Functions

Function: Function is a block of code designed to perform a particular task. When we invoke it then it will execute. It helps to reuse the code for number of times by passing different arguments to get different results when we want.

Syntax: function funcname(parameter) {body};funcname(arguments);

* Function deceleration by statement follows hoisted.

Hoisted: Hoisting is a js default behaviour of moving deceleration is the top of the current scope.

How to give input and return value to function:

function func(x){

return x\*5;

}

let a=10;

res=func(a) (->calling of function or caller)

console.log(res)

call by value: it means it transfer only value (10)not full variable(a) to parameter of function(x).

call by reference: it transfer the full variable it means original variable can modified

example: let a=5;a=10;now a value modified to 10

returning function: after define the function, when we call the function then only function will execute . After computing the function the return value is returned back to the caller. Then we stored the return value in a variable for further use.

Function deceleration by expression:

In js function can also be define using an expression. Here function expression can be stored in variable.

Syntax: let f1 =function (x){return body}; let res=fi(5);

Here above function is actually anonymus (it has no name).now onwards function can be stored in variable(f1).

Arrow function: Arrow function is the function that can write expression of function in one line. Arrow function are not hoisted means they must be defined before they are used.

Syntax: let f1=(x)=>x+1

Self invoking function: function expression can be made “self invoking”, function deceleration by statement can not make “self invoking” (self calles). Function calling by expression called (invoked)automatically with out being called .

Function deceleration by expression called automatically when add ()at last of function

Syntax: (function (){body})();

Self invoking function is anonymous (with out name).

Parameter and arguments its rules: function parameter are only the names listed in function definition.

Function arguments are the real value passed to the function.

Rules: js function definition don’t specify data types for parameters and arguments.

If the function is called with missing arguments the missing value assign with “undefined”.

Example: function func(x,y,z){body}func(2,3); here z assigns value of undefined.

Rest and spread:

Spread (…array): spread operator is used to spread all the elements in the array as individual elements.

Syntax: function func(x,y,z){console.log(x,y,z)}

Let a=[1,2,3,} func(…a) output is 1,2,3

Rest(…array): rest parameter is used to convert individual arguments in one array.

Syntax: function func(a,…b){console.log(a,b)}

Let a=[1,2,3,4,5] func(…a) output is 1,[2,3,4,5]

Call back function: it is used passed a function as arguments to another function. Call back function runs after higher order function finished.

Syntax: function f1(f){f()}

function f2(){console.log(“hai”)}

f1(f2) out put: hai

Nested function: one function defining in another function is called nested function.

Syntax:

array api or methods

API(application programming interface): software interface that allows two applications to interact with each other without user intervention.

1. a.length: .length property returns the length of a string or array.

Example: let a=[1,2,3,4,5]; let b=”hari”; console.log(a.length,b.length); output is 5,4

Example: storing of length in a variable.

Let a=[12,2,3,34]; res=a.length; console.log(res) output is 4

2. a.push(1 parameter or more): .push method is used to insert elements at end of an array and also return new length of array(after inserting element).change original array.

Example: let a=[]; res=a.push(23); console.log(a,res); output is [23],1 (1is length of array)

3.a.unshift(1 parameter or more): .unshift method is used to insert elements at begining of an array and also returns new length of array(after inserting element).change original array

Example: let a=[23]; res=a.unshift(4,5); console.log(a,res); output is [4,5,23],2

4.a.pop(): in brackets don’t give any parameter, .pop method is used to remove the last element of an array and also returns the popping(removing) element. Change original array

Example: let a=[1,2,3,4]; res=a.pop(); console.log(a,res) output is [1,2,3],4 (4 is the popping element)

5.a.shift():in brackets don’t give any parameter, .shift method is used to remove the first element of an array and also returns shifting(removing) element. Change original array

Example: let a=[1,2,3,4]; res=a.shift(); console.log(a,res) output is [2,3,4],1

6.a.join(parameter optional): .join method is used to convert array into string. original array not changed.

Example: let a=[1,2,3,4]; res=a.join(); console.log(res); output is 1,2,3,4(output is a string)

Here if parameter is not given then output is equal to input but it is a string.

