

1. Determine what this Javascript code will print out (without running it):

```
x = 1;

var a = 5;

var b = 10;

var c = function(a, b, c)
{
    var x = 10;

    document.write(x); // 10

    document.write(a); //8

    var f = function(a, b, c) {

        b = a;

        document.write(b); //8

        b = c;

        var x = 5;

    }

    f(a,b,c);

    document.write(b); //9

}

c(8,9,10);

document.write(b); //10

document.write(x); } //1
```

Ans: 10 8 8 9 10 1

2. What is the difference between a method and function?

Function is a code written to perform specific task. It can be invoked by calling functionName with ();
Syntax:

```
function functionName(parameters) {

    // Content

}
```

Method is a property of an object that contains a function definition.

Syntax:

```
object = {  
  methodName: function() {    // Content    } };
```

3. What does 'this' refer to when used in a Java method?

→ In Java method, this refers to object of current class. (**The object that contains the method.**)

4. What does 'this' refer to when used in a JavaScript method?

The JavaScript this keyword refers to the **object that contains the method**

It has different values depending on where it is used:

- In a method, this refers to the owner object.
- Alone, this refers to the global object.
- In a function, this refers to the global object.
- In a function, in strict mode, this is undefined.
- In an event, this refers to the element that received the event.
- Methods like call(), and apply() can refer this to any object.

5. What does 'this' refer to when used in a JavaScript constructor function?

Ans: The newly created object that called the function. When used in JavaScript constructor function, “this” does not have a value. It is a substitute for the new object. The value of this will become the new object when a new object is created.

6. Assume object x is the prototype for object y in Javascript. Object x has a method f() containing keyword 'this'. When f is called by x.f(), what does 'this' refer to?

→ Here, “this” refers to object x.

7. What is a free variable in JavaScript?

→ Free variable is a variable referred to by a function that is not **one of its parameters or local variables.**

8. Create an object that has properties with name = "fred" and major="music" and a property that is a function that takes 2 numbers and returns the smallest of the two, or the square of the two if they are equal.

```
var obj = {  
  name: "fred",  
  major: "music",  
  compute: function (x, y) {
```

```

    if (x < y) { return x; }

    else if (y > x) { return y; }

    else if (x === y) { return x * x; }

}

};

```

9. Write Javascript code for creating three Employee objects using the "new" keyword and a constructor function. Employee objects have the following fields: name, salary, position.

```

function Employee(name, salary, position) {
    this.name = name;
    this.salary = salary;
    this.position = position;
}

var emp1 = new Employee("Tom", 1000, "HR");
var emp2 = new Employee("Duke", 2000, "Developer");
var emp3 = new Employee("Hary", 3000, "Tester");

```

10. Write a Javascript function that takes any number of input arguments and returns the product of the arguments.

```

function product() {

    let product = 1;

    for (let key in arguments) {

        product *= arguments[key];

    }

    return product;

}

console.log(product(2, 3, 4,1,2)); //outputs 48

```

11. Write an arrow function that returns the maximum of its three input arguments.

```

var maxVal = (x, y, z) => { return Math.max(x, y, z) };

var a = maxVal(2, 5, 12)

alert(a);

```