**Basic:**

Types of Function:

* **Function:**

function func() {

//Your Code Here

}

* **Anonymous function:**

function() {

//Your Code Here

}

**. Inline function**

var func = function() {

//Your Code Here

};

**CallBack Function:** It is function (it can be Anonymous or Arraow Function) passed into another function as an argument, which is then invoked inside the outer function to complete some task;

**Passing function must be last in the argument.**

**1. Simple Callback Function without argument:**

function show(){

Console.log(“I am in function);

}

Function geeky(callback){ //taking as argument ‘cakllback’

Callback();  
}

geeky(show);//pass show as argument

**2. We can pass whole function as argument**

Function geeky(callback){

Callback();

}

Geeky(function show(){

Console.log(“I am in Function”);

});

**Simple callback with argument:**

1.

Function show(a)

{

Console.log(“I am in Function “+a);

}

Function geeky(a,callback)

{

Callback(a);

}

Geeky(101,show);//passing argument

**2. Another way**

Function show(a){

Console.log(“I am Function”);}

Geeky(callback)

{

Var a=101;

Callback(a);

}

Geeky(show);

**Anonymous Callback Function:**

Function geeky(callback)

{

Var a=101;

Callback(a);

}

geeky(function(a){

Console.log(“I am in Function “+a);

});

**Arrow callback function:**

Function geeky(callback)

{

Var a=101;

Callback(a);

}

Geeky(a=>console.log(“I am function “+a));**OR** Geeky((a)=>console.log(“I am function “+a));

**Synchronous And Asychronous Callback Function**

1. **Synchronous**:

**It wait for each operation to complete, after that it executes next operation.**

Example:

Function show()

{

Console.log(“I am Function”);

}

Function geeky(callback)

{

callback();

}

Geeky(show);

Console.log(“End”);

2.

**Asynchronous: It never wait for each operation to complete, rather it executes all operations in the first go only.**

setTimeout(function show(){

console.log(“I am Function”);

},5000);

Console.log(“End”);

**Synchronous And Asychronous (File Hindling)**

//const fs=require("fs");

//Synchronous

/\*const files=fs.readdirSync('./');

console.log(files);\*/

//Asychronous

/\*fs.readdir('./',function(err,files){

    if(err) console.log("Error"+err);

    else

    console.log(files);

});