**Note: Statements in the comments are results in this challenges.**

**Challenge1**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Expression vs Statement</title>

</head>

<body>

    <script src="challenge1.js"></script>

</body>

</html>

**Challenge1.js & result**

const myObj={

    a:10,

    b:"indra"

};

//delete is an expression

delete myObj.b; //Expression Statement

**Challenge2**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Expression vs Statement</title>

</head>

<body>

    <script src="challenge2.js"></script>

</body>

</html>

**Challenge2.js & result**

function fn(){

    console.log("Greeting from the 'fn' function");

    return (function(a){

        console.log(a);

    });

}

fn()  //function call is an expression. it will executes well and prints message on console

(true) // it will take argument for returned funtion and prints true. because semicolon is not there after funtion call.

**Challenge3**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Expression vs Statement</title>

</head>

<body>

    <script src="challenge3.js"></script>

</body>

</html>

**Challenge3.js & result**

//Why semicolon is added?

function firstFunction(a,b){

    return a+b;

} // function Declaration

const secondFunction= function(a,b){

    return a+b;

}; // Statement