

1. What would be the output of following code?

```
var strA = "hi there";  
  
var strB = strA;  
  
strB="bye there!";  
  
console.log (strA)
```

The output will 'hi there'

because we're dealing with strings here. Strings are passed by value, that is, copied.

2. What would be the output of following code?

```
var objA = {prop1: 42};  
  
var objB = objA;  
  
objB.prop1 = 90;  
  
console.log(objA)
```

The output will {prop1: 90}

because we're dealing with objects here. Objects are passed by reference, that is, objA and objB point to the same object in memory.

3. What would be the output of following code?

```
var objA = {prop1: 42};  
  
var objB = objA;  
  
objB = {};  
  
console.log(objA)
```

The output will {prop1: 42}.

When we assign objA to objB, the objB variable will point to the same object as the objB variable.

However, when we reassign objB to an empty object, we simply change where objB variable references to. This doesn't affect where objA variable references to.

4. What would be the output of following code?

```
var arrA = [0,1,2,3,4,5];  
  
var arrB = arrA;arrB[0]=42;  
  
console.log(arrA)
```

The output will be [42,1,2,3,4,5].

Arrays are object in JavaScript and they are passed and assigned by reference. This is why both arrA and arrB point to the same array [0,1,2,3,4,5]. That's why changing the first element of the arrB will also modify arrA: it's the same array in the memory.

5. What would be the output of following code?

```
var arrA = [0,1,2,3,4,5];
```

```
var arrB = arrA.slice();  
arrB[0]=42;  
console.log(arrA)
```

The output will be [0,1,2,3,4,5].

The slice function copies all the elements of the array returning the new array. That's why arrA and arrB reference two completely different arrays.

5. What would be the output of following code?

```
var arrA = [{prop1: "value of array A!!"}, {someProp: "also value of array A!!"},  
3,4,5]; var arrB = arrA; arrB[0].prop1=42; console.log(arrA);
```

The output will be [{prop1: 42}, {someProp: "also value of array A!!"}, 3,4,5].

Arrays are object in JS, so both variables arrA and arrB point to the same array. Changing arrB[0] is the same as changing arrA[0]

6. What would be the output of following code?

```
var arrA = [{prop1: "value of array A!!"},  
{someProp: "also value of array A!!"}, 3,4,5];  
var arrB = arrA.slice();  
arrB[0].prop1=42;  
arrB[3] = 20;  
console.log(arrA);
```

The output will be [{prop1: 42}, {someProp: "also value of array A!!"}, 3,4,5].

The slice function copies all the elements of the array returning the new array. However, it doesn't do deep copying. Instead it does shallow copying. You can imagine slice implemented like this:

```
function slice(arr) {  var result = [];  for (i = 0; i < arr.length; i++) {  
  result.push(arr[i]);  }  return result; }
```

Hoisting

1. console.log(employeeId);
 1. Some Value
 2. Undefined
 3. Type Error
 4. ReferenceError: employeeId is not defined

Answer: 4) ReferenceError: employeeId is not defined

2. What would be the output of following code?

```
console.log(employeeId);  
var employeeId = '19000';
```

5. Some Value

- 6. undefined
- 7. Type Error
- 8. ReferenceError: employeeId is not defined

Answer: 2) undefined

3. What would be the output of following code?

```
var employeeId = '1234abe';  
  
(function(){  
  console.log(employeeId);  
  var employeeId = '122345';})();
```

- 9. '122345'
- 10. undefined
- 11. Type Error
- 12. ReferenceError: employeeId is not defined

Answer: 2) undefined

4. What would be the output of following code?

```
var employeeId = '1234abe';  
  
(function() {  
  console.log(employeeId);  
  var employeeId = '122345';  
  
  (function() {  
    var employeeId = 'abc1234';  
  })();  
})();
```

- 13. '122345'
- 14. undefined
- 15. '1234abe'
- 16. ReferenceError: employeeId is not defined

Answer: 2) undefined

5. What would be the output of following code?

```
(function() {  
  console.log(typeof displayFunc);  
  
  var displayFunc = function(){  
    console.log("Hi I am inside displayFunc");  
  }  
})();
```

