1000-javascript-questions-answers

1. The web development environment (JavaScript) offers which standard construct for data validation of the input entered by the user.
a) Controlled loop constructs
b) Server page access
c) Client side Event
d) Permit server-side

Answer: a
Explanation: JavaScript provides with for, while loops and if, else, switch cases for checking the information entered by the user. Additionally, all development environments provide syntax to create and use memory variables, constants, and functions.

2. The main purpose of a "Live Wire" in NetScape is to ________
a) Create linkage between client side and server side
b) Permit server side, JavaScript code, to connect to RDBMS
c) Support only non relational database
d) To interpret JavaScript code

Answer: b
Explanation: A Live Wire database driver also supports a number of non-relational databases.

3. The script tag must be placed in __________
a) the head tag
b) the head or body
c) the title or head
d) after the body tag

Answer: b
Explanation: If the script tag is placed after the body tag, then, it will not be evaluated at all. Also, it is always recommended and effective to use the script snippet in the <head> tag.

4. A JavaScript program developed on a Unix Machine ________
a) will throw errors and exceptions
b) must be restricted to a Unix Machine only
c) will work perfectly well on a Windows Machine
d) will be displayed as a JavaScript text on the browser

Answer: c
Explanation: JavaScript can be executed on different operating systems therefore the program developed on UNIX will work perfectly fine on windows also.

5. JavaScript is ideal to ________
a) make computations in HTML simpler
b) minimize storage requirements on the web server
c) increase the download time for the client
d) increase the loading time of the website

Answer: b
Explanation: JavaScript helps in performing various tasks with minimum storage requirements. Therefore to minimize storage requirements, JavaScript is always a better say.

6. Which attribute is used to specify that the script is executed when the page has finished parsing? (only for external scripts)
a) parse
b) a sync

c) defer
d) type

Answer: c
Explanation: The defer attribute is a Boolean attribute. When present, it specifies that the script is executed when the page has finished parsing.

7. JavaScript Code can be called by using ____________
a) RMI
b) Triggering Event
c) Preprocessor
d) Function/Method

Answer: d
Explanation: JavaScript code can be called by making a function call to the element on which JavaScript has to be run. There are many other methods like onclick, onload, and onsubmit etc.

8. JavaScript can be written __________
a) directly into JS file and included into HTML
b) directly on the server page
c) directly into HTML pages
d) directly into the css file

Answer: a
Explanation: JavaScript files can be saved by .JS extension and can be included in the HTML files. Script tag along with src attribute is used to include the js files.

9. Which of the following Attribute is used to include External JS code inside your HTML Document?
a) src
b) ext
c) script
d) link

Answer: a
Explanation: Script "tag" is used to include the JavaScript code. To include external JavaScript files "src" attribute is used inside the script tag.

10. A proper scripting language is a __________
a) High level programming language
b) Assembly level programming language
c) Machine level programming language
d) Low level programming language

Answer: a
Explanation: JavaScript is a high-level programming language that is interpreted by another program at runtime rather than compiled by the computer's processor. Scripting languages, which can be embedded within HTML, commonly are used to add functionality to a Web page, such as different menu styles or graphics displays or to serve dynamic advertisements.

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

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

a) good day
b) goodday
c) error

d) undefined

Answer: b
Explanation: The + operator acts as a concatenation operator when used with string. The new string does not have any space between the two added string.

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

```
<p id="demo"></p>
<script>
var x = 5;
var y = 2;
var z = x % y;
document.getElementById("demo").innerHTML = z;
</script>
```

a) 0
b) 1
c) 2
d) 5

Answer: b
Explanation: The % operator returns the remainder between the two numbers. It is used many times with if condition to check whether the number is divisible or not.

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

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

a) 5
b) 10
c) 50
d) Error

Answer: c
*Explanation: The *= operator is a shorthand expression for multiplication of a particular number. It is a combination of two operators * and = .*

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

```
<p id="demo"></p>
<script>
txt1 = " one";
txt1 += "two";
document.getElementById("demo").innerHTML = txt1;
</script>
```

a) onetwo
b) one two
c) error
d) undefined

Answer: a
Explanation: The += operator acts in the same way as the concatenation operator in the string. There is no space added when two string are added together with += operator.

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

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = typeof "John"
</script>
```

a) integer
b) number
c) string
d) error

Answer: c
Explanation: The typeof operator returns the type of the argument passed to it. The typeof operator returns number for an integer and string for a character array.

1. JavaScript Code can be called by using ___________
a) RMI
b) Triggering Event
c) Preprocessor
d) Function/Method

Answer: d
Explanation: JavaScript code can be called by making a function call to the element on which JavaScript has to be run. There are many other methods like onclick, onload, and onsubmit etc.

2. The type of a variable that is volatile is _______________
a) Volatile variable
b) Mutable variable
c) Immutable variable
d) Dynamic variable

Answer: b
Explanation: The variables whose values can be changed are called mutable variable types. In JavaScript, only objects and arrays are mutable, not primitive values.

3. A hexadecimal literal begins with __________
a) 00
b) 0x
c) 0X
d) Both 0x and 0X

Answer: d
Explanation: Generally, X or x denotes hexadecimal values. So, any integer literal that begins with 0X or 0x denotes a hexadecimal number.

4. 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]

Answer: a
Explanation: Floating-point literals may also be represented using exponential notation: a real number followed by the letter e (or E), followed by an optional plus or minus sign, followed by an integer exponent. This notation represents the real number multiplied by 10 to the power of the exponent.

5. JavaScript _________ when there is an indefinite or an infinite value during an arithmetic computation.
a) Prints an exception error
b) Prints an overflow error
c) Displays "Infinity"

d) Prints the value as such

Answer: c
Explanation: When the result of a numeric operation is larger than the largest representable number (overflow), JavaScript prints the value as Infinity. Similarly, when a negative value becomes larger than the largest representable negative number, the result is negative infinity. The infinite values behave as you would expect: adding, subtracting, multiplying, or dividing them by anything results in an infinite value (possibly with the sign reversed).

6. Which of the following is not considered as an error in JavaScript?
a) Syntax error
b) Missing of semicolons
c) Division by zero
d) Missing of Bracket

Answer: c
Explanation: Division by zero is not an error in JavaScript: it simply returns infinity or negative infinity. There is one exception, however: zero divided by zero does not have a well defined value, and the result of this operation is the special not-a-number value, printed as NaN.

7. 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

Answer: d
Explanation: \f is the JavaScript escape sequence that stands for Form feed (\u000C). It skips to the start of the next page. (Applies mostly to terminals where the output device is a printer rather than a VDU).

8. 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)

Answer: d
Explanation: A strict comparison (e.g., ===) is only true if the operands are of the same type and the contents match. The more commonly-used abstract comparison (e.g. ==) converts the operands to the same type before making the comparison. The not-equal operator !== compares 0 to null and evaluates to either true or false.

9. 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

Answer: c
Explanation: "===" operator is known as the strict comparison operator. A strict comparison (===) is only true if the operands are of the same type and the contents match.

10. 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 false.toString() and String(false)

Answer: d

Explanation: The three approaches for converting to string are: value.toString(),"" + value and String(value). A non-string can be converted in two ways without using a new operator false.toString () and String(false).

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

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

a) true
b) false
c) runtime error
d) compilation error

Answer: a
Explanation: The == convert different type of operands to the same type before making the comparison. A strict comparison results into true value if the operands are of the same type and the contents match.

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

```
function equalto()
{
    int num=10;
    if(num===”10”)
        return true;
    else
        return false;
}
```

a) true
b) false
c) runtime error
d) compilation error

Answer: a
Explanation: A === operator is only true if the operands are of the same type and the contents match. Two strings are strictly equal when they have the same sequence of characters, same length, and same characters in corresponding positions.

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

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

a) true
b) false
c) runtime error
d) logical error

Answer: a

Explanation: A non string (integer) can be converted to string using .tostring() function. A strict comparison is only true if the operands are of the same type and the contents match. Hence the following code snippet would result into true output.

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

```
int a==2;
int b=4;
int ans=a+b;
print(ans);
```

a) 2
b) 6
c) 0
d) error

Answer: d
Explanation: The following code will generate into an error output as a comparator operator is used outside of if statement. A single equalto ('=') operator is used for initialization.

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

```
int a=1;
if(a!=null)
    return 1;
else
    return 0;
```

a) 1
b) 0
c) runtime error
d) compiler error

Answer: a
Explanation: != is the not equal to operator. It gives a value of 1 if the two values which are compared are not equal and give 0 if the two values are equal.

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

```
var string1 = "123";
var intvalue = 123;
alert( string1 + intvalue );
```

a) 123246
b) 246
c) 123123
d) Exception

Answer: c
Explanation: In JavaScript the alert function does the type casting and converts the integer value to string. After that it concatenates both the strings and shows the result as a concatenated string. Thus 123123 would be correct.

2. A function definition expression can be called as __________
a) Function prototype
b) Function literal
c) Function calling
d) Function declaration

Answer: b
Explanation: A function definition expression is a "function literal" in the same way that an object initializer is an "object literal." A Function definition expression typically consists of the keyword function followed by a comma-

separated list of zero or more identifiers (the parameter names) in parentheses and a block of JavaScript code (the function body) in curly braces.

3. The property of a primary expression is ____________
a) stand-alone expressions
b) basic expressions containing all necessary functions
c) contains variable references alone
d) contains only keywords

Answer: a
Explanation: The simplest expressions, known as primary expressions, are those that stand alone — they do not include any simpler expressions. Primary expressions in JavaScript are constant or literal values, certain language keywords, and variable references.

4. Consider the following JavaScript statements.

```
var text = "testing: 1, 2, 3"; // Sample text
var pattern = /\d+/g // Matches all instances of one or more digits
```

In order to check if the pattern matches with the string "text", the statement is ____________
a) text==pattern
b) text.equals(pattern)
c) text.test(pattern)
d) pattern.test(text)

Answer: d
Explanation: The given pattern is applied on the text given in the parenthesis.

5. The expression of calling (or executing) a function or method in JavaScript is called ________
a) Primary expression
b) Functional expression
c) Invocation expression
d) Property Access Expression

Answer: c
Explanation: An invocation expression is JavaScript's syntax for calling (or executing) a function or method). It starts with a function expression that identifies the function to be called.

6. What kind of expression is "new Point(2,3)"?
a) Primary Expression
b) Object Creation Expression
c) Invocation Expression
d) Constructor Calling Expression

Answer: b
Explanation: An object creation expression creates a new object and invokes a function (called a constructor) to initialize the properties of that object. Object creation expressions are like invocation expressions except that they are prefixed with the keyword new.

7. Which of the operator is used to test if a particular property exists or not?
a) in
b) exist
c) within
d) exists

Answer: a
Explanation: The operator "in" tests whether a particular property exists or not. In operator is usually added in looping statements to traverse the array or the object.

8. Among the following, which one is a ternary operator?
a) +
b) :
c) –
d) ?:

Answer: d
Explanation: JavaScript supports one ternary operator, the conditional operator ?:, which combines three expressions into a single expression. If else case can be replaced by the conditional operator

9. "An expression that can legally appear on the left side of an assignment expression." is a well known explanation for variables, properties of objects, and elements of arrays. They are called ___________
a) Properties
b) Prototypes
c) Lvalue
d) Definition

Answer: c
Explanation: lvalue is a historical term that means "an expression that can legally appear on the left side of an assignment expression." In JavaScript, variables, properties of objects, and elements of arrays are lvalues.

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

```
x = ~-y;
w = x = y = z;
q = a?b:c?d:e?f:g;
```

a)

```
x = ~(-y); w = (x = (y = z));
q = a?b:(c?d:(e?f:g));
```

b)

```
x = a?b:(c?d:(e?f:g));
q = ~(-y); w = (x = (y = z));
```

c)

```
x = (x = (y = z));w = ~(-y);
q = a?b:(c?d:(e?f:g));
```

d)

```
x = ~(-y); w = (x = (y = z));
q = (c?d:(e?f:g));
```

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

```
function output(option)
{
        return (option ?  "yes"  :  "no");
}
        bool ans=true;
console.log(output(ans));
```

a) Yes
b) No
c) Runtime error
d) Compilation error

Answer: a

Explanation: "?" is called the ternary operator which is used for choosing one choice from the given two choices. It is used instead of if else statement and makes the code shorter.

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

```
var  obj=
{
        length:20,
        height:35,
}
if ('breadth' in obj === false)
{
        obj.breadth = 12;
}

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

a) 20
b) 12
c) undefined
d) error

Answer: b
Explanation: The "in" operator checks for the presence of a particular property in object. It returns true if the property is present and returns false if the property is not present.

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

```
function height()
{
    var  height = 123.56;
    var type = (height>=190) ? "tall" : "short";
    return type;
}
```

a) 123.56
b) 190
c) tall
d) short

Answer: d
Explanation: The ternery operator is used as a comparison operator which works on three operands. The statement in the above code initializes type variable with the value short which is returned through the function.

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

```
string  a = "hi";
string  b ="there";
alert(a+b);
```

a) hi
b) there
c) hithere
d) undefined

Answer: c
Explanation: alert function is used to print the value passed as argument in a dialog box in a browser. The alert function adds both the string and prints the result as a combined string.

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

```
function output(object)
{
        var place=object ? object.place : “Italy”;
        return “clean:”+ place;
}
console.log(output({place:India}));
```

a) clean:India
b) clean:Italy
c) error
d) undefined

Answer: a
Explanation: ”?” operator is used to compare the values and place is initialized according to the true condition that whether it is true or false. The function is called in the console.log and the object value is passed.

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

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

a) 7.25
b) -7.25
c) 7
d) -7

Answer: a
Explanation: The abs() method returns the absolute value of a number. The method is find in the math library of Javascript.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
    document.getElementById("demo").innerHTML = Math.cbrt(125);
}
</script>
```

a) 125
b) 25
c) 5
d) Error

Answer: c
Explanation: cbrt return the cubic root of a number. The method is find in the math library of Javascript.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
    document.getElementById("demo").innerHTML = Math.acos(0.5);
}
</script>
```

a) 1.01
b) 1.047

c) 1.00
d) 1.4

Answer: b
Explanation: The acos() method returns the arccosine of a number as a value value between 0 and PI radians. If the parameter x is outside the range -1 to 1, the method will return NaN.

1. JavaScript is a _______________ language.
a) Object-Oriented
b) High-level
c) Assembly-language
d) Object-Based

Answer: d
Explanation: JavaScript is not a full-blown OOP (Object-Oriented Programming) language, such as Java or PHP, but it is an object-based language. The criteria for object orientation are the classic threesome of polymorphism, encapsulation and inheritance and JavaScript doesn't pass this.

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

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

a) 10
b) Error
c) 1
d) 5

Answer: c
Explanation: Firstly the interpreter checks obj for property b. But it doesn't find any property b so it takes the value from outside the object within the code snippet.

3. A conditional expression is also called a _______________
a) Alternative to if-else
b) Immediate if
c) If-then-else statement
d) Switch statement

Answer: b
Explanation: A conditional expression can evaluate to either True or False based on the evaluation of the condition. If the condition is true then the value following the operator is taken that's why it is called immediate if.

Code 2 :

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

a) Code 1
b) Code 2
c) Both Code 1 and Code 2
d) Cannot Compare

Answer: a
Explanation: Code 1 would be more efficient. Infact second code will go into runtime error as the value of num will

never reach less than or equal to one.

5. What is a block statement in JavaScript?
a) conditional block
b) block that contains a single statement
c) both conditional block and a single statement
d) block that combines multiple statements into a single compound statement

Answer: d
Explanation: A block statement groups zero or more statements. In languages other than JavaScript, it is known as a compound statement. A statement block is a block that combines more than one statements into a single compound statement for ease.

6. When an empty statement is encountered, a JavaScript interpreter __________
a) Ignores the statement
b) Prompts to complete the statement
c) Throws an error
d) Shows a warning

Answer: a
Explanation: The JavaScript interpreter takes no action when it executes an empty statement. The empty statement is occasionally useful when you want to create a loop that has an empty body.

7. The "var" and "function" are __________
a) Keywords
b) Declaration statements
c) Data types
d) Prototypes

Answer: b
Explanation: The var and function are declaration statements—they declare or define variables and functions. These statements define identifiers (variable and function names) that can be used elsewhere in your program and assign values to those identifiers.

8. In the following switch syntax, the expression is compared with the case labels using which of the following operator(s)?

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

a) ==
b) equals
c) equal
d) ===

Answer: d
Explanation: A strict comparison (e.g., ===) is only true if the operands are of the same type and the contents match. When a switch executes, it computes the value of the expression and then looks for a case label whose expression evaluates to the same value (where sameness is determined by the === operator).

9. What happens in the following javaScript code snippet?

```
switch(expression)
{
    statements
}
```

a) The values of count are logged or stored in a particular location or storage
b) The value of count from 0 to 9 is displayed in the console
c) An error is displayed
d) An exception is thrown

Answer: b
Explanation: Console.log is a predefined function in JavaScript which takes the value as an argument of its function.console.log prints this value in the argument in the console at the time of execution of the code.

10. The enumeration order becomes implementation dependent and non-interoperable if ___________
a) If the object inherits enumerable properties
b) The object does not have the properties present in the integer array indices
c) The delete keyword is never used
d) Object.defineProperty() is not used

Answer: a
Explanation: In computer programming, an enumerated type (also called enumeration or enum) is a data type consisting of a set of named values called elements, members or enumerators of the type. The enumerator names are usually identifiers that behave as constants in the language.

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

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

a) 10
b) 0
c) 1
d) Undefined

Answer: c
Explanation: The if else statement is a part of the javascript conditioning statements. The line of code inside the "if" statement is executed if the value passed to "if" is 1.

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

```
Int a=1;
if(a>10)
{
    document.write(10);
}
else
{
    document.write(a);
}
```

a) 10
b) 9
c) 8
d) 0

Answer: b
Explanation: The above code contains a switch loop. It is used as an alternative to if else statement. One of the given condition is satisfied according to the input of the switch case and the output is decided accordingly.

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

```
var grade='B';
var result;
switch(grade)
{
    case 'A':
    {
        result+="10";
        break;
    }
    case 'B':
    {
        result+=" 9";
        break;
    }
    case 'C':
    {
        result+=" 8";
        break;
    }
    default:
    result+=" 0";
}
document.write(result);
```

a) 10
b) 27
c) 8
d) 0

Answer: b
Explanation: The above code does not have a break statement after the cases in the switch loop. Therefore all of the cases following "A" will get executed.

