1. Which of the following is not JavaScript Data Types?

D. Float

2. Which company developed JavaScript?

A. Netscape

3. Inside which HTML element do we put the JavaScript?

A. <script>

4. Which of the following is correct about features of JavaScript?

B. JavaScript is a object-based scripting language.

5. Choose the correct JavaScript syntax to change the content of the following HTML code.

B. document.getElementById ("letsfindcourse").innerHTML = "I am a letsfindcourse";

6. which of the following are advantages of JavaScript?

A. Less server interaction

B. Increased interactivity

C. Richer interfaces

D. All of the above

7. Which of the following true about Javascript?

A. Client-side JavaScript does not allow the reading or writing of files

B. JavaScript cannot be used for networking applications

C. JavaScript doesn't have any multi-threading or multiprocessor capabilities

D. All of the above

8. Microsoft has developed a popular HTML editor called?

B. FrontPage

9. HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia.

A. True

10. JavaScript ignores?

A. spaces

B. tabs

C. newlines

D. All of the above

11. JavaScript is designed for following purpose -

C. to add interactivity to html pages

12. What will be the output of the following Javascript code?

```
var string1 = &quot;Letsfindcourse&quot;;
var intvalue = 30;
alert( string1 + intvalue );
```

C. Letsfindcourse30

13. Among the following, which one is a ternary operator in JavaScript?

D. ?:

14. What are the three important manipulations done in a for loop on a

loop variable in javascript?

A. the initialization, the Incrementation, and update

B. the initialization, the test, and the update

C. the initialization, the test, and Incrementation

D. All of the above

15. What does javascript use instead of == and !=?

B. It uses === and !== instead

16. What should appear at the very end of your JavaScript?

The <script language="javascript"> tag

A. <script>

B. </script>

C. </script language="javascript">

D. All of the above

17. Among the keywords below, which one is not a statement?

C. debugger

18. How do we define the term Thread?

C. Controlled execution of applications

19. Which symbol is used for comments in Javascript for single line?

B. //

20. Which of them is not the looping structures in JavaScript?

C. forwhich

21. What is defination of an undefined value in JavaScript?

A. Variable used in the code doesn't exist

B. Variable is not assigned to any value

C. Property doesn't exist

D. All of the above

22. What are the types of Pop up boxes available in JavaScript?

A. Alert

B. Prompt

C. Confirm

D. All of the above

23. what is the disadvantage of using innerHTML in JavaScript?

C. Even if you use +=like "innerHTML = innerHTML + 'html'" still the old content

is replaced by html

24. What are the two basic groups of dataypes in JavaScript?

A. Primitive

B. Reference types.

C. All of the above

25. Which of the following are the errors in JavaScript?

B. Run time errors:

26. Which of the following are the functional components in JavaScript?

A. First-class functions

27. Which of the following is not the properties of screen objects in JavaScript?

B. ColorsDepth

28. Javascript string using double quotes is exactly the same as a string using single quotes?

A. True

29. Find output of below code

var a = '20';

var b = a = 30;

document.write(a+b);

D. 60

30. What is divide by 0 in Javascript? var a = 10;

var b = 0;

document.write(a/b);

C. Infinity is printed

31.Which of the following function of Array object applies a function simultaneously against two values of the array (from right-to-left) as to reduce it to a single value?

D - reduceRight()

32. What is the output of following Javascript?

var a = 'letsfind';

var b = 'course';

var c = a/b;

document.write(c);

C. NaN

33. How ++ works in Javascript? Find output of below Javascript code.

var a = 1;

document.write(a--);

document.write(a);

D. 10

34. Find output of below Javascript addition code

document.write("1 plus 1 is " + 1 + 1);

C. 1 plus 1 is 11

35.In JavaScript, Arrays are data type. State True or False

B. False

36. Does JavaScript allow exception handling?

A. Yes, it provides try, catch as well as throw key word for exception handling

37. Which of the following method checks if its argument is not a number?

A. isNaN()

38. What if you use parseInt() to convert a string containing decimal value?

C. If returns only the integer portion of the number

39. What is the output of below Javascript code?

alert (typeof new Date());

B. object

40. What if we put ++ operator inside if condition? find the output of below code

< script>

var a = 10;

if(a == a++)

document.write(a);

D. 11

41.Which of the following is true?

