

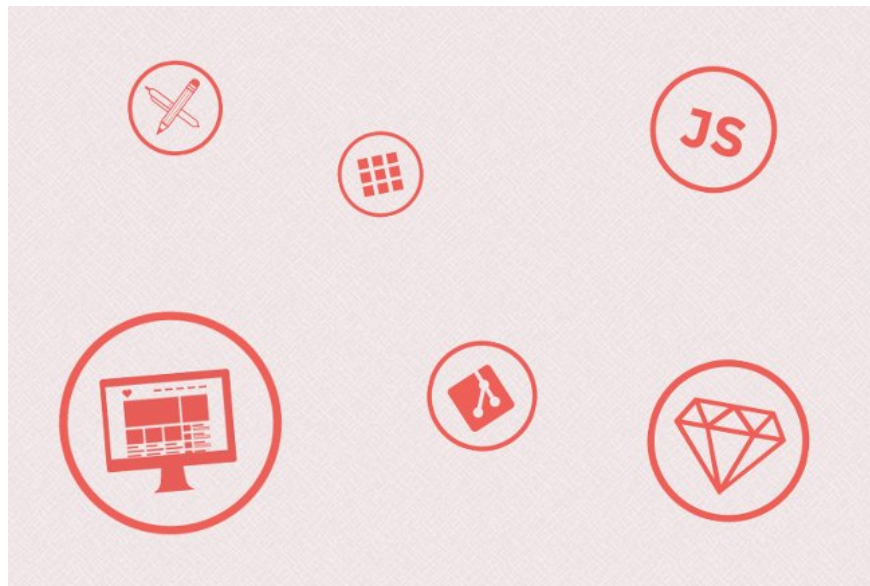
# 13 skills needed to become Frontend Developer

Master the 13 skills you need below and make sure you have a great front-end developer job!

1. Beginners of computer programming need to focus on what?
2. 11 basic principles that every programmer should follow
3. Why do I study code instead of pursuing a career in finance?

Have you ever visited your favorite website and noticed how it was arranged? You have to look at how it is set, how the ' button ' buttons work when you click on them or any other part and think ' I wonder how it gets complicated. so? 'Or' I wish I could do that ' or not?

All this is encapsulated in **front-end development** . **Web design** is how that website looks, how it is organized, and how the **front-end development** is how to implement those layouts on the website.



**Front-end developers** use HTML, CSS and JavaScript to code website and web application designs created by web designers. You can understand that the Front end developer is the interface design for websites and applications. The code that front-end developers write runs inside the user's browser ( *as opposed to back-end developers are code that runs on the web server* ).

You might think so simple: **Back-end developers** are like architects designing and creating systems that make a city work ( *electricity, water and sewers, .* ); Meanwhile, the front-end developer is the person who arranges the roads and ensures that everything in the system is properly connected to each other so that people can live their lives ( *a single analogy Simple when you have a rough idea* ). Moreover, the front-end developer is responsible for ensuring no errors or errors in the front-end; as well as ensuring that the design appears to meet the

requirements of different platforms and browsers.

I - the author of this article - thoroughly searched dozens of front end developer work lists to see which skills are most needed for a current front-end developer and that's what employers really are search ( *and will search in the near future* ). **Master the 13 skills you need below and make sure you have a great front-end developer job!**

## 1. HTML & CSS



**HTML** (Hyper Text Markup Language) and **CSS** (Cascading Style Sheets) are basic unified blocks in web coding. Without these two things, you can't design the website and all that is achieved is just a bunch of unformatted text on the screen. You can't even add images to a page without HTML!

**Before starting any web development career path, you need to master coding with HTML and CSS .** The good news is that you can master one of these two types of HTML and CSS skills in just a few weeks.

The most important part is that just having one of the two knowledge of HTML or CSS will build a basic website.

1. **17 simple HTML codes** you can learn in 10 minutes

## 2. JavaScript



JavaScript allows adding tons of functionality to the site. You can even create a lot of basic web applications that only need to use HTML, CSS and JavaScript ( *abbreviated as JS* ). At the most basic level, JS supports a lot of interactive elements for the site. The use of these elements to create things like maps, interactive films and online games. Sites like Pinterest use JavaScript a lot to make the user interface (UI) easier to use ( *in fact, the page doesn't reload whenever you pin something is thanks to JavaScript!* ).