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

```
var grade='A';
var result;
switch(grade)
{
    case 'A':
        result+="10";
    case 'B':
        result+=" 9";
    case 'C':
        result+=" 8";
    default:
        result+=" 0";
}
document.write(result);
```

a) 4
b) 1
c) Error
d) 0

Answer: d
Explanation: For checking more than one condition the if else if loop is used. It is the extension of if else loop which is also sometimes known as if else ladder.

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

```
int a=4;
int b=1;
int c=0;
If(a==b)
    document.write(a);
```

```
else if(a==c)
    document.write(a);
else
    document.write(c);
```

a) 10
b) 0
c) 18
d) 17

Answer: b
Explanation: The switch case contains a default case along with the other cases. The default case gets executed when no other case results into true.

1. 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
         {
             console.log(a[i]);
         } while (++i < len);
     }
}
```

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 statement creates a loop that executes a block of code once, before checking if the condition is true, then it will repeat the loop as long as the condition is true. Hence the iterator traverses through the array and print them in normal 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. Firstly the loop initialiases the variable then test the condition and then after executing the statement increments its value.

3. What will the following JavaScript code snippet work? If not, what will be the error?

```
function tail(o)
{
    for (; o.next; o = o.next) ;
    return o;
}
```

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. What will be the equivalent code of the following JavaScript code?

```
for(var p in o)
   console.log(o[p]);
```

5. One of the special features 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) the iteration is finite when an interpreter is used

Answer: a
Explanation: Interpreter translates the source code into machine code line by line, and stops when it encounters an error. Before each iteration, the interpreter evaluates the variable expression and assigns the name of the property (a string value) to it.

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) The property will be enumerated

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) The interpreter throws an error

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. What will be the role of the continue keyword in the following JavaScript code snippet?

```
while (a != 0)
{
   if (a == 1)
       continue;
   else
       a++;
}
```

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) The continue keyword breaks out of the loop

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

place encountered. While the break statement breaks out of the loop.

9. What could be the task of the statement debugger in the following JavaScript code?

```
function f(o)
{
     if (o === undefined) 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) It is used to find error in the statement

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

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 has appeared.*

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

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

a) 5
b) 555
c) 3
d) error

Answer: b
Explanation: for loop first initializes the variable and later on checks for the condition expression and after that execute the line of statements. The value of iterator i increase until it reaches the value of length.

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

```
var a = 10;
do {
        a += 1;
        console.log(a);
} while (a < 5);
```

a) 11121314
b) 1112

c) 12345
d) 11

Answer: d
Explanation: dowhile loop first executes the statements and after that checks for the condition. Therefore dowhile loop will be executed and after that the condition will become false and the loop will terminate.

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

```
var a= 0;
var b = 0;
while (a < 3)
{
        a++;
        b += a;
        console.log(b);
}
```

a) 135
b) 123
c) 013
d) 01

Answer: a
Explanation: A while loops checks for the condition first before executing the looping statements. A while loop increments the value at the end of the loop whereas for executes the statement at the starting of the loop.

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

```
int size=5;
int a=5;
int size=4;
for(int j=size;j>=0;j--)
{
        console.log(a);
        a=a-2;
}
```

a) 5555
b) 5321
c) 531-1
d) 531

Answer: c
Explanation: The value of a will decrease by 2 at every iteration. The for loop will iterate four times till the value of j will decrease to 0.

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

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

a) 0
b) error
c) 4
d) 5

Answer: d
Explanation: The value of a will increase until it will become equal to 5 after that the cursor will come out the loop. There are no statements for the for loop therefore only the value of a will increase. Hence the output will be five.

1. The unordered collection of properties, each of which has a name and a value is called ________
a) String
b) Object
c) Serialized Object
d) Array

Answer: b
Explanation: Objects in JavaScript may be defined as an unordered collection of related data, of primitive or reference types, in the form of "key:value" pairs. Hence each of the property has a name and value.

2. The object has three object attributes namely ________
a) Class, parameters, object's extensible flag
b) Prototype, class, objects' parameters
c) Prototype, class, object's extensible flag
d) Native object, Classes and Interfaces and Object's extensible flag

Answer: c
Explanation: Every object has three associated object attributes:

1. *An object's prototype is a reference to another object from which properties are inherited.*
2. *An object's class is a string that categorizes the type of an object.*
3. *An object's extensible flag specifies whether new properties may be added to the object.*

3. What will be the firstname and surname of the following JavaScript code?

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

a) properties
b) property values
c) property names
d) objects

Answer: c
Explanation: The above code snippet contains an object inside another object. firstname and surname are the property names. The value of that particular property is itself an object.

4. A linkage of series of prototype objects is called as ________
a) prototype stack
b) prototype chain
c) prototype class
d) prototypes

Answer: b
Explanation: Consider an example, Date.prototype inherits properties from Object.prototype, so a Date object created by new Date() inherits properties from both Date.prototype and Object.prototype. This linked series of prototype objects is known as prototype chain.

5. In the following syntax, the data type within the square brackets must be ___________

```
book[datatype]=assignment_value;
```

a) An integer

b) A String
c) An object
d) Floating point

Answer: b
Explanation: The value inside the square bracket is used to access that data element or the property of the object. When using square bracket notation, the expression inside the square brackets must evaluate to a string or a value that can be converted to a string.

6. To determine whether one object is the prototype of (or is part of the prototype chain of) another object, one should use the ____________
a) isPrototypeOf() method
b) equals() method
c) === operator
d) ==opertor

Answer: a
Explanation: Prototype is a global property which is available with almost all the objects. To determine whether one object is the prototype of (or is part of the prototype chain of) another object, one should use the isPrototype() method. To find out if p is the prototype of o write p.isPrototypeOf(o).

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

```
function f() {};
```

a) Function f
b) A custom constructor
c) Prototype of a function
d) Not valid

Answer: b
Explanation: All object instances have a constructor property that point to the constructor function that created it. A custom constructor is a constructor which requires no arguments and is created automatically by the compiler at the time of object creation if not created by the user.

8. The purpose of *extensible* attribute is to __________
a) make all of the own properties of that object non configurable
b) to configure and bring a writable property
c) "lock down" objects into a known state and prevent outside tampering
d) to include new properties into the object

Answer: c
Explanation: Extensible attributes determines whether the object can have new properties or not. Therefore extensible attribute will allow an object to include and write properties into them.

9. Identify the process done in the following JavaScript code snippet?

```
o = {x:1, y:{z:[false,null,""]}};
s = JSON.stringify(o);
p = JSON.parse(s);
```

a) Object Encapsulation
b) Object Serialization
c) Object Abstraction
d) Object Encoding

Answer: b
*Explanation: Object serialization is the process of converting an object's state to a string from which it can later be restored. The **JSON.parse()** method parses a JSON string, constructing the JavaScript value or object described by the*

string.

10. The basic purpose of the toLocaleString() is to _________
a) return a localised object representation
b) return a parsed string
c) return a local time in the string format
d) return a localized string representation of the object

Answer: d
Explanation: .toLocaleString is a predefined function in javascript which is used to return a localized string representation of the object. For example the ***date.toLocaleString()*** *is an inbuilt function in JavaScript which is used to convert a date and time to a string.*

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

```
const object1 = {};
a = Symbol('a');
b = Symbol.for('b');
object1[a] = 'harry';
object1[b] = 'derry';
const objectSymbols = Object.getOwnPropertySymbols(object1);
console.log(objectSymbols.length);
```

a) 0
b) 2
c) 1
d) Error

Answer: b
Explanation: The Object.getOwnPropertySymbols() method returns an array of all symbol properties found directly on an object. This method returns an empty array unless you have set symbol properties on your object.

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

```
const obj1 =
{
    property1: 21
}
const descriptor1 = Object.getOwnPropertyDescriptor(obj1, 'property1');
console.log(descriptor1.configurable);
console.log(descriptor1.enumerable);
```

a) true 21
b) true false
c) true true
d) false false

Answer: c
Explanation: The Object.getOwnPropertyDescriptor method allows to query the full information about a property. The method returns a property descriptor for an own property that is one directly present on an object and not in the object's prototype of a given object.

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

```
const obj1 = { property1: '10'};
const obj2 = Object.freeze(obj1);
obj2.property1 = '20';
console.log(obj2.property1);
```

a) 10
b) 20
c) Runtime error

d) Compilation error

Answer: a
Explanation: The Object.freeze() method "freezes" the properties of object and prevents new properties from being added to it. This method prevents the modification of existing property, attributes, and values.

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

```
const object1 = {
  property1: 20
};
console.log(Object.is(object1));
```

a) 20
b) true
c) false
d) error

Answer: c
Explanation: The Object.is() method of JavaScript is used to determine whether two values are the same value. There is a special built-in method that compares values. This method returns a Boolean indicating whether or not the two arguments are the same value.

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

```
const obj = {prop: 12};
Object.preventExtensions(obj);
console.log( Object.isExtensible(obj));
```

a) 12
b) false
c) true
d) error

Answer: b
Explanation: The Object.preventExtensions() only prevents the addition of new properties from ever being added to an object. This change is a permanent that means once an object has been made non-extensible, it cannot be made extensible again.

1. What is the observation made in the following JavaScript code?

```
var count = [1,,3];
```

a) The omitted value takes "undefined"
b) This results in an error
c) This results in an exception
d) The omitted value takes an integer value

Answer: a
Explanation: Array is defined with a null value when no value is mentioned. If you omit a value from an array literal, the omitted element is given an undefined value.

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

```
var a1 = [,,,];
var a2 = new Array(3);
0 in a1
0 in a2
```

a) true false
b) false true

c) true true
d) false true

Answer: a
Explanation: Array a1 is defined with null values. Therefore we can access the indexes 0, 1 and 2. But array a2 is only defined not declared. Therefore we cannot access 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) updates the element

Answer: a
Explanation: pop() function pops out that is delete the last element from the array. Hence pop() method (it works with push()) reduces the length of an array by 1.

4. What is the observation made in the following JavaScript code?

```
if (!a[i]) continue;
```

a) Skips the defined elements
b) Skips the existent elements
c) Skips the null elements
d) Skips the defined & existent elements

Answer: c
Explanation: The if loop in the above code checks whether the value of a[i] exists or not. For undefined, non existent and null values the if loop returns true.

5. 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) Stores the elements of an array in normal order

Answer: a
Explanation: The array.join() method is an inbuilt function in JavaScript which is used to join the elements of an array into a string. The reverse() followed by a join() will reverse the respective array and will store the reversed array in the memory.

6. What will be the possible output of the following JavaScript code?

```
var a = [1,2,3,4,5];
a.slice(0,3);
```

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: .slice() function is a predefined function in JavaScript used to keep the elements from starting point and ending point mentioned in the function argument. The elements after the ending point and before the starting point are not shown.

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

```
var a = [];
a.unshift(1);
```

```
a.unshift(22);
a.shift();
a.unshift(3,[4,5]);
a.shift();
a.shift();
a.shift();
```

a) 1
b) [4,5]
c) [3,4,5]
d) Exception is thrown

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.

8. 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) pass the elements of the array into another array

Answer: c
Explanation: map() is a predefined function in javascript used for mapping the array elements to be used for some other purpose. The map() method 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.

9. The reduce and reduceRight methods follow a common operation called __________
a) filter and fold
b) inject and fold
c) finger and fold
d) fold

Answer: b
Explanation: The reduceRight() method reduces the array to a single value. The reduceRight() method executes a provided function for each value of the array (from right-to-left). The return value of the function is stored in an accumulator (result/total). Hence it does the operation of injecting and folding.

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

Answer: d
Explanation: The typeof operator is used to get the data type (returns a string) of its operand. The operand can be either a literal or a data structure such as a variable, a function, or an object. The operator returns the data type.

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

```
var val1=[1,2,3];
var val2=[6,7,8];
var result=val1.concat(val2);
document.writeln(result);
```

a) 1, 2, 3

b) Error
c) 1, 2, 3, 6, 7, 8
d) 123

Answer: c
Explanation: concat is a predefined function in the array library in Javascript. The concat function is used to combine the value of two arrays.
i.e.1, 2, 3, 6, 7, 8

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

```
var values=[1,2,3,4]
var ans=values.slice(1);
document.writeln(ans);
```

a) 1, 2, 3, 4
b) 2, 3, 4
c) 1, 3, 4
d) error

Answer: d
Explanation: slice function is used to remove values from the array. The slice function accepts two arguments as the starting and ending values for slicing elements. As in the above function, only one argument is passed therefore it will result in an error.

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

```
int sum=0;

var arr = [10,15,20,30];

arr.forEach(function myFunction(element)
{
        sum= sum+element;
});
document.writeln(sum);
```

a) 70
b) 75
c) 10
d) error

Answer: b
Explanation: forEach is a predefined function in Javascript which used to traverse through the array. It works in a similar way to for loop and iterates through each value in array.

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

```
var values=["one","two","Three"];
var ans=values.shift();
document.writeln(ans);
```

a) one
b) two
c) three
d) error

Answer: a
Explanation: shift is a predefined function in array library and works like a pop function. It pops the first value of the array and returns its value. Therefore the answer will be one.

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

```
var arr=[1,2,3];
var rev=arr.reverse();
document.writeln(rev);
```

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

Answer: b
Explanation: reverse function is a predefined function in array library in Javascript. The function is used to reverse the element of the array.
i.e. 3, 2, 1.

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

Answer: c
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. What will be the output of the following JavaScript code?

```
function printprops(o)
{
    for(var p in o)
      console.log(p + ": " + o[p] + "\n");
}
```

a) Prints the contents of each property of o
b) Returns undefined
c) Prints only one property
d) Prints the address of elements

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) when the function is called

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

Answer: d
Explanation: The ***return*** *stops the execution of the function when it is encountered within the function. It returns the*

value to the statement where the function is called.

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) It will throw an error

Answer: c
Explanation: A function without a return statement will return a default value. If the return statement does not have an associated expression then it returns an undefined value.

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

Answer: a
Explanation: Void functions does not return a value. Functions with no return value are sometimes called ***procedures****.*

7. The function stops its execution when it encounters?
a) continue statement
b) break statement
c) goto statement
d) return statement

Answer: d
Explanation: Continue statement and break statement are used in the loops for skipping the iteration or going out of the loop. Whenever a return statement is encountered the function execution is stopped.

8. Which keyword is used to define the function in javascript?
a) void
b) int
c) function
d) main

Answer: c
Explanation: A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses(). Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

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);

Answer: d
Explanation: The two argument in a function are separated by a comma (,). The code above is an invocation expression: it includes a function expression o.m and two argument expressions, ***x*** *and* ***y****.*

10. What will be the equivalent code of the following JavaScript code?

```
o.m(x,y);
```

a) o.m(x) && o.m(y);
b) o[“m”](x,y);
c) o(m)[“x”,”y”];

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

Answer: b
Explanation: Another way to write o.m(x,y) is o["m"](x,y).o["m"] will access the property of the object and parenthesis will access the function present inside it.

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

```
function info()
{
    int a=1;
    int b=2;
    return a*b;
}
document.write(info());
```

a) 1
b) 2
c) 3
d) error

Answer: b
Explanation: document.write() is used to write the output on the console. In the above code document.write() passes a function as its argument and the function returns 2 which is printed as output.

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

```
var arr = [7, 5, 9, 1];
var value = Math.max.apply(null, arr);
document.writeln(value);
```

a) 7
b) 5
c) 1
d) 9

Answer: d
Explanation: apply() is a predefined function method in javascript. The argument of apply() method contains elements of an array. It contains two values as argument which are optional as input.

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

```
var person =
{
      name: "Rahul",
      getName: function()
      {
          nreturn this.name;
      }
}
var unboundName = person.getName;
var boundName = unboundName.bind(person);
document.writeln(boundName());
```

a) runtime error;
b) compilation error
c) Rahul
d) undefined

Answer: c
Explanation: The javascript bind() function is used to create a new function. The newly created function has its own set of keywords.

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

```
function code(id,name)
{
                this.id = id;
                this.name = name;
}
function pcode(id,name)
{
                code.call(this,id,name);
}
document.writeln(new pcode(101,"vivek").id);
```

a) vivek
b) 101
c) Runtime error
d) Compilation error

Answer: b
Explanation: The JavaScript call() method is used to call a function containing "this" value as an argument. It returns the result of calling function.

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

```
     var pow=new Function("num1","num2","return Math.pow(num1,num2)");
     document.writeln(pow(2,3));
```

a) 2
b) 3
c) 8
d) error

Answer: c
Explanation: pow function is a predefined function which is present in the math library of javascript. The pow function accepts two arguments of which power of the first argument is calculated with respect to the second.

1. What will be the output of the following JavaScript statement?

```
var grand_Total=eval("10*10+5");
```

a) 10*10+5
b) 105 as a string
c) 105 as an integer value
d) Exception is thrown

Answer: c
Explanation: eval() is a function property of the global object. The argument of the eval() function is a string. If the string represents an expression, eval() evaluates the expression. If the argument represents one or more JavaScript statements, eval() evaluates the statements.

2. Do functions in JavaScript necessarily return a value?
a) It is mandatory
b) Not necessary
c) Few functions return values by default
d) some functions do not return any value

Answer: c
Explanation: Functions generally have a return statement and hence usually returns a value. Some functions which does not have a return statement returns value by default during execution.

3. Will the following JavaScript code work?

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

a) Yes, perfectly
b) Error
c) Exception will be thrown
d) Memory leak

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

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

```
var string2Num=parseInt("123xyz");
```

a) 123
b) 123xyz
c) Exception
d) NaN

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

b)

```
   function concatenate()
   {
        return String.prototype.concat('', arguments);
   }
```

c)

```
   function concatenate()
   {
        return String.prototype.apply('', arguments);
   }
```

d)

```
   function concatenate()
   {
       return String.concat.apply('', arguments);
   }
```

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;

Answer: b
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. What will be the equivalent statement of the following JavaScript statement?

```
   function concatenate()
   {
       return String.prototype.concat.apply('', arguments);
   }
```

a) var o = Object();
b) var o;
c) var o= new Object;
d) Object o=new Object();

Answer: c
Explanation: As a special case, for the new operator only, JavaScript simplifies the grammar by allowing the parenthesis to be omitted if there are no arguments in the function call. Hence you can always omit a pair of empty parentheses in a constructor invocation.

