Angular Framework

What is angular

Used to develop client side applications for web, mobile, desktop and native applications.

It is mainly used to develop User Interfaces to create single page applications.

Single Page Application (SPA)

Everything happens in one page and any changes you do will update only part of the page instead of reloading the entire page,

SPA is much faster compare to multiple page applications, because SPA doesn’t need to pull the changes for the entire web page instead it has to pull content only for the part that needs to be updated.

Softwares required

Node.js - Runtime environment to run the angular applications

Editor - Visual Studio Code

Angular mainly uses two important technologies

* HTML
* Typescript

Typescript is a superset of Javascript it detects errors early and more reliable than Javascript

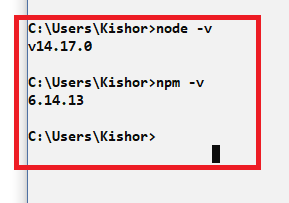
Inorder to develop angular applications, we need to install an Angular toolkit called as angular/cli

Angular CLI: It is a command line interface which provides commands using which you can create, run & build angular projects.

angular cli is downloaded from the internet which will have node modules which are javascript libraries, these node modules you can download only if you have node.js installed, node.js gives you one command called npm (Node Package Manager)

NPM: It is a tool to download any javascript libraries including Angular/CLI, React Toolkit.

Verifying node & npm



You can install angular/cli using the following command

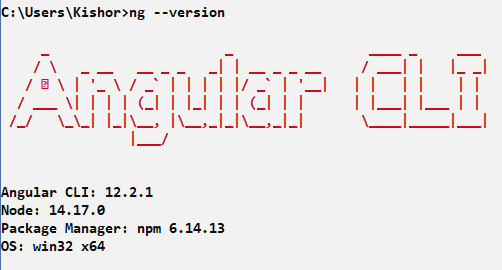
* npm install @angular/cli
* npm install -g @angular/cli

npm install @angular/cli: It installs angular toolkit locally, in the location your terminal is opened, you can create angular projects or run angular projects only in that location

npm install -g @angular/cli: It installs angular toolkit globally, you can create/run/build angular projects in any location

ng: this is command you will get once you install angular/cli, this command allows you to create angular projects, run and build the angular projects

Verifying the angular cli



Some of the useful commands in angular through ng

ng new app-name: it is used to create a new angular application, where app-name is the project name

ng serve: it is used to host your angular application & runs in a default server provided by angular in port 4200

Other than this you have commands like

ng generate component component-name (or) ng g c component-name

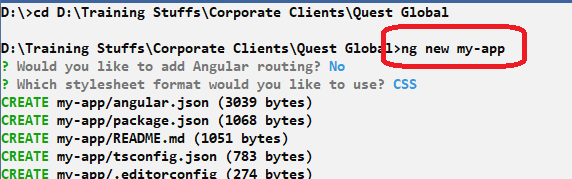
ng generate service service-name (or) ng g s service-name

ng build

ng test and so on.

Creating our first angular project

ng new my-app



After installation is complete, you will get a folder which is an angular project, you need to navigate inside the project from the terminal so that you can run your angular application or create angular programs.

What will the angular project have?

The angular project will have many features

* It will have auto-compilation feature which automatically compiler your angular application when you update the changes
* It will have an embedded server to host the angular application
* It will have a live-reload feature, which shows the output when you do changes in the application
* It will have all the supported Javascript libraries (node modules) required to develop angular application

