**1. The type that specifies what kind of event occured is**

a. event type

b. even target

c. Both a and b

d. None of the mentioned

**2. Which is the object on which the event occured or with which the event is associated?**

a. event type

b. event target

c. Both a and b

d. None of the mentioned

**3. In general, event handler is nothing but**

a. function

b. interface

c. event

d. handler

**4. When will the browser invoke the handler?**

a. Program begins

b. Any event occurs

c. Specified event occurs

d. None of the mentioned

**5. Which property specifies the property of the event?**

a. Type

b. Target

c. Manner

d. All of the mentioned

**6. The process by which the browser decides which objects to trigger event handlers on is**

a. Event Triggering

b. Event Listening

c. Event Handling

d. Event propogation

**7. Which form of event propogation handles the registered container elements?**

a. Event Propogation

b. Event Registration

c. Event Capturing

d. Default Actions

**8. The events that are directly tied to a specific input device are**

a. Device-independent input events

b. Device-dependent input events

c. User interface events

d. State change events

**9. The high-level events among the following events are**

a. User interface events

b. Device-independent events

c. Device-dependent events

d. All of the mentioned

**10. The events that are not directly tied to a specific input device are**

a. User interface events

b. Device-independent events

c. Device-dependent events

d. All of the mentioned

**11. 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”

**12. 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

**13. 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”

**14. 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]);

**15. 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

**16. 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

**17. 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

**18. Consider the following code snippet**

1625492465566293a6b8e1d\_000004

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

**19. 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

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

a. debugger

b. with

c. if

d. use strict

**21. 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

**22. 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

**23. 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 above

**24. 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

**25. What will happen if reverse() and join() methods are used simultaneously ?**

a. Reverses and stores in the same array

b. Reverses and concatenates the elements of the array

c. Reverses

d. All of the mentioned

**26. Consider the following code snippet :**

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]

**27. Consider the following code snippet :**

1283055714566435e0f27a4\_000004

The final output for the shift() is

a. 1

b. [4,5] c. [3,4,5] d. Exception is thrown

**28. The primary purpose of the array map() function is that it**

a. maps the elements of another array into itself

b. passes each element of the array and returns the necessary mapped elements

c. passes each element of the array on which it is invoked to the function you specify, and returns an array containing the values returned by that function.

d. None of the mentioned

**29. The reduce and reduceRight methods follow a common operation called**

a. filter and fold

b. inject and fold

c. finger and fold

d. fold

**30. The method or operator used to identify the array is**

a) isarrayType()

b) ==

c) ===

d) typeof

**31. The events that are not triggered directly by user activity are called**

a. Device-independent input events

b. Device-dependent input events

c. User interface events

d. State change events

**32. The video and the audio belong to the**

a. Timers and error handlers

b. API-Specific events

c. State change events

d. User interface events

**33. The client-side JavaScript’s asynchronous programming model contains**

a. Timers and error handlers

b. User interface events

c. State change events

d. API-specific events

**34. Which are the events that have default actions that can be canceled by event handlers?**

a. Submit and form-related events

b. Reset and form-related events

c. Submit and reset events

d. None of the mentioned

**35. The events that represents occurrences related to the browser window are**

a. Window

b. Element

c. Display

d. Handlers

**36. Which event is fired when a document and all of its external resources are fully loaded and displayed to the user?**

a. Window

b. Load

c. Element

d. Handler

**37. Which is the alternative to the load event?**

a. readychange

b. changestate

c. readystatechange

d. contentloader

**38. Which is the opposite to the load event in JavaScript?**

a. dontload

b. postload

c. preload

d. unload

**39. Which is the property that is triggered in response to JavaScript errors?**

a. onexception

b. onmessage

c. onerror

d. None of the mentioned

**40. Which event can be fired on any scrollable document element?**

a. Window

b. Scroll

c. Load

d. Unload

**41. 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

**42. 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 above

**43. 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

**44. Consider the following code snippet :**

var string2Num=parseInt("123xyz");

advertisements

The result for the above code snippet would be :

a. 123

b. 123xyz

c. Exception

d. NaN

**45. 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);

}

**46. 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;

**47. 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();

**48. What is the difference between the two lines given below ?**

!!(obj1 && obj2);

(obj1 && obj2);

advertisements

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

**49. 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

**50. 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 above

**51. JavaScript Code can be called by using**

a. RMI

b. Triggering Event

c. Preprocessor

d. Function/Method

**52. The type of a variable that is volatile is**

a. Volatile variable

b. Mutable variable

c. Immutable variable

d. Dynamic variable

**53. A hexadecimal literal begins with**

a. 00

b. 0x

c. 0X

d. Both b and c

**54. The generalised syntax for a real number representation is**

a. [digits][.digits][(E|e)[(+|-)]digits]

b. [digits][+digits][(E|e)[(+|-)]digits]

c. [digits][(E|e)[(+|-)]digits]

d. [.digits][digits][(E|e)[(+|-)]digits]

**55. When there is an indefinite or an infinity value during an arithmetic value computation, javascript**

a. Prints an exception error

b. Prints an overflow error

c. Displays “Infinity”

d. Prints the value as such

**56. Which of the following is not considered as an error in JavaScript?**

a. Syntax error

b. Missing of semicolons

c. Division by zero

d. All of the above

**57. The escape sequence ‘\f’ stands for**

a. Floating numbers

b. Representation of functions that returns a value

c. \f is not present in JavaScript

d. Form feed

**58. The snippet that has to be used to check if “a” is not equal to “null” is**

a. if(a!=null)

b. if (!a)

c. if(a!null)

d. if(a!==null)

**59. The statement a===b refers to**

a. Both a and b are equal in value, type and reference address

b. Both a and b are equal in value

c. Both a and b are equal in value and type

d. There is no such statement

**60. Assume that we have to convert “false” that is a non-string to string. The command that we use is** (without invoking the “new” operator)

a. false.toString()

b. String(false)

c. String newvariable=”false”

d. Both a and b