JAVASCRIPT function

A function is a set of statement that perform a task or calculate the value given to it;

Function declaration:

A function is defined by keyword function followed bu the functuion name in camelcased and a parameter inside the braces then statement inside it

Syntax:

Function functionName(parameter){

//statementsL

}

Example:

function add (number){

console.log(`the number is $(number + 2);

}

Calling a function:

Defining a function doesn’t execute it , defining function Name and specifies what to do when the function is called,calling the function actually perform the operation which are declared in the parameters

The argument of function is not limit it can be array or numbers or objects

Return function:

The return statement ends the function execution and specifies a value to be returned to the function caller.

Syntax:

Return[expression];

When a return statement is used function body,the execution of the function is terminated,the specified return value is returned to the function caller

Function scope:

Variables defined inside a function cannot be accwessed from anywhere outside the function,because the variable us defined only in scope if the function

The function defined in the global scope can access all variables defined in the global scope,a function defined inside another function can also acces all variables defined in its parent function

function foo(){

var fruit ='apple';

console.log('inside function: ',fruit);

}

foo(); //inside function: apple

console.log(fruit); //error: fruit is not defined

Block scope:

A block scope is the area within if,condition or for and while loops in general wherever u see {},it is a block.const and let keywords allow developers to declare variables in the block scope,which m,eams those variables exist only within the corresponding block

Example:

function foo(){

if(true){

var fruit1 = 'apple'; //exist in function scope

const fruit2 = 'banana'; //exist in block scope

let fruit3 = 'strawberry'; //exist in block scope

}

console.log(fruit1);

console.log(fruit2);

console.log(fruit3);

}

foo();

//result:

//apple

//error: fruit2 is not defined

//error: fruit3 is not defined

**Lexical Scope**

Another point to mention is the lexical scope. Lexical scope means the children scope have the access to the variables defined in the parent scope. The children functions are lexically bound to the execution context of their parents.

function foo1(){

var fruit1 = 'apple';

const fruit2 = 'banana';

let fruit3 = 'strawberry';

function foo2(){

console.log(fruit1);

console.log(fruit2);

console.log(fruit3);

}

foo2();

}

foo1();

//result:

//apple

//banana

//strawberry

Function expression :

The **function** keyword can be used to define a function inside an expression.

Syntax:

Function [name](paramaeters,[, paramaeters[,paramaeters…,]]]){

Statements

}