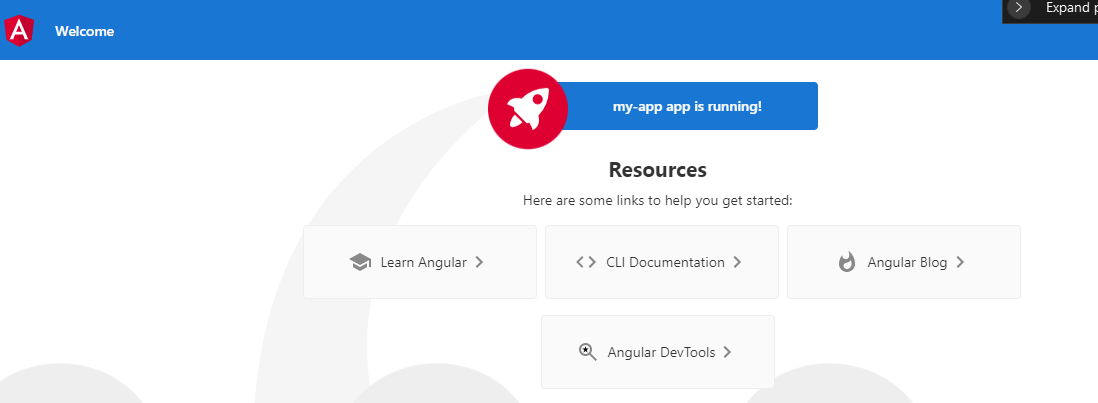
Note: Once you create the angular project using ‘ng new’ you will get a ready to run project with some default content

You can verify the project download completing by launching the application using ng serve

Note: *ng serve* should be entered from the project location i.e., parent directory, in my case it is my-app

Once you enter ng serve the command prompt shows compile successfully and angular application is running in port 4200, you can test the application is running in browser by entering <http://localhost:4200>

Output:



The above output is common for every new project which we need to change as per our requirement.

Summary:

* Verified node & npm using node -v & npm -v
* Installed angular/cli using npm install -g @angular/cli
* Created a new project ng new my-app

Note: Installing angular/cli is one time job

Angular uses two languages mainly

1. HTML - used to develop content for web pages
2. Typescript - used to write application code, which is a super set of Javascript

Javascript: It is a program written for web pages to make your web page more interactive, Angular uses Typescript which is a super set of javascript that is compiled by angular to convert to the javascript

Note: Angular converts Typescript to Javascript so that every browser can understand

Typescript make use of lot of new features of Javascript i.e, ES6 features

ES6 is also known as ECMAScript2015, released in 2015, ES6 is a standard which provides some rules that is implemented by Javascript & all the browsers.

ES6 has provided some features to improve the syntax of the Javascript to easily write the program, earlier Javascript used ES5 feature whose syntax were bit hard to write & understand.

Some of the features of ES6

* Declaring variables using let & const keywords
* Introduction of classes, extends, super keywords to make Javascript object oriented
* Rest & Spread operators
* Default function parameters
* Object Destructuring
* Arrow Functions
* Template string literals

Browsers understand 3 technologies

* HTML (.html)
* CSS (.css)
* Javascript (.js)

You can include css & javascript into HTML and open HTML in the browser to see the output

Purpose of let & const keywords

These are used to create block scoped variables in Javascript, earlier before ES6 javascript variables were global it means it can be accessed anywhere in the program even if the variables are created inside the loops or conditions or functions.

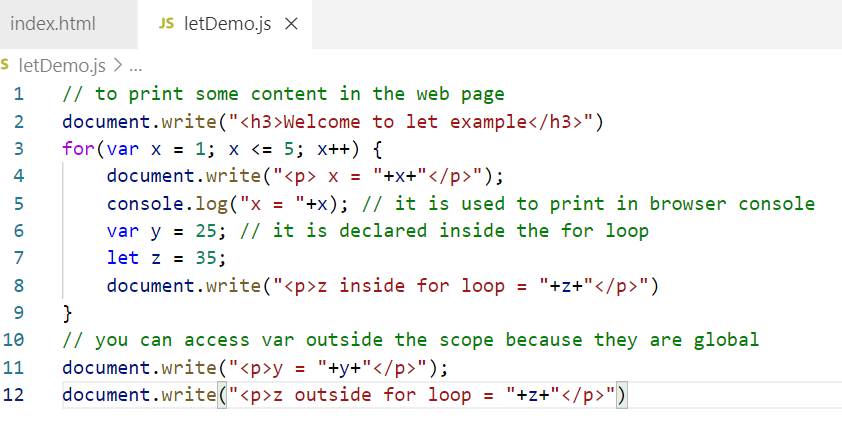
How earlier javascript variables were created

Using var keyword

var username = “Alex”;

var age = 35;

letDemo.js



index.html



Output:

