**First class functions:**

**What are pure functions?**

[**https://www.geeksforgeeks.org/pure-functions-in-javascript/**](https://www.geeksforgeeks.org/pure-functions-in-javascript/)

A**Pure Function** is a function (a block of code) that **always returns the same result if the same arguments are passed**. It does not depend on any state or data change during a program’s execution. Rather, it only depends on its input arguments.

If output is not depended on some outer variable, then the output will be fixed for fixed argument.

For eg- u can refer above geeksforgeeks link.

**What is an anonymous function?**

A function without a name is known as anonymous functions.

**What are first class functions?**

**Function stmt vs function expression vs function declaration?**

// First class functions:

// What are pure functions?

// https://www.geeksforgeeks.org/pure-functions-in-javascript/

// A Pure Function is a function (a block of code) that always returns the same result if the same arguments are passed. It does not depend on any state or data change during a program’s execution. Rather, it only depends on its input arguments.

//  If output is not depended on some outer variable, then the output will be fixed for fixed argument.

// For eg- u can refer above geeksforgeeks link.

// What is an anonymous function?

// A function without a name is known as anonymous functions.

// What are first class functions?

// Function stmt vs function expression vs function declaration?

//function statment

// function a() {

//   console.log("a called");

// }

//this way of creating a function is known as function stmt.

//The function statement declares a function. A declared function is "saved for later use", and will be executed later, when it is invoked (called).

//function expression

//in js, we have a feature to assign a function to a variable also- it is kwnon as function expression

//here function acts as a value to that variable. we are initializing b with some value.

// var b = function () {

//   console.log("b called");

// };

//these above 2 are the 2 ways to create a function.

//difference between the above 2- function stmt vs function expression is-

//major difference is hoisting -

// to call first function - u call  by using a(); and to call 2nd , u call by using b();

//how hoisting come into picture means

//lets take first type and call the function even before creating it.- similarly do for function expression and check.

a();

b();

function a() {

  console.log("a called");

}

var b = function () {

  console.log("b called");

};

//so from above ccode, we can observe that - function a  that is function stmt is supporting hoisting, worked same.

//  but  for b - got err - as b is not a function . so this function expression is not getting hoisted.

//so this is the difference between function stmt and function expression.

// so during hoisting phase or during memory creation phase., a is created a memory and this func is assigned to a.

// but during func expression , this b is treated as any other variable and it is  assigned undefined until it reaches  this assignmennt function exp line.

// until that definition line, it will be undefined.

//so here as b value is undefined , u cannot call b.

//function declaration:

//function stmt is also known as function declaration.

//anonymous functions:.

// a function without  name is knwon as anonymous function.

// function (){

//     //but this will give a syntax err -

// }

// above  func looks like a function stmt, but has no name,

// but according to ecmascript, function stmt should always have  a name,

//so above is invalid syntax

//run - syntax err- function stmts require  a function name

//so anonymous functions are used in a place where functions are used as values.

// so anonymous functions are used when funcs are used as values.

//u can use it to assign to some variables.

//so see the eg of function expression, there the value of b is a anonymous function used as a value to b.

// named function expression.:

// named  function expression is exactly same as function expression, but instead of anonymous func assigned to b there no, there we assign a named function,

var b = function xyz() {

  //now this becomes a named func expression

  console.log("b called");

};

//b() - we can call it as b(),

//xyz() - but if i do like this, then we get err -  reference err- xyz is not defined

// here  b will be created but xyz will not be created.

//difference  between parameters and arguments.

var b = function (param1, param2) {

  //here param1 and param2 are known as parameters , as they present in function def.

  //here param1 and param2 are local variables to this func , we cant access outside.

  console.log("b called");

};

// the arguments which we give to a function while calling it . they are known as arguments.

// b(arg1,arg2) ; here arg1 and arg2 are known as Parameters.

//first class functions:

// while calling a function, instead of values 1,2 like that we can pass a whole function as a argument., we can pass anonymou funtion and we can  receive a func as argumernt.it is perfect js .

// var c = function (param1) {

//   console.log(param1); //we get func def as output

// };

// c(function () {});

//there is one more way to pass this function inside the function  like a named function as below.

// var c = function (param1) {

//   console.log(param1); //we get func def as output

// };

// function xyz() {}

// c(xyz); //same as above way

//we can even return a anonymous function from a function

// var c = function (param1) {

//   // console.log(param1);

//   return function () {};

// };

// console.log(c()); //same as above way

// similarly we can return named function

var c = function (param1) {

  // console.log(param1);

  return function xyz() {};

};

console.log(c()); //same as above way

//so this is nothing but first class fucntions

//the ability to use functions as values is known as first class functions. and  can be passed as an arg to another funtionand can be returned  from a fucntion. this ability is known as first class fucntions.

//first class functions and first class citizens are same.

// es6 introduced arrow functions in 2015

//this function stmt, expression  all can be created using arrow functions also

//