A. If onKeyDown returns false, the key-press event is cancelled.

42. Syntax for creating a RegExp object:

1. var txt=new RegExp(pattern,attributes);

2. var txt=/pattern/attributes;

Which of the above mentioned syntax will correct?

C. Both 1 and 2

43. If para1 is the DOM object for a paragraph, what is the correct syntax to change the text within the paragraph?

B. para1.value="New Text";

44. The syntax of Eval is _____

B. [objectName.]eval(string)

45. The _____ method of an Array object adds and/or removes elements from an array.

D. Splice

46. Which tag(s) can handle mouse events in Netscape?

A. <IMG>

47. Consider the following code snippet

```
const pi=3.14;
var pi=4;
console.log(pi);
```

What will be the output for the above code snippet?

A. This will flash an error

48. What is the default value of the asyc attribute?

D. True

49. What is the purpose of the Attr object in the HTML DOM?

B. HTML Attribute

50. Which among the following POSIX signals generate events?

C. SIGINT

51. From which version of IE is canvas supported?

D. 9

52. How many parameters does the method plot() accept?

C. 9

53. What is the purpose of script loading?

C. Load JavaScript files programmatically

54. JavaScript can be written _____

A. directly into JS file and included into HTML

55. When a class B can extend another class A, we say that:

A. A is the superclass and B is the subclass

56. Which method receives the return value of setInterval() to cancel future invocations?

C. clearInterval()

57. The pop() method of the array does which of the following task ?

A. decrements the total length by 1

58. What is the need for bubble charts?

B. Represent 3D data

59. How to get a particular value using the tagged name?

C. getElementsbyTagName()

60. Which of the following is the correct syntax to display

"Letsfindcourse" in an alert box using JavaScript?

D. alert("Letsfindcourse");

61. What is the correct syntax for referring to an external script called

"LFC.js"?

A. <script src="LFC.js">

62. Which of the following is not Javascript frameworks or libraries?

C. Cassandra

63. Why so JavaScript and Java have similar name?

B. JavaScript's syntax is loosely based on Java's

64. What is the original name of JavaScript?

C. Mocha

1. The ” var” and “function” are known as _____.

Answer: (d) Declaration statements

2. Which of these is the correct way in which we can call the JavaScript code?

Answer: (c) Function/Method

3. Which of these functions of the Number Object would format a number with different numbers of digits to the decimal’s right?

Answer: (a) toFixed()

4. Out of the following functions of the string object, which one would return the character in any string via the specified number of characters starting at a specified position?

Answer: (b) substr()

5. Look at the snippets given below and check the one in which the variable “a” isn’t equal to the “NULL”.

Answer: (c) if(a!==null)

6. In JavaScript, what do we use for calling the expression for function definition?

Answer: (a) Function literal

7. Which of these is used in JavaScript for calling a method or a function?

Answer: (d) Invocation Expression

8. Which of these operators are used for checking if a specific property exists?

Answer: (a) in

9. “The expression that can appear legally on an assignment expression’s left side” is a common explanation for variables, elements of arrays, and properties of objects. These are known as _____:

Answer: (c) Lvalue

10. Which of these is a correct output for the JavaScript code given below?

string X= “Hey”;

string Y=”There”;

alert(X+Y);

Answer: (c) HeyThere

11. Which of these is known as the Equality operator used for checking whether both the values are equal?

Answer: (b) ==

12. In case a value of an operator is NULL, then the unary operator would return the _____ typeof.

Answer: (d) undefined

13. Which of these is not a keyword?

Answer: (b) use strict

14. Which of these symbols is used to create comments in JavaScript?

Answer: (a) //

15. In the line of code given below, what will the “datatype” written in brackets be called?

article[datatype]=assignment_value;

Answer: (b) A String

16. In the line of code given below, the prototype represents the _____.

functionx(){};

Answer: (a) Prototype of a function

17. Which of these methods or operators are used for identifying an array?

Answer: (a) isarrayType()

18. A function’s execution would stop whenever a program control would encounter the _____ statement in the function’s body.