- 17. undefined
- 18. function
- 19. 'Hi I am inside displayFunc'
- 20. ReferenceError: displayFunc is not defined

Answer: 1) undefined

6. What would be the output of following code?

```
var employeeId = 'abc123';function foo(){employeeId = '123bcd';return;}foo();console.log(employeeId);
```

- 21. undefined
- 22. '123bcd'
- 23. 'abc123'
- 24. ReferenceError: employeeId is not defined

Answer: 2) '123bcd'

7. What would be the output of following code?

```
var employeeId = 'abc123';  
  
function foo() {  
  employeeId = '123bcd';  
  return;  
}  
  
function employeeId() {}  
  
foo();  
  
console.log(employeeId);
```

- 25. undefined
- 26. '123bcd'
- 27. 'abc123'
- 28. ReferenceError: employeeId is not defined

Answer: 3) 'abc123'

8. What would be the output of following code?

```
var employeeId = 'abc123';  
  
function foo() {  
  employeeId();  
  return;  
}  
  
function employeeId() {  
  console.log(typeof employeeId);  
}  
  
foo();
```

- 29. undefined
- 30. function
- 31. string
- 32. ReferenceError: employeeId is not defined

Answer: 2) 'function'

9. What would be the output of following code?

```
function foo() {  
  employeeId();  
  var product = 'Car';  
  return;  
  function employeeId() {  
    console.log(product);  
  }  
}  
foo();
```

- 33. undefined
- 34. Type Error
- 35. 'Car'
- 36. ReferenceError: product is not defined

Answer: 1) undefined

10. What would be the output of following code?

```
(function foo() {  
  bar();  
  function bar() {  
    abc();  
    console.log(typeof abc);  
  }  
  function abc() {  
    console.log(typeof bar);}}})();
```

- 37. undefined undefined
- 38. Type Error
- 39. function function
- 40. ReferenceError: bar is not defined

Answer: 3) function function

Objects

1. What would be the output of following code ?

```
(function() {'use strict';  
  var person = {name: 'John'};  
  person.salary = '10000$';  
  person['country'] = 'USA';
```

```
Object.defineProperty(person, 'phoneNo', {value: '8888888888', enumerable: true})
console.log(Object.keys(person));
})();
```

- 41. Type Error
- 42. undefined
- 43. ["name", "salary", "country", "phoneNo"]
- 44. ["name", "salary", "country"]

Answer: 3) ["name", "salary", "country", "phoneNo"]

2. What would be the output of following code ?

```
(function() {'use strict';
var person = {name: 'John'};
person.salary = '10000$';
person['country'] = 'USA';
Object.defineProperty(person, 'phoneNo', {value: '8888888888', enumerable: false})
console.log(Object.keys(person)); })();
```

- 45. Type Error
- 46. undefined
- 47. ["name", "salary", "country", "phoneNo"]
- 48. ["name", "salary", "country"]

Answer: 4) ["name", "salary", "country"]

3. What would be the output of following code ?

```
(function() {var objA = {foo: 'foo', bar: 'bar'};
var objB = {foo: 'foo', bar: 'bar'};
console.log(objA == objB);
console.log(objA === objB);})();
```

- 49. false true
- 50. false false
- 51. true false
- 52. true true

Answer: 2) false false

4. What would be the output of following code ?

```
(function() {var objA = new Object({foo: "foo"});
var objB = new Object({foo: "foo"});
console.log(objA == objB);
```

```
console.log(objA === objB);}());
```

53. false true

54. false false

55. true false

56. true true

Answer: 2) false false

5. What would be the output of following code ?

```
(function() {var objA = Object.create({foo: 'foo'});
```

```
var objB = Object.create({foo: 'foo'});
```

```
console.log(objA == objB);
```

```
console.log(objA === objB);}());
```

57. false true

58. false false

59. true false

60. true true

Answer: 2) false false

6. What would be the output of following code ?

```
(function() {var objA = Object.create({foo: 'foo'});
```

```
var objB = Object.create(objA);
```

```
console.log(objA == objB);
```

```
console.log(objA === objB);}());
```

