

[References](#)
[DOM & BOM](#)
[Variables, Operators, Typeof](#)
[Equality Operators](#)
[Hoisting](#)
[Basic Functions](#)
[Events](#)
[Basic Objects](#)
[Scope](#)
[Advanced Concepts in Objects & functions](#)
[String, Array, Date functions](#)
[Power of 'this'](#)
[Prototype chain](#)
[Closures](#)
[IIFE](#)
[Regular Expression](#)
[Definitions and Solutions](#)
 [Lexical Scope](#)
 [IIFE](#)
 [Function "name" and "length"](#)
 [Miscellaneous](#)

References

1. <http://perfectionkills.com/javascript-quiz/>
2. <http://davidshariff.com/js-quiz/>
3. <http://www.codequizzes.com/>

DOM & BOM

1. What is the output?

`document.nodeType, document.documentElement.nodeType`

- a) 3, 1 b) 1, 9 c) 9, 1 d) 1, 1

2. How to get screen width?

- a) window.screen.width
b) window.innerWidth
c) screen.width
d) Both a & c

3. How to get width of HTML document without vertical scrollbar width?

- a) window.innerWidth
- b) document.clientWidth
- c) document.documentElement.width
- d) document.documentElement.clientWidth

4. What is the output?

```
typeof document.body.classList
```

- a) "string"
- b) "array"
- c) "object"
- d) "list"

5. Mention at least 5 objects of window. _____

Variables, Operators, Typeof

1. What are the values of **x** and **y**?

```
var y = 1, x = y = typeof x;
```

- a) "undefined", 1
- b) "undefined", "number"
- c) "undefined", "undefined"
- d) Error

2. What is the output

```
typeof null
```

- a) "null"
- b) "undefined"
- c) "object"
- d) "error"

3. What is return value of typeof

- a) "String"
- b) "number"
- c) "function"
- d) "string"

4. What is the output

```
typeof typeof(true)
```

- a) "boolean"
- b) "number"
- c) "string"
- d) "function"

6. What is the output ?

```
var m = "3";
```

```
typeof +m
```

- a) "boolean"
- b) "number"
- c) "function"
- d) "string"

7. What is the output ?

```
var m = "d";
```

```
typeof +m
```

- a) "string"
- b) "boolean"
- c) "number"
- d) "NaN"

8. What is the output ?

```
typeof !"Divami"
```

- a) "boolean" b) "number" c) "function" d) "string"

9. What is the value of variable a?

```
var a = 10, 20;
```

- a) 10 b) 20 c) Error

10. What is the value of variable a?

```
var a = (10, 20);
```

- a) Error b) 10 c) 20

11. What is the output of below code _____

```
Infinity / Infinity
```

12. What is the output ?

```
typeof Object
```

- a) "Object" b) "object" c) "Function" d) "function"

13. What is the output ?

```
typeof string
```

- b) "string" b) "function" c) "undefined" d) Error

14. How to print below line (with quotations) as output using **console.log**?

```
"Divami's Property"
```

Ans: _____

15. What is the result ?

```
(true + false) > 2 + true;
```

- a) true b) false

16. "1" - - "1"; ?

- a) 0 b) 2 c) 11 d) "11"

17. What is the result ?

```
String('Hello') === 'Hello';
```

- a) true b) false

18. `10 > 9 > 8 === true`

- a) true b) false

19. What is the alert value ?

```
function bar() {  
    return foo;  
    foo = 10;  
    function foo() {}  
    var foo = '11';  
}  
alert(typeof bar());
```

- a) "number" b) "string" c) "undefined" d) "function"

20. Output ?

```
(function(foo){  
    return typeof foo.bar;  
})({ foo: { bar: 1 } });
```

- a) "undefined" b) "object" c) "number" d) Error

21. What is the output ?

```
var result = 1+("six"/3);  
console.log(result);
```

- a) 1six/3 b) TypeError c) NaN

22. What is the output ?

```
var foo = {  
    bar: function() { return this.baz; },  
    baz: 1  
};  
(function(){  
    return typeof arguments[0]();  
})(foo.bar);
```