Answer: (d) return statement

19. Which of these codes is equivalent to the code given below?

Answer: (a) a [“x”] (g , h);

20. Which of these keywords is used to define various functions in JavaScript?

Answer: (a) function

JavaScript MCQ (Multiple Choice Questions)

1. What is JavaScript?

   Answer: JavaScript is a scripting language used to make the website interactive

2. Which of the following is correct about JavaScript?

   Answer: JavaScript is an Object-Based language

3. Among the given statements, which statement defines closures in JavaScript?

   Answer: JavaScript is a function that is enclosed with references to its lexical environment

4. What will be the output of the following JavaScript code snippet?

```
<p id="demo"></p>
var txt1 = "Sanfoundry_";
var txt2 = "Javascriptmcq";
document.getElementById("demo").innerHTML = txt1 + txt2;
```

Answer: d) Sanfoundry_Javascriptmcq

5. What will be the output of the following JavaScript code?

```
<p id="demo"></p>
<script>
var js = 10;
js *= 5;
document.getElementById("demo").innerHTML = js;
</script>
```

Answer: b) 50

6. Arrays in JavaScript are defined by which of the following statements?

Answer: a) It is an ordered list of values

7. What will be the output of the following JavaScript code?

```
// JavaScript Comparison Operators
function compare()
{
    int num=2;
    char b=2;
    if(a==b)
        return true;
    else
        return false;
}
```

Answer: b) true

8. What will be the output of the following JavaScript code snippet?

```
// JavaScript Equalto Operators
function equalto()
{
    int num=10;
    if(num==="10")
        return true;
    else
        return false;
}
```

Answer: a) false

9. Will the following JavaScript code work?

```
var js = (function(x) {return x*x;}(10));
```

Answer: d) Yes, perfectly

10. Which of the following is not javascript data types?

a) Null type
b) Undefined type
c) Number type
d) All of the mentioned

Answer: d

11. Where is Client-side JavaScript code is embedded within HTML documents?

Answer: b) A URL that uses the special javascript:protocol

12. What will be the output of the following JavaScript code snippet?

```
int a=1;
if(a!=null) // JavaScript not equal to Operators
    return 1;
else
    return 0;
```

Answer: b) 1

13. Which of the following object is the main entry point to all client-side JavaScript features and APIs?

Answer: b) Window

14. What will be the output of the following JavaScript program?

```
function sanfoundry(javascript)
{
         return (javascript ?  “yes” :  “no”);
}
         bool ans=true;
console.log(sanfoundry(ans));
```

Answer: c) Yes

15. What will be the output of the following JavaScript code?

```
// Javascript code snippet to compare the height
function height()
{
    var  height = 123.56;
    var type = (height>=190) ? "tall" : "short";
    return type;
}
```

Answer: a) short

16. Which of the following can be used to call a JavaScript Code Snippet?

Answer: a) Function/Method

17. What will be the output of the following JavaScript function?

```
<p id="demo"></p>
<script>
function javascript()
{
// javacript abs() method
    document.getElementById("demo").innerHTML = Math.abs(-7.25);
}
</script>
```

Answer: b) 7.25

18. What will be the output of the following JavaScript code?

```
var a=5 , b=1
var obj = { a : 10 }
// with keyword in JavaScript
with(obj)
{
      alert(b)
}
```

Answer: a) 1

19. Which of the following explains correctly what happens when a JavaScript program is developed on a Unix Machine?

Answer: a) will work perfectly well on a Windows Machine

20. Which is a more efficient JavaScript code snippet?
Code 1 :

```
// for loop in javascript
for(var num=10;num>=1;num--)
{
           document.writeln(num);
}
```

Code 2 :

```
var num=10;
while(num>=1)
{
       document.writeln(num);
       num++;
}
```

Answer: a) Code 1

21. What will be the output of the following JavaScript code?

```
function printArray(a)
{
     var len = a.length, i = 0;
     if (len == 0)
        console.log("Empty Array");
     else
```

```
    {
// do-while loop in javascript
        do
        {
            console.log(a[i]);
        } while (++i < len);
    }
}
```

Answer: c) Prints the numbers in the array in order

22. What happens in the following JavaScript code snippet?

```
var js = 0;
while (js < 10)
{
    console.log(js);
    js++;
}
```

Answer: c) The value of js from 0 to 9 is displayed in the console

23. What will be the output of the following JavaScript code?

```
function range(int javascript)
{
        int a=5;
        for(int i=0;i<javascript;i++)
        {
                console.log(a);
        }
}
range(3);
```

Answer: c) 555

24. Which of the following scoping type does JavaScript use?

Answer: c) Lexical

25. What is the basic difference between JavaScript and Java?

Answer: b) Functions are values, and there is no hard distinction between methods and fields