8. What is the difference between the two lines in the following JavaScript code?

```
var o = new Object();
```

a) Both the lines result in a boolean value "True"
b) Both the lines result in a boolean value "False"
c) Both the lines check 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

Answer: d
Explanation: The first returns a "real" boolean value, because you first negate what is inside the parenthesis, but then immediately negate it again. So, it's like saying something is "not not" truth-y, making it true. The second example simply checks for the existence of the obj1 and obj2, but might not necessarily return a "real" boolean value, instead returning something that is either truth-y or false-y. This can be problematic, because false-y can be the number 0, or an empty string, etc. Simple existence can be truth-y. A "real" boolean will only be true or false.

9. What will be the state stored in d in the following JavaScript code?

```
!!(obj1 && obj2);
(obj1 && obj2);
```

a) 1
b) 0
c) Null
d) Undefined

Answer: a
Explanation: Counter function increments the value of the variable by 1 and reset function sets the value of the variable back to 0.as d is incremented twice and reset function is not called on d therefore the value of d is 2.

10. What will be the last statement return in the following JavaScript code?

```
var c = counter(), d = counter();
c.count()
d.count()
c.reset()
c.count()
d.count()
```

a) 9
b) 0
c) 10
d) 12

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.

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

```
function constfuncs()
{
    var funcs = [];
    for(var i = 0; i < 10; i++)
        funcs[i] = function() { return i; };
    return funcs;
}
var funcs = constfuncs();
funcs[5]()
```

a) 7
b) 5
c) 1
d) 9

Answer: c
Explanation: The function apply() is used to call a function that contains "this" value and the argument contains elements of an array. Unlike call() method, it contains a single array of arguments.

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

```
var arr = [7, 5, 9, 1];
var min = Math.min.apply(null, arr);
document.writeln(min);
```

a) 2
b) 5
c) Error
d) 7

Answer: d
Explanation: add function is defined in the first line of the code using the new property. Add function returns the sum of the two arguments passed to it.

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

```
var add=new Function("num1","num2","return num1+num2");
document.writeln(add(2,5));
```

a) 6
b) 7.4
c) 7.5
d) 8

Answer: d
Explanation: The ciel function in javascript round offs the value passed in its argument to the next higher closest value. Therefore the value of a will be converted to 4 and hence the output will be 8.

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

```
var a=3.7;
var b=2;
a=ciel(a)
document.writeIn(a*b);
```

a) 9
b) 8.31
c) Error
d) 4

Answer: d
Explanation: The floor method round offs the value of the decimal number to the lower limit. Therefore floor of a will be 2 which will result in the output of 4.

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

```
var a=2.99;
var ans=floor(a)*floor(a)
console.log(ans);
```

a) 225
b) 15
c) Error
d) Undefined

Answer: b
Explanation: The JavaScript math sqrt() method returns the square root of a number. If the provided number is negative, it returns NaN.

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

```
var a=225;
document.writeln(Math.sqrt(a));
```

a) 0.80
b) 0.88
c) 0.50
d) 0.78

Answer: b
Explanation: The asinh() method returns the hyperbolic arcsine of a number. The method is find in the math library of Javascript.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
    document.getElementById("demo").innerHTML = Math.asinh(1);
}
</script>
```

a) 1.00
b) 1.01
c) 1.05
d) 1.10

Answer: d
Explanation: The atan2() method returns the arctangent of the quotient of its arguments, as a numeric value between PI and -PI radians. The number returned represents the counterclockwise angle in radians (not degrees) between the positive X axis and the point (x, y).

1. What kind of scoping does JavaScript use?
a) Literal
b) Lexical
c) Segmental
d) Sequential

Answer: b
*Explanation: Like most modern programming languages, JavaScript uses **lexical scoping**. This means that functions are*

executed using the variable scope that was in effect when they were defined, not the variable scope that is in effect when they are invoked.

2. What must be done in order to implement Lexical Scoping?
a) Get the object
b) Dereference the current scope chain
c) Reference the current scope chain
d) Return the value

Answer: c
*Explanation: In order to implement lexical scoping, the internal state of a JavaScript function object must include not only the code of the function but also **a reference to the current scope chain**.*

3. What is closure?
a) Function objects
b) Scope where function's variables are resolved
c) Both Function objects and Scope where function's variables are resolved
d) Function return value

Answer: c
Explanation: A combination of a function object and a scope (a set of variable bindings) in which the function's variables are resolved is called a closure.

4. Which of the following is not an example of closures?
a) Objects
b) Variables
c) Functions
d) Graphics

Answer: d
Explanation: In JavaScript, closures are created every time a function is created, at function creation time. Technically, all JavaScript functions are closures: they are objects, and they have a scope chain associated with them.

5. Which of the following uses a lot of CPU cycles?
a) GUI
b) Statically generated graphics
c) Dynamically generated graphics
d) Generic scoping

Answer: c
Explanation: Dynamic graphics for data, means simulating motion or movement using the computer. It may also be thought of as multiple plots linked by time. Hence dynamically generating graphics from real-time data uses a lot of CPU cycles.

6. What will be the function of the following JavaScript code?

```
var scope = "global scope";
function checkscope()
{
    var scope = "local scope";
    function f()
    {
         return scope;
    }
    return f;
}
```

a) Returns value null
b) Returns exception
c) Returns the value in scope

d) Shows an error message

Answer: c
Explanation: In JavaScript, every running function, code block, and the script as a whole have an associated object known as the Lexical Environment. The above code snippet returns the value in scope.

7. What is the fundamental rule of lexical scoping?
a) Functions are declared in the scope
b) Functions are executed using scope chain
c) Functions are declared outside the scope
d) Variables are declared within the function

Answer: b
Explanation: The fundamental rule of ***lexical scoping*** *is that the JavaScript functions are executed using the scope chain that was in effect when they were defined.*

8. What is the opposite approach to the lexical scoping?
a) Literal scoping
b) Static scoping
c) Dynamic scoping
d) Generic scoping

Answer: c
Explanation: The opposite approach to the lexical scoping is the dynamic scoping. Dynamic scoping does not care how the code is written, but instead how it executes. Each time a new function is executed, a new scope is pushed onto the stack. This scope is typically stored with the function's call stack. When a variable is referenced in the function, the scope in each call stack is checked to see if it provides the value.

9. What is the purpose of the dynamic scoping?
a) Variables can be declared outside the scope
b) Variables must be declared outside the scope
c) Variables cannot be declared outside the scope
d) Variable cannot be declared within the function

Answer: a
Explanation: ***Dynamic scoping*** *creates variables that can be called from outside the block of code in which they are defined. A variable declared in this fashion is sometimes called a public variable.*

10. Which of the algorithmic languages is not lexical scoping standardized in?
a) Ada
b) Pascal
c) Modula2
d) Html

Answer: d
Explanation: Lexical scoping is standardized in all algorithmic languages (ALGOL), such as Ada, Pascal, and Modula2. Additionally, it is used in modern functional languages like ML and Haskell.

1. The behaviour of the instances present of a class inside a method is defined by __________
a) Method
b) Classes
c) Interfaces
d) Classes and Interfaces

Answer: b
Explanation: Objects of the class are also known as instances of the class. The behaviour of the instance of a class is defined by the class and is shared by all instances.

2. The keyword or the property that you use to refer to an object through which they were invoked is _________
a) from
b) to
c) this
d) object

Answer: c
Explanation: 'this' keyword is used to refer to the object through which the properties or methods were invoked. This use of 'this' is a fundamental characteristic of the methods of any class.

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

```
var o = new F();
o.constructor === F
```

a) false
b) true
c) 0
d) 1

Answer: b
Explanation: Constructor is a function property of the class which is used to create objects of that class. In the above code, both statements create an instance of the class.

4. The basic difference between JavaScript and Java is _________
a) There is no difference
b) Functions are considered as fields
c) Variables are specific
d) Functions are values, and there is no hard distinction between methods and fields

Answer: d
Explanation: Java is an OOP programming language while JavaScript is an OOP scripting language. The basic difference between JavaScript and Java is that the functions are values, and there is no hard distinction between methods and fields.

5. The meaning for *Augmenting classes* is that ___________
a) objects inherit prototype properties even in a dynamic state
b) objects inherit prototype properties only in a dynamic state
c) objects inherit prototype properties in the static state
d) object doesn't inherit prototype properties in the static state

Answer: a
Explanation: JavaScript's prototype-based inheritance mechanism is dynamic an object inherits properties from its prototype, even if the prototype changes after the object is created. This means that we can augment JavaScript classes simply by adding new methods to their prototype objects.

6. The property of JSON() method is __________
a) it can be invoked manually as object.JSON()
b) it will be automatically invoked by the compiler
c) it is invoked automatically by the JSON.stringify() method
d) it cannot be invoked in any form

Answer: c
Explanation: A common use of JSON is to exchange data to/from a web server. When sending data to a web server, the data has to be a string. In that case json.strigify() is used to convert a javascript object into a string.

7. When a class B can extend another class A, we say that?
a) A is the superclass and B is the subclass

b) B is the superclass and A is the subclass
c) Both A and B are the superclass
d) Both A and B are the subclass

Answer: a
Explanation: Superclass is the class from which subclasses are defined. Subclasses are also called extensions of superclass.therefore in the above scenario A will be superclass and B will be subclass.

8. If A is the superclass and B is the subclass, then subclass inheriting the superclass can be represented as _________
a) B=inherit(A);
b) B=A.inherit();
c) B.prototype=inherit(A);
d) B.prototype=inherit(A.prototype);

Answer: c
Explanation: inherit() function is a predefined function in javascript which is used to inherit properties of another class. The subclass B inherits the prototype of the class A.

9. The snippet that filters the filtered set is __________
a) var t=new FilteredSet(s, {function(s) {return !(x instanceof Set);}});
b) var t=new FilteredSet{function(s) {return !(x instanceof Set);}});
c) var t=new FilteredSet(s, {function(s) {return (x instanceof Set);}});
d) var t=new FilteredSet(s, {function(s) {return x;}});

Answer: a
Explanation: The instanceof operator is used to check the type of an object at run time. The instanceof operator returns a boolean value that indicates if an object is an instance of a particular class.

10. The method that can be used to create new properties and also to modify the attributes of existing properties is _________
a) Object.defineProperty()
b) Object.defineProperties()
c) Both Object.defineProperty() and Object.defineProperties()
d) Object.inherit()

Answer: c
Explanation: The method Object.defineProperty() defines a new property directly on an object, or modifies an existing property on an object, and returns the object.Both Object.defineProperty() and Object.defineProperties() can be used todefine new properties.

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

```
const obj1 =
{
        a: 10,
        b: 15,
        c: 18
};
const obj2 = Object.assign({c: 7, d: 1}, obj1);
console.log(obj2.c, obj2.d);
```

a) 7,1
b) 18,1
c) Undefined
d) Error

Answer: a
Explanation: The Object.assign() method is used to copy the properties and values of one object to another. Objects are

assigned and copied by reference.

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

```
function person()
{
        this.name = 'rahul';
}
function obj()
{
       obj.call(this)
}
obj.prototype = Object.create(person.prototype);
const app = new obj();
console.log(app.name);
```

a) undefined
b) runtime error
c) compilation error
d) rahul

Answer: d
Explanation: Object.create() method is used to create a new object with the specified properties. This is another way of creating a new object using specified object type. The particular function accepts two values as an argument.

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

```
const object1 = {};
Object.defineProperties(object1,
{
        property1:
        {
             value: 10
        }
 });
console.log(object1.property1);
```

a) 0
b) 10
c) undefined
d) error

Answer: b
Explanation: Object.defineProperties() is a predefined function in the object library of javascript. The Object.defineProperties() method defines new or modifies existing properties directly on an object and returns the object.

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

```
const prototype1 = {};
const object1 = Object.create(prototype1);
console.log(Object.getPrototypeOf(object1) === prototype1);
```

a) true
b) false
c) error
d) 0

Answer: a
Explanation: The Object.getPrototypeOf() method of JavaScript returns the prototype value of the specified object. There are no inherited properties in this object.

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

```
const obj1 = {
  property1: 2
};
Object.seal(object1);
obj1.property1 =4;
console.log(obj1.property1);
delete obj1.property1;
```

a) 2
b) 4
c) Error
d) Undefined

Answer: c
Explanation: The seal property does not allow the object to be deleted. The object to be sealed is passed as an argument, and the method returns the object which has been sealed.

1. The four kinds of class members are ________
a) Instance methods, Instance fields, Static method, Dynamic method
b) Instance fields, Instance methods, Class fields, Class methods
c) Instance fields, Non-instance fields, Dynamic methods, Global methods
d) Global methods, Local methods, Dynamic methods, Static methods

Answer: b
Explanation: A class mainly contains data members and associated member functions. Therefore over all different kinds of class members are instance field, instance method, class fields and class methods.

2. Different kinds of the object involved in a class definition are ________
a) Public object, Private object, Protected object
b) Constructor object, Function object, Destructor object
c) Constructor object, Prototype object, Instance object
d) Instance method, Static object, Dynamic object

Answer: c
Explanation: In JavaScript, there are three different objects involved in any class definition, and the properties of these three objects act like different kinds of class members namely, Constructor object, Prototype object, and Instance object.

3. The object whose properties are inherited by all instances of the class, and properties whose values are functions behaving like instance methods of the class, is ________
a) Instance object
b) Constructor object
c) Destructor object
d) Prototype object

Answer: d
Explanation: The properties of the prototype object are inherited by all instances of the class, and properties whose values are functions behave like instance methods of the class.

4. Which variables are used internally in object methods and are also globally visible?
a) Object properties
b) Variable properties
c) Method properties
d) Internal properties

Answer: b
Explanation: The variable properties are usually variables that are used internally in the object's methods, but can also be globally visible variables that are used through the page.

5. The class that represents the regular expressions is ________

a) RegExpObj
b) RegExpClass
c) RegExp
d) StringExp

Answer: c
*Explanation: Regular expression is an object that describes a pattern of characters. The JavaScript RegExp class represents regular expressions, and both string and **RegExp** define methods that use regular expressions.*
Answer: b
Explanation: Date() is a predefined object type in javascript. There are 4 ways to create a new date object:new Date(), new Date(year, month, day, hours, minutes, seconds, milliseconds), new Date(milliseconds), new Date(date string).

7. Which is the correct code that returns a complex number that is the complex conjugate of this one?
a) Complex.prototype.conj = function() { return new Complex(this.r, -this.i); };
b) Complex.prototype.conj = function() { return Complex(this.r, -this.i); };
c) Complex.prototype.conj = function() { return (this.r, -this.i); };
d) Complex.prototype.conj = function() { new Complex(this.r, -this.i); };

Answer: a
Explanation: Object.prototype.constructor specifies the function that creates the object prototype. The above code snippet returns a complex number that is the complex conjugate of this one.

8. How can we make methods available on all objects?
a) Object.add(methods)
b) Object.methods(add)
c) Object.add.methods(…)
d) Object.prototype

Answer: d
*Explanation: It is possible to add methods to **Object.prototype**, making them available on all objects. This is not recommended, however, because prior to ECMAScript5, there is no way to make these add-on methods non enumerable, and if you add properties to **Object.prototype**, those properties will be reported by all **for/in** loops.*

9. What is the procedure to add methods to HTMLElement so that they will be inherited by the objects that represent the HTML tags in the current document?
a) HTMLElement.prototype(…)
b) HTMLElement.prototype
c) HTML.addmethods()
d) HTML.elements(add)

Answer: b
*Explanation: It is implementation-dependent whether classes defined by the host environment (such as the web browser) can be augmented using **Object.prototype**. In many web browsers, for example, you can add methods to **HTMLElement.prototype** and those methods will be inherited by the objects that represent the HTML tags in the current document.*

10. You can refresh the webpage in JavaScript by using ________
a) window.reload
b) location.reload
c) window.refresh
d) page.refresh

Answer: b
*Explanation: The reload method is used to reload the current file. User can reload the page from the server through **location. Reload**.*

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

```
   emp={id:102,name:"Shyam Kumar",salary:40000}
   document.write(emp.id+" "+emp.name+" "+emp.salary);
```

a) 102 4000 Shyam Kumar
b) 102 4000
c) 102 Shyam Kumar 40000
d) 102 shyam kumar 40000

Answer: b
Explanation: In this method, the object is created using literals. Property and Value are separated by colon.

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

```
function emp(id,name)
{
    this.id=id;
    this.name=name;
}
e=new emp(103,"Vimal Jaiswal");

document.write(e.id+" "+e.name");
```

a) 103 vimal jaiswal
b) 103
c) 103 Vimal Jaiswal
d) Vimal jaiswal

Answer: c
Explanation: Object in the above code is created by passing arguments to the function. "this" keyword is used for defining the values.

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

```
var emp=new Object();
emp.name="Ravi Malik";
emp.salary=50000;
document.write("emp.name+" "+emp.salary);
```

a) Ravi malik 5000
b) Ravi Malik 50000
c) Ravi malik
d) 50000

Answer: b
Explanation: In the above code an object of class emp is made using the new operator. The above code prints the value of name and salary as defined in the object of the class.

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

```
function emp(name,salary)
{
     this.name=name;
     this.salary=salary;

     this.changeSalary=changeSalary;
     function changeSalary(otherSalary)
     {
         this.salary=otherSalary;
     }
}
e=new emp("Rahul",30000);
e.changeSalary(45000);
document.write("e.name+" "+e.salary);
```

a) Rahul 30000
b) Rahul
c) Rahul 45000
d) 45000

Answer: c
Explanation: The above code defines an additional method in JavaScript object. But before defining method, the property should be added in the function with the same name as method.

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

```
const obj = { 10: 'arry', 21: 'barry', 23: 'carry' };
console.log(Object.entries(obj)[2]);
```

a) [“arry”, “10”]
b) [“10”,”arry”]
c) [“21”,barry”]
d) [“23”,”carry”]

Answer: d
Explanation: Object.entries() method is used to return an array of a given object’s own [key, value] pairs. The ordering of the properties is the same as that given by looping over the property values of the object manually.

1. The functions provide() and require() of Dojo toolkit and Google’s Closure library are used for ___________
a) declaring and loading modules
b) declaring functions
c) declaring modules
d) loading modules

Answer: a
Explanation: Both the Dojo toolkit and Google’s Closure library define provide() and require() functions for declaring and loading modules.

2. The maximum number of global symbols a module can define is ____________
a) 2
b) 3
c) 1
d) 4

Answer: c
Explanation: Generally, the various modules are allowed to run in the pristine (or near pristine) environment that it expects. The modules should minimize the number of global symbols they define – ideally, no module should define more than one.

3. To define each of the set classes as a property of the sets object (namespace) for the module, the statement is
a) sets = sets.AbstractEnumerableSet.extend();
b) sets.SingletonSet = sets.AbstractEnumerableSet.extend(…);
c) sets.SingletonSet = sets.extend(…);
d) sets = sets.extend(…);

Answer: b
Explanation: Singleton is an object which can only be instantiated once. The extend keyword is used in class declarations or class expressions to create a class which is a child of another class. The sets object is the namespace for the module, and we define each of the set classes as a property of this object.

4. What will be the efficiency quotient of the following JavaScript statements?