61. false true

62. false false

63. true false

64. true true

Answer: 2) false false

7. What would be the output of following code ?

```
(function() {var objA = Object.create({foo: 'foo'});
```

```
var objB = Object.create(objA);
```

```
console.log(objA.toString() == objB.toString());
```

```
console.log(objA.toString() === objB.toString());}());
```

65. false true

66. false false

67. true false

68. true true

Answer: 4) true true

8. What would be the output of following code ?

```
(function() {  
var objA = Object.create({foo: 'foo'});  
var objB = objA; console.log(objA == objB);  
console.log(objA === objB);  
console.log(objA.toString() == objB.toString());  
console.log(objA.toString() === objB.toString());})();
```

- 69. true true true false
- 70. true false true true
- 71. true true true true
- 72. true true false false

Answer: 3) true true true true

9. What would be the output of following code ?

```
(function() {var objA = Object.create({foo: 'foo'});  
var objB = objA; objB.foo = 'bar';  
console.log(objA.foo);  
console.log(objB.foo);  
})();
```

- 73. foo bar
- 74. bar bar
- 75. foo foo
- 76. bar foo

Answer: 2) bar bar

10. What would be the output of following code ?

```
(function() {var objA = Object.create({foo: 'foo'}); var objB = objA; objB.foo =  
'bar'; delete objA.foo; console.log(objA.foo); console.log(objB.foo);})();
```

- 77. foo bar
- 78. bar bar
- 79. foo foo
- 80. bar foo

Answer: 3) foo foo

11. What would be the output of following code ?

```
(function() {var objA = {foo: 'foo'};  
var objB = objA;  
objB.foo = 'bar';
```



```
delete objA.foo;
console.log(objA.foo);
console.log(objB.foo);
})();
```

- 81. foo bar
- 82. undefined undefined
- 83. foo foo
- 84. undefined bar

Answer: 2) undefined undefined

Arrays

1. What would be the output of following code?

```
(function() {var array = new Array('100');
console.log(array);
console.log(array.length);})();
```

- 85. undefined undefined
- 86. [undefined × 100] 100
- 87. ["100"] 1
- 88. ReferenceError: array is not defined

Answer: 3) ["100"] 1

2. What would be the output of following code?

```
(function() {
var array1 = [];
var array2 = new Array(100);
var array3 = new Array(['1',2,'3',4,5.6]);
console.log(array1);
console.log(array2);
console.log(array3);
console.log(array3.length);})();
```

- 89. [] [] [Array[5]] 1
- 90. [] [undefined × 100] Array[5] 1
- 91. [] [] ['1',2,'3',4,5.6] 5
- 92. [] [] [Array[5]] 5

Answer: 1) [] [] [Array[5]] 1

3. What would be the output of following code?

```
(function () {
var array = new Array('a', 'b', 'c', 'd', 'e');
```

```
array[10] = 'f';  
delete array[10];  
console.log(array.length);})();
```

- 93. 11
- 94. 5
- 95. 6
- 96. undefined

Answer: 1) 11

4. What would be the output of following code?

```
(function(){  
var animal = ['cow','horse'];  
animal.push('cat');  
animal.push('dog','rat','goat');  
console.log(animal.length);})();
```

- 97. 4
- 98. 5
- 99. 6
- 100. undefined

Answer: 3) 6

5. What would be the output of following code?

```
(function(){var animal = ['cow','horse'];  
animal.push('cat');  
animal.unshift('dog','rat','goat');  
console.log(animal);})();
```

- 101. ['dog','rat','goat','cow','horse','cat']
- 102. ['cow','horse','cat','dog','rat','goat']
- 103. Type Error
- 104. undefined

Answer: 1) ['dog','rat','goat','cow','horse','cat']

6. What would be the output of following code?

```
(function(){  
var array = [1,2,3,4,5];  
console.log(array.indexOf(2));  
console.log([ {name: 'John'}, {name: 'John'} ].indexOf({name: 'John'}));  
console.log([ [1],[2],[3],[4] ].indexOf([3]));  
console.log("abcdefgh".indexOf('e'));})();
```