26. What will be the output of the following JavaScript code?

```
var quiz=[1,2,3];
var js=[6,7,8];
var result=quiz.concat(js);
document.writeln(result);
```

Answer: a) 1, 2, 3, 6, 7, 8

27. Why JavaScript Engine is needed?

Answer: c) Interpreting the JavaScript

28. What will be the function of the following JavaScript program?

```
var scope = "js scope";
function checkscope()
{
    var scope = "javascript scope";
    function f()
    {
         return scope;
    }
    return f;
}
```

Answer: a) Returns the value in scope

29. What will be the output of the following JavaScript code?

```
int a=0;
for(a;a<5;a++);
```

```
console.log(a);
```

Answer: b) 5

30. Which of the following methods/operation does javascript use instead of == and !=?

Answer: d) JavaScript uses === and !== instead

31. What will be the result or type of error if p is not defined in the following JavaScript code snippet?

```
console.log(p)
```

Answer: b) Reference Error

32. What is the prototype represents in the following JavaScript code snippet?

```
function javascript() {};
```

Answer: d) A custom constructor

33. Why event handlers is needed in JS?

Answer: a) Allows JavaScript code to alter the *behaviour* of windows

34. Which of the following is not a framework?

Answer: b) JavaScript

35. Which of the following is the property that is triggered in response to JS errors?

Answer: b) onerror

36. What will be the output of the following JavaScript code?

```
function compare()
{
    int sanfoundry=1;
    char javascript=1;
    if(sanfoundry.tostring()===javascript)
        return true;
    else
        return false;
}
```

Answer: c) true

37. What will be the firstname and surname of the following JavaScript program?

```
var book = {
              "main title": "JavaScript",
              'sub-title': "The Definitive Guide",
              "for": "all audiences",
              author: {
                          firstname: "David",
                          surname: "Flanagan"
                       }
          };
```

Answer: b) property names

38. Which of the following is not an error in JavaScript?

Answer: b) Division by zero

39. Consider the following JavaScript statement containing regular expressions and check if the pattern matches.

```
var text = "testing: 1, 2, 3";
var pattern = /d+/g;
```

Answer: b) pattern.test(text)

Q1. What are the three important manipulations done in for loop on a loop variable?

Ans: Updation,incremantaion,initialization.
Q2. A function definition expression can be called as:

Ans: Function literal.

Q3. The expression of calling (or executing) a function or method in JavaScript is called :

Ans:Functional expressation.

Q4. The script tag must be placed in _____?

Ans:The head or body.

Q5. The four kinds of class members are _____?

Ans: Instance fields,Instance methods,Class fields,Class Q6.

Q6. The scope of a function is also called as _____methods?

Ans:Module function.

Q7. **The basic difference between JavaScript and Java is _____?**
Ans:Functions are values,and there is no hard distinction between methods and fields.

Q8.The regular expression to match any one character not between the brackets is:
Ans: [^…]

Q9. What are the events generated by the Node objects called?
Ans:Emitters.

Q10. Which of the following is an event emitter?
Ans:process.

Q11. Which is the function used to retrieve a value?
Ans:getItem().

Q12. What is the method used to pause “data” events?
Ans:s.pause().

Q13. Which is the handler method used to invoke when uncaught JavaScript exceptions occur?
Ans:Oneerror.

Q14.The paragraph “p” is a part of _____?
Ans:both body and html.

Q15. The URL property belongs to which of the following object?
Ans:Document.

Q16. JavaScript is a single-threaded language?
Ans:True.

Q17. JavaScript uses?
Ans: Lexical Scoping.

Q18. What will be the output of the following code?

```
console.log("50" + 50 - 50)
```

Ans: 5000.

Q19. What will be the output of the following?

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

Ans:20.

Q20. Which of the following statements is correct?
Ans: `splice()` changes original array, `slice()` doesn't.

Q21. What will be the output of the following two codes?

```
function TnS_add(a,b)
{
return a + b;
}
function TnS_spreadadd(...nums){
var count = 0 ;
for(let num of nums)
{
count += num ;
}
return count ;
}

console.log(TnS_add(10,20,40) , TnS_spreadadd(10,20,40))
```

Ans: 30, 70.

Q22. What will be the order of execution?

```
console.log("A")
setTimeout(function(){ console.log("B") }, 0);
console.log("C")
setTimeout(function(){ console.log("D") }, 1000);
```

Ans: A C B D.