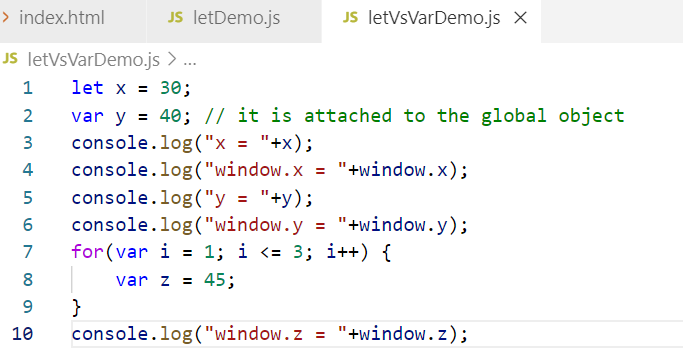


You can notice that let variable is not accessible outside the scope, the reason var variables are accessible is because it is attached to the global object of the program i.e., window

You can access the var variables using window object also, because these are added as a property of global object

ie.., window.y

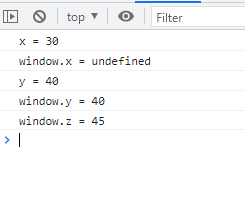
letVsVarDemo.js



index.html



Output:



the var is attaching the property into the window object hence they become global, but let variables are not added to the window object.

const keyword

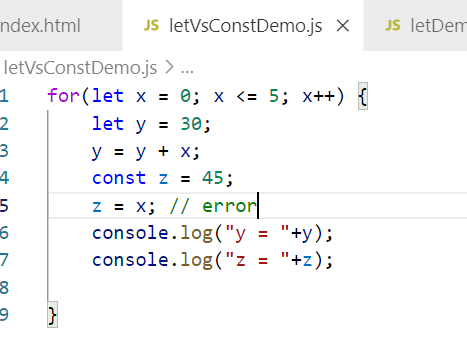
const is another block scope variable you can create, but its value can’t be modified, it is read-only once declared

const PI = 3.14

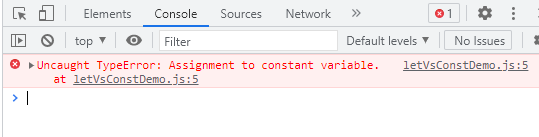
PI = 4.4; // error

let x = 30;

x = 45; // OK



Output:



var Vs let Vs const

var will become global and you can re-declare the var variables

i.e.,

var x = 30;

var x = 40;

let & const are block scoped variables, let can be modified however const can’t be modified, re-declaration is not allows in both let & const

let x = 40;

x = 15; // OK

let x = 50; // error

const y = 10;

y = 5; // error, trying to modify the const

const y = 10; // error

Using const for Javascript objects

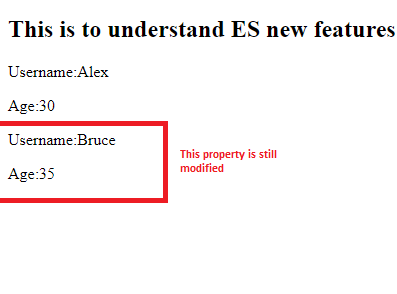
const emp = {name : “Alex”, age : 35, salary : 40000};

emp = {name: “Bruce”, age : 40, salary : 45000} // error

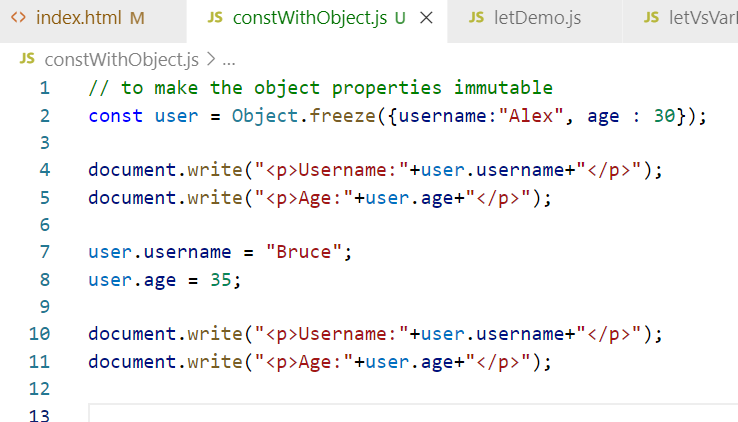
The const keyword restricts modification, however you can still change the value of the object.



