Quiz-5, CCSL, Round-39

1. Write down two window and history method.
2. What is the indexOf()?
3. What is object and write 3 built in objects?
4. What is group function? Write down types of group function.
5. What is order by clause?
6. What are the differences between setIntervale() and setTimeout()?
7. What is event and eventhandler?
8. What are the differences between substring() and substr()?
9. What is function? Write down role of return statement.
10. What do you mean by cookie?

Quiz-6, CCSL, Round-39

1. Consider the code snippet given below

var count = [1,,3];

What is the observation made?  
a) The omitted value takes “undefined”  
b) This results in an error  
c) This results in an exception  
d) None of the mentioned

Answer: a  
Explanation: If you omit a value from an array literal, the omitted element is given the value.

2. Consider the following code snippet

var a1 = [,,,];

var a2 = new Array(3);

0 in a1

0 in a2

The result would be  
a) true false  
b) false true  
c) true true  
d) false true

Answer: a  
Explanation: a1 has an element with index 0 and a2 has no element with index 0.

**3. The pop() method of the array does which of the following task ?**  
a) decrements the total length by 1  
b) increments the total length by 1  
c) prints the first element but no effect on the length  
d) None of the mentioned

Answer: a  
Explanation: Arrays have a pop() method (it works with push()) that reduces the length of an array by 1 but also returns the value of the deleted element.

4. Consider the following code snippet :

if (!a[i]) continue;

What is the observation made ?  
a) Skips the undefined elements  
b) Skips the non existent elements  
c) Skips the null elements  
d) All of the mentioned

Answer: d  
Explanation: None.

6. Consider the following code snippet :

advertisement

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

a.slice(0,3);

What is the possible output for the above code snippet ?  
a) Returns [1,2,3].  
b) Returns [4,5].  
c) Returns [1,2,3,4].  
d) Returns [1,2,3,4,5].

Answer: a  
Explanation: None.

7. Consider the following code snippet :

var a = [];

a.unshift(1);

a.unshift(22);

a.shift();

a.unshift(3,[4,5]);

a.shift();

a.shift();

a.shift();

The final output for the shift() is  
a) 1  
b) [4,5].  
c) [3,4,5].  
d) Exception is thrown  
View Answer

Answer: a  
Explanation: The unshift() and shift() methods behave much like push() and pop(), except that they insert and remove elements from the beginning of an array rather than from the end. unshift() adds an element or elements to the beginning of the array, shifts the existing array elements up to higher indexes to make room, and returns the new length of the array. shift() removes and returns the first element of the array, shifting all subsequent elements down one place to occupy the newly vacant space at the start of the array.

10. The method or operator used to identify the array is  
a) isarrayType()  
b) ==  
c) ===  
d) typeof

Answer: d  
Explanation: The typeof property is used to identify the array type.

1. Consider the following code snippet

function printArray(a)

{

var len = a.length, i = 0;

if (len == 0)

console.log("Empty Array");

else

{

do

{

console.log(a[i]);

} while (++i < len);

}

}

What does the above code result?  
a) Prints the numbers in the array in order  
b) Prints the numbers in the array in the reverse order  
c) Prints 0 to the length of the array  
d) Prints “Empty Array”

Answer: a  
Explanation: The do/while loop is less commonly used when compared to the while loop. Here, it prints from the array in the given order.

2. What are the three important manipulations done in a for loop on a loop variable?  
a) Updation, Incrementation, Initialization  
b) Initialization,Testing, Updation  
c) Testing, Updation, Testing  
d) Initialization,Testing, Incrementation

Answer: b  
Explanation: In a for loop, the initialization, the test, and the update are the three crucial manipulations of a loop variable.

3. Consider the following code snippet

function tail(o)

{

for (; o.next; o = o.next) ;

return o;

}

Will the above code snippet work? If not, what will be the error?  
a) No, this will throw an exception as only numerics can be used in a for loop  
b) No, this will not iterate  
c) Yes, this will work  
d) No, this will result in a runtime error with the message “Cannot use Linked List”

Answer: c  
Explanation: The above code uses a for loop to traverse a linked list data structure and return the last object in the list. This will perfectly work.

4. Consider the following code snippet

for(var p in o)

console.log(o[p]);

The above code is equivalent to which code?

a) for (var i = 0;i < a.length;i++)

console.log(a[i]);

b) for (int i = 0;i < a.length;i++)

console.log(a[i]);

c) for (var i = 0;i <= a.length;i++)

console.log(a[i]);

d) for (var i = 1;i < a.length;i++)

console.log(a[i]);

Answer: a  
Explanation: The for/in loop makes it easy to do the same that we do using a for.

