE COCOA
BEGIN FRIENDSHIP
COUNTER
RECREATIONAL ACTIVITY? TELL HE ONE OF YOUR INTERESTS?
DO I SHARE THAT INTEREST?
NO
YES
LOOP
COUNT
LEAST OBJECTIONABLE ACTION
"WHY DON'T WE DO THAT TOGETHER?"
PARTAKE IN INTEREST

Sample Algorithm design

Here are some resources for you to read more about Data Structures and algorithms:

- Fundamentals of Algorithms
- Data Structures - Geeks for Geeks
- Introduction to Algorithms - MIT OpenCourseWare

Chapter 14

GIT: Version Control

Well, in a nutshell, it’s an open source version control system that was created to help people collaborate with code. Of course it’s more complicated than that, but that describes the basics.

What is version control?

It’s a convenient and powerful way to safely make changes to something (with Git, this something is most commonly code of some kind) without worrying that you, or someone else working with you, might break something.

For example, let’s say you’re editing a bunch of documents with others. Version control allows you to do that safely, without fear of your changes overwriting someone else’s. In addition, it offers you a history (and backups) of those changes, just in case you need to restore back to an earlier state.

How can you collaborate with Git?

Git can be used as a simple single-player version control system, but it’s most powerful when used with others. Through concepts like “forking” and “branching” you, and your collaborators, will be able to work on the same project, even in the same file, without any fear of writing over each other’s work.

Here are some resources for you to read more about GIT:

- Pro Git Book by Scott Chacon and Ben Straub
- Git Guru Guide by Atlassian
- Git Immersion

Chapter 15

SEO: Search Engine Optimization

SEO stands for Search Engine Optimisation. This refers to the process of improving a website's rankings within a search engine like Google. It's one of the most important skills in online business.

But you may be thinking, "If I'm a developer, why do I need to worry about how well a site ranks in the search engines? Isn't that the job of the bloggers and content creators?"

Well yes, a large part of a website's SEO has to do with the content. But the actual structure and code of the website plays a role as well.

For example, heading tags (HTML) are extremely important for SEO. They tell search engines what's really important on a website.

Now, do you need to be an SEO expert? Absolutely not. But you should definitely learn the basics of SEO, and keep SEO in mind while you're creating each website. In doing so, you'll give the website a much greater chance of success, and make your web development client that much happier.

Here are some resources for learning the basics of SEO:

- The Beginner's Guide to SEO
- The Web Developer's SEO Cheat Sheet 2.0
- Link Building: The Definitive Guide

It’s not over yet!

Here we have provided you with an overview of the most essential technologies and requirements of the development cycle. There is still a lot left to learn that goes on in software development, but the information given here will set some strong foundations for you to build upon.

We hope this ebook helped you understand Web Development Better. If you have any queries, questions of you just want to talk to us, Please drop us an email at: **saurabh@hackr.io**

This ebook is brought to you by **hackr.io.** Hackr.io is a platform to **find and share best online tutorials** of any programming language or technologies.
Visit us at: **https://hackr.io**

>hackr.io

Acknowledgements

Some of the resources that helped us whle creating this ebook.

Front End and Programming Intro: Medium

[Learn to code in 2018, get hired, and have fun along the way](https://hackernoon.com/learn-to-code-in-2018-get-hired-and-have-fun-along-the-way-b338247eed6a)

Back End Intro: Upwork

[A Beginner’s Guide to Back-end Development](https://www.upwork.com/hiring/development/a-beginners-guide-to-back-end-development/)

Back End Explanation: thinkapps.com

[Back-End Development: A Guide to the Basics](http://thinkapps.com/blog/development/basics-back-end-development/)

Git basics-1

[Learn to love Git. Part one: The basics. – Designing Atlassian – Medium](https://medium.com/designing-atlassian/learn-to-love-git-part-one-the-basics-90429f456ace

HTTP:

[An overview of HTTP - HTTP | MDN](https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview)

SSH:

https://semaphoreci.com/community/tutorials/getting-started-with-ssh

SEO

[SEO Made Simple: A Step-by-Step Guide](https://neilpatel.com/what-is-seo/)