- 105. 1 -1 -1 4
- 106. 1 0 -1 4
- 107. 1 -1 -1 -1
- 108. 1 undefined -1 4

Answer: 1) 1 -1 -1 4

7. What would be the output of following code?

```
(function(){var array = [1,2,3,4,5,1,2,3,4,5,6];
console.log(array.indexOf(2));
console.log(array.indexOf(2,3));
console.log(array.indexOf(2,10));})();
```

- 109. 1 -1 -1
- 110. 1 6 -1
- 111. 1 1 -1
- 112. 1 undefined undefined

Answer: 2) 1 6 -1

8. What would be the output of following code?

```
(function(){var numbers = [2,3,4,8,9,11,13,12,16];
var even = numbers.filter(
function(element, index){
return element % 2 === 0; });
console.log(even);
var containsDivisibleby3 = numbers.some(function(element, index){return element %
3 === 0;});
console.log(containsDivisibleby3);})();
```

- 113. [2, 4, 8, 12, 16] [0, 3, 0, 0, 9, 0, 12]
- 114. [2, 4, 8, 12, 16] [3, 9, 12]
- 115. [2, 4, 8, 12, 16] true
- 116. [2, 4, 8, 12, 16] false

Answer: 3) [2, 4, 8, 12, 16] true

9. What would be the output of following code?

```
(function(){var containers = [2,0,false,"", '12', true];
var containers = containers.filter(Boolean);
console.log(containers);
var containers = containers.filter(Number);
console.log(containers);
var containers = containers.filter(String);
```

```

console.log(containers);
var containers = containers.filter(Object);
console.log(containers);})();
117.      [ 2, '12', true ] [ 2, '12', true ] [ 2, '12', true ] [ 2, '12', true ]
118.      [false, true] [ 2 ] ['12'] [ ]
119.      [2,0,false,"", '12', true] [2,0,false,"", '12', true] [2,0,false,"", '12', true]
          [2,0,false,"", '12', true]
120.      [ 2, '12', true ] [ 2, '12', true, false ] [ 2, '12', true,false ] [ 2, '12', true,false]

```

Answer: 1) [2, '12', true] [2, '12', true] [2, '12', true] [2, '12', true]

10. What would be the output of following code?

```

(function(){var list = ['foo','bar','john','ritz'];
    console.log(list.slice(1));
console.log(list.slice(1,3));
    console.log(list.slice());
console.log(list.slice(2,2));
    console.log(list);})();
121.      ['bar','john','ritz'] ['bar','john'] ['foo','bar','john','ritz'] [] ['foo','bar',
          'john','ritz']
122.      ['bar','john','ritz'] ['bar','john','ritz'] ['foo','bar','john','ritz'] [] ['foo',
          'bar','john','ritz']
123.      ['john','ritz'] ['bar','john'] ['foo','bar','john','ritz'] [] ['foo','bar','john',
          'ritz']
124.      ['foo'] ['bar','john'] ['foo','bar','john','ritz'] [] ['foo','bar','john','ritz']

```

Answer: 1) ['bar','john','ritz'] ['bar','john'] ['foo','bar','john','ritz'] [] ['foo','bar','john','ritz']

11. What would be the output of following code?

```

(function(){var list = ['foo','bar','john'];
    console.log(list.splice(1));
console.log(list.splice(1,2));
    console.log(list);})();
125.      ['bar','john'] [] ['foo']
126.      ['bar','john'] [] ['bar','john']
127.      ['bar','john'] ['bar','john'] ['bar','john']
128.      ['bar','john'] [] []

```

Answer: 1. ['bar','john'] [] ['foo']

12. What would be the output of following code?

```

(function(){var arrayNumb = [2, 8, 15, 16, 23, 42];

```

```
arrayNumb.sort();
```

```
console.log(arrayNumb);})();
```

- 129. [2, 8, 15, 16, 23, 42]
- 130. [42, 23, 26, 15, 8, 2]
- 131. [15, 16, 2, 23, 42, 8]
- 132. [2, 8, 15, 16, 23, 42]