- a) "undefined" b) "object" c) "number" d) "function"

23. What is the return value ?

```
var phone = {  
    ring: function() {  
        return this === phone;  
    }  
}
```

- a) true b) false

24. What is the output printed in console ?

```
var emp = {  
  name: 'Prakash',  
  age: 35,  
  getName: function() {  
    this = {name: 'Kiran'}  
    console.log(this.name);  
  }  
}
```

- a) Error b) Prakash c) Kiran d) undefined

25. What is the output ?

```
var a = 2, b = 3;  
a && b ?
```

- a) true b) false c) 3 d) 2

26. What is the output ?

```
42..toString() _____
```

27. What is the value printed in console ?

```
var a = 1;  
function b() {  
  a = 10;  
  return;  
  function a() {}  
}  
b();  
console.log(a);
```

- a) 1 b) 10 c) undefined;

Equality Operators

1. What is the output in console.

```
var foo = function foo() {  
  console.log(foo === foo);  
};  
foo();
```

- a) true
b) false
c) Reference Error

2. What is the output of a == b? _____ .

```
var foo = function foo() {
    console.log(foo === foo);
};
var a = new foo();
var b = new foo();
```

3. What is the value of alert ?

```
function foo() {
    return
    {
        name: 'divami'
    };
}
alert(typeof foo());
```

- a) "number" b) "object" c) "function" d) "undefined"

4. What is the result ?

```
Number("1") - 1 == 0;
```

- a) true b) false c) Error

5. What is the result?

```
NaN === NaN;
```

- a) true b) false

Hoisting

1. What is Hoisting ?

2. How the function will look like after hoisting ?

```
function prints(){
    console.log(a);
    a = 10;
    b = 20;
    console.log(b);
    var a = 30;
    var b = 40;
    console.log( a + b);
}
```

3. What is the output of the following function?

```
function(){
```

```

        console.log(typeof a);
        a = 10;
        function a(){
        }
        console.log(typeof a);
        var a = 40;
        console.log(typeof a);
    }

```

4. What is the alert value?

```

function bar() {
    return foo;
    foo = 10;
    function foo() {}
    var foo = '11';
}
alert(typeof bar());

```

- a) "number" b) "string" c) "undefined" d) "function"

5. Output?

```

(function f(){
    function f(){ return 1; }
    return f();
    function f(){ return 2; }
}) ();

```

- a) 1 b) 2 c) Error d) undefined

6. What does the following code print to the console? [c]

```

var func = function () {
    return a;
    a = "hi";
};
console.log(func());

```

- a) null b) "hi" c) ReferenceError d) undefined

6. What does the following code print to the console? [b]

```

var abc = function () {
    return abc;
    abc = "hi";
};
console.log(abc());

```

- a) "hi" b) abc Function c) ReferenceError d) undefined

7. What is the value printed in console ?

[d]

```
function foo() { return "foo" }  
console.log(foo);  
function foo() { return "bar" }
```

- a) foo
- b) bar
- c) function foo() { return "foo" }
- d) function foo() { return "bar" }

8. Choose the correct answer ?

[a]

```
var message = "hello";  
function greet() {  
    alert(message + "");  
    var message = "bye";  
}  
greet();  
alert(message);
```

- a) Alerts 'undefined' and then alerts 'hello'
- b) Throws runtime error
- c) Alerts 'hello' then alerts 'bye'
- d) Alerts 'bye' then alerts 'hello'

Basic Functions

```
1. function func1(){  
    return typeof arguments;  
};  
console.log(func1());
```

- a) "undefined"
- b) "array"
- c) Error
- d) "object"

```
2. var f = function g(){ return 23; };  
typeof g();
```

- a) Error
- b) "function"
- c) "number"
- d) "undefined"

```
3. function func2(x){  
    delete x;  
    return x;  
};
```



```
func2(20);
```

- a) undefined b) null c) 1 d) Error e) _____

4. What is the output of following?

```
function add(x, y, z){  
    return x + y + z;  
}
```

add(10, 20, '30') = _____
add(10) = _____
add(10, null, 40) = _____
add('10', null, 40) = _____
add(10, undefined, undefined) = _____
add('10') = _____
add('10', undefined) = _____