```
var Set = sets.Set;
var s = new Set(1,2,3);
```

a) The programmer imports at once the frequently used values into the global namespace
b) There is no efficiency quotient, the programmer tries to make it inefficient
c) The programmer needs to import the Sets everytime he wants to use it
d) the programmer imports the set everytime the statement is encountered

Answer: a
Explanation: A programmer can import frequently used values into the global namespace. A programmer who was going to make frequent use of the Set class from the sets namespace might import the class like that.

5. The scope of a function is also called as ________
a) Predefined function
b) Module function
c) Public function
d) Private function

Answer: b
*Explanation: The scope of a function can be used as a private namespace for a module. Therefore, the scope of a function is called a **module function**.*

6. Modules that have more than one item in their API can ________
a) Assign itself to a global variable
b) Invoke another module of the same kind
c) Return a namespace object
d) Invoke another module of the same kind

Answer: c
Explanation: Namespace is a container for a set of identifiers, functions, methods and all that. It gives a level of direction to its contents so that it will be well distinguished and organized. Modules that have more than one item in their API can return a namespace object.

7. The provides() function and the exportsobject are used to _________
a) Register the module's API and Store their API
b) Call the modules api
c) Store the module's API
d) Register the modules api

Answer: a
*Explanation: Frameworks that define module loading systems may have other methods of exporting a module's API. There may be a **provides()** function for modules to register their API, or an **exports** object into which modules must store their API.*

8. What could be achieved by running the JavaScript code snippet below?

```
var sets = com.davidflanagan.collections.sets;
```

a) Importing a single module
b) Importing a module partially
c) Importing a namespace
d) Importing the entire module

Answer: d
Explanation: Rather than importing individual classes, a programmer might import the entire module to the global namespace.

9. The properties() method is a ________
a) Enumerable method
b) Non-enumerable method
c) Operational method

d) calling method

Answer: b
Explanation: The ***properties()*** *is a method to get information of the properties of a particular object. properties() method is a non-enumerable method.*

10. What can be done in order to avoid the creation of global variables in JavaScript?
a) To use a method that defines all the variables
b) To use an object that has the reference to all the variables
c) To use an object as its namespace
d) To use global functions

Answer: c
Explanation: One way for a module to avoid the creation of global variables is to use an object as its namespace. Instead of defining global functions and variables, it stores the functions and values as properties of an object (which may be referenced to a global variable).

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

```
<p id="demo"></p>
<script>
var x = 123e5;
document.getElementById("demo").innerHTML = x ;
</script>
```

a) 0.0123
b) 12300
c) Error
d) Undefined

Answer: b
Explanation: Extra large or extra small numbers can be written with scientific (exponential) notation. The number followed by e tells about the number of digits in the number.

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

```
<p id="demo"></p>
 <script>
function myFunction()
{
   var res = "";
   res = res + Number.isFinite(5-2) ;
   document.getElementById("demo").innerHTML = res;
}
</script>
```

a) 3
b) true
c) false
d) error

Answer: b
Explanation: The Number.isFinite() method determines whether a value is a finite number. This method returns true if the value is of the type Number, and equates to a finite number. Otherwise, it returns false.

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

```
<p id="demo"></p>
 <script>
function myFunction()
{
   var res = "";
   res = res + Number.isFinite(0 / 0);
   document.getElementById("demo").innerHTML = res;
}
</script>
```

a) 3
b) true
c) false
d) error

Answer: c
Explanation: The Number.isFinite() method determines whether a value is a finite number. The above code results into an infinte number therefore the output will be false.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var res = "";
   res = res + Number.isInteger('123');
   document.getElementById("demo").innerHTML = res;
}
</script>
```

a) True
b) False
c) Error
d) Undefined

Answer: b
Explanation: The Number.isInteger() method determines whether a value an integer. This method returns true if the value is of the type Number.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var res = "";
   res = res + Number.isInteger(0.5) + ": 0.5<br>";
   document.getElementById("demo").innerHTML = res;
}
</script>
```

a) true
b) false
c) error
d) undefined

Answer: b
Explanation: isInteger method returns true if the value is of the type Number, and an integer (a number without decimals). Otherwise it returns false.

1. The ‘$’ present in the RegExp object is called a ____________
a) character
b) matcher
c) metacharacter
d) metadata

Answer: c
Explanation: The 'S' is a special metacharacter that matches the end of a string. It is used to define or access elements in jquery.

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

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

a) text==pattern
b) text.equals(pattern)
c) text.test(pattern)
d) pattern.test(text)

Answer: d
Explanation: The given pattern is applied to the text given in the parenthesis. The test() method tests for a match in a string. This method returns true if it finds a match, otherwise it returns false.

3. The regular expression to match any one character not between the brackets is __________
a) […]
b) [^]
c) [^…]
d) [\D]

Answer: c
Explanation: RegExp defines a special set of character that is used to do manipulation on strings and other variables. The [^...] character class is used to match or draw any one character not between the brackets.

4. What does /[^(]* regular expression indicate?
a) Match one or more characters that are not open parenthesis
b) Match zero or more characters that are open parenthesis
c) Match zero or more characters that are not open parenthesis
d) Match one or more characters that are open parenthesis

Answer: c
*Explanation: The [^...] character class is used to match or draw any one character not between the brackets. One should always be careful while using * and ? as repetition characters as they may match zero instances of whatever precedes them, they are allowed to match nothing.*

5. What will be the result when non greedy repetition is used on the pattern /a+?b/?
a) Matches the letter b preceded by the fewest number of a's possible
b) Matches the letter b preceded by any number of a
c) Matches letter a preceded by letter b, in the stack order
d) Matches letter a present in the string

Answer: a
Explanation: Using non greedy repetition may not always produce the results you expect. /a+?b/ matches the letter b preceded by the fewest number of a's possible.

6. What does the subexpression /java(script)?/ result in?
a) It matches "java" followed by the optional "script"
b) It matches "java" followed by any number of "script"
c) It matches "java" followed by a minimum of one "script"
d) It matches "java" followed by a single "script"

Answer: a
Explanation: paranthesis () check for the optional presence of the argument in the string. subexpression /java(script)?/ matches "java" followed by the optional "script".

7. What is the most essential purpose of parentheses in regular expressions?
a) Define pattern matching techniques
b) Define subpatterns within the complete pattern
c) Define portion of strings in the regular expression
d) matching the complete string

Answer: b
Explanation: When a regular expression is successfully matched against a target string, it is possible to extract the portions of the target string that matched any particular paranthesized subpattern. The essential purpose of parantheses in regular expressions is to define subpatterns within the complete pattern.

8. The method that performs the search-and-replace operation to strings for pattern matching is _______
a) searchandreplace()
b) add()
c) edit()
d) replace()

Answer: d
Explanation: The replace() method performs a search-and-replace operation. It takes a regular expression as its first argument and a replacement string as its second argument.

9. What would be the result of the following statement in JavaScript using regular expression methods?
a) Returns ["123″"456″"789"]
b) Returns ["123″,"456″,"789"]
c) Returns [1,2,3,4,5,6,7,8,9]
d) Throws an exception

Answer: b
Explanation: The split() method can take regular expressions as its arguments. The split() method generally breaks the string on which it is called into an array of substrings, using the argument as a separator.

10. What will be the purpose of exec() in the following JavaScript code?

```
    var pattern = /Java/g;
    var text = "JavaScript is more fun than Java!";
    var result;
    while ((result = pattern.exec(text)) != null)
    {
        alert("Matched '" + result[0] + "'" +" at position " + result.index +";
              next search begins at " + pattern.lastIndex);
    }
```

a) Returns the same kind of array whether or not the regular expression has the global g flag
b) Returns different arrays in the different turns of iterations
c) Return a sub part of the array
d) Returns a null value

Answer: a
Explanation: exec() returns the same kind of array whether or not the regular expression has the global g flag. Recall that match() returns an array of matches when passed a global regular expression. exec(), by contrast, always returns a single match and provides complete information about that match. When exec() is called on a regular expression that has the g flag, it sets the lastIndex property of the regular expression object to the character position immediately following the matched substring. When exec() is invoked a second time for the same regular expression, it begins its search at the character position indicated by the lastIndex property. If exec() does not find a match, it resets lastIndex to 0.

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

```
       System.out.println(Pattern.matches("[amn]", "abcd"));
```

a) true

b) false
c) undefined
d) a

Answer: b
Explanation: The pattern.matches method tests whether the regular expression matches the pattern. The above code will result into false as string "abcd" is not among a, m, or n.

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

```
System.out.println(Pattern.matches("[amn]?", "a"));
```

a) true
b) false
c) undefined
d) bcd

Answer: a
Explanation: The above code tests for single occurrence of the character in a particular string. The symbol ? is used along with the "[]" for testing single occurrence.

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

```
     System.out.println(Pattern.matches("\\d", "1"));
```

a) true
b) false
c) undefined
d) 1

Answer: a
Explanation: The above code tests for single occurrence of digit. //d returns true when there is any digits occurring one time.

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

```
   System.out.println(Pattern.matches("[adf]+", "a"));
```

a) true
b) false
c) undefined
d) 0

Answer: a
Explanation: "+" sign in regex checks for more than one time occurrence of a particular character. It returns true if a character from the set is present more than once.

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

```
    System.out.println(Pattern.matches("[^abc]", "aemngq"));
```

a) true
b) false
c) undefined
d) 1

Answer: b
Explanation: "^" is used as a negation operator. The above code will print false as "a" is present in the string passed in the argument.

1. The Crockford's subset does not include which function in JavaScript?
a) eval()
b) coeval()
c) equal()
d) equivalent()

Answer: a
Explanation: The Crockford's subset does not include the with and continue statements or the eval() function. It defines functions using function definition expressions only and does not include the function definition statement.

2. What does javascript use instead of == and !=?
a) It uses bitwise checking
b) It uses === and !== instead
c) It uses equals() and notequals() instead
d) It uses equalto()

Answer: b
Explanation: The subset does not include the comma operator, the bitwise operators, or the ++ and — operators. It also disallows == and != because of the type conversion they perform, requiring use of === and !== instead.

3. What is being imposed on each subset to ensure that it conforms to the subset?
a) A parser to parse the code
b) A parser that parses and adds to the subset
c) A static verifier that parses code
d) A predefined function to parse the code

Answer: c
Explanation: Each subset is coupled with a static verifier that parses code to ensure that it conforms to the subset.

4. Why was "The Good Parts" designed as a language subset in JavaScript?
a) To improve programmer flexibility
b) To balance the workload of the programmer
c) To create an in-built compiler and interpreter
d) To improve programmer productivity

Answer: d
Explanation: The Good Parts is a language subset designed for aesthetic reasons and with a desire to improve programmer productivity. There is a larger class of subsets that have been designed for the purpose of safely running untrusted JavaScript in a secure container or "sandbox".

5. Which is the subset that is a secure container designed for the purpose of safely running untrusted JavaScript?
a) Sandbox
b) The Good Parts
c) Both Sandbox and Good Parts
d) Web browser

Answer: a
Explanation: There is a larger class of subsets that have been designed for the purpose of safely running untrusted JavaScript in a secure container or "sandbox".

6. Why is this keyword not preferred in JavaScript?
a) Highly memory consuming
b) Functions should access the global objects
c) Functions should not access the global objects
d) Very inefficient to use

Answer: c
*Explanation: The **this** keyword is forbidden or restricted because functions (in non-strict mode) can access the global*

*object through **this**. Preventing access to the global object is one of the key purposes of any sandboxing system.*

7. Which are the two functions that are not allowed in any secure subset?
a) evaluate() and restrict()
b) eval() and the Function() constructor
c) debugger() and test()
d) eval() and debugger()

Answer: b
*Explanation: **eval()** and the **Function()** constructor are not allowed in any secure subset because they allow the execution of arbitrary strings of code, and these strings cannot be statically analyzed.*

8. Which is the object that defines methods that allow complete control over page content?
a) The client-side document object
b) The server-side document object
c) Both client-side and server-side document object
d) Web document object

Answer: a
Explanation: A web page is divided into two object documents in which one is client-side document object and the other is server-side document object. The client-side document object defines methods that allow complete control over page content

9. Which was one of the first security subsets proposed?
a) FBJS
b) Caja
c) dojox.secure
d) ADSafe

Answer: d
Explanation: ADsafe was one of the first security subsets proposed) It was created by Douglas Crockford (who also defined The Good Parts subset). ADsafe relies on static verification only, and it uses JSLint as its verifier. It forbids access to most global variables and defines an ADSAFE variable that provides access to a secure API, including special-purpose DOM methods. ADsafe is not in wide use, but it was an influential proof-of-concept that influenced other secure subsets.

10. Which is the subset that transforms web content into secure modules that can be safely hosted on a web page?
a) Microsoft Web Sandbox
b) ADsafe
c) Caja
d) dojox.secure

Answer: d
Explanation: Caja is Google's open-source secure subset. Caja (Spanish for "box") defines two language subsets. Cajita ("little box") is a narrow subset like that used by ADsafe and dojox.secure. Valija ("suitcase" or "baggage") is a much broader language that is close to regular ECMAScript 5 strict mode (with the removal of eval()). Caja itself is the name of the compiler that transforms (or "cajoles") web content (HTML, CSS, and JavaScript code) into secure modules that can be safely hosted on a web page without being able to affect the page as a whole or other modules on the page.

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

```
var set = new Set();
set.add("one");
set.add("two");
for (let elements of set)
{
    document.writeln(elements+" ");
}
```

a) one
b) two
c) one two
d) undefined

Answer: c
Explanation: The JavaScript Set add() method is used to add an element to Set object with a specified value. Each element must have a unique value.

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

```
set.add("AngularJS");
set.add("Bootstrap");
set.delete("Bootstrap");
document.writeln(set.size);
```

a) 2
b) 1
c) 0
d) Undefined

Answer: b
Explanation: The delete() method is used to remove all the elements from the Set object which are passed as an argument to the delete function. Since the delete function is called once the size will be reduced to 1.

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

```
set.add("one");
set.add("two");
set.add("three");
set.clear();
document.writeln(set.size);
```

a) one
b) 1
c) 2
d) 0

Answer: d
Explanation: JavaScript Set clear() method is used to remove all the elements from Set object. The clear method will remove all the entries and hence the size of the set will be zero.

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

```
set.add("one");
set.add("two”);
var itr=set.values();
document.writeln(itr.next().value);
```

a) one
b) two
c) error
d) undefined

Answer: a
Explanation: values() method returns an object of new Set iterator. This object contains the value for each element. It maintains insertion order.

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

```
set.add("1");
set.add("2");
```

```
document.writeln(set.has("3"));
```

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

Answer: c
Explanation: The Set has() method indicates whether the Set object contains the specified value. It returns true if the specified value is present, otherwise false.

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

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

a) This will flash an error
b) Prints 4
c) Prints 3.14
d) Ambiguity

Answer: a
Explanation: Const keyword fixes the value of the variable. Const keyword can not be redefined. Therefore attempts to alter the value or re-declaration causes errors.

2. The let keyword cannot be used ___________
a) as a substitute of var
b) as a block statement to define new variables
c) to define variables that are scoped to a single expression
d) in a else if loop, as a substitute for var

Answer: d
Explanation: The let keyword can be used in four ways :

1. *as a variable declaration like var;*
2. *in a for or for/in loop, as a substitute for var;*
3. *as a block statement, to define new variables and explicitly delimit their scope; and*
4. *to define variables that are scoped to a single expression.*

3. The main difference between the variables declared with var and with let is __________
a) var is confined to a particular function but let is not
b) let is confined to a particular function but var is not
c) var defines values based on conditions but let does not
d) let doesn't let you change the value of the variable

Answer: b
Explanation: Variables declared with var have global scope whereas variable declared with let have block scope. Variables declared with let are defined only within the closest enclosing block (and any blocks nested within it, of course).

4. What will be the output of the following JavaScript code if oddsums(5); is executed after the following code snippet?

```
function oddsums(n)
{
     let total = 0, result=[];
     for(let x = 1; x <= n; x++)
     {
        let odd = 2*x-1;
        total += odd;
        result.push(total);
```

```
    }
    return result;
}
```

a) Returns [1,4,9,16,25]
b) Returns [1,2,3,4,5]
c) Returns [3,6,9,12,15]
d) Returns [1,3,5,7,9]

Answer: a
Explanation: Let keyword has block scope thus in the above code snippet the total variable will be defined inside the for loop. The above code returns 1, 4, 9, 16, 25 which is the square of the first five natural numbers.

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

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

a) Zero
b) Null
c) Reference Error
d) Value not found Error

Answer: c
Explanation: Console.log() is a predefined function in javascript which is used to print data or message on the console. If the argument of the console.log is not defined it will give a reference error.

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

```
let x=x+1;
console.log(x);
```

a) 0
b) Null
c) Reference error
d) NaN

Answer: d
Explanation: x has not been defined with any value hence the value of x+1 will be Nan. Therefore the console.log function will print NaN.

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

```
[x,y]=[y,x];
```

a) Throws exception
b) Swap the value of the two variables
c) Flashes an error
d) Creates a new reference object

Answer: c
Explanation: The above code snippet will flash an error message. This is due to the fact that the variables x and y are not defined.

8. Which looping statement allows XML tags to appear in JavaScript programs and adds API for operating on XML data?
a) for loop
b) while loop
c) for/each loop
d) do while loop

Answer: c
Explanation: The ***for/each*** *loop is a new looping statement standardized by E4X. E4X (ECMAScript for XML) is a language extension that allows XML tags to appear literally in JavaScript programs and adds syntax and API for operating on XML data.*

9. Which exception does the Iterators throw from their next() method when there are no more values to iterate, that work on finite collections?
a) Exit Iteration
b) Abort Iteration
c) Abort
d) Stop Iteration

Answer: d
Explanation: Iterators that work on finite collections throw Stop Iteration from their next() method when there are no more values to iterate. Stop Iteration is a property of the global object in JavaScript 1.7. Its value is an ordinary object (with no properties of its own) that is reserved for this special purpose of terminating iterations. Note, in particular,that Stop Iteration is not a constructor function like TypeError() or RangeError().

10. Which method of the iterable object returns an iterator object for the collection?
a) iterator()
b) _iterator_()
c) _iteration_()
d) _return_iterator_()

Answer: b
Explanation: An iterable object represents a collection of values that can be iterated. An iterable object must define a method named __iterator__() (with two underscores at the start and end of the name) which returns an iterator object for the collection.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var res = "";
   res=res + Number.isSafeInteger(Math.pow(2, 53)-1)+": 2<sup>53</sup>-1<br>";
   document.getElementById("demo").innerHTML = res;
}
</script>
```

a) True
b) False
c) Error
d) Undefined

Answer: a
Explanation: The Number.isSafeInteger() method determines whether a value is a safe integer. A safe integer is an integer that can be exactly represented as an IEEE-754 double precision number (all integers from (2^{53} – 1) to -(2^{53} – 1)).