Example2: let a=[1,2,3,4]; res=a.join(“”);console.log(res,a);output is 1234(string),[1,2,3,4]

Example3: let a=[1,2,3,4];res=a.join(“ “);console.log(res,a);output is 1 2 3 4(string),[1,2,3,4]

Example4: let a=[1,2,3,4];res=a.join(“#”);console.log(res);output is 1#2#3#4(string)

7.a.split(parameter is optional): .split method is used to convert string into array. Always parameter is part(letter or character)of a string. otherwise it convert as empty parameter and print same output(op=ip)

Example: let a=”hari”;res=a.split();console.log(res); output is [“hari”](array)

Here we don’t give any parameter then output is same as input but out is change as array.

Example2:reverse a single string?

Let a=”hari”; res=a.split(“”);console.log(res); output is [h,a,r,i](array)

console.log(res.reverse()); output is [i,r,a,h](array) Console.log(res.join(“”); output is irah(string)

example3: let a=”I am a good boy”;res=a.split(“a”); console.log(res); output is [ 'I ', 'm ', ' good boy' ]

8.a.reverse():in brackets don’t give any parameter, .reverse method is used to reverse the array.it changes the original array.

Example: let a=[1,2,3,4,5]; res=a.reverse();console.log(a,res); output is [5,4,3,2,1][5,4,3,2,1]

9. a.slice(start index, ending index): .slice method is used to returns selected elements in an array ,as a new array.in parameter first parameter start from that index to 2nd parameter ends before(excluding) that index. original array not changed.

Example: let a=[1,2,3,4,5]; res =a.slice(1,4);console.log(res,a); output is [2,3,4] [1,2,3,4,5]

Example2:let a=[1,2,3,4,5];res=a.slice(1,5);console.log(res); output is [2,3,4,5]

Example 3:let a=[1,2,3,4];res=a.slice(-3,-1);console.log(res);output is [2,3]

Note: negative numbers select from the end of the array.[1,2,3,4]

10. a.splice(startIndex,quantity,insertVlaues): .splice method is used to add or remove array elements. It changes the original array.

Example: let a=[1,2,3,4]; res=a.splice(1,2); console.log(res,a); output is[2,3] [1,4]

Here when we give only two parameters (start index and quantity) it used to remove elements. it returns the removing elements and original array is overwriting.

Example2: let a=[1,2,3,4,5];res=a.splice(1,2,10,20);console.log(res,a); output is[2,3][1,10,20,4,5]

Example3: let a=[1,2,3,4,5];res=a.splice(1,0,10,20);console.log(res,a);output is [][1,10,20,2,3,4,5].

11. a.indexOf(number or char): .indexOf method return the returns the index number first occurrence value of array or string. If the value is not fund then returns -1,method is case sensitive.

Example: let a=[1,2,3,4,5,1];res=a.indexOf(2);console.log(res);output is 1

Example: let a=”hari is good boy”;res=a.indexOf(“o”);console.log(res);output is 8;

Example: let a=”hari is good boy”;res=a.indexOf(“good”);consolelog(res);output is 7

Here in parameter given is “good” but it finds the position of first letter in “good” that is “g”

Example: let a=”hari is good boy”;res=a.indexOf(“z”);console.log(res);output is -1; z is not found

Example: let a=”hari is good noy”;res=a.indexOf(“o”,11);console.log(res);output is 14

Here we give two parameter 2nd parameter refers that to find element “o” after 11 index of a string.

12. a.lastIndexOf(number or char): this method returns the index(position) of the last occurrence of a specified value in a array or string.it returns -1 if value is not found, this method is case sensitive.

Example: let a=[1,2,3,4,5,1];res=a.lastIndexOf(1);console.log(res);output is 5

Example: let a=”hari is good boy”;res=a.lastIndexOf(“o”);console.log(res);output is 14

Example: let a=”hari is good boy”;res=a.lastIndexOf(“O”);console.log(res);output is -1

Here capital O is not found.

13.Math.max(number):this method returns the number with highest value .