**JavaScript is also the most popular programming language in the world, so no matter what your career plan is, JS is a valuable content to learn.**

### 3. jQuery



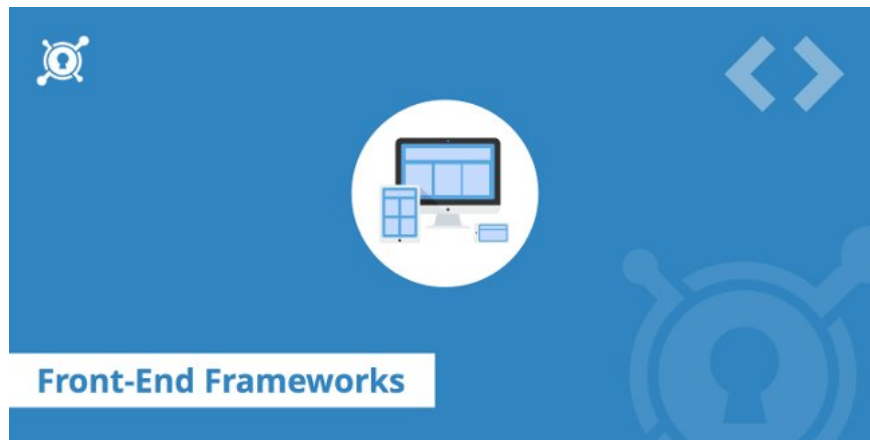
**jQuery** is a JavaScript library: a set of plugins and extensions that make JavaScript development faster and easier. Instead of coding everything from scratch, jQuery allows you to add existing elements to the project and customize it as needed ( *the reason for knowing JavaScript is important* ). You can use jQuery for things like timers *countdown* , search form autocomplete, and even grid layouts that are automatically rearranged and resized.

### 4. JavaScript Frameworks



Javascript frameworks ( including *AngularJS, Backbone, Ember, and ReactJS* ) provide available structures for Javascript code. There are many different frameworks suitable for different requirements, but the above 4 frameworks are the top priorities in meeting the requirements of today's jobs. These frameworks really help **speed up programming** and can be used with libraries like jQuery to **minimize the need to re-code from scratch** .

## 5. Front End Frameworks



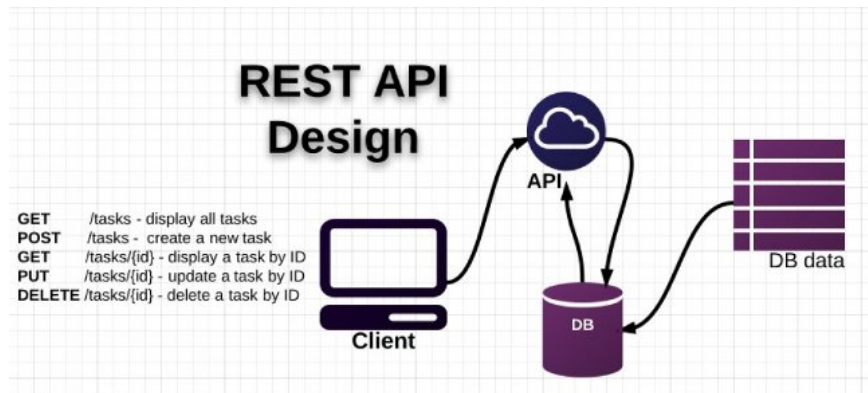
CSS and front-end frameworks ( *most commonly called Bootstrap* ) support CSS as much as what JS frameworks support for JavaScript: the starting point for faster code. Because CSS often starts with the same elements from project to project, we need a framework to define these elements in advance. Most of the current work for front end developers requires you to understand and use these frameworks well.

## 6. Experience with CSS Preprocessors



**Preprocessors** are another factor that can speed up CSS coding. A CSS Preprocessors (CSS *extended scripting language* ) adds functionality to CSS to make CSS scalable and easier to work with. It handles code before you upload it to the website and turns it into a user-friendly, formatted CSS that has a good browser. According to the current list of jobs, **SASS and LESS are in fact two preprocessors with the highest demand** .