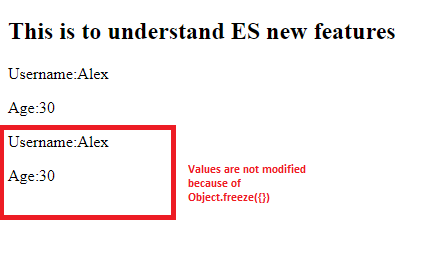
Output:



You need to use Object.freeze({}) to make the object properties immutable



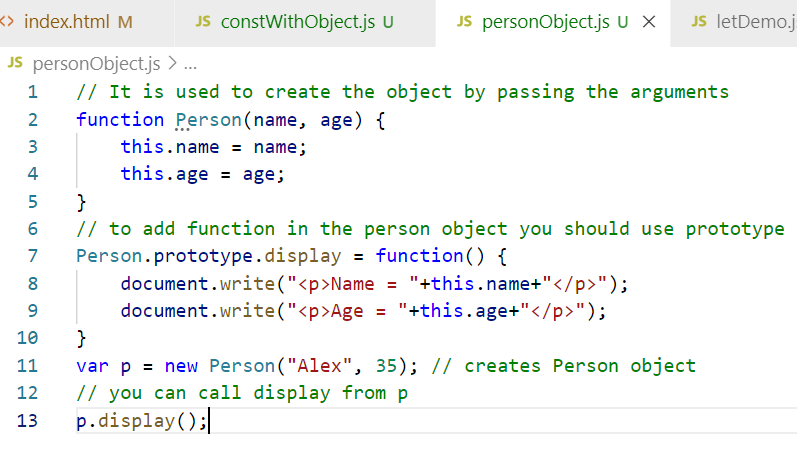
Output:



How to add methods to the object

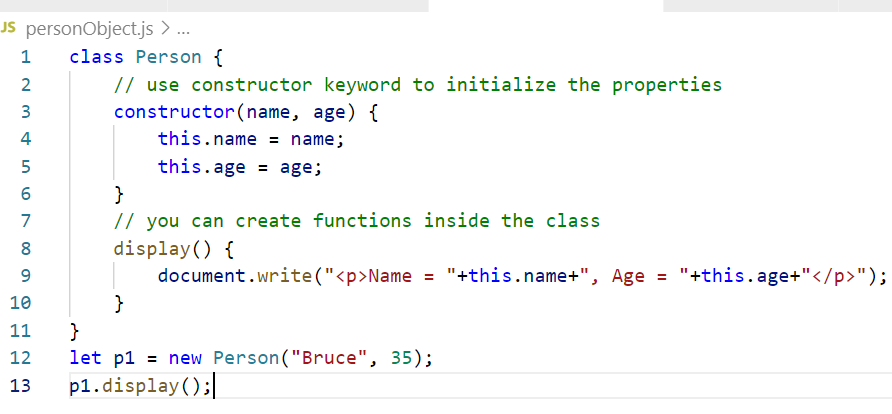
Earlier you need to use prototype property to add the methods to the javascript object

Before ES6 you need to use prototype to create function for the object



But from ES6 onwards you can use class keyword to create the constructor & function inside the class.

From ES6 onwards you can create class as below



Inheritance: Process of acquiring the properties & behaviour of an object from another object

Before ES6 you need to use prototype to achieve inheritance

function Person(name, age) { }   
Person.prototype.display = function() { }

function Employee(id, name, age) { }

If You want display() function in employee then you need to inherit using prototype as below

Employee.prototype = Object.create(Person.prototype)

e = new Employee(1, “Raj”, 35);