Answer: 3. [15, 16, 2, 23, 42, 8]

Functions

1. What would be the output of following code ?

```
function funcA(){console.log("funcA ", this);
```

```
(function innerFuncA1(){
```

```
console.log("innerFunc1", this);
```

```
(function innerFunA11(){
```

```
console.log("innerFunA11", this);})();})();}
```

```
console.log(funcA());
```

- 133. funcA Window {...} innerFunc1 Window {...} innerFunA11 Window {...}
- 134. undefined
- 135. Type Error
- 136. ReferenceError: this is not defined

Answer: 1)

2. What would be the output of following code ?

```
var obj = {message: "Hello",innerMessage: !(function()  
{console.log(this.message);})()};
```

```
console.log(obj.innerMessage);
```

- 137. ReferenceError: this.message is not defined
- 138. undefined
- 139. Type Error
- 140. undefined true

Answer: 4) undefined true

3. What would be the output of following code ?

```
var obj = {message: "Hello",innerMessage: function() {return this.message;}};
```

```
console.log(obj.innerMessage());
```

- 141. Hello
- 142. undefined
- 143. Type Error
- 144. ReferenceError: this.message is not defined

Answer: 1) Hello

4. What would be the output of following code ?

```
var obj = { message: 'Hello', innerMessage: function () { (function () {  
console.log(this.message); })();  
}};  
console.log(obj.innerMessage());
```

- 145. Type Error
- 146. Hello
- 147. undefined
- 148. ReferenceError: this.message is not defined

Answer: 3) undefined

5. What would be the output of following code ?

```
var obj = { message: 'Hello', innerMessage: function () {  
var self = this;  
(function () {  
console.log(self.message);  
})(); }};  
console.log(obj.innerMessage());
```

- 149. Type Error
- 150. 'Hello'
- 151. undefined
- 152. ReferenceError: self.message is not defined

Answer: 2) 'Hello'

6. What would be the output of following code ?

```
function myFunc(){console.log(this.message);}  
myFunc.message = "Hi John";  
console.log(myFunc());
```

- 153. Type Error
- 154. 'Hi John'
- 155. undefined
- 156. ReferenceError: this.message is not defined

Answer: 3) undefined

7. What would be the output of following code ?

```
function myFunc(){console.log(myFunc.message);}  
myFunc.message = "Hi John";  
console.log(myFunc());
```

- 157. Type Error
- 158. 'Hi John'
- 159. undefined
- 160. ReferenceError: this.message is not defined

Answer: 2) 'Hi John'

8. What would be the output of following code ?

```
function myFunc() {  
  myFunc.message = 'Hi John';  
  console.log(myFunc.message);}  
console.log(myFunc());
```

- 161. Type Error
- 162. 'Hi John'
- 163. undefined
- 164. ReferenceError: this.message is not defined

Answer: 2) 'Hi John'

9. What would be the output of following code ?

```
function myFunc(param1,param2) {  
  console.log(myFunc.length);}  
console.log(myFunc());  
console.log(myFunc("a","b"));  
console.log(myFunc("a","b","c","d"));
```

- 165. 2 2 2
- 166. 0 2 4
- 167. undefined
- 168. ReferenceError

Answer: a) 2 2 2

10. What would be the output of following code ?

```
function myFunc() {  
  console.log(arguments.length);}  
console.log(myFunc());  
console.log(myFunc("a","b"));  
console.log(myFunc("a","b","c","d"));
```

- 169. 2 2 2
- 170. 0 2 4
- 171. undefined
- 172. ReferenceError

Answer: 2) 0 2 4

Object Oriented

1. What would be the output of following code ?

```
function Person(name, age){  
  this.name = name || "John";  
  this.age = age || 24;  
  this.displayName = function(){console.log(this.name);}}  
Person.name = "John";  
Person.displayName = function(){console.log(this.name);}  
var person1 = new Person('John');  
person1.displayName();  
Person.displayName();
```

