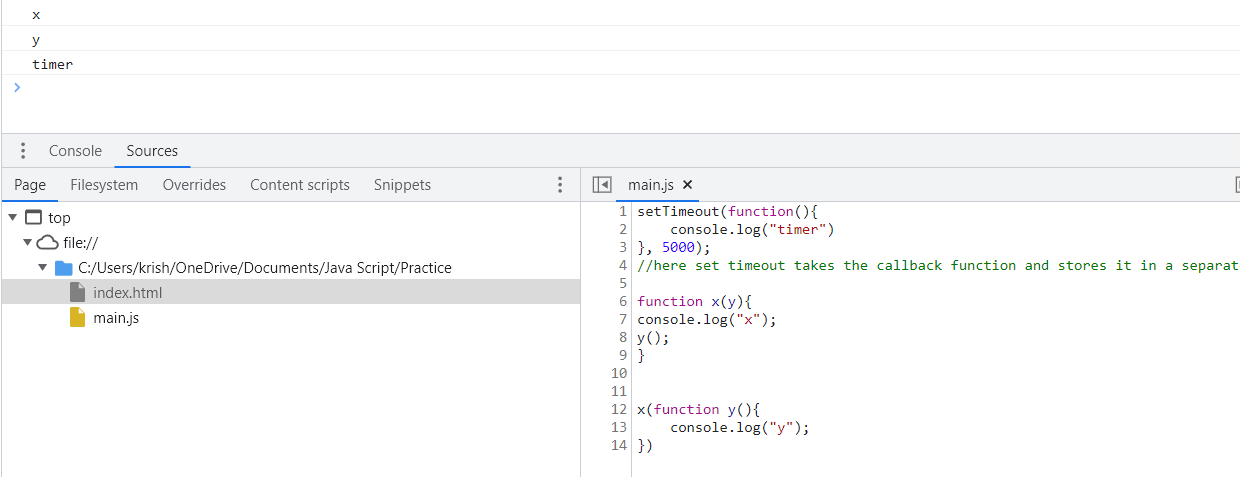
**Call Back Functions in JavaScript**

What is a callback function?

As functions are first class citizens in java script, we can pass functions are arguments, call function inside a function, return function as a value. The functions which we pass as arguments are called callback functions.

Java script is a synchronous single threaded language.

These callback functions give us access to the asynchronous world.



If you observe initially whole function is executed and there is nothing inside call stack and suddenly call back gets loaded into global call stack after 5000ms.

This it brings the asynchronous feature to the java script.

Graphical user interface, text, application, email

Description automatically generated

**Blocking the main thread**

Any operation which blocks the call stack for more time is called blocking the main thread.

Since every operation is executed in call stack to avoid delay we should use async operations for things which take time just like we did above using setTimeout().

Event Listeners

document.getElementById("clickMe")

.addEventListener("click",function xyz(){

console.log("button clicked")

});

So whenever you click a button function is called into the call stack.

Demo of a closure with event listener

Here function is xyz is a callback function.

Lets suppose we want to count the number of times button is clicked

function attachEventListners(){

    let count=0;

    document.getElementById("clickMe")

.addEventListener("click",function xyz(){

console.log("button", ++count);

});

}

Here function xyz() is forming a closure with its outer scope.

How is it forming a closure?

Closure=function along with its lexical scope.

Here we need the value of count inside the function xyz(). But count is global variable and outside its scope so we need a closure to access it lexical environment.

Graphical user interface, text, application, email

Description automatically generated

Timeline

Description automatically generated

The handler we see in the above picture is the call back function.

Garbage collection and remove Event listeners.

It is recommended to remove Event listeners because they are heavy as take memory even when the function is not being executed and memory is blocked. The reason it takes memory is we never know the other variable will try to access the lexical scope.