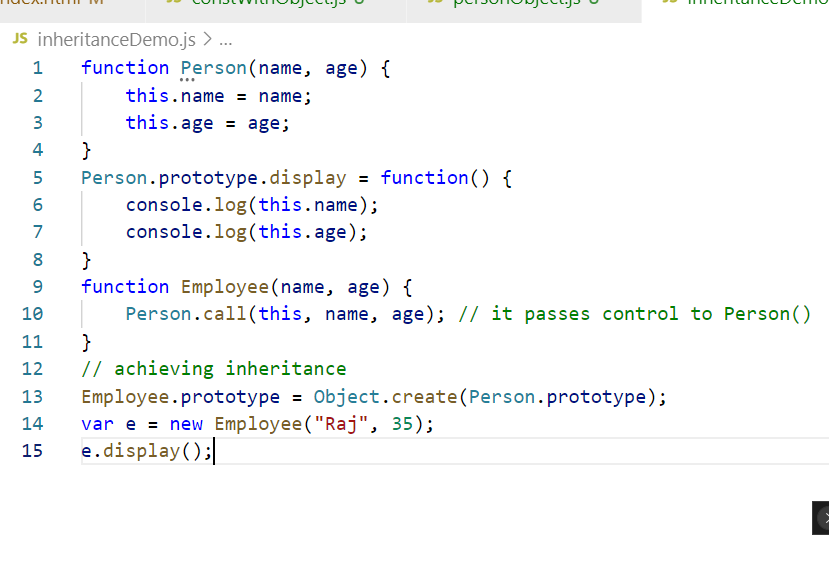
e.display();

From ES6 onwards you can use extends keywords

class Person { … }

class Employee extends Person { }

Old approach of inheriting the functions



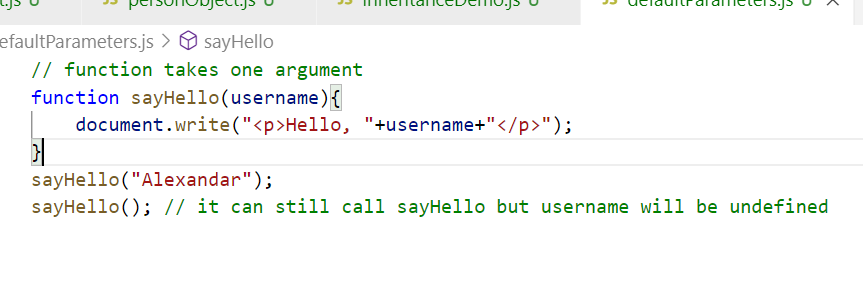
New approach of inheriting the functions



Default parameters to the function

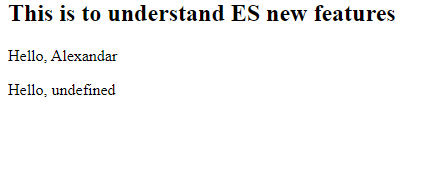
In Javascript you can call the function having arguments without passing right number of arguments, in that case you don’t get error instead the missing parameters will be undefined

Ex:

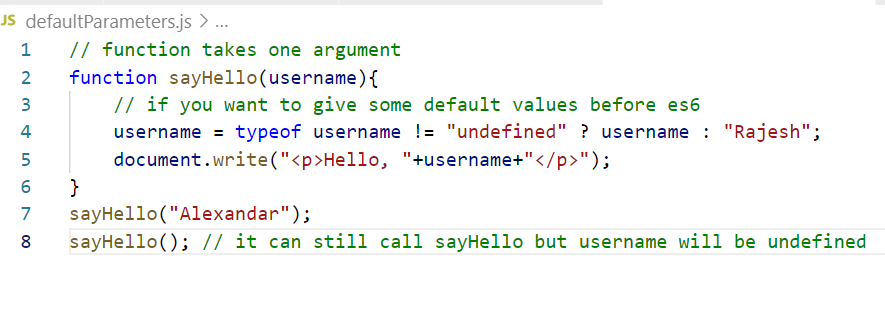


Here you are calling sayHello() without passing parameter in that case username will be undefined.

Output:

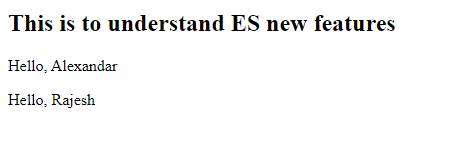


Earlier to ES6 developers need to write some logics to avoid undefined



The above code checks the type of username is undefined or not, if yes then assigns the default value Rajesh, if not then assigns the value you passed as the argument.