- 173. John Person
- 174. John John
- 175. John undefined
- 176. John John

Answer: 1) John Person

Scopes

1. What would be the output of following code ?

```
function passWordMngr() {  
  var password = '12345678';  
  this.userName = 'John';  
  return {pwd: password};}  
var userInfo = passWordMngr();  
console.log(userInfo.pwd);  
console.log(userInfo.userName);
```

- 177. 12345678 Window
- 178. 12345678 John
- 179. 12345678 undefined
- 180. undefined undefined

Answer: 3) 12345678 undefined

2. What would be the output of following code ?

```
var employeeId = 'aq123';  
function Employee() {  this.employeeId = 'bq1uy';}
```



```
console.log(Employee.employeeId);
```

```
181.      Reference Error
182.      aq123
183.      bq1uy
184.      undefined
```

Answer: 4) undefined

3. What would be the output of following code ?

```
var employeeId = 'aq123';

function Employee() {this.employeeId = 'bq1uy';}

console.log(new Employee().employeeId);

Employee.prototype.employeeId = 'kj182';

Employee.prototype.JobId = '1BJKSJ';

console.log(new Employee().JobId);

console.log(new Employee().employeeId);
```

```
185.      bq1uy 1BJKSJ bq1uy undefined
186.      bq1uy 1BJKSJ bq1uy
187.      bq1uy 1BJKSJ kj182
188.      undefined 1BJKSJ kj182
```

Answer: 2) bq1uy 1BJKSJ bq1uy

4. What would be the output of following code ?

```
var employeeId = 'aq123';

(function Employee() {try {throw 'foo123';} catch (employeeId) {
console.log(employeeId);}
console.log(employeeId);}());
```

```
189.      foo123 aq123
190.      foo123 foo123
191.      aq123 aq123
192.      foo123 undefined
```

Answer: 1) foo123 aq123

Call, Apply, Bind

1. What would be the output of following code ?

```
(function() {var greet = 'Hello World';

var toGreet = [].filter.call(greet, function(element, index) {return index > 5;});

console.log(toGreet);}());
```

```
193.      Hello World
```

- 194. undefined
- 195. World
- 196. ['W', 'o', 'r', 'l', 'd']

Answer: 4) ['W', 'o', 'r', 'l', 'd']

2. What would be the output of following code ?

```
(function() {var fooAccount = {
name: 'John',amount: 4000,deductAmount: function(amount) {
this.amount -= amount;
return 'Total amount left in account: ' + this.amount;}};
var barAccount = {name: 'John',amount: 6000};
var withdrawAmountBy = function(totalAmount) {return
fooAccount.deductAmount.bind(barAccount, totalAmount)};
console.log(withdrawAmountBy(400)());
console.log(withdrawAmountBy(300)());});
```

- 197. Total amount left in account: 5600 Total amount left in account: 5300
- 198. undefined undefined
- 199. Total amount left in account: 3600 Total amount left in account: 3300
- 200. Total amount left in account: 5600 Total amount left in account: 5600

Answer: 1) Total amount left in account: 5600 Total amount left in account: 5300

3. What would be the output of following code ?

```
(function() {var fooAccount = {name: 'John',amount: 4000,deductAmount:
function(amount) {this.amount -= amount;return this.amount;}};
var barAccount = {name: 'John',amount: 6000};
var withdrawAmountBy = function(totalAmount) {return
fooAccount.deductAmount.apply(barAccount, [totalAmount])};
console.log(withdrawAmountBy(400));
console.log(withdrawAmountBy(300));
console.log(withdrawAmountBy(200));});
```

- 201. 5600 5300 5100
- 202. 3600 3300 3100
- 203. 5600 3300 5100
- 204. undefined undefined undefined

Answer: 1) 5600 5300 5100

4. What would be the output of following code ?

```
(function() {
var fooAccount = {name: 'John',amount: 6000,deductAmount: function(amount)
{this.amount -= amount;return this.amount;}};
```

```

var barAccount = {name: 'John', amount: 4000};

var withdrawAmountBy = function(totalAmount) {return
fooAccount.deductAmount.call(barAccount, totalAmount)};

console.log(withdrawAmountBy(400));
console.log(withdrawAmountBy(300));
console.log(withdrawAmountBy(200));})();

205.      5600 5300 5100
206.      3600 3300 3100
207.      5600 3300 5100
208.      undefined undefined undefined

