**Closures in JavaScript**

**What is a Closure?**

Closure is a function along with its lexical scope.

function x(){

    var a=7;

    function y(){

        console.log(a);

    }

    return y;

}

var z=x();

console.log(z);

Here return y literally returns the function itself.

ƒ y(){

console.log(a);

}

JavaScript is synchronous language and once z=x() is invoked and executed, x is removed from the execution context.

Now lets suppose after 1000 lines if run z()

The output is 7.

How is this happening

Z just refers to below code.

ƒ y(){

console.log(a);

}

How y is referring to the value of ‘a’ and printing 7.

**Answer**:

As we know Lexical scope is local memory + lexical scope of outer execution context. So function y remembers the memory address of var ‘a’.

Graphical user interface, text, application

Description automatically generated

Now let’s dig deeper and find the Nuances of this

function z(){

    var b=100;

function x(){

    var a=7;

    function y(){

        console.log(a,b);

    }

    a=200;

    y();

}

x();

}

var k=z();

console.log(k);

Guess the output of a and b?

One would think a is assigned with 7 there answer is 7,100.

But function y only refers to memory address of var ‘a’ but not the value.

Shadowing of var ‘a’ happens and answer is 200,100.

In the above program inside the func y() we are accessing its parent’s scope func x() and x() parents scope which z(). Will there be a closure here?

Yes there would be two closures for each function separately.

Graphical user interface, text, application, email

Description automatically generated