Output:



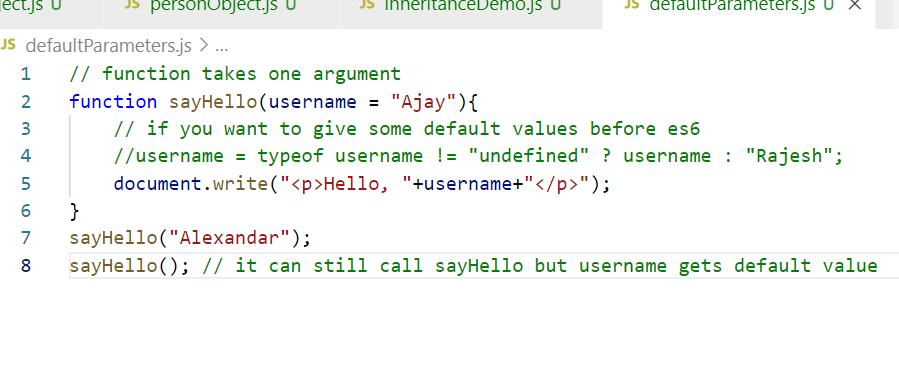
But in ES6 its much easier to provide the default values, i.e., in the parameters itself you can assign the value to the parameter so if the argument is passed then the parameter takes the value passed in the argument, else the parameter takes the default value.

function sayHello(username = “Ajay”) // default value is Ajay  
{   
}

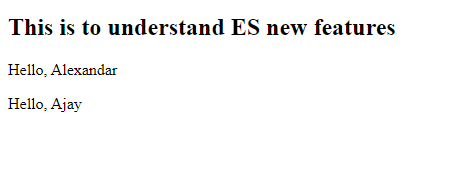
sayHello(“Kiran”); username will be Kiran

sayHello(); username will be Ajay

Default parameters in ES6 is much easier as below

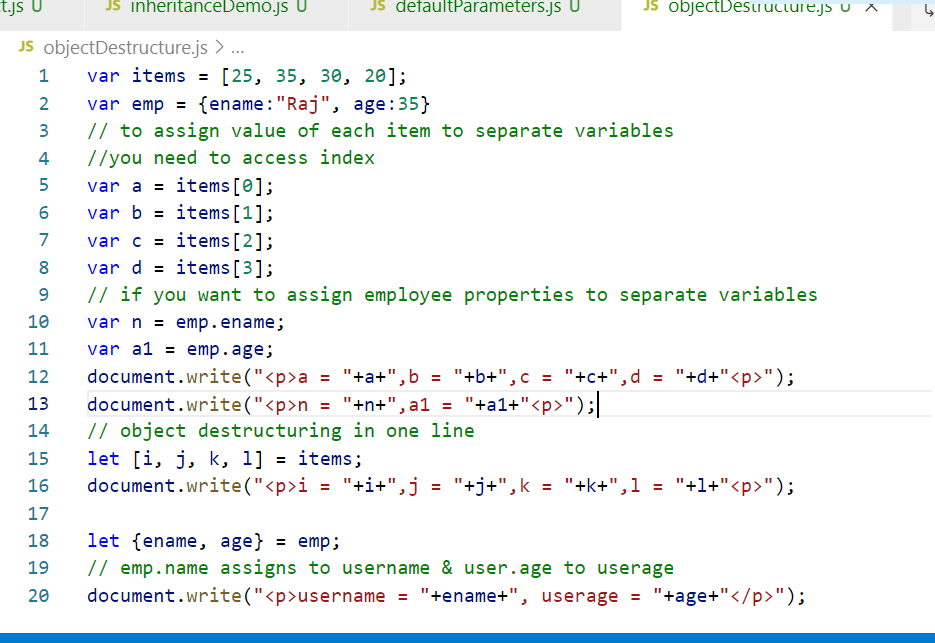


Output:



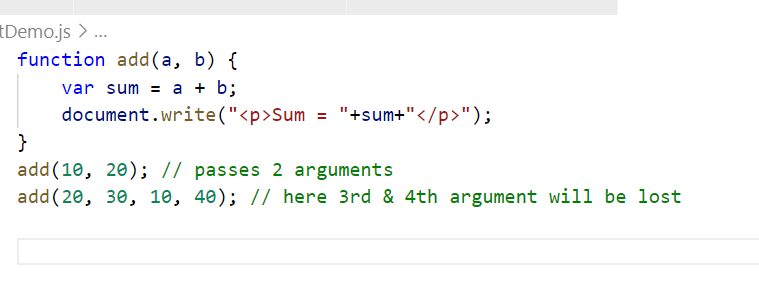
Object Destructuring

It allows you to assign values to multiple variables without accessing each index of a complex object