Example: let a=[1,2,3,4];res=Math.max(…a);console.log(res); output is 4

Example: res= Math.max(2,5);console.log(res);output is 5

Example: res= Math.max(1,2,3,-1,-2,-3,0,5);console.log(res); output is 5

14. Math.min(number): this method returns the minimum value.

Example: let a=[1,23,4,5];res=Math.min(…a);console.log(res); output is 1

Example: res=Math.min(2,55);console.log(res);output is 2

15.for of loop: this loop runs on bases of elements in array not on index of elements. Loop is used to run block of code for number of times.

Syntax: let arr=[1,2,3,4];

For( let ele of arr){ body};

Example: let a=[1,2,3,4,5,6]; for(ele of a){if(ele%2==0);

Console.log(ele);} output is 2,4,6

Function api or methods

16. a.forEach(function name): this method calls a function for each elements in an array and it not execute for empty elements. Its main drawback is not returns any value.

It doesn’t change the original array.

Example: let sum=0; let a=[1,2,3,4,5]; function func(x){sum=sum+x; }

a.forEach(func); console.log(sum); out put is 15

example: let a=[1,2,3,4]; function func(x){console.log(x)} a.forEach(func); output is 1,2,3,4(not an array)

17. a.map(function name): this method calls a function for each element in an array and returns in array and also returns the value and don’t execute the function of empty elements. It doesn’t change the original array.

Example: let a=[1,4,9,16,25]; res=a.map(Math.sqrt); console.log(res); out put is [1,2,3,4,5](op is always array.)

Example: let a=[1,2,3,4,5];function func(x){return x\*10} res=a.map(func); console.log(res,a) op is [10,20,30,40,50] [1,2,3,4,5]

18. a.filter(function name): this method creates a new array filled with elements that pass a test(condition) provide by a function.it doesn’t execute the function for empty elements.it doesn’t change the original array.

Example: let a=[32,33,16,40];function func(x){return x>=18;} res=a.filter(func); console.log(res,a)

Output is [32,33,40] [32,32,16,40]

19. a.reduce(function name,number):here number parameter is optional, method is used to run a function for number of times and finally returns a single value.it doesn’t change the original array.this function works from left to right

a.reduceRight() works from right to left.

Example: let a=[8,5,6,3]; function func(total,val){return total+val}

res =a.reduce(func,0); console.log(res); op is 22

Example: let a=[8,5,6,3]; function func(total,val){return total+val}

res =a.reduce(func,10); console.log(res); op is 32

Example: let a=[1,2,3,4]; function func(total,val){return total-val}

res =a.reduce(func); console.log(res);

20. a.sort(function name): here parameter is optional if we decelar function then use parameter, use only array no ned to use parameter. This method is used to sort the int array either ascending order or descending order ,in case of string array sort based on alphabetical and dictoniary order.it changes the original array.

Example: let a=[“hari”,”giri”,”zebra”,”Lokesh”]; res=a.sort();console.log(res,a)

Output is [ 'Lokesh', 'giri', 'hari', 'zebra' ] [ 'Lokesh', 'giri', 'hari', 'zebra' ]

Example: for sorting numbers in array always use functions concept ,other wise it give wrong ans.

let a=[1,20,4,3,5]; function func(a,b){ return a-b}

res =a.sort(func); console.log(res); out put is [1,3,4,5,20]

note: here function always takes two parameters. Return (a-b) is for ascending order,(b-a) returns descending order.

Example: let a=[1,20,4,3,5]; function func(a,b){ return b-a}

res =a.sort(func); console.log(res); out put is [20,5,4,3,1]

Example: let a=[2,4,3,1]; console.log(a.sort()) ;

21.typeof(variable): this operator is used to find the datatype of js variable.

Example: let a=”hari”;(console.log(typeof(a)); output is string.

Datatypes: in js there are 5 different types of datatypes they are

1.string 2.number 3.bollean 4.object 5.function

Two datatypes that cannot contain values they are 1.null 2.undefined

Example: console.log(typeof(NaN)); op is number

console.log(typeof(null)); op is object

console.log(typeof(undefined)); op is undefined

note: difference between null and undefined are equal in value but different in types.