5. What are local and global variable here?

```
var x = 10;  
function(a, b){  
    var y = 20;  
    var x = 30;  
    c = x + y;  
    d = x - y;  
    e = a*b;  
}
```

6. What value is alerted. ?

```
function func(a, b) {  
    arguments[1] = 13;  
    alert(b);  
}  
func(1);
```

- a) 13 b) undefined c) ReferenceError

7. Output?

```
var f = (function f(){ return "1"; }, function g(){ return 2;  
}) ();  
typeof f;
```

- a) "string" b) "undefined" c) "number" d) "function"

Events

1. Which of the following event object properties indicates which key was pressed? (In chrome and mozilla browsers)

- a) event.which
- b) event.keyCode
- c) event.onKeyPress

2. Choose the suited option from the following options. [c]

```
<div id="test">
  <input id="myButton" type="button" value="Click me" />
</div>
<script>
  document.getElementById("myButton").onclick = function(e) {
    console.log("I'm the button");
    //TODO: CODE TO PREVENT EVENT BUBBLING
  }

  document.getElementById("myButton").onclick = function(e) {
    console.log("I'm Div");
  }
</script>
```

- a) return false b) e.preventDefault() c) e.stopPropagation()

3. What is the output of below code when we click on div#div2.

```
<div id="div1" class="div"> 1
  <div id="div2" class="div"> 2
    <div id="div3" class="div"> 3 </div>
  </div>
</div>
<script>
  function highlight(e) {
    console.log(this.id);
  }
  var divs = document.getElementsByClassName('div');
  for(var i=0; i<divs.length; i++) {
    divs[i].addEventListener('click', highlight, false);
    divs[i].addEventListener('click', highlight, true);
  }
</script>
```

- a) div1, div2, div2, div1 c) div2, div1, div1, div2
b) div2, div3, div3, div2, div1 d) div3, div2, div2, div3

Basic Objects

1. What is the output ?

```
var foo = {  
    "name": "divami",  
    "location": "Hyd"  
}  
foo.length
```

- a) 2 b) 4 c) Error d) undefined

2. What is the output ?

```
var foo = {  
    "name": "divami",  
    "location": "Hyd"  
}  
typeof foo.valueOf()
```

- a) "string" b) "object" c) Error d) "undefined"

3. Mention at least 2 builtin properties/methods of an any Javascript object ?

Ans: _____

4. "divami" instanceof String

- a) true b) false

5. How to access first-name from the foo object?

```
var foo = {  
    first-name: "Bill",  
    company: "microsoft"  
}
```

- a) foo.first-name b) foo['first-name'] c) Both a & b d) Error in code

6. What is the result?

```
new Array(3).toString();
```

- a) [] b) [""] c) "undefined,undefined,undefined" d) ",,"

7. What is the output?

```
var x = 0;  
function foo() {  
    x++;  
    this.x = x;  
    return foo;  
}  
var bar = new new foo;
```

```
console.log(bar.x);
```

- a) **ReferenceError** b) **TypeError** c) **undefined** d) **0**

8. What is the result?

```
var bar = 1,  
    foo = {};  
foo: {  
  bar: 2;  
  baz: ++bar;  
};  
foo.baz + foo.bar + bar;
```

- a) **ReferenceError** b) **NaN** c) **undefined** d) **0**

9. What is the output?

```
function f(){ return f; }  
new f() instanceof f;
```

- a) **true** b) **false**

Scope

1. What is the output?

```
var foo = {  
  bar: function(){ return this.baz; },  
  baz: 1  
}  
typeof (f = foo.bar)();  
typeof foo.bar();
```

- a) **"number", "number"**
b) **"undefined", "number"**
c) **"undefined", "undefined"**
d) **"function", "number"**

2. What is the return value?

```
function f1(){var a = 1; f2();}  
function f2(){return a;}  
f1();
```

- a) **undefined** b) **1** c) **ReferenceError**

Advanced Concepts in Objects & functions

1. What does the following code prints?

```
function sum(a, b) { };  
console.log(sum.name);  
console.log(sum.length);
```

