Node.js

Software’s requirement

1. Node.js
2. VS Code

Javascript is used to develop websites that adds interactivity to achieve dynamic behaviour.

Javascript fundamentals

1. Functions
2. Variables
3. Operators
4. Loops
5. Conditional Statements
6. Objects
7. Arrays

Variables: You can use let or const keywords to create variables, avoid using var keyword

var x = 20; // not recommended

let y = 30; // block scoped, but you can modify let variables

const z = 40; // block scoped, but you can’t modify const variables

Functions: In Javascript you can create functions using function keyword

function getUsers() {

…..

}

function multiply(a, b) {  
 return a \* b;   
}

multiply(20, 30);

Arrays, forEach & map



forEach: It is only to iterate

map: it converts an array to another array.

Handling events in Javascript

DOM (Document Object Model) generates some events when you interact with the HTML element, then you can perform some task for those events which is called as event handling, you can call some functions based on the events.

document.getElementById(“id”): This is used to access an HTML element based on the id.

document.getElementsByTagName(“tag”): This is used to access all the elements having same tag name, it returns array of elements

document.getElementsByClassName(“class”): This is used to access all the elements having same class name, it returns array of elements

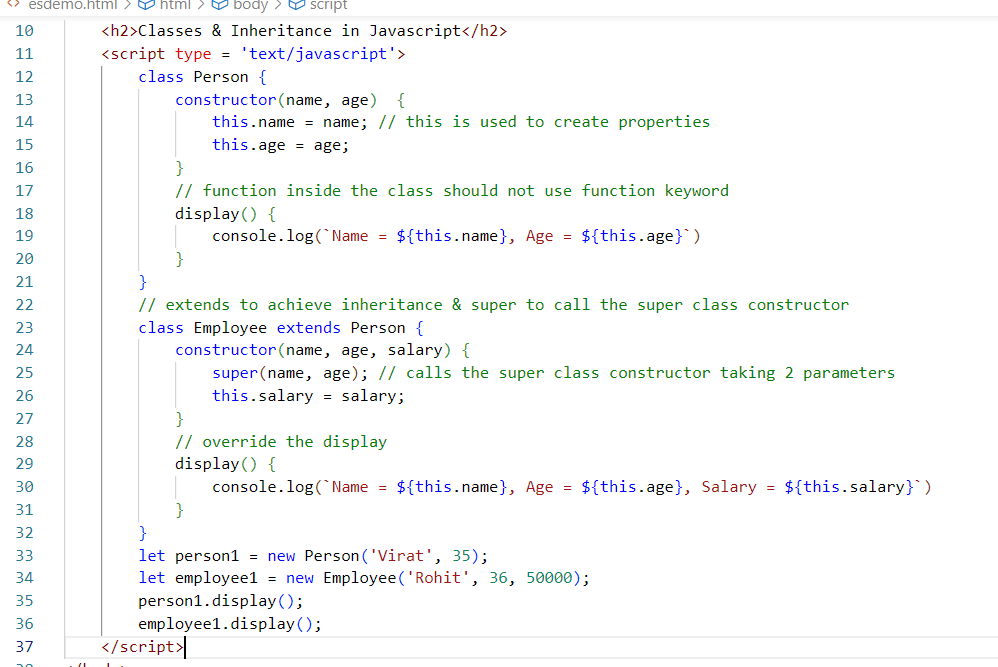
document.querySelector(“id or class or tag name”): This is also used to access the elements having either an id or class name or tag name

New features of Javascript

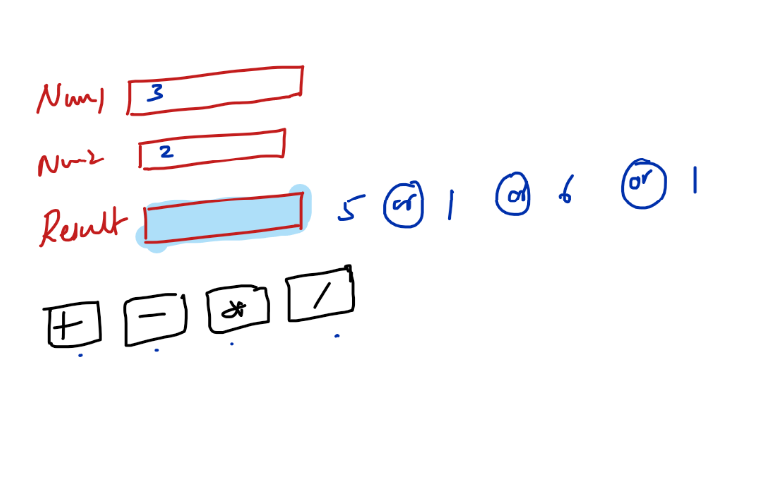
ECMA Script is the standard to specify the features for the Javascript, in ES6 they introduced lot of changes in the syntax to simplify the javascript writing

1. let, const keywords
2. class, super, extends keywords
3. arrow function – simplified form of writing the callbacks
4. template strings – back tick with ${expression}
5. rest & spread operators – to work with multiple parameters in an easy way
6. generators – to get multiple values from the function without completing the function execution
7. optional chain
8. exponential operators

Classes & Inheritance in new feature



Activity



Arrow functions

It simplifies writing callback functions by removing lot of things in the callback like function keyword, { } if its one line statement, return statement if its one line statement

Callback:

function(a, b) {   
 return a + b;  
}

Arrow function

(a, b) => a + b

(or)