```

Answer: 2) 3600 3300 3100

5. What would be the output of following code ?

```

(function greetNewCustomer() {
console.log('Hello ' + this.name);
}.bind({name: 'John'}))();

```

209. Hello John
210. Reference Error
211. Window
212. undefined

Answer: 1) Hello John

Callback Functions

1. What would be the output of following code ?

```

function getDataFromServer(apiUrl){
var name = "John";
return {then : function(fn){
fn(name);
}}}

getDataFromServer('www.google.com').then(function(name){console.log(name)});

```

213. John
214. undefined
215. Reference Error
216. fn is not defined

Answer: 1) John

2. What would be the output of following code ?

```

(function(){var arrayNumb = [2, 8, 15, 16, 23, 42];
Array.prototype.sort = function(a,b){return a - b}};

```

```

arrayNumb.sort();
console.log(arrayNumb);})();
(function(){var numberArray = [2, 8, 15, 16, 23, 42];
numberArray.sort(function(a,b){
if(a == b){return 0;}
else{return a < b ? -1 : 1;}});
console.log(numberArray);})();
(function(){var numberArray = [2, 8, 15, 16, 23, 42];
numberArray.sort(function(a,b){return a-b;});
console.log(numberArray);})();
217.      [ 2, 8, 15, 16, 23, 42 ] [ 2, 8, 15, 16, 23, 42 ] [ 2, 8, 15, 16, 23, 42 ]
218.      undefined undefined undefined
219.      [42, 23, 16, 15, 8, 2] [42, 23, 16, 15, 8, 2] [42, 23, 16, 15, 8, 2]
220.      Reference Error

```

Answer: 1) [2, 8, 15, 16, 23, 42] [2, 8, 15, 16, 23, 42] [2, 8, 15, 16, 23, 42]

Return Statement

1. What would be the output of following code ?

```

(function(){function sayHello(){var name = "Hi John";
return {fullName: name}}
console.log(sayHello().fullName);})();
221.      Hi John
222.      undefined
223.      Reference Error
224.      Uncaught TypeError: Cannot read property 'fullName' of undefined

```

Answer: 4) Uncaught TypeError: Cannot read property 'fullName' of undefined

2. What would be the output of following code ?

```

function getNumber(){
return (2,4,5);
}var numb = getNumber();
console.log(numb);
225.      5
226.      undefined
227.      2
228.      (2,4,5)

```

Answer: 1) 5

3. What would be the output of following code ?

```
function getNumber(){return;}var numb = getNumber();console.log(numb);
```

- 229. null
- 230. undefined
- 231. ""
- 232. 0

Answer: 2) undefined

4**. What would be the output of following code ?

```
function mul(x){  
  return function(y){  
    return [x*y, function(z){return x*y + z;}];  
  }  
}console.log(mul(2)(3)[0]);  
console.log(mul(2)(3)[1](4));
```

- 233. 6, 10
- 234. undefined undefined
- 235. Reference Error
- 236. 10, 6

Answer: 1) 6, 10

5**. What would be the output of following code ?

```
function mul(x) {  
  return function(y) {  
    return {result: x * y, sum: function(z) {  
      return x * y + z;}}};  
}console.log(mul(2)(3).result);  
console.log(mul(2)(3).sum(4));
```

- 237. 6, 10
- 238. undefined undefined
- 239. Reference Error
- 240. 10, 6

Answer: 1) 6, 10

6. What would be the output of following code ?

```
function mul(x) {  
  return function(y) {  
    return function(z) {
```

```
return function(w) {  
  return function(p) {  
    return x * y * z * w * p;  
  };  
};  
console.log(mul(2)(3)(4)(5)(6));
```

- 241. 720
- 242. undefined
- 243. Reference Error
- 244. Type Error

Answer: 1) 720