## 7. Experience with RESTful Services and APIs

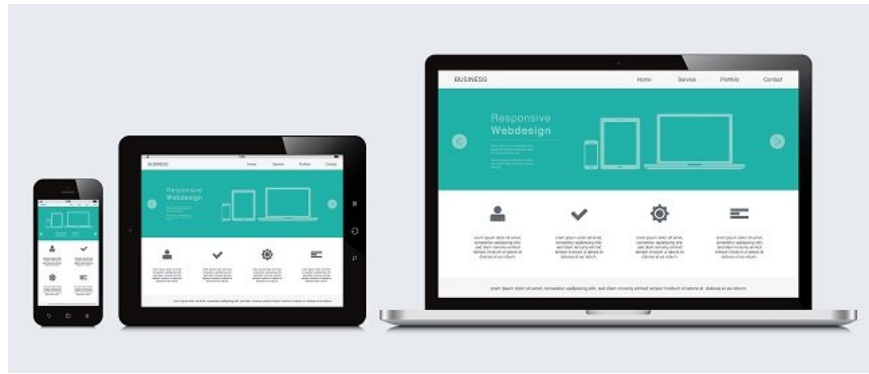


**REST** stands for **Representational State Transfer** . Basically, it is a lightweight architecture that simplifies the communication network on the website. RESTful service and APIs are web services that conform to REST architecture.

For example, you want to write an application with the goal of displaying all your friends on social networks in order to make friends, who will make friends first and who will make friends later. You can call Facebook's **RESTful API** to read your friends list and return that data, similar to **Twitter** ( *social networks also use RESTful APIs* ). The general process is the same for any service that uses RESTful APIs and only returns data is different.

This may sound a bit complicated and professional, but it is simply a set of instructions and practices that set requirements and expectations, so you know how to communicate with a web service. These guidelines and practices help make web services work more efficiently, localize the scope better, be more reliable and easier to adjust or move.

## 8. Mobile Responsive Design interface design



In the United States alone, mobile Internet traffic is higher than the amount of traffic from the desktop, so it's no surprise that **mobile design skills** play an important role in the eyes of employers. **Responsive design** is the site layout ( *and sometimes the function and content* ) changes based on the screen size and the device being used.

For example, when a website is accessed from the desktop with a large screen, users will receive multiple columns, large and interactive graphics created specifically for the mouse and keyboard used. On mobile devices, the same website will appear with a single column **optimized** for touch interaction but using the same base files.

**Mobile designs** can include responsive designs, as well as specific mobile designs. Sometimes the user experience you want when they visit the website on the desktop will be completely different from the experiences that occur on mobile. For example, a banking website with online banking will benefit from a separate mobile site that allows users to view things like the nearest bank location and check accounts ( *by small-sized phone screens*). *more* ).

## 9. Cross-Browser Development

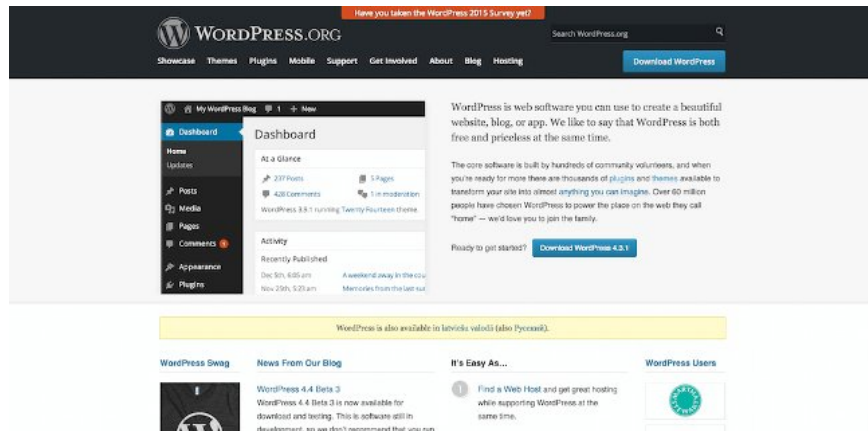
```

1 <!--
2 <head>
3 <title>RSS Reader</title>
4 <link href="/favicon.ico" rel="shortcut icon" type="image/vnd.microsoft.icon" />
5 <style type="text/css">
6 {
7   margin: 0;
8   padding: 0;
9   border: none;
10 }
11
12 .toolbar {
13   background-color: #2c99ce;
14   background: -webkit-gradient(linear, 50% 100%, 50% 0%, color-stop(0%, #2c99ce), color-stop(95%, #74b9e0), color-stop(100%, #abelfa));
15   background: -webkit-linear-gradient(to top, #2c99ce, #459dcf 76%, #74b9e0 95%, #abelfa);
16   background: -moz-linear-gradient(to top, #2c99ce, #459dcf 76%, #74b9e0 95%, #abelfa);
17   background: -o-linear-gradient(to top, #2c99ce, #459dcf 76%, #74b9e0 95%, #abelfa);
18   background: linear-gradient(to top, #2c99ce, #459dcf 76%, #74b9e0 95%, #abelfa);
19 }
20 </style>
21 <link href="/application.css" rel="stylesheet" />
22 </head>
23 </pre>

```

Modern browsers are displaying the website consistently, but there are still differences in how to "interpret" the code. Until all modern browsers work perfectly with web standards, you need to know how to make each type of browser follow its will. This is the focus of multi-browser programming.

