**Things you should know before moving to any JavaScript framework -** [Vijay Tembugade](https://vijaytembugade.medium.com/?source=post_page-----466f328f18b4--------------------------------) Nov 22, 2021

When people start their journey as a developer, everyone wants to learn everything. This is so common mentality, that every learner has! But, it is so important that we should know the basics of whatever we are learning.  
For example, before going to learn trigonometry, everyone should know the basics of angles and triangles. Before learning integration and differentiation, we should know what are the functions and logs. Uff! Too much maths. Don’t worry, I just want to say that, before diving into the sea, we should first dive into a swimming pool or lake to learn to swim.

Let’s start the journey of knowing some important things in JavaScript. I am assuming, that the reader is a bit comfortable in JavaScript or any other programming language, and also familiar with coding. JavaScript has very popular frontend and backend frameworks, such as Nodejs, ReactJs, VueJs, AngularJs, etc. To learn these frameworks, you should have a good grasp of JavaScript. So, this blog contains some of the most required concepts that we should know in Modern JavaScript, before moving into those frameworks.

**Let , Const and Var :**

This let, const and var are keywords used in JavaScript to declare a variable. Everyone has its scope and functionality. Before ES6 , var is the only keyword used to declare the variable in JavaScript.

***What is this var?***

Keyword var is a variable declaration that has global scope if it is declared outside the function and it has a functional scope if we declare inside the function. var can be re-declared and updated with the same name.

Text

Description automatically generated

this is an example of var.

In above example, value of var hello is different inside the function and outside the function.

***What is let?***

Keyword let is blocked scope, which means its value is limited within particular block of code. let can be updated but cannot be re-declared.

A screenshot of a computer

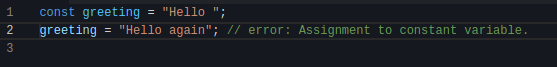
Description automatically generated with medium confidence

example of let.

In the above example, we can see , let firstname has a global value and then, in function changedName , we have let capitalName, we can see that firstname is accessible inside the function. But, let capitalName which has a block scope of function only, is not accessible outside the function.

***What is const?***

Like let , const also have block scope. But const cannot be updated nor be re-declared.(There are some cases like you can update the const if it is array or object, you can read it [here](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const).)



updating const gives an error



re-declaring const gives an error

These are some of the basics of let, const and var, that you should before diving into more things about JavaScript. To know more about it you can check out [freecodecamp blog by clicking here.](https://www.freecodecamp.org/news/var-let-and-const-whats-the-difference/" \t "_blank)

**Destructuring of object:**

Let’s understand what is an object in JavaScript. The object is something that has properties collected at one place, which has key-value pairs. You can read [**this**](https://vijaytembugade.medium.com/this-in-javascript-part-i-31ce09082f33)blog to know more about objects. But, in short, the object looks something like the below, which is representing *a student* having properties like *name, age*, and *education*.

let student = {  
 name : "Vijay",  
 age : 23,  
 education : "graduate",  
}

We know that we can access the this object and its value, as show below.

Text

Description automatically generated

We can see that we have to use a lot of lines to get value using keys. But what if, there is another method that we can use to get it done with a single line of code. That’s where, object destructuring works.

A screenshot of a computer

Description automatically generated with medium confidence

So, as in the above snippet, we can separately take the value of each key of an object by declaring them inside the curly brackets(line no.7) and we can implicitly use that values.

If we need only one value and other values are not needed the also we can destructure the object and take that value. For example,

let {name} = student;  
//here value of name will be "Vijay"

Here, we just have to take care of naming conventions that should be identified as keys in the object. [Click](https://dmitripavlutin.com/javascript-object-destructuring/) here to read more about it.

**Arrow Function**

As far, we all are familiar with functions in any programming language. JavaScript has it too. But, there are some methods or ways to declare a function in JavaScript. Let’s see, regular way to create a function as shown below.

Graphical user interface, application

Description automatically generated

regulat function declaration in JavaScript

But in JavaScript we can use function as a variable and create its instance. For example , as shown below.

A screenshot of a computer screen

Description automatically generated with medium confidence

function declaration with help of variable

But, still, don’t you think it is confusing? Yes! But, yeah, we can get rid of function keyword from the above example, and still out code will work. How? That’s where the arrow function comes into play. Instead of function keyword , give arrow after parameters.

Text

Description automatically generated with medium confidence

arrow function in JavaScript

So, we can see, we put arrow after passing parameter and everything works fine. That’s what we call an arrow function.

Now, if we have only a single line inside the function we can get rid of that return keyword and brackets also. Your function will look like this.

A picture containing text, clock, screen, dark

Description automatically generated

arrow function without return

This is how the arrow function looks like, most of the time we create an anonymous function in JavaScript as the callback, that way, this arrow function plays an important role. Subsequently, it has various other properties that regular functions don’t have, you can check that out [here](https://www.digitalocean.com/community/tutorials/understanding-arrow-functions-in-javascript#:~:text=An%20arrow%20function%20expression%20is,%E2%80%9D%20syntax%20%28%20%3D%3E%20%29.&text=Like%20traditional%20function%20expressions%2C%20arrow,to%20name%20an%20arrow%20function.).