Q23. What will be output on the console?

```
for(var i=0 ;i<5;i++)
{
setTimeout(function(){ console.log(i) }, i);
}
```

Ans: 0 1 2 3 4.

Q24. What will be the output?

```
for(let i=0 ;i<5;i++)
  {
   setTimeout(function(){ console.log(i) }, i);
  }
```

Ans: 0 1 2 3 4.

Q25. What will be the output of `TnS_Traingle`?

```
function TnS_Rectangle()
{
var height = 10 ;
let base = 20 ;

if(height > 5)
{
let base = 10 ;
}

return base*height ;
}
```

Ans: **200.**

Q1: Explain what a *callback* function is and provide a simple example?

Answer

A `callback` function is a function that is passed to another function as an argument and is executed after some operation has been completed. Below is an example of a simple callback function that logs to the console *after* some operations have been completed.

```
function modifyArray(arr, callback) {
  // do something to arr here
  arr.push(100);
  // then execute the callback function that was passed
  callback();
}

var arr = [1, 2, 3, 4, 5];

modifyArray(arr, function() {
  console.log("array has been modified", arr);
});
```

Q2: Given a string, reverse each word in the sentence?

Answer

```
var string = "Welcome to this Javascript Guide!";

// Output becomes !ediuG tpircsavaJ siht ot emocleW
var reverseEntireSentence = reverseBySeparator(string, "");
```

```
// Output becomes emocleW ot siht tpircsavaJ !ediuG
var reverseEachWord = reverseBySeparator(reverseEntireSentence, " ");

function reverseBySeparator(string, separator) {
  return string.split(separator).reverse().join(separator);
}
```

Q3: How to check if an object is an array or not? Provide some code?

Answer

The best way to find whether an object is instance of a particular class or not using `toString` method from `Object.prototype`

```
var arrayList = [1 , 2, 3];
```

One of the best use cases of type checking of an object is when we do method overloading in JavaScript. For understanding this let say we have a method called `greet` which take one single string and also a list of string, so making our `greet` method workable in both situation we need to know what kind of parameter is being passed, is it single value or list of value?

```
function greet(param) {
  if() {
    // here have to check whether param is array or not
  }
  else {
  }
}
```

However, in above implementation it might not necessary to check type for array, we can check for single value string and put array logic code in else block, let see below code for the same.

```
 function greet(param) {
   if(typeof param === 'string') {
   }
   else {
     // If param is of type array then this block of code would execute
   }
 }
```

Now it's fine we can go with above two implementations, but when we have a situation like a parameter can be `single value`, `array`, and `object` type then we will be in trouble.
Coming back to checking type of object, As we mentioned that we can use `Object.prototype.toString`

```
if(Object.prototype.toString.call(arrayList) === '[object Array]') {
  console.log('Array!');
}
```

If you are using `jQuery` then you can also used jQuery `isArray` method:

```
if($.isArray(arrayList)) {
  console.log('Array');
} else {
  console.log('Not an array');
}
```

FYI jQuery uses `Object.prototype.toString.call` internally to check whether an object is an array or not.
In modern browser, you can also use:

```
Array.isArray(arrayList);
```

`Array.isArray` is supported by Chrome 5, Firefox 4.0, IE 9, Opera 10.5 and Safari 5

Q4: How to empty an array in JavaScript?

Problem

```
var arrayList =  ['a', 'b', 'c', 'd', 'e', 'f'];
```

How could we empty the array above?

Answer

Method 1

```
arrayList = [];
```

Above code will set the variable `arrayList` to a new empty array. This is recommended if you don't have **references to the original array** `arrayList` anywhere else because It will actually create a new empty array. You should be careful with this way of empty the array, because if you have referenced this array from another variable, then the original reference array will remain unchanged, Only use this way if you have only referenced the array by its original variable `arrayList`.

For Instance:

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f']; // Created array
var anotherArrayList = arrayList;  // Referenced arrayList by another variable
arrayList = []; // Empty the array
console.log(anotherArrayList); // Output ['a', 'b', 'c', 'd', 'e', 'f']
```

Method 2

```
arrayList.length = 0;
```

Above code will clear the existing array by setting its length to 0. This way of empty the array also update all the reference variable which pointing to the original array. This way of empty the array is useful when you want to update all the another reference variable which pointing to `arrayList`.

For Instance:

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f']; // Created array
var anotherArrayList = arrayList;  // Referenced arrayList by another variable
arrayList.length = 0; // Empty the array by setting length to 0
console.log(anotherArrayList); // Output []
```

Method 3

```
arrayList.splice(0, arrayList.length);
```

Above implementation will also work perfectly. This way of empty the array will also update all the references of the original array.