## 10. Content Management system and E-commerce Platforms platform



Most websites are built on the content management system ( *CMS - Content Management System* ). **E-commerce platform is a specific type of CMS** . The most popular CMS worldwide is WordPress behind millions of websites (including Skillcrush!). Almost 60% of websites that use a CMS are going to use WordPress.

Other popular CMSs include Joomla, Drupal and Magento. While knowing that these will not make you professional about WordPress, they can provide you with a proper position among companies that use those systems (and there are many things on it).

Skillcrush's Freelance WordPress Developer Blueprint is a great place to learn what you need to know when you first start!

## 11. Testing and Debugging



There is one fact in web programming that can't be avoided: that's the **bug** . Therefore, you must be familiar with **testing and debugging processes**.

**Unit testing** is the process of testing each source code block ( *instructions that tell how the site works* ) and **unit testing frameworks** provide a specific method and structure to do that ( *there are other ways each other for each programming language* ).

The most common type of testing is the **UI testing** ( *also known as acceptance testing, browser testing or functional testing* ) - this is for you to make sure that the websites meet the user's needs. You can check posts to find things like HTML on a web page ( *like a user forgetting to fill in a form, the form error box will appear* ).

Debugging simply does all the " *bug - check*" errors ( *or users find out when your site is running* ), put " *detective hats* " to find out why, how they happen and troubleshoot. Different companies use different debug processes but if you are familiar with debugging then it is easier to adapt.

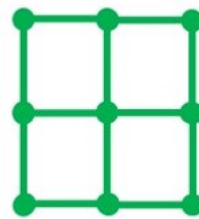
## 12. Version management system and Git



### Version Control Systems



Centralized Version Control Systems



Distributed Version Control Systems

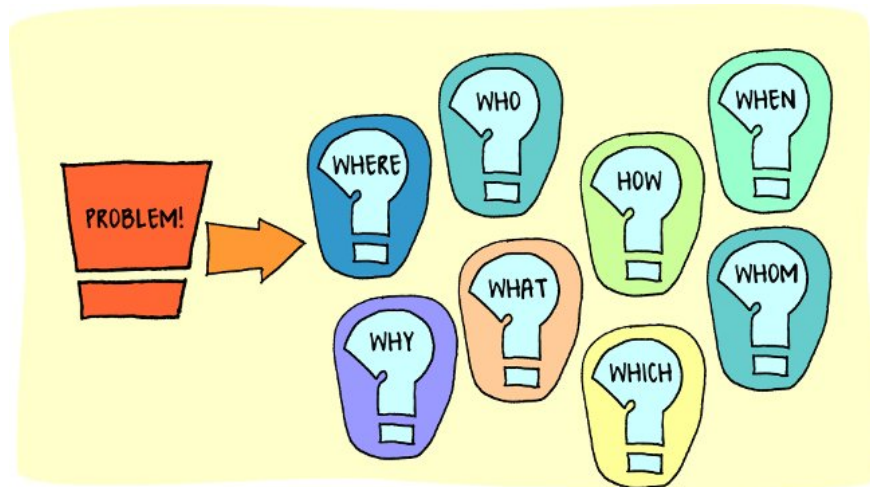
Confidential

7

*Version control systems* allow you to track changes that have been made when the code is over time; It's also easier to return to the previous version if you add something. So suppose you add a custom jQuery plugin and suddenly half of the code is broken. Instead of trying to remove it and fix all errors, you can go back to the previous version and try again with another solution.

**Git** is most widely used in version management systems. **Knowing how to use Git is a mandatory requirement in programming** . This is one of the important job skills that developers need, but few people talk about it.

## 13. Problem solving skills



If there is one thing that all UI developers need to have, regardless of job descriptions or official names, it is problem solving skills. From finding the best design implementation to fix the error, know how to identify the operation of the frontend code with the code backend being implemented . All related to problem solving skills.

For example, you have created a perfect front-end website and given it to back-end developers so they integrate it with content management systems. But then, half of your great features stop working. A good front-end developer will see this as a puzzle to solve. Of course a good front-end developer will anticipate these problems and try to stop them from the beginning.

*Author: Cameron Chapman*

See also: 9 reasons you should be equipped with a little knowledge of HTML and CSS

Having fun!

You finished reading the article "**13 skills needed to become Frontend Developer**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.