(a, b) => { return a + b; }

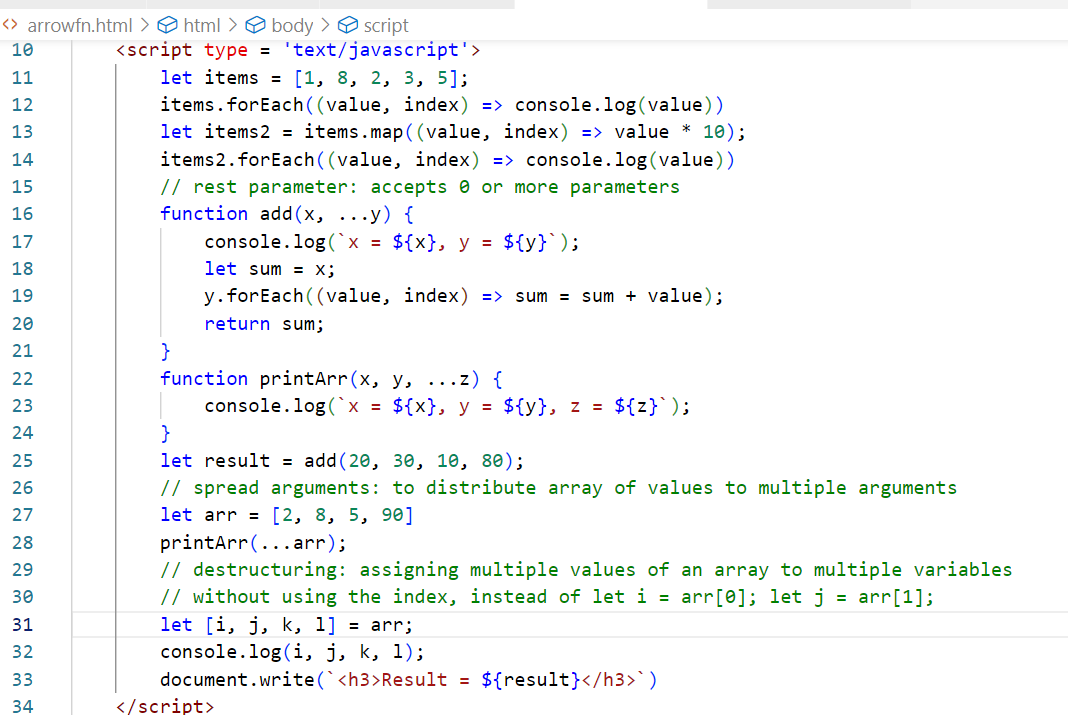
Using in forEach or map function

let items = [2, 1, 5, 4, 3];

items.forEach( (value, index) => console.log(value); }

let items2 = items.map( ( value, index) => value \* 10);

Rest, Spread & Destructuring



Rest: to pass 0 or more values

Spread: to distribute multiple values to multiple parameters

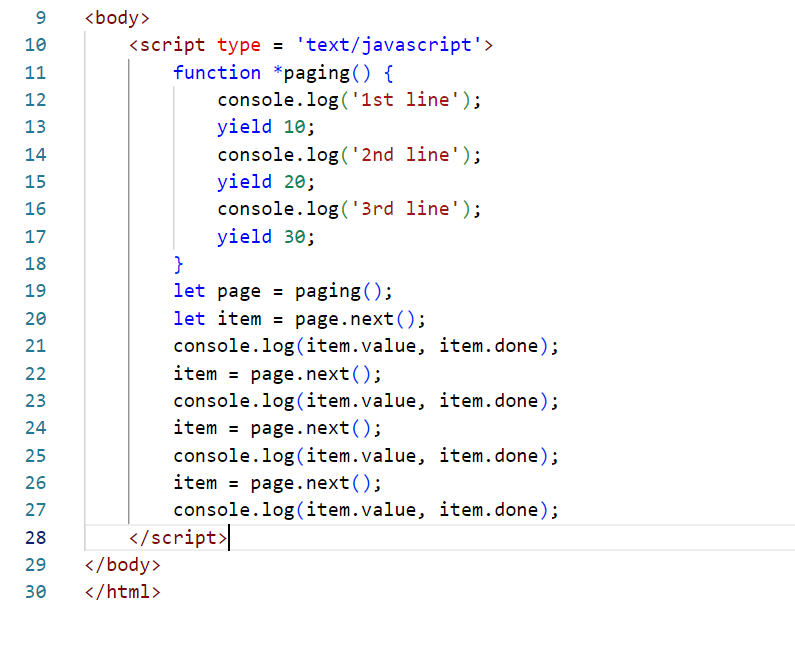
Destructuring: to assign values to multiple variables in a single line

Generators:

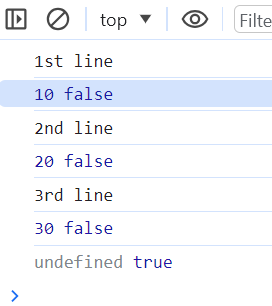
They can return multiple values on a single function call on installment

function \*foo() {  
 calling backend   
 yield data;  
 calling backend  
 yield data;  
 calling backend  
 yield data;  
}

let f = foo();  
f.next();  
f.next();  
f.next();



Output:



Optional chain

It is to access the nested property in an easier without checking for undefined condition.



Exponential operator

2 \* 3 = 6

2 \*\* 3 = 8

2 \*\* 4 = 16

Asynchronous actions

It initiates the action now but finishes later

setTimeout(callback, timing);

Ajax calls using XMLHttpRequest : call the backend api’s

let xhr = new XMLHttpRequest();

xhr.send( “api/posts” );

xhr.onreadystatechange = function() {   
 // updates the document when the response comes  
}

Promise: It is used to perform an asynchronous operation whose result could be either successful or failure

promise  
 .then( callback )  
 .catch( callback )