**Template String (Template literals)**

Template Strings are strings which are denoted inside of backticks. `` (Mostly button below Esc on keyboad)

In this literals , we can access the variable by putting them inside ${} simply.

let name = "Vijay";  
let age = 23;  
console.log(`My name is ${name} and I am ${age}.`)  
output:  
My name is Vijay and I am 23.

We can see that strings with backticks can access the variable or value declared simply by putting them inside $ and {}.  
Click [here](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals) to read more about it.

**Import and Export**

Building any software takes a lot of time and a lot of code. But, It is heavy to handle the entire code of software written inside a single file. So, the major solution for this kind of problem is dividing such a large amount of code into a smaller function. But, still, entire functions in one single file are not a feasible way to write a program. Here, the concept of modules comes into play, in JavaScript.

For example, we can write a code in one file and we can use that in another file. JavaScript provides us the environment to do so. Let’s look into an example.

Let’s create two files called index.js and addition.js  
addition.js will have logic for addition of two numbers and console.log them and we will take that addition function into index.js and execute it.

Graphical user interface, application

Description automatically generated

Two files.

Now , create an addition function in addition.js file and export that function inside of curly brackets as shown below.

Text

Description automatically generated with medium confidence

addition.js

Now, we can import this addition function inside index.js and pass the parameter into it.

Graphical user interface, text

Description automatically generated

index.js

In this way we can use , import and export in JavaScript. There is one more thing which is used with default keyword.

Exporting something as default will export that default module. You don’t have to use curly bracketsto import and export for default cases.  
Let’s see this by example.

Graphical user interface, text

Description automatically generated

addition.js

In above addition.js , you can see, multiply function is exported but addition function is exported as default. So, it means , when we import addition.js in any other program first thing , it will by default take, in addition function.

Graphical user interface, text, chat or text message

Description automatically generated

index.js

Here, we imported addition and multiply. But as you can see, we have not used brackets for addition. Because it is exported as default. By here, if you are doubting that, if name while importing of default export can be changed, and will it work? Yes. You are going well. You can name anything for default export and it will work but this is not the case in ordinary export. For example,

A screenshot of a computer screen

Description automatically generated with medium confidence

index.js

Here, we imported addition module as add, and as addition is default export it will get assigned to add and it will work perfectly. But it will not work in case of multiply.

This is how, export and import works in JavaScript. For daily use in JavaScript, and to get more about it, you can [click here](https://javascript.info/import-export) to read more.

***Note****: We need to do changes in package.json to work with import-export of ES6 . We have to add , “type”: “module” in package.json . And please note that ES5 has a different syntax for importing exporting modules.*

**Spread syntax:**

Spread operator is denoted by ... (three dots). It has much significance while using in JavaScript but we are going to understand the basic rule of it. It is used with iterable things like object and arrays. It can provide a function call with an array (or any other iterable) where 0 or more arguments were expected.

Various use cases of spread sysntax as of follows:

* ***Concatenation of Array.***

Adding or merging two arrays or object is a basic thing n any programming language. We can use spread operator to so. For example. Let’s declare a variable city which is an array of city name. Now, we have to another city inside that array. we can use spread on city and then add another city name as shown below on *line no. 3*. We can also push elements inside the array using a similar method.

Graphical user interface, text

Description automatically generated

concatenation of array using spread

* ***Concatenation of Object.***

Now as we used it on the array we can also use it on object and add key-value pairs in an object. For example, let’s declare an object called student having keys and value are as {name : “Sam”, age : 22} . Now we have to add {mark : 100} in this object. We can use spread operator to do it. As shown below, mark property is added in object by spreading student object as …student this and adding mark : 100 in it.

Graphical user interface, text, application, chat or text message

Description automatically generated

concatenation of object using spread

* These are some basic cases of spread syntax. We can use it in passing parameters inside a function and copying an array. To know more, read [this](https://www.freecodecamp.org/news/an-introduction-to-spread-syntax-in-javascript-fba39595922c/).

**Map and Filter**

Map and Filter are both the most important and most used methods in JavaScript. Mostly, these are used with iterables like objects, arrays, or strings. These are non-destructive methods, which means the original value doesn’t get affected, but it creates a new value. Let’s see those with examples.

Suppose , we have one object called students which has two properties callled name and age.

Text

Description automatically generated

students Object

***Case I : Create an array which content all name of students in upper case format.***

Here, we have to create from given values and we have iterate and modify the values and fetch something from a given object. So, here, we need to use MAP method.

So, we have to use student.map() method and return the names with putting upper case on it.

A picture containing timeline

Description automatically generated

Map method

Above we have declared variable names and iterated over the student’s object and passed a callback function in it. student which is inside a map function is an individual value on the main object and we are iterating over it object and returning it.

***Case II : Filter out the students who are below of age 18.***

Here, we have to filter out and check the condition. Filter method is used to do it.

Text

Description automatically generated

Filter Method

The filter method checks the given condition and returns the values which are true as asked(depends on condition).  
In JavaScript there are other methods are also like reduce, find, some, every, etc. Look at the following image to take a brief grasp of that method.

Graphical user interface, icon

Description automatically generated

These are some important concepts we should know, before moving to any JavaScript framework for beginners. Along with this, there are other things too that, I’ve put below as a checklist for understanding JavaScript deeply, which are most used when we move towards beginner to advanced level.

* Fetch Api
* Async/ Await
* Primitive vs Reference Data Types
* Date and Time Concepts
* Try and Catch
* Promise and CallBack
* JSON data
* Local Storage