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   document.getElementById("demo").innerHTML = Number.MAX_VALUE;
}
</script>
```

a) 1.7976931348623157e+308
b) 1.7976931348623157e+305
c) 1.7976931348623157e+307
d) Error

Answer: a
Explanation: The MAX_VALUE property returns the largest number possible in JavaScript. This static property has a value of 1.7976931348623157e+308.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   document.getElementById("demo").innerHTML = Number.NEGATIVE_INFINITY;
}
</script>
```

a) -1000
b) -infinity
c) infinity
d) undefined

Answer: b
Explanation: The NEGATIVE_INFINITY property represents negative infinity. Negative infinity can be explained as something that is lower than any other number.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var x = 100;
   document.getElementById("demo").innerHTML = x.NEGATIVE_INFINITY;
}
</script>
```

a) True
b) False
c) Error
d) Undefiend

Answer: d
Explanation: NEGATIVE_INFINITY is a static property of the JavaScript Number object. Using x.NEGATIVE_INFINITY, where x is a number or a Number object, will return undefined.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var num = 5.56789;
   var n = num.toExponential();
   document.getElementById("demo").innerHTML = n;
}
</script>
```

a) 5.56789e+0
b) 5.57e+0
c) 5.568e+0

d) Error

Answer: a
Explanation: The toExponential() method converts a number into an exponential notation. It displays the number of digit according to the input passed to it.

1. What convenience does the following JavaScript code snippet provide?

```
let succ = function(x) x+1, yes = function() true, no = function() false;
```

a) Functional behaviour
b) Modular behaviour
c) No convenience
d) Shorthand expression

Answer: a
Explanation: The functions defined in this way behave exactly like functions defined with curly braces and the ***return*** *keyword. The above code makes the expression short and reduces the line of code.*

2. What does the following JavaScript code snippet do?

```
data.sort(function(a,b),b-a);
```

a) Sort in the alphabetical order
b) Sort in the chronological order
c) Sort in reverse alphabetical order
d) Sort in reverse numerical order

Answer: d
Explanation: 'a-b' is used to sort the array in ascending order whereas 'b-a' is used to sort the array in descending order. Therefore the above code snippet sorts an array in reverse numerical order.

3. What is the code to be used to trim whitespaces?
a) let trimmed = (l.trim() for (l in lines));
b) let trimmed = (trim(l));
c) let trimmed = l.trim();
d) let trimmed = for(l in lines));

Answer: a
Explanation: The various types of trim functions are trimLeft(), trimRight() and trim().can use the above code to trim whitespaces and filter out comments and blank lines.

4. What will be the reaction when a catch clause has no conditionals?
a) Takes it to be 0
b) Takes it to be 1
c) Takes it to be true
d) Takes it to be false

Answer: c
Explanation: The try and catch statement handles some or all of the errors that may occur in a block of code, while still running code. If a catch clause has no conditional, it behaves as if it has the conditional if true, and it is always triggered if no clause before it was triggered.

5. When will the finally block be called?
a) When there is no exception
b) When the catch does not match
c) When there is exception
d) After try-catch execution

Answer: d
Explanation: The try and catch statement handles some or all of the errors that may occur in a block of code, while still running code. A finally block is called after try-catch execution.

6. What is the return type of typeof for standard JavaScript objects?
a) xml
b) object
c) dom
d) html

Answer: b
Explanation: The ***typeof*** *operator returns "object" for all standard JavaScript objects. It returns "object" for a null, "number" for NaN, "number" for Infinity, "object" for a "new Number(1)" and "object" for an array.*

7. Which method to use while working with XML fragments, instead of XML()?
a) XMLInterface()
b) XMLClass()
c) XMLList()
d) XMLArray()

Answer: c
Explanation: An XML fragment is an XML document with no single top-level root element. When working with XML fragments, use XMLList() instead of XML().

8. Which of the following is the descendant operator?
a) ..
b) …
c) *
d) @

Answer: a
Explanation: While the . operator accesses direct children of the given node, the .. operator accesses all children no matter how deeply nested: The .. operator is the descendant operator; you can use it in place of the normal. member-access operator:

```
var names = pt..name;
```

9. Which of the following is an example to perform the most common XML manipulations using the XML objects invocation?
a) insertChildBefore()
b) insertChildAfter()
c) appendChildAfter(…)
d) appendChildBefore(…)

Answer: a
Explanation: E4X is designed so that you can perform most common XML manipulations using language syntax. E4X also defines methods you can invoke on XML objects. Here, for example, is the ***insertChildBefore()*** *method:*

```
pt.insertChildBefore(pt.element[1],<element id="1"><name>Deuterium</name></element>);
```

10. What is the code required to delete all "weight" tags?
a) delete weight(pt).all;
b) delete pt.element[all];
c) delete pt;
d) delete pt..weight;

Answer: d
Explanation: Delete is a keyword in javascript which is used for deleting objects ,pointers and variables. Removing

attributes and tags is very easy with the standard delete operator :

```
delete pt..weight; //delete all <weight> tags
```

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

```
<p id="demo"></p>
<script>
var x = 10;
var y = "20";
var z = x + y;
document.getElementById("demo").innerHTML = z;
</script>
```

a) 30
b) 10 20
c) 1020
d) Error

Answer: c
Explanation: If a number and a numeric string is added, then the result will be a concatenated string. The two numbers will not be added as they are of different types.

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

```
<p id="demo"></p>
<script>
var x = 10;
var y = 20;
document.getElementById("demo").innerHTML = x + y;
</script>
```

a) 1020
b) 10 20
c) 30
d) Error

Answer: a
Explanation: The two numbers will not get and rather get concatenated as string. Therefore the output will be 1020.

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

```
<p id="demo"></p>
<script>
var x = "100";
var y = "10";
var z = x * y;
document.getElementById("demo").innerHTML = z;
</script>
```

a) 1000
b) 10
c) 10010
d) Error

Answer: a
Explanation: JavaScript will try to convert strings to numbers when multiplying. Therefore the two values will be converted to numbers and multiplied.

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

```
<p id="demo"></p>
<script>
```

```
var x = 500;         // x is a number
var y = new Number(500);  // y is an object
document.getElementById("demo").innerHTML = (x===y);
</script>
```

a) true
b) false
c) error
d) undefined

Answer: b
Explanation: In the above code an object and a number are compared. When different data types are compared then the answer is false.

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

```
<p id="demo"></p>
<script>
var x = 9.656;
document.getElementById("demo").innerHTML =
  x.toFixed(0)
</script>
```

a) 10
b) 9.65
c) 9.6
d) 9.656

Answer: a
Explanation: toFixed() method rounds off the number to specified number of decimal places. Since the argument is passed with 0 value therefore the output will be 10.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
     document.getElementById("demo").innerHTML = Math.trunc(8.76);
}
</script>
```

a) 8.70
b) 8.76
c) 8.00
d) Error

Answer: c
Explanation: The trunc() method returns the integer part of a number. This method will NOT round the number up/down to the nearest ingeger, but simply remove the decimals.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
     document.getElementById("demo").innerHTML = Math.LN10;
}
</script>
```

a) 2.00
b) 2.10

c) 2.20
d) 2.30

Answer: d
Explanation: The LN10 property returns the natural logarithm of 10, approximately 2.302. The method is find in the math library of Javascript.

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

```
<p id="demo"></p>

<script>
function myFunction()
{
     document.getElementById("demo").innerHTML = Math.acosh(2);
}
</script>
```

a) 1.31
b) 1.20
c) 1.11
d) 1.41

Answer: a
Explanation: The acosh() method returns the hyperbolic arccosine of a number. If the parameter x is less than 1, the method will return NaN.

1. What are the events generated by the Node objects called?
a) generators
b) emitters
c) dispatchers
d) highevents

Answer: b
Explanation: There are two classes of events one is called event listener and the other is called event emitter. Node objects that generate events (known as event emitters) define an on() method for registering handlers.

2. What is the function used to deregister event handler 'f'?
a) deleteAllListeners(name)
b) deleteListener(name,f)
c) removeListener(name,f)
d) removeAllListeners(name)

Answer: c
Explanation: The removeEventListener() method removes an event handler that has been attached with the addEventListener() method. The removeListeners(name,f) is used to deregister event handler f represented as : ***emitter.removeListener(name,f)***

3. What is the function used to remove all handlers for name events?
a) deleteAllListeners(name)
b) deleteListener(name,f)
c) removeListener(name,f)
d) removeAllListeners(name)

Answer: d
Explanation: The removeAllListeners(name) is used to remove all handlers from name events represented as : ***emitter.removeAllListeners(name)***

4. Which function is a synonym for on()?
a) addListener()

b) listeners()
c) once()
d) add()

Answer: a
Explanation: The on() method is used for registering handlers. addListener() is a synonym for on().

5. Which of the following is an event emitter?
a) once
b) process
c) listeners
d) on

Answer: b
Explanation: The process object is an event emitter. The Node defines other important globals under the process namespaces that contain properties of that object like version, argv, env, pid, getuid(), cwd(), chdir() and exit().

6. When do uncaught exceptions generate events?
a) When handlers are registered
b) When handlers are deregistered
c) When handler functions are called
d) When handlers do not have a matching catch clause

Answer: a
Explanation: The on() method and addListener() method perform the same task of acting as an event emitter. The on() and addListener() method is used for registering handlers.

7. Which among the following POSIX signals generate events?
a) SIGDOWN
b) SIGFLOAT
c) SIGINT
d) SIGSHORT

Answer: c
Explanation: The SIGINT is a POSIX signal that generates event. Simple code like below can do a proper clean up and exit on CTRL-C or SIGINT passed from command line / other application to the nodejs app's ProcessID.

```
process.on('SIGINT', function()
      console.log('SIGINT');
      cleanup()
      process.exit(1));
```

8. What is the method used to pause "data" events?
a) s.pause();
b) s.stop();
c) s.halt();
d) s.wait();

Answer: a
Explanation: Data events are the events which are performed by either the user or the browser. The above code snippet is used to pause data events, for throttling uploads.

9. When the "end" event fires on EOF when no more data will arrive, which function is called?
a) s.on("data",f);
b) s.on("end",f);
c) s.on("error",f);
d) s.on("default",f);

Answer: b

Explanation: ”EOF” stands for end of file.The above code snippet gets “end” event fired on EOF when no more data will arrive.

10. What will be the return value of the write() method when the Node cannot write the data immediately and has to buffer it internally?
a) 0
b) 1
c) True
d) False

Answer: d
Explanation: The write() method writes HTML expressions or JavaScript code to a document. The write() method is mostly used for testing: If it is used after an HTML document is fully loaded, it will delete all existing HTML. write() method never blocks. If Node cannot write the data immediately and has to buffer it internally, the write() method returns false.

11. If the user presses “ok” in the dialog box then what will be the output of the following JavaScript code?

```
function msg()
{
    var v= confirm("Are u sure?");
    if(v==true)
    {
        alert("yes");
    }
    else
    {
        alert("no");
    }
}
```

a) true
b) yes
c) no
d) undefined

Answer: b
Explanation: The function confirm is present in the window object. confirm() method displays the confirm dialog box containing a message with ok and cancel button.

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

```
document.writeln("<br/>navigator.appCodeName: "+navigator.appCodeName);
```

a) Browser name
b) Version
c) Error
d) Undefined

Answer: a
Explanation: The JavaScript navigator object is used for browser detection. appCodeName returns the browser name.

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

```
document.writeln("<br/>navigator.language: "+navigator.language);
```

a) Broswer name
b) Browser language
c) Browser version
d) Error

Answer: c
Explanation: navigator.language returns the language of the browser. It is supported in Netscape and Firefox only.

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

```
document.writeln("<br/>navigator.appVersion: "+navigator.appVersion);
```

a) Browser version
b) Browser name
c) Browser language
d) Error

Answer: a
Explanation: navigator.version is present in the navigator object. appVersion returns the version of the browser.

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

```
document.writeln("<br/>screen.width: "+screen.width);
```

a) Browser length
b) Browser width
c) Browser area
d) Error

Answer: b
Explanation: screen.width method is provided in the screen object of Javascript. It returns the width of the browser screen

1. Rhino is originated by _______
a) Microsoft
b) Mozilla
c) Apple
d) Chrome

Answer: b
Explanation: Rhino is a JavaScript engine written fully in Java and managed by the Mozilla Foundation as open source software. Rhino is free software from Mozilla. You can download a copy from http://www.mozilla.org/rhino/.

2. Which of the following are global functions that are not part of core JavaScript?
a) spawn(f);
b) trim();
c) exult();
d) eval()

Answer: a
Explanation: The spawn(f) runs f() or loads and executes file f in a new thread.

3. Which of the following reads the textual contents of a URL and returns as a string?
a) spawn(f);
b) load(filename,…);
c) readFile(file);
d) readUrl(url);

Answer: d
Explanation: Rhino defines a handful of important global functions that are not part of core JavaScript in which readUrl(url) reads the textual contents of a URL and return as a string.

4. Which Rhino command quits Rhino environment?
a) terminate()

b) exit()
c) quit()
d) close()

Answer: c
Explanation: quit() is a predefined command in rhino. The quit() command makes Rhino exit.

5. Which is a useful way to try out small and simple Rhino programs and one-liners?
a) Starting an interactive shell
b) Starting a one to one shell
c) Creating a thread to do simple programs
d) Starting a multiple shell

Answer: a
Explanation: Rhino is distributed as a JAR archive. Start it with a command line like this :
java -jar rhino1_7R2/js.jar program.js
*If you omit **program.js**, Rhino starts an interactive shell, which is useful for trying out simple programs and one-liners.*

6. Which is a more formal way of importing packages and classes as JavaScript objects?
a) import(java.util.*);
b) importClass(java.util.*);
c) import.Class(java.util.*);
d) Class.import(java.util.*);

Answer: b
Explanation: Because packages and classes are represented as JavaScript objects, you can assign them to variables to give them shorter names. But you can also more formally import them, if you want to:

```
importClass(java.util.HashMap); // Same as : var HashMap = java.util.HashMap
```

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

```
var f = new java.io.File("/tmp/test");
var out = new java.io.FileWriter(f);
out instanceof java.io.Reader
```

a) Error
b) True
c) Exception
d) False

Answer: d
*Explanation: The output for the above code snippet is **false** as it is a writer and not a Reader.*

8. What does Rhino do when the getter and setter methods exist?
a) It becomes JavaScript properties
b) Java classes are used to avoid them
c) Java classes & JavaScript properties
d) It act as javascript function

Answer: a
Explanation: Rhino allows JavaScript code to query and set the static fields of Java classes and the instance fields of Java objects. Java classes often avoid defining public fields in favor of getter and setter methods. When getter and setter methods exist, Rhino exposes them as JavaScript properties.

9. The JavaScript classes can be instantiated using _____ operator.
a) create
b) new
c) instantiate

d) create.new

Answer: b
Explanation: New is a keyword in JavaScript which is used to define new models. New operator is used to create new instance of the class.

10. The new Java arrays can be created into JavaScript programs using which of the following classes?
a) java.Array
b) java.lang.*
c) java.lang.Array
d) java.lang.reflect.Array

Answer: d
*Explanation: There is no natural JavaScript syntax that Rhino can extend to allow JavaScript programs to create new Java arrays, so you have to do that using the **java.lang.reflect.Array** class.*

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

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
 Math.sin(90 * Math.PI / 180);
</script>
```

a) 1
b) 0
c) 1.6
d) 0.5

Answer: a
Explanation: Math.sin function evaluates the sine value of a particular input.
Math.PI function generates a value of 3.14.

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

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.abs(-4.5);
</script>
```

a) -4.5
b) 4.5
c) 0
d) Error

Answer: b
Explanation: The Math.abs method returns the absolute positive value of the argument passed to it. The above code will hence generate an output of 4.5.

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

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
Math.min(0, 150, 30, 20, -8, -200);
</script>
```

a) 0
b) -8
c) -200
d) 20

Answer: c
Explanation: Math.min() returns the lowest value in a list of arguments. The lowest value is -200, hence the output will be -200.

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

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.floor(4.7);
</script>
```

a) 4.5
b) 4
c) 4.7
d) 5

Answer: b
Explanation: Math.floor(x) returns the value of x rounded down to its nearest integer. The value is decreased in the decimal is removed.