5. One of the special feature of an interpreter in reference with the for loop is that  
a) Before each iteration, the interpreter evaluates the variable expression and assigns the name of the property  
b) The iterations can be infinite when an interpreter is used  
c) The body of the loop is executed only once  
d) All of the mentioned  
View Answer

Answer: a  
Explanation: Before each iteration, the interpreter evaluates the variable expression and assigns the name of the property (a string value) to it.

advertisement

6. What will happen if the body of a for/in loop deletes a property that has not yet been enumerated?  
a) The property will be stored in a cache  
b) The loop will not run  
c) That property will not be enumerated  
d) All of the mentioned  
View Answer

Answer: c  
Explanation: If the body of a for/in loop deletes a property that has not yet been enumerated, that property will not be enumerated. If the body of the loop defines new properties on the object, those properties will generally not be enumerated.

7. What will be the step of the interpreter in a jump statement when an exception is thrown?  
a) The interpreter stops its work  
b) The interpreter throws another exception  
c) The interpreter jumps to the nearest enclosing exception handler  
d) None of the mentioned  
View Answer

Answer: c  
Explanation: When an exception is thrown in a jump statement, the interpreter jumps to the nearest enclosing exception handler, which may be in the same function or up the call stack in an invoking function.

8. Consider the following code snippet

while (a != 0)

{

if (a == 1)

continue;

else

a++;

}

What will be the role of the **continue** keyword in the above code snippet?  
a) The continue keyword restarts the loop  
b) The continue keyword skips the next iteration  
c) The continue keyword skips the rest of the statements in that iteration  
d) None of the mentioned  
View Answer

Answer: c  
Explanation: Instead of exiting a loop like the **break** keyword, the **continue** keyword moves to the next iteration from the place encountered.

9. Consider the following code snippet

function f(o)

{

if (o === undefined) debugger;

}

What could be the task of the statement debugger?  
a) It does nothing but a simple breakpoint  
b) It debugs the error in that statement and restarts the statement’s execution  
c) It is used as a keyword that debugs the entire program at once  
d) All of the mentioned  
View Answer

Answer: a  
Explanation: The **debugger** statement normally does nothing. If, however, a debugger program is available and is running, then an implementation may (but is not required to) perform some kind of debugging action. In practice, this statement acts like a breakpoint: execution of JavaScript code stops and you can use the debugger to print variable’s values.

10. Among the keywords below, which one is not a statement?  
a) debugger  
b) with  
c) if  
d) use strict  
View Answer

Answer: d  
Explanation: **use strict** is a directive introduced in ECMAScript5. Directives are not statements because it does not include any language keywords. Also, it can appear only at the start of a script or at the start of a function body, before any real statement have appeared.

1. The function definitions in JavaScript begins with  
a) Identifier and Parentheses  
b) Return type and Identifier  
c) Return type, Function keyword, Identifier and Parentheses  
d) Identifier and Return type  
View Answer

Answer: a  
Explanation: The function definitions begin with the keyword function followed by an identifier that names the function and a pair of parentheses around a comma-separated list of zero or more identifiers.

2. Consider the following code snippet

function printprops(o)

{

for(var p in o)

console.log(p + ": " + o[p] + "\n");

}

What will the above code snippet result ?  
a) Prints the contents of each property of o  
b) Returns undefined  
c) All of the mentioned  
d) None of the mentioned  
View Answer

Answer: b  
Explanation: The above code snippet returns undefined.

3. When does the function name become optional in JavaScript?  
a) When the function is defined as a looping statement  
b) When the function is defined as expressions  
c) When the function is predefined  
d) All of the mentioned  
View Answer

Answer: b  
Explanation: The function name is optional for functions defined as expressions. A function declaration statement actually **declares** a variable and assigns a function object to it.

4. What is the purpose of a **return** statement in a function?  
a) Returns the value and continues executing rest of the statements, if any  
b) Returns the value and stops the program  
c) Returns the value and stops executing the function  
d) Stops executing the function and returns the value  
View Answer

Answer: d  
Explanation: The **return** statement causes the function to stop executing and to return the value of its expression (if any) to the caller.

5. What will happen if a return statement does not have an associated expression?  
a) It returns the value 0  
b) It will throw an exception  
c) It returns the **undefined** value  
d) None of the mentioned  
View Answer

Answer: c  
Explanation: If the **return** statement does not have an associated expression, it returns the **undefined** value.

6. A function with no return value is called  
a) Procedures  
b) Method  
c) Static function  
d) Dynamic function  
View Answer

Answer: a  
Explanation: Functions with no return value are sometimes called **procedures**.

7. Consider the following code snippet

function hypotenuse(a, b)

{

function square(x)

{

return x\*x;

}

return Math.sqrt(square(a) + square(b));

}

What does the above code result?  
a) Sum of square of a and b  
b) Square of sum of a and b  
c) Sum of a and b square  
d) None of the mentioned  
View Answer

Answer: a  
Explanation: The above code snippet contains nested function in which the function hypotenuse(a,b) has another function inside its scope, function square(x). The interesting thing about nested functions is their variable scoping rules. They can access the parameters and variables of the function (or functions) they are nested within.