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f']; // Created array
var anotherArrayList = arrayList;  // Referenced arrayList by another variable
arrayList.splice(0, arrayList.length); // Empty the array by setting length to 0
console.log(anotherArrayList); // Output []
```

Method 4

```
while(arrayList.length) {
  arrayList.pop();
}
```

Above implementation can also empty the array. But not recommended to use often.

Q5: How would you check if a number is an integer?

Answer

A very simply way to check if a number is a decimal or integer is to see if there is a remainder left when you divide by 1.

```
function isInt(num) {
  return num % 1 === 0;
}

console.log(isInt(4)); // true
console.log(isInt(12.2)); // false
console.log(isInt(0.3)); // false
```

Q6: Implement enqueue and dequeue using only two stacks?

Answer

Enqueue means to add an element, *dequeue* to remove an element.

```
var inputStack = []; // First stack
var outputStack = []; // Second stack

// For enqueue, just push the item into the first stack
function enqueue(stackInput, item) {
  return stackInput.push(item);
}

function dequeue(stackInput, stackOutput) {
  // Reverse the stack such that the first element of the output stack is the
  // last element of the input stack. After that, pop the top of the output to
  // get the first element that was ever pushed into the input stack
  if (stackOutput.length <= 0) {
    while(stackInput.length > 0) {
      var elementToOutput = stackInput.pop();
      stackOutput.push(elementToOutput);
    }
  }

  return stackOutput.pop();
}
```

Q7: Make this work

Problem

```
duplicate([1, 2, 3, 4, 5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(arr) {
  return arr.concat(arr);
}

duplicate([1, 2, 3, 4, 5]); // [1,2,3,4,5,1,2,3,4,5]
```

Q8: Write a "mul" function which will properly when invoked as below syntax?

Problem

```
console.log(mul(2)(3)(4)); // output : 24
console.log(mul(4)(3)(4)); // output : 48
```

Answer

```
function mul (x) {
```

```
  return function (y) { // anonymous function
    return function (z) { // anonymous function
      return x * y * z;
    };
  };
}
```

Here `mul` function accept the first argument and return anonymous function which take the second parameter and return anonymous function which take the third parameter and return multiplication of arguments which is being passed in successive
In JavaScript function defined inside has access to outer function variable and function is the first class object so it can be returned by function as well and passed as argument in another function.

- A function is an instance of the Object type
- A function can have properties and has a link back to its constructor method
- Function can be stored as variable
- Function can be pass as a parameter to another function
- Function can be returned from function

Q9: Write a function that would allow you to do this?

Problem

```
var addSix = createBase(6);
addSix(10); // returns 16
addSix(21); // returns 27
```

Answer

You can create a closure to keep the value passed to the function `createBase` even after the inner function is returned. The inner function that is being returned is created within an outer function, making it a closure, and it has access to the variables within the outer function, in this case the variable `baseNumber`.

```
function createBase(baseNumber) {
  return function(N) {
    // we are referencing baseNumber here even though it was declared
    // outside of this function. Closures allow us to do this in JavaScript
    return baseNumber + N;
  }
}

var addSix = createBase(6);
addSix(10);
addSix(21);
```

Q10: FizzBuzz Challenge?

Problem

Create a for loop that iterates up to `100` while outputting **"fizz"** at multiples of `3`, **"buzz"** at multiples of `5` and **"fizzbuzz"** at multiples of `3` and `5`.

Answer

Check out this version of FizzBuzz:

```
for (let i = 1; i <= 100; i++) {
  let f = i % 3 == 0,
    b = i % 5 == 0;
  console.log(f ? (b ? 'FizzBuzz' : 'Fizz') : b ? 'Buzz' : i);
}
```

Q11: Given two strings, return true if they are anagrams of one another?

Problem

For example: `Mary` is an anagram of `Army`

Answer

```
var firstWord = "Mary";
var secondWord = "Army";

isAnagram(firstWord, secondWord); // true

function isAnagram(first, second) {
  // For case insensitivity, change both words to lowercase.
  var a = first.toLowerCase();
  var b = second.toLowerCase();

  // Sort the strings, and join the resulting array to a string. Compare the results
  a = a.split("").sort().join("");
  b = b.split("").sort().join("");

  return a === b;
}
```

Q12: How would you use a closure to create a private counter?

Answer

You can create a function within an outer function (a closure) that allows you to update a private variable but the variable wouldn't be accessible from outside the function without the use of a helper function.