- a) "", "" c) undefined, undefined
b) Error (sum is not an object) d) "sum", 2 e) "sum", ""

2. What does the following code print to the console?

```
var cat = function(x) {  
    return 100;  
}  
console.log(cat.name);
```

- a) undefined b) "" c) abc is not an object d) "abc"

3. What does the following code print to the console?

```
var cat = function xyz(x) {  
    return 100;  
}  
console.log(cat.name);
```

- b) undefined b) "" c) abc is not an object d) "xyz"

4. What is the output?

```
function foo () {  
    return bar();  
    function bar() {  
        return "Poppin' bottles";  
    }  
}  
console.log(foo());
```

- a) undefined b) ReferenceError c) "Poppin' bottles" d) Function

5. What is the output? **Ans: monsters**

```
function c(foo) {  
    return foo();  
}  
function foo() {  
    return "monsters";  
}  
c(foo);
```

7. What is the output?

```
function foo() {  
    return this === window;  
}();  
console.log(result);
```

- a) true b) false c) Error

7. What is the output?

```
var result = function foo() {  
    return this === window;  
}();  
console.log(result);
```

- b) true b) false c) Error

8. What is the output? [b]

```
function foo() { console.log("divami"); }  
foo.name ?
```

```
var bar = function foo() { console.log("divami"); }  
bar.name ?
```

```
var baz = function() { console.log("divami"); }  
baz.name ?
```

- a) "foo", "bar", "baz" b) "foo", "foo", "" c) "foo", "baz", "" d) "foo", "", ""

9. What is the output? **Ans: abc function, null**

```
function abc() {  
    console.log(arguments.callee);  
    console.log(arguments.callee.caller);  
}  
abc();
```

10. What is the output? **Ans: xyz function, null, abc function, xyz function**

```
function abc() {  
    console.log(arguments.callee);  
    console.log(arguments.callee.caller);  
}  
function xyz() {  
    console.log(arguments.callee);  
    console.log(arguments.callee.caller);  
    abc();  
}  
xyz();
```

11. Which of the following is used to make immutable object [b]
a) Object.seal b) Object.freeze c) Object.immutable d) Both a & b

12. How to check whether the object “obj” is frozen or not ?

- a) Object.isFrozen()
- b) By checking the return value of delete

```
if(delete obj.name) {  
    // Not Frozen  
} else {  
    // Frozen  
}
```

- c) Both a & b
- d) None of the above

13. How to unfreeze the object ?

- a) Object.unFreeze()
- b) Object.freeze(null, false)
- c) There is way

14. How to sort array of numbers using sort() method ?

Ans:

```
var a = [1, 10, 2, 15, 21, 17];  
a.sort(function(x, y) {  
    return x-y  
});
```

15. What is the output?

```
var a = {  
    'name': 'venkat',  
    'age': 27  
};  
a.forEach(function(prop, k, obj) {  
    console.log(obj[k]);  
});
```

- a) “venkat”, 27 b) undefined, undefined c) name, age d) Error

16. What is printed in console ?

```
function abc(a) {  
    console.log(a * 2);  
}(10);
```

- a) 10 b) 20 c) Error d) Nothing

17. What is printed in console ?

```
var foo = ["divami", "divami"+"apervi", function(){return  
"ux"}(),];  
console.log(foo);
```

18. What is the output ? [a]

```
var foo = new Object();  
var bar = new Object();  
var baz = new Object();  
baz[foo] = "foo";  
baz[bar] = "bar";  
console.log(baz[foo]);
```

- a) foo b) bar c) Object {} d) ReferenceError

19. Choose correct option ? [d]

```
function foo(name) {  
    this.name = "sai";  
    this.name = name;  
    return { name: "vasu" }  
}  
var bar = new foo;  
console.log(new foo().name);  
console.log(new foo('naveen').name);  
console.log(bar.name);
```

- a) vasu, naveen, sai
b) vasu, vasu, error
c) naveen, naveen, undefined
d) vasu, vasu, vasu