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

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.sqrt(49);
</script>
```

a) 6
b) 7
c) 49
d) Error

Answer: b
Explanation: Math.sqrt function returns the square root of the argument passed to it. It is found in the math library of Javascript.

1. Which is a fast C++ based JavaScript interpreter?
a) Node
b) Sockets
c) Processors
d) Closures

Answer: a
Explanation: Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Node is a fast C++-based JavaScript interpreter with bindings to the low-level Unix APIs for working with processes, files, network sockets, etc., and also to HTTP client and server APIs.

2. Why does the Node rely on event handlers?
a) APIs are synchronous
b) APIs are asynchronous
c) APIs are reusable
d) APIs are modular

Answer: b
Explanation: For handling the spontaneous events occurring on the web page the event handlers are important. Because the APIs are asynchronous, Node relies on event handlers, which are often implemented using nested functions and closures.

3. What is the command to run the node programs?
a) node(program.js)

b) program.js
c) node program.js
d) node.program.js

Answer: c
*Explanation: The node programs can be run with the command:**node program.js**. The command can be written more simply like node program.*

4. What is the alternative command used in Node for load()?
a) store()
b) module()
c) log()
d) require()

Answer: d
Explanation: require() is used for including other javascript files. Use require() instead of load(). It loads and executes (only once) the named module, returning an object that contains its exported symbols.

5. What is the command used for debugging output in Node?
a) print();
b) console.log(…);
c) debug(…);
d) execute(…);

Answer: b
Explanation: Console.log() prints the content in the argument on to the output screen. Node defines console.log() for debugging output like browsers do.

6. What is the code to print hello one second from now?
a) setTimeout(function() { console.log("Hello World"); }, 1000);
b) setTimeout(function() { 1000, console.log("Hello World"); });
c) setTimeout(function(1000) { console.log("Hello World"); });
d) setTimeout(function() { console.log("Hello World"); });

Answer: a
Explanation: SetTimeout function is used to hold the execution of the code with the required amount of time. The argument of the setTimeout includes the function which is to be executed followed by the time after which the code is to be executed.

7. Among the below given functions, Node supports which of the following client-side timer functions?
a) getInterval()
b) Interval()
c) clearTime()
d) clearTimeout()

Answer: d
*Explanation: Client-side timer functions are used to perform applications based on time constraints. Node supports the client-side timer functions set **setTimeout(), setInterval(), clearTimeout(),** and **clearInterval()**.*

8. The necessary globals of a node are defined under which namespace?
a) variables
b) system
c) process
d) using

Answer: c
*Explanation: The process object is a global that provides information about, and control over, the current Node.js process. Node defines other important globals under the **process** namespace.*

9. Why does Node not block while waiting for operations to complete?
a) Static
b) Asynchronous
c) Synchronous
d) Recursive

Answer: b
Explanation: Node executes the function the in one go without any waiting or blocking. Because the Node's functions and methods are asynchronous, they do not block while waiting for operations to complete.

10. Which is the method used for registering handlers?
a) on()
b) register()
c) add()
d) include()

Answer: a
Explanation: The on() method attaches one or more event handlers for the selected elements and child elements. Node objects that generate events (known as event emitters) define an on() method for registering handlers.

1. Which is not a form of client-side storage?
a) Web Databases
b) FileSystem API
c) Offline Web Applications
d) Online Web Applications

Answer: d
Explanation: Client-side storage allows the creater to store data on the users system for faster loading of the website. The various forms of client-side storage are web databases, filesystem API, Offline web applications and cookies.

2. Which is the storage that allows the caching of web pages and their associated resources?
a) Web Databases
b) FileSystem API
c) Offline Web Applications
d) Cookies

Answer: c
Explanation: HTML5 defines an "Offline Web Applications" API that allows the caching of web pages and their associated resources (scripts, CSS files, images, and so on). This is client-side storage for web applications themselves rather than just their data, and it allows web apps to install themselves so that they are available even when there is no connection to the Internet.

3. Which is Microsoft's own proprietary client-side storage?
a) IE User Data
b) Offline Web Applications
c) Cookies
d) Offline Apis

Answer: a
Explanation: Microsoft implements its own proprietary client-side storage mechanism, known as "userData," in IE5 and later. userData enables the storage of medium amounts of string data and can be used as an alternative to Web Storage in versions of IE before IE8. This makes loading of programs and software faster.

4. Which object supports Filesystem API?
a) Element
b) File
c) Window
d) DOM

Answer: b
Explanation: These objects can be obtained from the filesystem property on any file system entry. Some browsers offer additional APIs to create and manage file systems, such as Chrome's requestFileSystem() method.

5. Which is the most appropriate database for developers requiring a huge amount of data?
a) Database
b) Datawarehouse
c) Web databases
d) Access

Answer: c
Explanation: Developers who need to work with really huge amounts of data like to use databases, and the most recent browsers have started to integrate client-side database functionality into their browsers. Client data base helps in making the website faster and handling the data easier.

6. The localStorage and sessionStorage belongs to ___________
a) Window object
b) Element object
c) Hash object
d) DOM object

Answer: a
*Explanation: Browsers that implement the "Web Storage" draft specification define two properties on the Window object: **localStorage** and **sessionStorage**. Local storage and Session storage are the web storage objects. Session storage is destroyed once the user closes the browser whereas, Local storage stores data with no expiration date.*

7. What is the main difference between localStorage and sessionStorage?
a) Lifetime
b) Scope
c) Both Lifetime and Scope
d) Storage Location

Answer: c
*Explanation: The difference between **localStorage** and **sessionStorage** has to do with lifetime and scope: how long the data is saved for and who the data is accessible to. Session storage is destroyed once the user closes the browser whereas, Local storage stores data with no expiration date.*

8. What is the lifetime of the data stored through localStorage?
a) Permanent
b) Temporary
c) Both Permanent and Temporary at times
d) Cannot store

Answer: a
*Explanation: Data stored through **localStorage** is permanent. it does not expire and remains stored on the user's computer until a web app deletes it or the user asks the browser (through some browser-specific UI) to delete it. This data is stored on the client side server and is used for faster access of data.*

9. Which is the function used to retrieve a value?
a) get()
b) retrieve()
c) getItem()
d) retrieveItem()

Answer: c
*Explanation: To retrieve a value, pass the name to **getItem()**. The getItem() method of the Storage interface, when passed a key name, will return that key's value, or null if the key does not exist, in the given Storage object.*

10. Which is the function used to store a value?
a) setItem()
b) set()
c) storeItem()
d) store()

Answer: a
*Explanation: To store a value, pass the name and value to **setItem()**. The setItem() method of the Storage interface, when passed a key name and value, will add that key to the given Storage object, or update that key's value if it already exists.*

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

```
<p id="demo"></p>
<script>
var carName = "Volvo";
var carName;
document.getElementById("demo").innerHTML = carName;
</script>
```

a) Error
b) Undefined
c) Volvo
d) Garbage value

Answer: c
Explanation: A variable does not lose its value if it is re-declared. The Javascript variable will store the value and the output will be Volvo.

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

```
<p>The result of adding "5" + 2 + 3:</p>
<p id="demo"></p>
<script>
x = "5" + 2 + 3;
document.getElementById("demo").innerHTML = x;
</script>
```

a) 523
b) 10
c) 5
d) Error

Answer: a
Explanation: When a string and integer is added in Javascript then the resulting output is string. Javascript will type caste integer to string and would then concatenate to produce the output.

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

```
<p id="demo">code</p>
<script>
function myFunction()
{
   var text = document.getElementById("demo").innerHTML;
   document.getElementById("demo").innerHTML = text.toUpperCase();
}
</script>
```

a) Code
b) CODE
c) code
d) Error

Answer: b
Explanation: toUpperCase function is present in the string Library of Javascript. It converts the string to uppercase.

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

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Number(true);
</script>
```

a) 1
b) 0
c) true
d) undefined

Answer: a
Explanation: The Number() method converts variables to numbers. True value is converted to 1 and a false value is converted to 0.

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

```
<p id="demo2"></p>
<script>
Var arr = ["one", "two", "three",];
arr.shift();
document.getElementById("demo2").innerHTML = arr;
</script>
```

a) two three
b) one two
c) one three
d) error

Answer: a
Explanation: The shift() method removes the first element of an array. It "shifts" all other elements to the left.

1. Which of the following is a way of embedding Client-side JavaScript code within HTML documents?
a) From javascript:encoding
b) External file specified by the src attribute of a "script" tag
c) By using a header tag
d) By using body tag

Answer: b
Explanation: The Client-side JavaScript code is embedded within HTML documents in four ways :

1. *Inline, between a pair of "script" tags*
2. *From an external file specified by the src attribute of a "script" tag*
3. *In an HTML event handler attribute, such as onclick or onmouseover*
4. *In a URL that uses the special javascript: protocol.*

2. When does JavaScript code appear inline within an HTML file?
a) Between the "script" tag
b) Outside the "script" tag
c) Between or Outside the "script" tag
d) Between the header tag

Answer: a
Explanation: JavaScript code can appear inline within an HTML file between the "script" tags. Javascript can also be included from an external file specified by the src attribute of a "script" tag.

3. Which character in JavaScript code will be interpreted as XML markup?
a) !
b) >
c) &
d) .

Answer: c
Explanation: If your JavaScript code contains the < or & characters, these characters are interpreted as XML markup. Element tags must begin with the < character, and entities and character references in an XML document must begin with the & character.

4. Which is the root element in a HTML document?
a) HTML
b) HEAD
c) SCRIPT
d) BODY

Answer: a
Explanation: The "html" tag is the root element of any HTML document regardless of it containing a JavaScript code or not. Body tag includes the main content that is shown on the website.

5. What is the code for getting the current time?
a) now = new Date();
b) var now = new Date();
c) var now = Date();
d) var now = new Date(current);

Answer: b
Explanation: Date() is a predefined function in javascript which returns the date in string form. The above code determines the current time and stores it in the variable "now".

6. What is the code to start displaying the time when the document loads?
a) onload = displayTime;
b) window. = displayTime;
c) window.onload = displayTime;
d) window.onload = start;

Answer: c
Explanation: window.onload is used to access the screen while the page is loading. The above code starts displaying the time when the document loads.

7. One of the main advantage of using src attribute is ____________
a) It becomes self-cached
b) It makes the HTML file modular
c) It restricts manipulation in the HTML file
d) It simplifies the HTML files

Answer: d
Explanation: The main advantage of using the src attribute is that it simplifies your HTML files by allowing you to remove large blocks of JavaScript code from them. Hence separate files for css and javascript files are made to make the code modular and readable.

8. What will be done if more than one page requires a file of JavaScript code?
a) Downloads that many times
b) Retrives from the browser cache
c) Must be re executed
d) Must be included in all the pages

Answer: b
Explanation: If a file of JavaScript code is shared by more than one page, it only needs to be downloaded once, by the first page that uses it—subsequent pages can retrieve it from the browser cache. This makes the loading process easier and hence faster.

9. What is the default value of the type attribute?
a) text/css
b) text/javascript
c) html
d) xml

Answer: b
Explanation: The default value of the ***type*** *attribute is "text/javascript". You can specify this type explicitly if you want, but it is never necessary.*

10. The language is commonly used to ___________________
a) Specify the user's language
b) Specify the language going to be scripted
c) No longer in use
d) Specify the programmer's favourable language

Answer: c
Explanation: The ***language*** *attribute specifies the natural language of the content of a web page. The language attribute is deprecated and should no longer be used.*

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

```
<p id="demo"></p>
<script>
var numbers1 = [4, 9];
var numbers2 = numbers1.map(myFunction);
document.getElementById("demo").innerHTML = numbers2;
function myFunction(value, index, array)
{
  return value * 2;
}
</script>
```

a) 8 9
b) 8 18
c) 4 9
d) Error

Answer: b
Explanation: Map function creates a new array by performing a function on each array element. myFunction multiplies each value with 2.

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

```
<p id="demo"></p>
<script>
var numbers = [45, 4, 9, 16, 25];
var ans = numbers.reduce(myFunction);
document.getElementById("demo").innerHTML = sum;
function myFunction(total, value, index, array)
{
  return total + value;
}
</script>
```

a) 100
b) 99

c) 0
d) error

Answer: b
Explanation: Reduce function reduces the array values to a single variable. The function returns the sum of array elements.

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

```
<p id="demo"></p>
<script>
var numbers = [45, 4, 9, 16, 25];
var arr= numbers.every(myFunction);
document.getElementById("demo").innerHTML =arr;
function myFunction(value, index, array)
{
   return value > 18;
}
</script>
```

a) true
b) false
c) 0
d) error

Answer: b
Explanation: The every() method checks if all array values pass a test. The function tests if all the values are greater than 18 or not.

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

```
<p id="demo"></p>
<script>
var numbers = [45, 4, 9, 16, 25];
var someOver18 = numbers.some(myFunction);
document.getElementById("demo").innerHTML = "Some over 18 is " + someOver18;
function myFunction(value, index, array)
{
  return value > 10;
}
</script>
```

a) True
b) False
c) Error
d) Undefined

Answer: a
Explanation: The some() method checks if some array values pass a test. Since some of the values are greater than 10 the answer will be true.

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

```
<p id="demo"></p>
<script>
var arr = ["1", "1", "2", "1"];
var a = arr.lastIndexOf("1");
document.getElementById("demo").innerHTML = (a + 1);
</script>
```

a) 2
b) 3
c) 4

d) 0

Answer: c
Explanation: The lastindexof method returns the last occurrence of element in the array. The last occurrence of 1 is at index 4.

1. The word "document" mainly refers to ______________
a) Dynamic Information
b) Static Information
c) Both Dynamic and Static Information
d) Temporary information

Answer: b
Explanation: Some pages present static information and can be called documents. The document is a keyword which is used in selecting any particular element in the document.

2. Which object is the main entry point to all client-side JavaScript features and APIs?
a) Standard
b) Location
c) Window
d) Position

Answer: c
Explanation: The Window object is the main entry point to all client-side JavaScript features and APIs. It represents a web browser window or frame, and you can refer to it with the identifier window.

3. Which identifier is used to represent a web browser window or frame?
a) frames
b) window
c) location
d) frame

Answer: b
Explanation: The window object represents a web browser window or frame, and you can refer to it with the identifier window. Global variables are properties of the window object and Global functions are methods of the window object.

4. Which property in the Window object is used to refer to a Location object?
a) position
b) area
c) window
d) location

Answer: d
*Explanation: The Window object defines properties like **location**, which refers to a Location object that specifies the URL currently displayed in the window and allows a script to load a new URL into the window. The window.location object can be used to get the current page address (URL) and to redirect the browser to a new page.*

5. Which Window object method is used to display a message in a dialog box?
a) alert()
b) prompt()
c) message()
d) console.log

Answer: a
Explanation: The Window object defines methods like alert(), which displays a message in a dialog box. A prompt message can be displayed on the screen using alert().

6. The setTimeout() method is used to ______________

a) Make the event sleep
b) Register a function to be invoked after a certain time
c) Invoke an event after a certain time
d) Time for iteration

Answer: b
*Explanation: The **setTimeout()**, which registers a function to be invoked after a specified amount of time. The **setTimeout()** method calls a function or evaluates an expression after a specified number of milliseconds.*

7. Which of the following is a global object?
a) Register
b) Location
c) Window
d) Position

Answer: c
Explanation: In client-side JavaScript, the Window object is also the global object. This means that the Window object is at the top of the scope chain and that its properties and methods are effectively global variables and global functions.

8. When will the window property come into play?
a) Representation convenience
b) Use as an extension of other objects
c) Use objects in the Window object
d) Refer to window object itself

Answer: d
Explanation: The Window object has a property named window that always refers to itself. You can use this property if you need to refer to the window object itself.

9. Which is the property that represents the content displayed in the window?
a) document
b) content
c) window
d) frame

Answer: a
*Explanation: One of the most important properties of the Window object is **document**: it refers to a Document object that represents the content displayed in the window.*

10. How to pick a document element based on the value of its id attribute?
a) getElementsbyId()
b) getElementbyId()
c) both getElementsbyId() and getElementbyId()
d) getElement

Answer: b
*Explanation: The Document object has important methods such as **getElementById()**, which returns a single document element (representing an open/close pair of HTML tags and all of the content between them) based on the value of its **id** attribute. Using this property different operation can be performed on the selected element.*

1. What is the property to access the first child of a node?
a) timestamp.Child1
b) timestamp.Child(1)
c) timestamp.Child(0)
d) timestamp.firstChild

Answer: d
*Explanation: The first child of a node can be accessed using the **firstChild** property. firstChild returns the first child node*

as an element node, a text node or a comment node (depending on which one's first).

2. What are the properties supporting CSS styles for a document element?
a) style and font
b) style and className
c) size and style
d) className and font

Answer: b
*Explanation: Each Element object has **style** and **className** properties that allow scripts to specify CSS styles for a document element or to alter the CSS class names that apply to the element. firstChild returns the first child node as an element node, a text node or a comment node (depending on which one's first).*

3. Which of the following object belongs to the style property?
a) Element
b) Window
c) Location
d) Dynamic

Answer: a
*Explanation: Each Element object has **style** and **className** properties that allow scripts to specify CSS styles for a document element or to alter the CSS class names that apply to the element.*

4. What is the purpose of the event handlers in the JavaScript?
a) Adds innerHTML page to the code
b) Performs handling of exceptions and occurrences
c) Allows JavaScript code to alter the *behaviour* of windows
d) Change the server location

Answer: c
Explanation: Event handlers allow JavaScript code to alter the behavior of windows, of documents, and of the elements that make up those documents.

5. Which handler is triggered when the content of the document in the window is stable and ready for manipulation?
a) onload
b) manipulate
c) create
d) onkeypress

Answer: a
Explanation: One of the most important event handlers is the onload handler of the Window object. It is triggered when the content of the document displayed in the window is stable and ready to be manipulated. JavaScript code is commonly wrapped within an onload event handler.

6. When a program contains extensive use of event handlers, which of the following is necessary?
a) Modular functions
b) Nested functions
c) Split up programs
d) Global variables

Answer: b
Explanation: Nested functions are those functions in which one function is defined inside another function. Nested functions are common in client-side JavaScript, because of its extensive use of event handlers.

7. What is the JavaScript code snippet to find all container elements with class "reveal"?
a) var elements = document.getElementsByClassName("reveal");
b) var elements = document.getElementByClassName("reveal");
c) var elements = document.getElementByName("reveal");

d) var elements = document.getElementsClassName("reveal");

Answer: a
Explanation: The getElementsByClassName() method returns a collection of all elements in the document with the specified class name, as a NodeList object. The above code snippet finds all container elements with class "reveal".

8. What is the JavaScript code snippet to update the content of the timestamp element when the user clicks on it?
a) timestamp.onLoad = function() { this.innerHTML = new Date().toString(); }
b) timestamp.onclick = function() { this.innerHTML = new Date().toString(); }
c) timestamp.onload = function() { this.innerHTML = new Date().toString(); }
d) timestamp.onclick = function() { innerHTML = new Date().toString(); }

Answer: b
Explanation: onclick() function is used to handle events when the user clicks on the mouse. The above code snippet updates the content of the timestamp element when the user clicks on it.