```
function counter() {
  var _counter = 0;
  // return an object with several functions that allow you
  // to modify the private _counter variable
  return {
    add: function(increment) { _counter += increment; },
    retrieve: function() { return 'The counter is currently at: ' + _counter; }
  }
}

// error if we try to access the private variable like below
// _counter;
```

```
// usage of our counter function
var c = counter();
c.add(5);
c.add(9);

// now we can access the private variable in the following way
c.retrieve(); // => The counter is currently at: 14
```

Q13: Provide some examples of non-bulean value coercion to a boolean one?

Answer
The question is when a non-boolean value is coerced to a boolean, does it become `true` or `false`, respectively?
The specific list of "falsy" values in JavaScript is as follows:

- `""` (empty string)
- `0`, `-0`, `NaN` (invalid number)
- `null`, `undefined`
- `false`

Any value that's not on this "falsy" list is "truthy." Here are some examples of those:

- `"hello"`
- `42`
- `true`
- `[ ]`, `[ 1, "2", 3 ]` (arrays)
- `{ }`, `{ a: 42 }` (objects)
- `function foo() { .. }` (functions)

Q14: What will be the output of the following code?

Problem

```
var y = 1;
if (function f() {}) {
  y += typeof f;
}
console.log(y);
```

Answer

Above code would give output `1undefined`. If condition statement evaluate using `eval` so `eval(function f() {})` which return `function f() {}` which is true so inside if statement code execute. `typeof f` return undefined because if statement code execute at run time, so statement inside `if` condition evaluated at run time.

```
var k = 1;
if (1) {
  eval(function foo() {});
  k += typeof foo;
```

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

Above code will also output `1undefined`.

```
var k = 1;
if (1) {
  function foo() {};
  k += typeof foo;
}
console.log(k); // output 1function
```

Q15: What will the following code output?

Problem

```
(function() {
  var a = b = 5;
})();

console.log(b);
```

Answer

The code above will output 5 even though it seems as if the variable was declared within a function and can't be accessed outside of it. This is because

```
var a = b = 5;
```

is interpreted the following way:

```
var a = b;
b = 5;
```

But `b` is not declared anywhere in the function with `var` so it is set equal to 5 in the *global scope*.

Q16: Write a function that would allow you to do this?

Problem

```
multiply(5)(6);
```

Answer

You can create a *closure* to keep the value of a even after the inner function is returned. The inner function that is being returned is created within an outer function, making it a closure, and it has access to the variables within the outer function, in this case the variable `a`.

```
function multiply(a) {
  return function(b) {
    return a * b;
  }
}

multiply(5)(6);
```

Q17: How does the `this` keyword work? Provide some code examples

Answer

In JavaScript *this* always refers to the “owner” of the function we're executing, or rather, to the object that a function is a method of.

Consider:

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

```
var bar = "global";

var obj1 = {
          bar: "obj1",
          foo: foo
};

var obj2 = {
          bar: "obj2"
};

foo();                          // "global"
obj1.foo();             // "obj1"
foo.call( obj2 );  // "obj2"
new foo();               // undefined
```

Q18: How would you create a private variable in JavaScript?

Answer

To create a private variable in JavaScript that cannot be changed you need to create it as a local variable within a function. Even if the function is executed the variable cannot be accessed outside of the function. For example:

```
function func() {
  var priv = "secret code";
}

console.log(priv); // throws error
```

To access the variable, a helper function would need to be created that returns the private variable.