20. What is the output ?

```
function Actor(name) {  
    this.name = name;  
    this.industry = 'Telugu';  
    this.getName = function() {  
        console.log("I am " + this.name + " from " + this.industry  
+ " Industry.");  
    }  
}  
var a = Actor("mahesh");  
typeof a;
```

- a) "string" b) "object" c) "function" d) "undefined"

21. Can we actually change **window** and **document** objects like this ?


```
window = {};  
document = 20;
```

- a) Yes b) No

String, Array, Date functions

1. What is the result?

```
[] + [] + 'foo'.split('');
```

- a) "f, o, o" b) TypeError c) ["f", "o", "o"] d) [] [] ["f", "o", "o"]

2. What is the output in console?

```
var myArr = ['foo', 'bar', 'baz'];  
myArr.length = 0;  
myArr.push('bin');  
console.log(myArr);
```

- a) ['foo', 'bar', 'baz']
b) ['foo', 'bar', 'baz', 'bin']
c) ['bin']
d) Can't change length of an array

3. What is the result?

```
var myArr = ['foo', 'bar', 'baz'];  
myArr[2];  
console.log('2' in myArr);
```

- a) true b) false d) ReferenceError

4. What is the result?

```
var arr = [];  
arr[0] = 'a';  
arr[1] = 'b';  
arr.foo = 'c';  
alert(arr.length);
```

- a) 1 b) 2 c) 3 d) undefined

5. What does the following code print to the console? [c]

```
function sum() {  
  result = 0;  
  arguments.forEach(function (num) {  
    result += num;  
  });  
  return result;  
}
```

```
sum(1, 2, 3);
```

- a) 0 b) 6 c) TypeError d) toStringValueOf

6. Sort the values in the following array in ascending report: [1, 10, 2, 15, 17, 21]

```
[1, 10, 2, 15, 17, 21].sort(function (x, y) {  
    return x - y;  
})
```

7. What is the output ?

```
parseInt('12abc45')
```

- a) NaN b) 12 c) 1245 d) 0

8. What is the output ?

```
['1', '2', '3'].map(parseInt)
```

- a) [1, 2, 3] b) [123] c) ['1', '2', '3'] d) [1, NaN, NaN] e) Error

9. What is the output ?

```
parseInt("3.1428"); _____  
parseFloat("3.1428"); _____
```

10. What is the return value of **splice()** function [c]

```
var a = [1, 2, 3, 4, 5, 6, 7, 8];  
a.splice(2, 4, 10, 11, 12);
```

- a) Error b) [3, 4, 5, 6, 10, 11, 12] c) [3, 4, 5, 6]

11. What is the return value of **push()** of array

- a) Pushed values
b) Current length of an array
c) true

12. What is the return value of below code ? **Ans: undefined**

```
[ ].pop()
```

13. What is the value of array "a" after execution of below code ?

```
var a = [1, 2, 3, 4, 5]  
a.pop(3);
```

- a) [1, 2, 4, 5] b) [1, 2, 3, 4] c) [2, 3, 4, 5] d) Error

14. What is the return value and final fruits array value [a]

```
var fruits = ["apple", "banana", "orange", "grapes"];  
fruits.splice(2, 0, "mango");
```

- a) [], ["apple", "banana", "mango", "orange", "grapes"]
b) [], ["apple", "banana", "orange", "mango", "grapes"]

- c) `[], ["apple", "banana", "orange", "grapes"]`
- d) None of the above

15. Which of the following methods effects original array [a]

- a) splice, push, pop, reverse
- b) splice, push, pop, shift, concat
- c) splice, push, pop, map, filter
- d) splice, push, shift, map

16. How to get last 2 characters of a string using **slice()** method?

Ans:

```
"divami".slice(-2);  
"010".slice(-2);
```

17. What is the output ? [b]

```
var a = [1, 2, 3]  
delete a[2-1];  
console.log(a.length);
```

- a) 2
- b) 3
- c) Error

18. What is the output ?

```
var a = [1, 2, 3]  
var b = a;  
delete b[1];  
console.log(b.length);
```

- a) 3
- b) 2
- c) Error

19. What is the output ? [d]

```
["a", "b", "null", null].every(function(item, i, arr) {  
    return item == null;  
});
```