9. Which of the following is not an object?
a) Element
b) Location
c) Position
d) Window

Answer: c
Explanation: There is no object called Position. Whereas elements, location and window are a predefined object in JavaScript.

10. What is the JavaScript code snippet to change the class and let the stylesheet specify the details?
a) timestamp.className = "highlight";
b) timestamp.className = "change";
c) timestamp.className = "specify";
d) timestamp.className = "move";

Answer: a
Explanation: Each Element object has style and className properties that allow scripts to specify CSS styles for a document element or to alter the CSS class names that apply to the element. The above code snippet changes the class and lets the stylesheet specify the details.

1. The libraries that build a new higher-level API for client-side programming is ____________
a) Library
b) Framework
c) APIs
d) Script

Answer: b
Explanation: Many web developers find it useful to build their web applications on top of a client side framework library. These libraries are "frameworks" in the sense that they build a new higher-level API for client-side programming on top of the standard and proprietary APIs offered by web browsers: once you adopt a framework, your code needs to be written to use the APIs defined by that framework.

2. Which of the following is not a framework?
a) jQuery
b) .NET
c) JavaScript
d) Cocoa

Answer: c
Explanation: One of the most popular frameworks is jQuery which is used in web development. Here, JavaScript is a scripting language and not a framework.

3. Which of the following frameworks focuses on DOM and Ajax utilities?
a) jQuery
b) Prototype
c) Dojo
d) Both jQuery and Prototype

Answer: d
Explanation: DOM stands for data object modulation. The Prototype library focuses on DOM and Ajax utilities like jQuery does, and adds quite a few core-language utilities as well.

4. What is the purpose of the Dojo framework?
a) Focuses on DOM and Ajax utilities
b) Advertises incredible depth
c) Ajax utilities
d) Modular

Answer: b
Explanation: The Dojo toolkit is an open-source modular toolkit containing a JavaScript library that is designed for rapidly creating JavaScript/Ajax-based websites and cross-platform applications. Dojo is a large framework that advertises its "incredible depth." It includes an extensive set of UI widgets, a package system, a data abstraction layer, and more.

5. Which is the in-house library of Yahoo!?
a) Dojo
b) YUI
c) Prototype
d) Closure

Answer: b
Explanation: YUI is the in-house library of Yahoo!, and it is used on their home page. The Yahoo User Interface Library is a discontinued open-source JavaScript library for building richly interactive web applications using techniques such as Ajax, DHTML, and DOM scripting.

6. What does Dojo and YUI have in common?
a) Facilitates DOM utilities and UI Widgets
b) Does not facilitates DOM utilities and UI Widgets
c) Client-side library
d) Server-side library

Answer: a
Explanation: Like Dojo, it is a large, all-encompassing library with language utilities, DOM utilities, UI widgets, and so on. There are actually two incompatible versions of YUI, known as YUI 2 and YUI 3.

7. What are the two incompatible versions of YUI?
a) YUI1 and YUI2
b) YUI2 and YUI4
c) YUI1 and YUI3
d) YUI2 and YUI3

Answer: a
Explanation: YUI stands for yahoo user interface. Like Dojo, it is a large, all-encompassing library with language utilities, DOM utilities, UI widgets, and so on. There are actually two incompatible versions of YUI, known as YUI 2 and YUI 3.

8. Which of the following framework was used by Google for Gmail?
a) Dojo
b) GWT
c) Closure

d) YUI

Answer: c
Explanation: The Closure library is the client-side library that Google uses for Gmail, Google Docs, and other web applications. This library is intended to be used with the Closure compiler, which strips out unused library functions.

9. Which of the following is a web application API framework?
a) Dojo
b) YUI
c) GWT
d) jQuery

Answer: c
Explanation: GWT, the Google Web Toolkit defines a web application API in Java and provides a compiler to translate your Java programs into compatible client-side JavaScript whereas Google Closure Tools is a set of tools to help developers build rich web applications with JavaScript.

10. Which is more widely used than GWT in Google?
a) Closure
b) Dojo
c) Procedure
d) jQuery

Answer: a
Explanation: Google Closure Tools is a set of tools to help developers build rich web applications with JavaScript. GWT is used in some of Google's products, but it is not as widely used as their Closure library.

1. Which function among the following lets to register a function to be invoked once?
a) setTimeout()
b) setTotaltime()
c) setInterval()
d) settime()

Answer: a
Explanation: setTimeout() and setInterval() allow you to register a function to be invoked once or repeatedly after a specified amount of time has elapsed. Both these function are used to do time manipulation in javascript.

2. Which function among the following lets to register a function to be invoked repeatedly after a certain time?
a) setTimeout()
b) setTotaltime()
c) setInterval()
d) settime()

Answer: c
Explanation: setTimeout() and setInterval() allow you to register a function to be invoked once or repeatedly after a specified amount of time has elapsed. Both these function are used to do time manipulation in javascript.

3. Which is the handler method used to invoke when uncaught JavaScript exceptions occur?
a) Onhalt
b) Onerror
c) Both onhalt and onerror
d) Onsuspend

Answer: b
Explanation: The ***onerror*** *handler method can be registered to be invoked when uncaught JavaScript exceptions occur. The onerror event is triggered if an error occurs while loading an external file (e.g. a document or an image).*

4. Which property is used to obtain browser vendor and version information?

a) modal
b) version
c) browser
d) navigator

Answer: d
*Explanation: The **navigator** property is used to obtain browser vendor and version information. Various navaigator property includes appname, appversion, geolocation, language etc.*

5. Which method receives the return value of setInterval() to cancel future invocations?
a) clearInvocation()
b) cancelInvocation()
c) clearInterval()
d) clear()

Answer: c
Explanation: Like setTimeout(), setInterval() returns a value that can be passed to clearInterval() to cancel any future invocations of the scheduled function. The ID value returned by setInterval() is used as the parameter for the clearInterval() method.

6. The setTimeout() belongs to which object?
a) Element
b) Window
c) Location
d) Event

Answer: b
Explanation: The setTimeout() method of the Window object schedules a function to run after a specified number of milliseconds elapses. setTimeout() and setInterval() are used for time manipulations in javascript.

7. Which method receives the return value of setTimeout() to cancel future invocations?
a) clearTimeout()
b) clearInterval()
c) clearSchedule()
d) cancelInvocation()

Answer: a
Explanation: setTimeout() returns a value that can be passed to clearTimeout() to cancel the execution of the scheduled function. The ID value returned by setTimeout() is used as the parameter for the clearTimeout() method.

8. What will happen if we call setTimeout() with a time of 0 ms?
a) Placed in stack
b) Placed in queue
c) Will run continuously
d) Will execute immediately

Answer: b
Explanation: If you call setTimeout() with a time of 0 ms, the function you specify is not invoked right away. Instead, it is placed on a queue to be invoked "as soon as possible" after any currently pending event handlers finish running.

9. To which object does the location property belong?
a) Window
b) Position
c) Element
d) Location

Answer: a
Explanation: The location property of the Window object refers to a Location object, which represents the current URL

of the document displayed in the window. The window.location object can be used to get the current page address (URL) and to redirect the browser to a new page.

10. What is the result of the following code snippet?

```
window.location === document.location
```

a) False
b) True
c) 0
d) 1

Answer: b
Explanation: The window.location object can be used to get the current page address (URL) and to redirect the browser to a new page. The Document.location read-only property returns a Location object, which contains information about the URL of the document and provides methods for changing that URL and loading another URL.

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

```
function getcube()
{
    var number=document.getElementById("number").value;
    alert(number*number*number);
}
<form>
     Enter No:<input type="text" id="number" value="3" name="number"/><br/>
     <input type="button"  value="ok" onclick="getcube()"/>
</form>
```

a) 9
b) 27
c) Error
d) Undefined

Answer: b
Explanation: The document.getElementById() is used to get value of the input text. But we need to define id for the input field.

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

```
function totalelements()
{
    var allgenders=document.getElementsByName("gender");
    alert("Total Genders:"+allgenders.length);
}
<form>
     <input type="radio" name="gender" value="male">
     <input type="radio" name="gender" value="female">
     <input type="button" onclick="totalelements()" value="Total Genders">
</form>
```

a) 0
b) Error
c) 2
d) 1

Answer: c
Explanation: The document.getElementsByName() method returns all the element of specified name. The above code counts the total number of output mentioned in the form.

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

```
function counth2()
```

```
{
    var totalh2=document.getElementsByTagName("h2");
    alert("totalh2.length);
}
<h2>hello</h2>
<h2>hello</h2>
```

a) 0
b) hello
c) h2
d) 2

Answer: d
Explanation: The document.getElementsByTagName() method returns all the element of specified tag name. The above code counts the total number of specific tags.

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

```
function validate()
{
    var msg;
    if(document.myForm.userPass.value.length>5)
    {
        msg="good";
    }
    else
    {
         msg="poor";
    }
    document.getElementById('mylocation').innerText=msg;
}
<form name="myForm">
<input type="password" value="rhuld"  onkeyup="validate()">
Strength:<span id="mylocation">no strength</span>
</form>
```

a) Strength: good
b) Strength: poor
c) Strength: no strength
d) Undefined

Answer: b
Explanation: The innerText property can be used to write the dynamic text on the html document. It is used mostly in the web pages to generate dynamic content such as writing the validation message, password strength etc.

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

```
function showcommentform()
{
    var data="new text"
    document.getElementById('mylocation').innerHTML=data;
}
<form name="myForm">
<input type="button" value="comment" onclick="showcommentform()">
<div id="mylocation"></div>
</form>
```

a) Comment
b) new text
c) Error
d) Undefined

Answer: a
Explanation: The innerHTML property can be used to write the dynamic html on the html document. It is used mostly in

the web pages to generate dynamic html such as registration form, comment form, links etc.

1. The URL property belongs to which of the following object?
a) Document
b) Element
c) Location
d) Event

Answer: a
Explanation: The Document object has a ***URL*** *property, which is a static string that holds the URL of the document when it was first loaded. If you want to access any element in an HTML page, you always start with accessing the document object.*

2. What does the location property represent?
a) Current DOM object
b) Current URL
c) Both DOM object and URL
d) Document

Answer: b
Explanation: The ***location*** *property of a window is a reference to a Location object; it represents the current URL of the document being displayed in that window.*

3. Which among the following is not a property of the Location object?
a) protocol
b) host
c) hostee
d) hostname

Answer: c
Explanation: The location object is part of the window object and is accessed through the window.location property. The various properties of the location object are the ***protocol, host, hostname, port, search,*** *and* ***hash****.*

4. What is the return type of the hash property?
a) Query string
b) Packets
c) String
d) Fragment identifier

Answer: d
Explanation: The hash property sets or returns the anchor part of a URL. The hash property returns the "fragment identifier" portion of the URL if there is one a hash mark (#) followed by an element ID. It is accessed by using the statement location.hash.

5. What is the function used to extract arguments from the search property of a URL?
a) urlArgs()
b) url()
c) hash()
d) geturl()

Answer: a
Explanation: The ***urgArgs()*** *function can be used to extract arguments from the* ***search*** *property of a* ***URL****. Search property can be accessed through the location object.*

6. The decodeURIComponent() is defined by ______________
a) Server-side JavaScript
b) Client-side JavaScript
c) Both Server-side and Client-side JavaScript

d) Service side JavaScript

Answer: b
Explanation: The ***decodeURIComponent()*** *is a global function defined by client-side JavaScript. The decodeURIComponent() function decodes a Uniform Resource Identifier (URI) component previously created by encodeURIComponent or by a similar routine.*

7. Which is the method that removes the current document from the browsing history before loading the new document?
a) modify()
b) assign()
c) replace()
d) remove()

Answer: c
Explanation: The replace() removes the current document from the browsing history before loading the new document. The difference between assign method and replace(), is that replace() removes the URL of the current document from the document history, meaning that it is not possible to use the "back" button to navigate back to the original document.

8. Why is the replace() method better than the assign() method?
a) Reliable
b) Highly manageable
c) More efficient
d) Handles unconditional loading

Answer: d
Explanation: The difference between assign method and replace(), is that replace() removes the URL of the current document from the document history, meaning that it is not possible to use the "back" button to navigate back to the original document. When a script unconditionally loads a new document, the replace() method is often a better choice than assign().

9. What is the purpose of the assign() method?
a) Only loading
b) Loading of window and display
c) Displays already present window
d) Unloading of window

Answer: b
Explanation: The ***assign()*** *method of the Location object makes the window load and display the document at the URL you specify. The difference between this method and replace(), is that replace() removes the URL of the current document from the document history, meaning that it is not possible to use the "back" button to navigate back to the original document.*

10.The history property belongs to which object?
a) Element
b) Window
c) History
d) Location

Answer: c
Explanation: The ***history*** *property of the Window object refers to the History object for the window. The history object is part of the window object and is accessed through the window.history property.*

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

```
<p>1</p>
<p>2</p>
<p>3</p>
function myFunction()
```

```
{
    var x = document.getElementsByTagName("P").item(0);
    alert(x.innerHTML);
}
```

a) 1
b) 2
c) 3
d) Error

Answer: a
Explanation: Item object Returns the element at the specified index in an HTMLCollection. The function gets the HTML content of the first <p> element of the document.

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

```
<p>1</p>
<p>2</p>
<p>3</p>
function myFunction()
{
   var l = document.getElementsByTagName("P").length;
   alert(l);
}
```

a) 1
b) 2
c) 3
d) Error

Answer: c
Explanation: The length property returns the number of elements in a HTMLCollection. This element is useful when you want to loop through a HTMLCollection.

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

```
<p id="Element">Head</p>
function myFunction()
{
   var x = document.getElementsByTagName("P").namedItem("Element");
   alert(x.innerHTML);
}
```

a) Head
b) Element
c) Error
d) Undefined

Answer: a
Explanation: The namedItem() method returns the element with the specified ID, or name, in an HTMLCollection. A shorthand method can also be used, and will produce the same result.

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

```
<p id="demo">head</p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction()
{
   var x = document.getElementById("demo");
   x.innerHTML = x.attributes[0].isId;
}
</script>
```

a) head
b) true
c) false
d) 0

Answer: b
Explanation: The isId property returns true if the attribute is of type ID, otherwise it returns false. It is found in the DOM attribute object.

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

```
<head id="myHead">
  <title>My title</title>
</head>
<p id="demo"></p>
<script>
function myFunction()
{
   var x = document.head.id;
   document.getElementById("demo").innerHTML = x;
}
</script>
```

a) demo
b) myHead
c) error
d) undefined

Answer: a
Explanation: The head property returns the <head> element of the current document. The id property returns the id of the head.

1. What is the code snippet to go back to a history twice?
a) history(2);
b) history(-2);
c) history.go(-2);
d) history.go(2);

Answer: c
Explanation: The go() method loads a specific URL from the history list. The above code snippet goes back 2, like clicking the Back button twice.

2. If the window has child windows, how will the browsing histories be affected?
a) Numerically interleaved
b) Chronologically interleaved
c) Both Numerically and Chronologically interleaved
d) Numerically or Chronologically interleaved

Answer: b
Explanation: If a window contains child windows, the browsing histories of the child windows are chronologically interleaved with the history of the main window. The opener property returns a reference to the window that created the window.

3. The length property belongs to which of the following objects?
a) Window
b) Element
c) History
d) Document

Answer: c

Explanation: The length property of the History object specifies the number of elements in the browsing history list. The property returns at least 1, because the list includes the currently loaded page.

4. What is the datatype of the go() method's parameter?
a) String
b) Integer
c) Double
d) Float

Answer: b
*Explanation: The **go()** method takes an integer argument and can skip any number of pages forward and backward in the history list.*

5. What is the special feature of modern web applications?
a) Can alter contents without loading document
b) Must load the document to manipulate
c) Remains static
d) Can't be altered at all

Answer: a
Explanation: Modern web applications can dynamically alter their own content without loading a new document.

6. The navigator property belongs to which of the following object?
a) Document
b) Window
c) Navigator
d) Location

Answer: c
*Explanation: The **navigator** property of a Window object refers to a Navigator object that contains browser vendor and version number information. Navigator object property includes appCodeName, appVersion, appName etc.*

7. What is the vendor-neutral synonym for navigator?
a) staticData
b) purposeInformation
c) dataInformation
d) clientInformation

Answer: d
*Explanation: IE supports **clientInformation** as a vendor-neutral synonym for a navigator. The navigator property of a Window object refers to a **Navigator** object that contains browser vendor and version number information.*

8. Which is the preferred testing nowadays for scripting?
a) Software testing
b) Feature testing
c) Blackbox testing
d) Whitebox testing

Answer: b
Explanation: The "browser-sniffing" approach is problematic because it requires constant tweaking as new browsers and new versions of existing browsers are introduced. Today, feature testing is preferred rather than making assumptions about particular browser versions and their features, you simply test for the feature (i.e., the method or property) you need.

9. Which of the below properties can be used for browser sniffing?
a) platform
b) appVersion
c) both platform and appVersion

d) appName

Answer: c
*Explanation: The **platform** and **appVersion** can be found out in the navigator object properties.*

10. Where is the information of the userAgent property located?
a) appId
b) appName
c) platform
d) appVersion

Answer: d
Explanation: The string that the browser sends in its USER-AGENT HTTP header. This property typically contains all the information in appVersion and may contain additional details as well.

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

```
function myFunction()
{
    document.getElementById("demo").innerHTML = Boolean(10 > 9);
}
```

a) true
b) false
c) error
d) 0

Answer: a
Explanation: The boolean function returns the boolean values. Since 10 is greater than 9 the boolean function returns true.

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

```
var b5 = Boolean('false');
document.getElementById("demo").innerHTML =b5;
```

a) False
b) True
c) Error
d) Undefined

Answer: b
Explanation: The boolean function returns the boolean values. The boolean function returns true for any non empty string even if the string is false.

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

```
function myFunction()
{
  var x = "";
  document.getElementById("demo").innerHTML = Boolean(x);
}
```

a) true
b) false
c) 0
d) 1

Answer: b
Explanation: When an empty string is passed to the boolean function then the function returns false. The boolean function returns true or false according to the input passed to it.

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

```
function myFunction()
{
   var x = 10 / "H";
   document.getElementById("demo").innerHTML = Boolean(x);
}
```

a) True
b) False
c) Error
d) Undefined

Answer: b
Explanation: The value return by the boolean method depends on the input passed to it. The NaN value when passed to the boolean function returns false.

