**The Best Way to Learn JavaScript Faster -** [Inthisar Hamza](https://inthisar-hamza.medium.com/?source=post_page-----90318c26785b--------------------------------) Jan 20, 2022

The 3 methods of learning a new programming language.

I am a self-taught Web Developer and I say — **the best way to learn JavaScript is to learn by yourself**. As with any other programming language, the learning process of JavaScript should be started from the **beginners** level to the **intermediate** level and then finally the **advanced** level.

What many **JavaScript beginners**make **mistake** is that they keep on **reading JavaScript** tutorials one by one, and **do not implement** any of them. This way of learning is totally wrong and futile.

Beginners should read **one tutorial then implement** what the tutorial has taught them, in their own code. This will make them **understand** the code correctly, otherwise, I guarantee they will **forget** what they just read within the next **30 minutes** or before.

**Three different approaches work for me when learning a new programming language.**

**The first method**is to look at examples of code in books or on websites. This is good because you get snippets of code to try out, but it doesn’t always provide an explanation of how each part works. You can then experiment with modifying bits of the example code to see what effect it has on the rest of the program. This is a nice way to learn one specific thing at a time without having to switch between many pages and files to figure out how everything fits together as a whole.

You can also read books about JavaScript, but this doesn’t let you experiment with the code to see what happens. For some books, though, it is helpful to read a chapter or section and then try out what you learned in an online program editor that lets you type the code and see the output without having to run a program. Most of these types of editors also let you add in your own code snippets for future reference if you think of a way to do something later on. I feel like I learn more by actually typing out examples than just reading about them because when typing I am forced to get the syntax correct otherwise the interpreter will give me an error message.

**The second method** is using tutorials that show how to configure tools such as text editors, browser developer tools, and web servers. This lets you write code to create HTML pages that load JavaScript files for testing. I like this method because you can quickly see your changes without having to make a whole new build of the program every time you want to test one small change.

You could also use tutorials that show how to set up an integrated development environment (IDE) like NetBeans/VS Code with the correct plug-ins, or install node.js and then download different versions of the required JavaScript file packages into them. Then you would be able to execute programs in real-time as opposed to just running it once after building your program into an executable file like most IDE’s require. Unfortunately, not all tutorial authors do this for their readers so some tutorials can be more difficult to follow along with than others.

**The third method** is using the debugger that usually comes with your browser’s JavaScript console. If you define a function and call it right away, then this will cause an error message under most circumstances because you haven’t defined what the function does yet so the interpreter doesn’t know how to carry out its task. You can use breakpoints in most browsers’ JavaScript consoles to stop where you want execution to halt before a function gets called so that you can check the values of variables at specific points in your program as it runs. This lets you see if certain parts of your code are causing errors by stopping at various points throughout the execution of your program and looking at things from various angles.

The first two methods are good to use by themselves, but the third method is best used in combination with one of the other two. By using a combination of all three, you can learn any programming language much faster and understand it better than if you only use one or two methods on your own.

***What should be covered on the Beginners level of JavaScript***

*The topics which you should cover are –*

1. JavaScript Syntax — [JavaScript Syntax](https://www.w3schools.com/js/js_syntax.asp)

2. Variables — [JavaScript Variables](https://www.w3schools.com/js/js_variables.asp)

3. Data Types — [JavaScript Data Types](https://www.w3schools.com/js/js_datatypes.asp)

4. Operators — [JavaScript Operators](https://www.w3schools.com/js/js_operators.asp)

5. Arithmetic Operations — [JavaScript Arithmetic](https://www.w3schools.com/js/js_arithmetic.asp)

6. Assignment — [JavaScript Assignment](https://www.w3schools.com/js/js_assignment.asp)

7. JS Output — [JavaScript Output](https://www.w3schools.com/js/js_output.asp)

8. Objects — [JavaScript Objects](https://www.w3schools.com/js/js_objects.asp)

9. Break & Continue — [JavaScript Break and Continue](https://www.w3schools.com/js/js_break.asp)

***Examples of JavaScript Programs for Beginners***

*Some programs Beginners should implement.*

1. **Create a program** where the user inputs 2 int values in 2 input boxes and you display the greatest of them on the alert box when a button is clicked.

2. **Create a program** where the user enters a number in a text box and you should display its table (up to 10) on a div. The table should be created on the button click event. Also, prevent users to enter strings on the text box (give appropriate error on the alert box when input is a string).

3. **Create a small form** for the Job Applications. It should have 5 fields — name, age, education, address, and expected salary. These 5 fields should have HTML controls where the user enters his/their information, there should be a button which on clicking shows the user information on div controls.

4. **Finding whether**a number is a **prime number or not**.

5. **Create a calculator** with common functions — add, subtract, divide & multiply

***What should be covered on the Intermediate level of JavaScript***

**1. AJAX**

a. [AJAX Introduction](https://www.w3schools.com/xml/ajax_intro.asp)

b. [AJAX Database](https://www.w3schools.com/xml/ajax_database.asp)

c. [XML Applications](https://www.w3schools.com/xml/ajax_applications.asp)

**2. Functions**

a. [Arguments object](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/arguments)

b. [Arrow functions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions)

c. [Default parameters](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Default_parameters)

d. [Method definitions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Method_definitions)

e. [Rest parameters](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/rest_parameters)

f. [getter](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/get)

g. [setter](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/set)

h. [async function](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function)

**3. Working with Classes**

a. [class](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/class)

b. [constructor](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes/constructor)

c. [extends](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes/extends)

d. [static](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes/static)

4. J[avaScript Cookies](https://www.w3schools.com/js/js_cookies.asp)

***What should be covered on the Advanced level of JavaScript***

When you are on the Advanced level, this means you basically know how to work on JavaScript. In the Advanced level, you have to deal with the integration of JavaScript with applications created on NodeJS, Python, Ruby, PHP, ASP NET, etc.

Here your job would be to make features on JavaScript that communicate efficiently with the Server technology like databases, NodeJS, Python, Ruby, PHP, ASP.NET Core, etc

**Some examples of advanced features with JavaScript are:**

1. **Partially updating parts of a web page from the database** every few seconds using AJAX.

2. **Get and show real-time stock prices** on the page. You can get the stock price by implementing the API with your JavaScript code.

3. **Creating interactive feature** on your website so that conversion rates increases. An example can be an **autocomplete**feature that you can create with JavaSript.

4. **External API implementations** like **TMDB API** in your JavaScript and helping users to get details of any movie, actor & series

5. **Infinite scroll or auto-paging feature**

**Conclusion**

Most of the time not knowing the fundamentals is the reason why developers get errors as well as not knowing why the program is running instead of giving errors as well as they get stuck on frameworks like react angular etc because of not knowing the fundamentals.

Practicing is extremely important, but at the same time, you cannot get the best of it by just applying simple tasks you find in tutorials and books. To truly master a programming language you need a long period where you are actually working in a company or real projects with your language, the thing that makes you face **real-life problems** which you will not find in books.