8. Which of the following is the correct code for invoking a function without **this** keyword at all, and also too determine whether the strict mode is in effect?

advertisement

a) var strict = (function { return this; });

b) mode strict = (function() { return !this; }());

c) var strict = (function() { return !this; }());

d) mode strict = (function { });

View Answer

Answer: c  
Explanation: The above code defines and invokes a function to determine if we’re in strict mode.

   
9. Which is an equivalent code to invoke a function **m** of class **o** that expects two arguments x and y?

a) o(x,y);

b) o.m(x) && o.m(y);

c) m(x,y);

d) o.m(x,y);

View Answer

Answer: d  
Explanation: The code above is an invocation expression: it includes a function expression **o.m** and two argument expressions, **x** and **y**.

   
10. Consider the following code snippet

o.m(x,y);

Which is an equivalent code for the above code snippet?

a) o.m(x) && o.m(y);

b) o["m"](x,y);

c) o(m)["x","y"];

d) o.m(x && y);

View Answer

Answer: b  
Explanation: Another way to write o.m(x,y) is o[“m”](x,y).

1. Consider the following code snippet :

var grand\_Total=eval("10\*10+5");

The output for the above statement would be :  
a) 10\*10+5  
b) 105 as a string  
c) 105 as an integer value  
d) Exception is thrown  
View Answer

Answer: c  
Explanation: Even if the string value passed as a parameter to **eval** does represent a numeric value the use of **eval()** results in an error being generated.

2. Do functions in JavaScript necessarily return a value ?  
a) It is mandatory  
b) Not necessary  
c) Few functions return values by default  
d) All of the mentioned  
View Answer

Answer: c  
Explanation: None.

3. Consider the following code snippet :

var tensquared = (function(x) {return x\*x;}(10));

Will the above code work ?  
a) Yes, perfectly  
b) Error  
c) Exception will be thrown  
d) Memory leak  
View Answer

Answer: a  
Explanation: Function name is optional for functions defined as expressions. Function expressions are sometimes defined and immediately invoked.

4. Consider the following code snippet :

var string2Num=parseInt("123xyz");

The result for the above code snippet would be :  
a) 123  
b) 123xyz  
c) Exception  
d) NaN  
View Answer

Answer: b  
Explanation: The parseInt() function returns the first integer contained in the string or 0 if the string does not begin with an integer.

5. The one-liner code that concatenates all strings passed into a function is

a) function concatenate()

{

return String.prototype.concat('', arguments);

}

b) function concatenate()

{

return String.prototype.apply('', arguments);

}

c) function concatenate()

{

return String.concat.apply('', arguments);

}

d) function concatenate()

{

return String.prototype.concat.apply('', arguments);

}

View Answer

Answer: d  
Explanation: None

6. If you have a function f and an object o, you can define a method named m of o with  
a) o.m=m.f;  
b) o.m=f;  
c) o=f.m;  
d) o=f;  
View Answer

Answer: a  
Explanation: A method is nothing more than a JavaScript function that is stored in a property of an object. If you have a function f and an object o, you can define a method named m of o with the following line:  
**o.m = f;**

7. For the below mentioned code snippet:

var o = new Object();

The equivalent statement is:

a) var o = Object();

b) var o;

c) var o= new Object;

d) Object o=new Object();

View Answer

Answer: c  
Explanation: You can always omit a pair of empty parentheses in a constructor invocation.

8. What is the difference between the two lines given below ?

advertisement

!!(obj1 && obj2);

(obj1 && obj2);

a) Both the lines result in a boolean value “True”  
b) Both the lines result in a boolean value “False”  
c) Both the lines checks just for the existence of the object alone  
d) The first line results in a *real* boolean value whereas the second line merely checks for the existence of the objects  
View Answer

Answer: d  
Explanation: None.

9. Consider the following code snippet :

var c = counter(), d = counter();

c.count()

d.count()

c.reset()

c.count()

d.count()

The state stored in d is :  
a) 1  
b) 0  
c) Null  
d) Undefined  
View Answer

Answer: a  
Explanation: The state stored in d is 1 because d was not reset.

10. Consider the following code snippet :

function constfuncs()

{

var funcs = [];

for(var i = 0; i < 10; i++)

funcs[i] = function() { return i; };

return funcs;

}

var funcs = constfuncs();

funcs[5]()

What does the last statement return ?  
a) 9  
b) 0  
c) 10  
d) None of the mentioned  
View Answer

Answer: c  
Explanation: The code above creates 10 closures, and stores them in an array. The closures are all defined within the same invocation of the function, so they share access to the variable i. When constfuncs() returns, the value of the variable i is 10, and all 10 closures share this value. Therefore, all the functions in the returned array of functions return the same value.