- a) [null]
- b) null
- c) true
- d) false

20. What is the output? **Ans: false**

```
var foo = [1, 2, 3];  
var bar = [1, 2, 3];  
foo == bar
```

21. What is the output ?

```
["apple", "banana", "orange", "mango", "grapes"][1, 2, 3]
```

- a) "orange"
- b) 3
- c) "banana", "orange", "mango"
- d) "mango"

22. What is the output ?

```
var foo = [1, 2, 3, 4, 5, 6];
foo.length = 3;
console.log(foo);
foo.length = 6;
foo.push(4);
console.log(foo);
```

- a) [1, 2, 3]
- b) [undefined, undefined, undefined, undefined, undefined, undefined, 4]
- c) [1, 2, 3] [1, 2, 3]
- d) [1, 2, 3] [1, 2, 3, undefined, undefined, undefined, 4]

23. What is the output ? [c]

```
"divami".substr(2, 3);
```

- a) "v" b) "va" c) "vam"

24. What is the output ? [b]

```
var a = 10;
a.toString(2);
```

- a) "10" b) "1010" c) 2 Error

25. What is the output ? Ans: _____

```
var n = 3.142857;
n.toFixed(2)    ?
n.toPrecision(3)    ?
n.toFixed()    ?
n.toPrecision()    ?
```

26. How to reverse a string without **for loop** and **split()** function ?

Ans:

```
var str = '12345';
Array.prototype.map.call(str, function(x) {
    return x;
}).reverse().join('');
```

27. How to check the variable is array or not ? Write all possible solutions.

Ans:

1. Array.isArray([1,2,3]);
2. [1,2,3].constructor == Array
3. [1,2,3] instanceof Array

Power of 'this'

1. What is the output?

```
var x = 3;
var foo = {
  x: 2,
  baz: {
    x: 1,
    bar: function() {
      return this.x;
    }
  }
}
var go = foo.baz.bar;
alert(go());
alert(foo.baz.bar());
```

- a) 1, 2 b) 1, 3 c) 2, 1 d) 2, 3 e) 3, 1

2. What is the output?

```
var x = 3;
var elem = document.getElementById('gbw');
function foo() {
  var x = 2;
  this.val = 10;
  function print(e) {
    console.log(this.x, this.val, x);
  }
  elem.addEventListener('click', print);
}
foo();
elem.click();
```

- b) 3, 10, 2 b) 2, 10, 3 c) undefined, undefined, 2 d) 3, undefined, 2

3. What is the output?

```
var x = 3;
var elem = document.getElementById('gbw');
function foo() {
  var x = 2;
  this.x = 4;
  this.val = 10;
  function print(e) {
```

```

        console.log(this.x, this.val, x);
    }
    elem.addEventListener('click', print.bind(this))
}
foo();
elem.click();

```

- c) 3, 10, 2 b) 2, 10, 3 c) 4, 10, 2 d) undefined, undefined, 2

4. What the value of this ?

```

function abc() {
    console.log(this);
}
abc();

```

- a) undefined object b) document object c) window d) null

5. Name the invocation pattern ?

```

function xyz() {
    console.log(this);    // Name the invocation pattern
}
function foo(a) {
    this.name = a;
    xyz();
}
var a = new foo(10);

```

- a) Constructor b) Method c) Function d) Apply

6. What is the output ?

```

var name = "Google";
var obj = {
    name: 'divami',
    getName: function() {
        return this.name;
    }
}
var c = { name: 'Microsoft'};
c.d = obj.getName;
c.d();

```

- a) divami b) Google c) Microsoft

7. Name the invocation pattern(s) in the below code ?

```

var name = "Google";
var obj = {

```

```

        name: 'divami',
        getName: function() {
            return this.name;
        }
    }
    obj.getName();           // Invocation pattern 1
    var x = obj.getName;     // Invocation pattern 2
    x();

```

- a) Method, Function c) Method, Method d) Function, Method
 b) Function, Function