Rest Operators

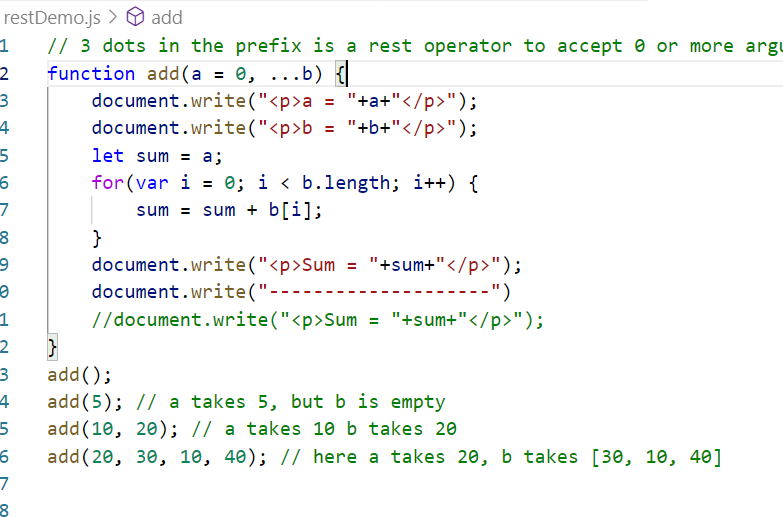
Before ES6 a function can only take specified number of arguments based on the number of parameters



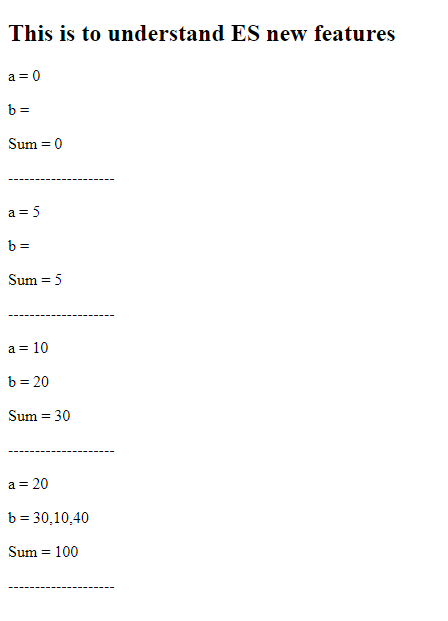
In ES6 you have a rest operator where a variable can accept 0 or more arguments like an array

With Rest operator you can avoid losing the data

restDemo.js



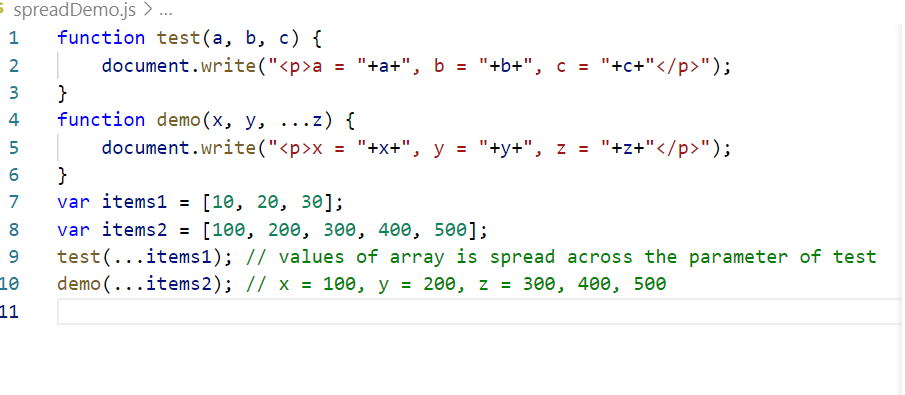
Output:



Spread Operator

It is used to spread the arguments to multiple parameters of a function

spreadDemo.js



Output:

