React

It is a Javascript library used to develop single page applications, it follows component based approach to develop the UI.

Single page application

It is an application where you can see everything in one page, if user clicks on a button or a link or any element only part of the page will be loaded instead of loading the entire page.

ex: Facebook, Git, Twitter, Git hub and so on.

Component: These are reusable UI’s which you can see in the page and they can be nested in other components, you can develop and maintain the components independently and fit the components wherever you need.

Note: React.js is used only to develop User Interface.

Advantages of React.js

1. It is easy to learn and understand
2. It uses components which are easy to maintain and test
3. It can be used to develop UI’s for many platforms like Mobile, Web and DVR.
4. It is a library and encourages you to add any javascript libraries you wish ex: axios to call back end, bootstrap js to style or materials to style and so on.
5. It provides javascript libraries to perform unit testing.

Pre-requisites

1. HTML
2. CSS
3. Javascript
4. New Features of Javascript (ECMAScript or ES6 and latest)
5. JSX (Javascript XML, extended version of Javascript).

Softwares required

1. Visual Studio Code editor
2. Node.js (Javascript Runtime Environment)
3. Browser

React.js is developed using JSX, which is an extended form of Javascript.

Why JSX

JSX is easy to write compare to Javascript when comes to developing UI’s i.e., writing HTML code is easier in JSX compare to Javascript

Brief Overview of Contents

* React.js
* JSX
* ES6+ Features
* VDOM
* Components
* Controlled Components
* Props & States
* HOC
* Render Props
* Advanced concepts
* Router
* Axios
* Redux
* Testing
* Materials
* Javascript or Node.js libraries i.e., Express

Understanding the new features of Javascript i.e., ES6+

ES6 onwards the javascript syntax has slightly changed.

ECMAScript or ES

* It is standard that will have set of rules & regulations which is implemented by Javascript, Node.js, Browser
* It has different versions like ES6, ES7, ES8 and so on
* In each versions new changes are added.

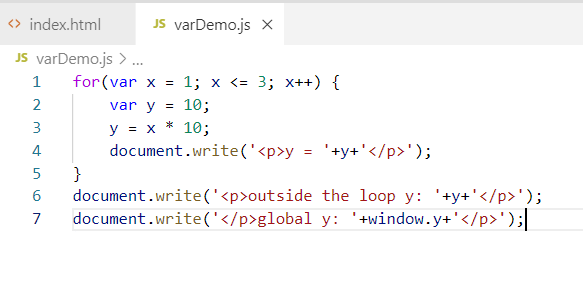
What are the main changes introduced in ES6 or later

* New keywords in javascript like let, const, class, constructor, super
* Template String literals
* Arrow functions
* Static members
* New functions in Object like entries & value
* New functions in String like padStart & padEnd
* Generators
* Object Destructuring
* Rest & Spread operators
* Default arguments
* Optional Chain
* Trailing commas
* Exponential Operator
* includes in Arrays

Understanding let & const keyword

let & const keywords are used to create block scoped variables, because earlier javascript had var keyword to create variables which would become global variable when you create.

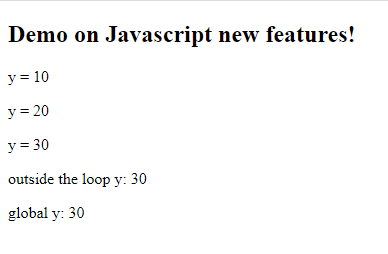
varDemo.js



index.html



Output:



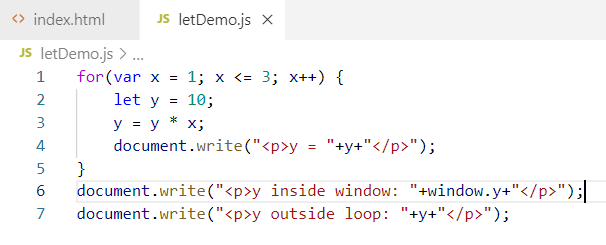
Here the problem is you can access the variable which is declared inside the loop i.e., outside the loop also they are accessible, which is not secure, hence we have let & const to declare variables.

let & const variables wouldn’t be added to window object.

let is used to create block scoped variables & you can modify it within the scope, but you can’t access outside the scope

const is used to create constants, it is also block scoped variable, but you can’t modify it.

letDemo.js

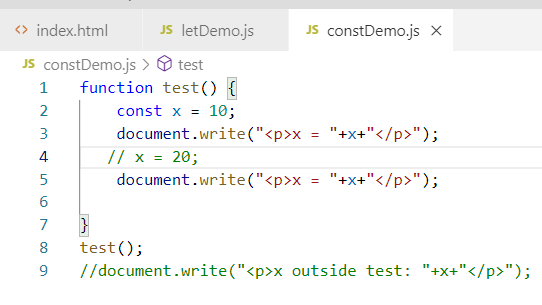


Output:



Since let keyword doesn’t add the variables inside the global object, the window.y gives undefined & also accessing the ‘y’ outside the scope gives error in the browser console.

constDemo.js

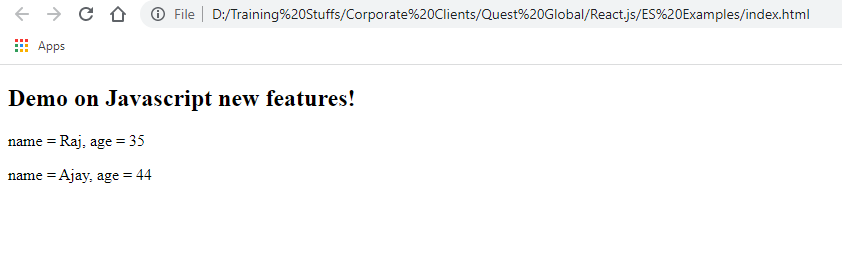


The above code gives error at the line no. 4.

But the object properties are not constants.



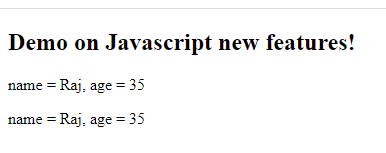
Output:



The above code lets you modify object properties though it is const, but to avoid modifying properties you need to freeze the object using Object.freeze.



Output:



The above code makes the entire object freeze so that you can’t modify this was there even before ES6.

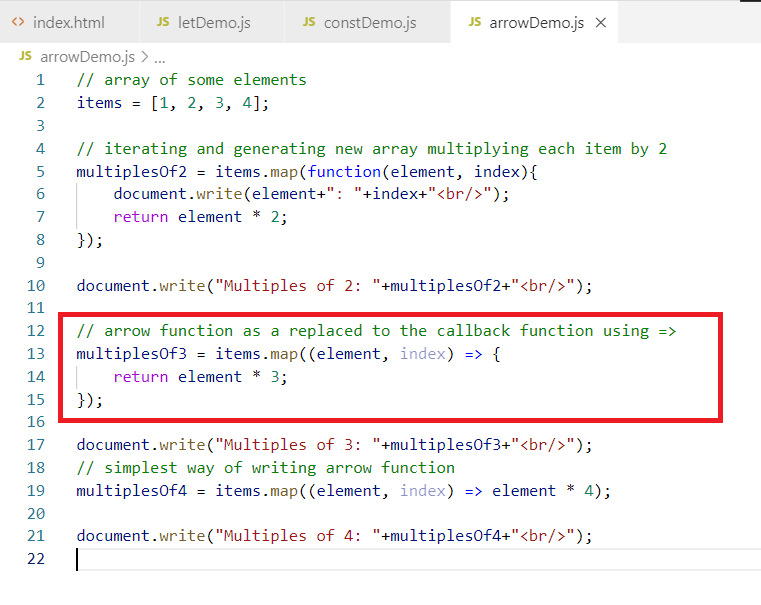
Differences in var and let

|  |  |
| --- | --- |
| var | let |
| It is global | It is block scoped |
| by default variables without any keyword takes var | by default variables doesn’t take let |
| You can redeclare same variable | You can’t redeclare same variable |



Arrow functions

These are used to simply writing callback functions with a simpler syntax that eliminates using function keyword, {} if it’s a single line statement, return keyword if it’s a single line statement.



Note: If in case the callback has only one line statement then you can avoid writing {}.

Valid arrow functions

() => “Hello World”: This takes no arguments & returns string

(x, y) => x + y: This takes 2 arguments & returns addition of 2 arguments

(x, y) => return (x + y): Not valid, use return inside { }

(x, y) => { x + y; }: This is also not valid, you must return the expression or create some valid statement, since in javascript a function need not to return always.