8. What is the console output ?

```

var value = 100;
var obj = {
    value: 10,
    getValue: function() {
        return this.value
    },
    setValue: function(v) {
        this.value = v;
    },
    double: function() {
        var helper = function() {
            this.value = this.value * 2;
        }
        helper();
    }
}
obj.setValue(30);
obj.double();
var go = obj.double;
go();

console.log(obj.getValue());
console.log(value);

```

- a) 20, 200 c) 60, 200
 b) 30, 400 d) 120, 100

9. What is the output ?

```

var name = "google";
var myObj = {
    name: 'divami',
    loc: 'hyd'
}

```

```

    }
    function getName() {
        console.log(this.name);
    }
    var obj = {
        name: "Informatica",
        getName: function() {
            return this.name;
        }
    }

    var myGetName1 = getName.bind(myObj);
    var myGetName2 = obj.getName.bind(myObj);
    getName();
    obj.getName();
    myGetName1();
    myGetName2();

```

10. What is the value of arguments array ?

```

function sum(a, b) {
    console.log(arguments);
}
go = sum.bind(null, 3, 4, 5);
go(1, 2);

```

- a) [1, 2, 3, 4, 5] b) [3, 4, 5] c) [3, 4, 5, 1, 2] d) [1, 2]

11. What is the output ?

```

function employee(name, profession) {
    console.log(this);
}
employee.call(null, "Sumathi", "Architect");

```

- a) null b) undefined c) window d) Error

12. What is the output ?

```

var value = 100;
function calc() {
    this.value = 10,
    this.getValue = function() {
        setTimeout(function() {
            console.log(this.value);
        }, 2000);
    }
}

```



```
var r = new calc();
r.getValue();
```

- a) 100 b) 10 c) undefined d) Error

13. If the 12th program output is not “10” then how can I get “10” ? Change the program accordingly.

Ans: Two ways to do that.

1. var that = this;
2. use bind

14. Modify the function such that it should reverse the parameters. Use Array prototype reverse function.

```
function change() {
    // Code here
}
change(1,2,3,4);           // Should return [4,3,2,1]
change(9,8,7);             // Should return [7,8,9]
```

Ans: return Array.prototype.reverse.apply(arguments);

15. What is the output ?

```
function abc() {
    console.log('divami');
}
var a1 = new abc();
```

```
a1 instanceof abc      ??
a1 instanceof Object   ??
```

- a) false, true b) true, false d) true, true e) false, false

Prototype chain

1. What is the final link in prototype chain.

[a]

- a) null b) Object c) Function d) No final link

2. Is prototype property of any function enumerable? What is the output ?

```
function person() {
    this.location = 'hyd';
}
person.name = 'Sudeep';
person.prototype.designation = "Developer"
person.age = 25;
```

```

for(var k in person) {
    console.log(person[k]);
}

```

- a) hyd, Sudeep, 25 b) hyd c) hyd, age d) Sudeep, Developer, 25
e) 25

3. What is “dunder proto” ? _____

4. Fill in the following blank.

```

function Foo(who) {
    this.name = who;
}
var a1 = new Foo();
a1._____ == Foo.prototype

```

- a) [[Prototype]] b) __proto__ c) prototype d) Both a & b

5. Fill in the following blanks.

```

function Person(who) {
    this.name = who;
    this.location = "hyd";
    return {
    };
}
var p1 = new Person('Naresh');

```

```

p1.name          = _____
p1.constructor   = _____
p1.__proto__     = _____

```

6. What are the 4 steps when we create object with ‘new’ keyword? Please write a code for the four steps when we use new keyword.

7. What is the output ?

```

typeof a1.__proto__

```

- a) “function” b) “object” c) “string” d) “undefined”

8. What are the values printed in console ?

```

function person(name) {
    this.name = name;
    this.location = 'Kerala';
}
person.prototype.officeLocation = 'hyd';

```

```

var p1 = new person('Jinu');
for(var k in p1) {
    console.log(p1[k]);
}

```

- a) Jinu, Kerala, hyd b) Jinu, Kerala c) Error

9. What is the output ?

```

function Shape(id, x, y) {
    this.id = id;
    this.move(x, y);
};
Shape.prototype.move = function (x, y) {
    this.x = x;
    this.y = y;
};
function Rectangle(id, x, y, width, height) {
    Shape.call(this, id, x, y);
    this.width = width;
    this.height = height;
};
Rectangle.prototype = Object.create(Shape.prototype);
Rectangle.prototype.constructor = Rectangle;
Rectangle.prototype = new Shape();
var r1 = new Rectangle(1, 10, 20, 200, 100);

```

Closures

1. What is clouser? Write different ways to create closures ?

2. What is the output ?

```

<div class="name"> Name1 </div>
<div class="name"> Name2 </div>
var name = 'Divami';
function Class(){
    var name = "Software";
    function type(e){
        $(this.currentTarget).html(name);
    }
    return type;
}

```

```
function name(e){
    $(this.currentTarget).html(name);
}

$('.name')[0].on('click', name);
$('.name')[1].on('click', Class());

$('.name')[0].click(); _____
$('.name')[1].click(); _____
```

IIFE

1. What is the output ?

```
function countdown (num) {
    for (var i = 0; i <= num; i += 1) {
        setTimeout(function () {
            console.log(num - i);
        }, i * 2000);
    }
}

countdown(5); _____
```

2. What is the problem in above question? Please give us the solution.

Ans:

3. What is the return value ?

```
(function f(f){
    return typeof f();
})(function(){ return 1; });
```

- a) "number" b) "undefined" c) "function" d) Error

4. What is the return value ?

```
var foo = {
    bar: function() { return this.baz; },
    baz: 1
};

(function(){
    return typeof arguments[0]();
})(foo.bar);
```

- a) "undefined" b) "object" c) "number" d) "function"

5. What is the output ?

```
var f = (function f(){ return "1"; }, function g(){ return 2; })();  
typeof f;
```

- a) "undefined" b) "number" c) "string" d) "function"

6. What is the output ?

```
var foo = (function() {  
    var x = {  
        bar: "bar"  
    };  
    return {obj: x};  
})();  
typeof foo;  
foo.obj.bar;
```

- a) "function", undefined b) "object", bar c) "object", undefined

7. What is the output & which function is closure ?

```
var bar = "bar100";  
function foo() {  
    var bar = "bar";  
    return function baz() {  
        console.log(bar);  
    }  
}  
  
function bam() {  
    foo()();  
}  
  
bam();
```

- a) bar100, baz b) bar, baz c) bar, bam d) bar, foo

Regular Expression

Definitions and Solutions

Lexical Scope

In Javascript, functions have lexical scope. This means that functions create their scope when they are defined not when they are executed.

IIFE

There are no. of ways to call a function immediately. All IIFE requires any kind of enclosure.

example #1

```
(function abc() {  
    console.log('divami');  
})();
```

example #2

```
(function abc() {  
    console.log('divami');  
})();
```

example #3

```
[function abc() {  
    console.log('divami');  
}]();
```

example #4

```
var func = function abc() {  
    console.log('divami');  
}();
```

Function “name” and “length”

1. Functions have a length property that corresponds with the number of named parameters.
2. Functions have a name property that corresponds with the name of the function.
3. Anonymous functions also have a name property, but it returns the empty string.

example #1

```
function abc(a, b) {  
    console.log("hello");  
}  
abc.length; // Gives 2  
abc.name    // "abc"
```

example #2

```
var abc = function(a, b) {  
    console.log("hello");  
}  
abc.length; // Gives 2  
abc.name    // ""
```

example #3

```
var abc = function xyz(a, b) {  
    console.log("hello");  
}  
abc.length; // Gives 2  
abc.name    // "xyz"
```

Miscellaneous

The following works differently. Still we don't know the exact answer.

```
1. 20.toFixed()           // Won't work  
   (20).toFixed()         // Works  
   (or)  
   var a = 20;  
   a.toFixed();           // Works  
  
2. 0.1+0.2 == 0.3         // false
```

Log Prefix

```
function log(){  
    var args = Array.prototype.slice.call(arguments);  
    args.unshift('(app)');  
    console.log.apply(console, args);  
}  
  
log('my message'); // (app) my message  
log('my message', 'your message'); // (app) my message your message
```