```
function func() {
  var priv = "secret code";
  return function() {
    return priv;
  }
}

var getPriv = func();
console.log(getPriv()); // => secret code
```

Q19: What is *Closure* in JavaScript? Provide an example

Answer

A *closure* is a function defined inside another function (called parent function) and has access to the variable which is declared and defined in parent function scope.

The closure has access to variable in three scopes:

- Variable declared in his own scope
- Variable declared in parent function scope
- Variable declared in global namespace

```
var globalVar = "abc";

// Parent self invoking function
(function outerFunction (outerArg) { // begin of scope outerFunction
  // Variable declared in outerFunction function scope
  var outerFuncVar = 'x';
  // Closure self-invoking function
```

```
  (function innerFunction (innerArg) { // begin of scope innerFunction
    // variable declared in innerFunction function scope
    var innerFuncVar = "y";
    console.log(
      "outerArg = " + outerArg + "\n" +
      "outerFuncVar = " + outerFuncVar + "\n" +
      "innerArg = " + innerArg + "\n" +
      "innerFuncVar = " + innerFuncVar + "\n" +
      "globalVar = " + globalVar);
  // end of scope innerFunction
  })(5); // Pass 5 as parameter
// end of scope outerFunction
})(7); // Pass 7 as parameter
```

`innerFunction` is closure which is defined inside `outerFunction` and has access to all variable which is declared and defined in outerFunction scope. In addition to this function defined inside function as closure has access to variable which is declared in `global namespace`.
Output of above code would be:

```
outerArg = 7
outerFuncVar = x
innerArg = 5
innerFuncVar = y
globalVar = abc
```

Q20: What will be the output of the following code?

Problem

```
var output = (function(x) {
  delete x;
  return x;
})(0);

console.log(output);
```

Answer
Above code will output `0` as output. `delete` operator is used to delete a property from an object. Here `x` is not an object it's **local variable**. `delete` operator doesn't affect local variable.

Q21: What will be the output of the following code?

Problem

```
var Employee = {
  company: 'xyz'
}
var emp1 = Object.create(Employee);
delete emp1.company
console.log(emp1.company);
```

Answer
Above code will output `xyz` as output. Here `emp1` object got company as **prototype** property. `delete` operator doesn't delete prototype property. `emp1` object doesn't have **company** as its own property. You can test it like:

```
console.log(emp1.hasOwnProperty('company')); //output : false
```

However, we can delete company property directly from `Employee` object using `delete Employee.company` or we can also delete from `emp1` object using `__proto__` property `delete emp1.__proto__.company`.

Q22: What will the following code output?

Problem

```
0.1 + 0.2 === 0.3
```

Answer

This will surprisingly output `false` because of floating point errors in internally representing certain numbers. `0.1 + 0.2` does not nicely come out to `0.3` but instead the result is actually `0.30000000000000004` because the computer cannot internally represent the correct number. One solution to get around this problem is to round the results when doing arithmetic with decimal numbers.

Q23: When would you use the `bind` function?

Answer
The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called.
A good use of the `bind` function is when you have a particular function that you want to call with a specific this value. You can then use `bind` to pass a specific object to a function that uses a `this` reference.

```
function fullName() {
  return "Hello, this is " + this.first + " " + this.last;
}

console.log(fullName()); // => Hello this is undefined undefined

// create a person object and pass its values to the fullName function
var person = {first: "Foo", last: "Bar"};
console.log(fullName.bind(person)()); // => Hello this is Foo Bar
```

Q24: Write a recursive function that performs a binary search?

Answer

```
function recursiveBinarySearch(array, value, leftPosition, rightPosition) {
  // Value DNE
  if (leftPosition > rightPosition) return -1;

  var middlePivot = Math.floor((leftPosition + rightPosition) / 2);
  if (array[middlePivot] === value) {
    return middlePivot;
  } else if (array[middlePivot] > value) {
    return recursiveBinarySearch(array, value, leftPosition, middlePivot - 1);
  } else {
    return recursiveBinarySearch(array, value, middlePivot + 1, rightPosition);
  }
}
```

Q25: Describe the Revealing Module Pattern design pattern?

A variation of the **module pattern** is called the **Revealing Module Pattern**. The purpose is to maintain encapsulation and reveal certain variables and methods returned in an object literal. The direct implementation looks like this:

```
var Exposer = (function() {
  var privateVariable = 10;

  var privateMethod = function() {
    console.log('Inside a private method!');
    privateVariable++;
  }

  var methodToExpose = function() {
    console.log('This is a method I want to expose!');
```

```
  }

  var otherMethodIWantToExpose = function() {
    privateMethod();
  }

  return {
      first: methodToExpose,
      second: otherMethodIWantToExpose
  };
})();

Exposer.first();        // Output: This is a method I want to expose!
Exposer.second();       // Output: Inside a private method!
Exposer.methodToExpose; // undefined
```

An obvious disadvantage of it is unable to reference the private methods

END