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

```
function myFunction()
{
   var x = null;
   document.getElementById("demo").innerHTML = Boolean(x);
}
```

a) True
b) False
c) Error
d) Undefined

Answer: b
Explanation: The value return by the boolean method depends on the input passed to it. The NULL value when passed to the boolean function returns false.

1. Each tab in the single web browser window is called as ____________
a) Browser Information
b) Browsing context
c) Both Browser Information & Browsing context
d) Browser Log

Answer: b
Explanation: A single web browser window on your desktop may contain several tabs. Each tab is an independent browsing context. browser context is also defined as the environment in which the browser displays a document.

2. Nested documents inside the HTML documents can be created using ___________
a) frame
b) nest
c) iframe
d) into

Answer: c
*Explanation: HTML documents may contain nested documents using an **iframe** element. An **iframe** creates a nested browsing context represented by a Window object of its own.*

3. How are windows, tabs, iframes, and frames treated according to client-side javascript?
a) They are all browsing contexts
b) They are all browsing information
c) They are all Window contexts
d) They are all Window objects

Answer: a
Explanation: Client-side JavaScript makes very little distinction between windows, tabs, iframes, and frames they are all browsing contexts, and to JavaScript, they are all Window objects.

4. How are windows, tabs, iframes, and frames treated according to javascript?
a) They are all browsing contexts
b) They are all browsing information
c) They are all Window contexts
d) They are all Window objects

Answer: d
Explanation: Client-side JavaScript makes very little distinction between windows, tabs, iframes, and frames they are all browsing contexts, and to JavaScript, they are all Window objects.

5. A new web browser window can be opened using which method of the Window object?
a) createtab()
b) Window.open()
c) open()
d) create()

Answer: b
*Explanation: You can open a new web browser window with the **open()** method of the Window object. **Window.open()** loads a specified URL into a new or existing window and returns the Window object that represents that window.*

6. What will happen if the first argument of open() is omitted?
a) Error Page
b) Remains in the same page
c) about:blank
d) Reloads the page

Answer: b
Explanation: The first argument in the open function is for the url of the page which is to be opened. When the first argument of the open() is omitted, the about:blank is opened.

7. Which object serves as the global object at the top of the scope chain?
a) Hash
b) Property
c) Element
d) Window

Answer: d
Explanation: The Window object serves as the global object at the top of the scope chain in client-side JavaScript. All global JavaScript objects, functions, and variables automatically become members of the window object.

8. Which is the property of a Window object that holds the name of the frame?
a) name
b) title
c) description
d) style

Answer: a
*Explanation: The **name** property of a Window object holds the name of the frame if it has one. This property is writable, and scripts can set it as desired. An iframe creates a nested browsing context represented by a Window object of its own.*

9. When will the fourth argument to open() useful?
a) When the second argument names a retired window
b) When the first argument names an existing window
c) When the second argument names an existing window

d) When the first argument names a retired window

Answer: c
Explanation: The fourth argument to ***open()*** *is useful only when the second argument names an existing window. This fourth argument is a boolean value that indicates whether the URL specified as the first argument should replace the current entry in the window's browsing history (true) or create a new entry in the window's browsing history (false). Omitting this argument is the same as passing* ***false****.*

10. The inner frame within a top-level window can be referred to as _____________
a) parent(parent)
b) parent.parent
c) parent*parent
d) parent/parent

Answer: b
Explanation: Frames returns the window itself, which is an array-like object, listing the direct sub-frames of the current window. If a frame is contained within another frame that is contained within a top-level window, that frame can refer to the top-level window as ***parent.parent****.*

1. The central object in a larger API is known as __________
a) Document Object Material
b) Document Object Model
c) Binary Object Model
d) Data object model

Answer: b
Explanation: Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document. It is the central object in a larger API, known as the Document Object Model, or DOM, for representing and manipulating document content.

2. The paragraph "p" is a part of __________
a) h1
b) body
c) html
d) both body and html

Answer: d
Explanation: The paragraph tag belongs to both html and body tag. It is used to write paragraph on html pages.

3. The node directly above a node is called __________
a) sibling
b) child
c) parent
d) ancestors

Answer: c
Explanation: The node directly above a node is the parent of that node. In HTML, the document itself is the parent node of the HTML element, HEAD and BODY are child nodes of the HTML element.

4. The Text and Comment is part of __________
a) CharacterData
b) Document
c) Attr
d) Element

Answer: a
Explanation: The ***CharacterData*** *abstract interface represents a Node object that contains characters. This is an abstract interface, meaning there aren't any object of type CharacterData. The Text and Comment is part of the*

CharacterData Element.

5. The nodes that represent HTML elements are the __________
a) Subclass nodes
b) HTML nodes
c) Window nodes
d) Element nodes

Answer: d
Explanation: The nodes that represent HTML elements are Element nodes. The various html elements include h1, p, div etc.

6. Which of the following is/are of Text nodes?
a) Text
b) Comment
c) Both Text and Comment
d) h1

Answer: c
*Explanation: Both **Text** and **Comment** are basically strings of text, and these nodes are much like the Text nodes that represent the displaying text of a document.*

7. Which is not the way to query a document for an element or elements?
a) With a specified id attribute
b) Matching the specified CSS selector
c) With the specified tag name
d) without the specified CSS class or classes

Answer: d
Explanation: The DOM defines a number of ways to select elements; you can query a document for an element or elements:

1. *with a specified id attribute;*
2. *with a specified name attribute;*
3. *with the specified tag name;*
4. *with the specified CSS class or classes; or*
5. *matching the specified CSS selector*

8. Which of the following can be used to select HTML elements based on the value of their name attributes?
a) getElementByName()
b) getElementsByName()
c) getElementsName()
d) getElementName()

Answer: b
Explanation: The getElementsByName() method returns a collection of all elements in the document with the specified name (the value of the name attribute), as a NodeList object.

```
var radiobuttons = document.getElementsByName("favorite_color");
```

9. Which of the following property refers to the root element of the document?
a) documentElement
b) elementdocument
c) rootdocument
d) rootelement

Answer: a
*Explanation: The **documentElement** property of the Document class refers to the root element of the document. This is*

always an HTML element. The documentElement property returns the documentElement of the document, as an Element object.

10. The return type of getElementsByClassName() is __________
a) DOM
b) Document
c) Node
d) NodeList

Answer: d
*Explanation: The **getElementsByClassName()** method returns a collection of all elements in the document with the specified class name, as a NodeList object. The other methods of returning nodelist objects are **getElementsByTagName()**, **getElementbyId()** etc.*

1. Which syntax is used to describe elements in CSS?
a) Protectors
b) Selectors
c) Both Protectors and Selectors
d) Protectors or Selectors

Answer: b
Explanation: CSS stylesheets have a very powerful syntax, known as selectors, for describing elements or sets of elements within a document.

2. What does the following JavaScript code snippet mean?

```
#log>span
```

a) Span child after log declaration
b) Specific span child of id greater than log
c) Span child less than log
d) Any span child of the element with id as log

Answer: d
Explanation: The above code snippet means that any span child of the element with id=”log”.

3. Which of the following is the ultimate element selection method?
a) querySelectorAll()
b) querySelector()
c) queryAll()
d) query()

Answer: a
*Explanation: **querySelectorAll()** is the ultimate element selection method: it is a very powerful technique by which client-side JavaScript programs can select the document elements that they are going to manipulate.*

4. Which of the following is the Web application equivalent to querySelectorAll()?
a) #()
b) &()
c) $()
d) !()

Answer: c
*Explanation: Web applications based on jQuery use a portable, cross-browser equivalent to **querySelectorAll()** named **$()**.*

5. The C in CSS stands for _______________
a) Continuous

b) Cascaded
c) Contentional
d) Cascading

Answer: d
Explanation: The C in CSS stands for "cascading". This term indicates that the style rules that apply to any given element in a document can come from a "cascade" of different sources.

6. The first version of CSS is _____________
a) CSS1
b) CSS2
c) CSS3
d) CSS

Answer: a
Explanation: The first version of CSS is CSS1. It was officially released in the year 1996.

7. Which of the following is not an example of a Shortcut Property?
a) border
b) font
c) text
d) value

Answer: d
Explanation: border, font & text are shortcut properties. For example, the font-family, font-size, font-style, and font-weight properties can all be set at once using a single font property with a compound value:

```
font: bold italic 24pt helvetica;
```

8. Which of the following is the default positioning elements with CSS?
a) relative
b) static
c) absolute
d) fixed

Answer: b
*Explanation: **static** is the default value and specifies that the element is positioned according to the normal flow of document content (for most Western languages, this is left to right and top to bottom).*

9. Which property lays the element according to the normal flow?
a) relative
b) absolute
c) fixed
d) static

Answer: a
*Explanation: When the position property is set to **relative**, an element is laid out according to the normal flow, and its position is then adjusted relative to its position in the normal flow.*

10. Which of the following property allows you to specify an element's position with respect to the browser window?
a) relative
b) fixed
c) static
d) absolute

Answer: b
*Explanation: The **fixed** value allows you to specify an element's position with respect to the browser window. Elements with **fixed** positioning are always visible and do not scroll with the rest of the document. Like absolutely positioned*

elements, fixed-position elements are independent of all others and are not part of the document flow.

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

```
<p id="demo"></p>
<script>
var x = 'It\'s';
document.getElementById("demo").innerHTML = x ;
</script>
```

a) It\’s
b) ‘It\’s’
c) It’s
d) Error

Answer: c
Explanation: If an apostrophe is present in the string then a backslash is added before it. The string skips the execution of the character after a backslash.

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

```
<p id="demo"></p>
<script>
var pos = str.indexOf("locate");
document.getElementById("demo").innerHTML = pos;
</script>
```

a) 5
b) 7
c) 0
d) Error

Answer: b
Explanation: The indexOf() method returns the position of the first occurrence of a specified text. locate occurs first time at 7 positions.

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

```
<button onclick="myFunction()">Try it</button>
<p id="demo">one</p>
<script>
function myFunction()
{
   var str = document.getElementById("demo").innerHTML;
   var txt = str.replace("one","two");
   document.getElementById("demo").innerHTML = txt;
}
</script>
```

a) one
b) two
c) error
d) undefined

Answer: b
Explanation: The replace function replaces string data with a specified value. The innerHtml function changes the value of data in the paragraph.

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

```
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
```

```
function myFunction()
{
   var str = "The rain in SPAIN stays mainly in the plain";
   var res = str.match(/z/);
   If(res)
      res=”true”;
   Else
      res=”false”;
   document.getElementById("demo").innerHTML = res;
}
</script>
```

a) True
b) False
c) Error
d) Handling

Answer: b
Explanation: The match method finds for a match in the given string. Since z is not present in the string, the output will be false.

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

```
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "a,b,c,d,e,f";
   var arr = str.split(",");
   document.getElementById("demo").innerHTML = arr[3];
}
</script>
```

a) d
b) a
c) b
d) c

Answer: a
Explanation: The split function splits the string according to the argument passed to it. The third index of the array will be initialized with d, hence the output will be d.

1. The type that specifies what kind of event occurred is ____________
a) event type
b) even target
c) both event type and even target
d) interface

Answer: a
Explanation: The event type is a string that specifies what kind of event occurred. The type “mousemove”, for example, means that the user moved the mouse.

2. Which is the object on which the event occurred or with which the event is associated?
a) event type
b) event target
c) both event type and even target
d) interface

Answer: b
Explanation: The event target is the object on which the event occurred or with which the event is associated. When we speak of an event, we must specify both the type and the target. The target property of the Event interface is a reference

to the object that dispatched the event.

3. In general, event handler is nothing but ____________
a) function
b) interface
c) event
d) handler

Answer: a
Explanation: An event handler is in general, a function that handles or responds to an event. For example onclick, onkeypress, onload etc are event handler functions.

4. When will the browser invoke the handler?
a) Program begins
b) Any event occurs
c) Specified event occurs
d) Webpage loads

Answer: c
Explanation: When an event of the specified type occurs on the specified target, the browser invokes the handler. For example onclick function is executed when mouse is clicked.

5. Which property specifies the property of the event?
a) Type
b) Target
c) Manner
d) Program

Answer: a
Explanation: All event objects have a ***type*** *property that specifies the event type. Event type is a string that specifies what kind of event occurred.*

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 propagation

Answer: d
Explanation: Event propagation is the process by which the browser decides which objects to trigger event handlers. Event propagation is a way to describe the "stack" of events that are fired in a web browser.

7. Which form of event propagation handles the registered container elements?
a) Event Propagation
b) Event Registration
c) Event Capturing
d) Default Actions

Answer: c
Explanation: Event bubbling and capturing are two ways of event propagation in the HTML DOM API. With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements. With capturing, the event is first captured by the outermost element and propagated to the inner elements.

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

Answer: b
Explanation: These are events that are directly tied to a specific input device, such as the mouse or keyboard) They include legacy event types such as "mousedown", "mousemove", "mouseup", "keydown", "keypress", and "keyup" and also new touch specific events like "touchmove" and "gesturechange".

9. The high-level events among the following events are ____________
a) User interface events
b) Device-independent events
c) Device-dependent events
d) Stage event change

Answer: a
Explanation: UI events are higher-level events, often on HTML form elements that define a user interface for a web application. They include the focus event (when a text input field gains keyboard focus), the change event when the user changes the value displayed by a form element, and the submit event when the user clicks a Submit button in a form.

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) Stage event change

Answer: b
Explanation: These are input events that are not directly tied to a specific input device. The click event, for example, indicates that a link or button (or other document element) has been activated) This is often done via a mouse click, but it could also be done by keyboard or (on touch-sensitive devices) by gesture.

1. 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

Answer: d
Explanation: Some events are not triggered directly by user activity, but by network or browser activity, and indicate some kind of lifecycle or state-related change. The load event, fired on the Window object when the document is fully loaded, is probably the most commonly used of these events.

2. 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

Answer: b
Explanation: The HTML5 video and audio elements define a long list of associated event types such as "waiting", "playing", "seeking", "volumechange", and so on. These events are usually only of interest to web apps that want to define custom controls for video or audio playback.

3. The client-side JavaScript asynchronous programming model contains _________
a) Timers and error handlers
b) User interface events
c) State change events
d) API-specific events

Answer: a
Explanation: Timers and error handlers are part of client-side JavaScript asynchronous programming model and are similar to events.

4. 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) form-related events

Answer: c
Explanation: The submit and reset events have default actions that can be canceled by event handlers, and some click events do, too. The focus and blur events do not bubble, but all the other form events do.

5. The events that represent occurrences related to the browser window are _________
a) Window
b) Element
c) Display
d) Handlers

Answer: a
Explanation: Window events represent occurrences related to the browser window itself, rather than any specific document content displayed inside the window.

6. 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

Answer: b
Explanation: The load event is the most important of these events: it is fired when a document and all of its external resources (such as images) are fully loaded and displayed to the user.

7. Which is the alternative to the load event?
a) readychange
b) changestate
c) readystatechange
d) contentloader

Answer: c
*Explanation: **DOMContentLoaded** and **readystatechange** are alternatives to the load event: they are triggered sooner, when the document and its elements are ready to manipulate, but before external resources are fully loaded.*

8. Which is the opposite of the load event in JavaScript?
a) dontload
b) postload
c) preload
d) unload

Answer: d
Explanation: The unload event is the opposite of load: it is triggered when the user is navigating away from a document. An unload event handler might be used to save the user's state, but it cannot be used to cancel navigation.

9. Which is the property that is triggered in response to JavaScript errors?
a) onexception
b) onmessage
c) onerror
d) onclick

Answer: c
*Explanation: The **onerror** property of the Window object is something like an event handler, and it is triggered in*

response to JavaScript errors. It isn't a true event handler, however, because it is invoked with different arguments.

10. Which event can be fired on any scrollable document element?
a) Window
b) Scroll
c) Load
d) Unload

Answer: b
Explanation: Scroll events can also be fired on any scrollable document element, such as those with the CSS overflow property set.

1. When are the mouse events generated?
a) When user clicks the mouse over a document
b) When user moves over a document
c) On pressing a key
d) When user clicks or moves the mouse over a document

Answer: d
Explanation: Mouse events are generated when the user moves or clicks the mouse over a document. These events are triggered on the most deeply nested element that the mouse pointer is over, but they bubble up through the document.

2. The properties that specify the position and button state of the mouse are _________
a) clientX and clientY
b) clientY and clientX
c) altKey and ctrlKey
d) metaKey and shiftKey

Answer: a
Explanation: The clientX and clientY properties specify the position of the mouse in window coordinates. The clientX property returns the horizontal coordinate (according to the client area) of the mouse pointer when a mouse event was triggered. Similarly clientY returns the vertical coordinates.

3. Which of the following keys are not set to true when the keyboard modifier keys are held down?
a) altKey
b) ctrlKey
c) metaKey
d) delkey

Answer: d
*Explanation: A modifier key is a key that modifies the action of another key when the two are pressed together. The **altKey**, **ctrlKey**, **metaKey**, and **shiftKey** properties are set to true when the corresponding keyboard modifier keys are held down.*

4. How to detect and respond to mouse drags?
a) Registering a mouseover handler
b) Releasing a mousedown handler
c) Registering a mousedown handler
d) Releasing a mouseover handler

Answer: c
Explanation: By registering a mousedown handler that registers a mousemove handler, you can detect and respond to mouse drags. Doing this properly involves being able to capture mouse events so that you continue to receive mousemove events even when the mouse has moved out of the element it started in.

5. When is the mouseover event fired?
a) When mouse is moved over a new element
b) When mouse is clicked

c) When mouse is both moved and clicked
d) When mouse is released

Answer: a
Explanation: When the user moves the mouse so that it goes over a new element, the browser fires a mouseover event on that element. The onmouseover event occurs when the mouse pointer is moved onto an element, or onto one of its children.

6. When is the mouseout event fired?
a) When mouse is no longer over an element
b) When mouse is over an element
c) When mouse is hovered
d) When mouse is clicked

Answer: a
Explanation: When the mouse moves so that it is no longer over an element, the browser fires a mouseout event on that element. The mouseout() method triggers the mouseout event, or attaches a function to run when a mouseout event occurs

7. The focus and blur events are also part of _________
a) Element events
b) Handler events
c) Window events
d) Scroll events

Answer: c
Explanation: The focus and blur events are also used as Window events: they are triggered on a window when that browser window receives or loses keyboard focus from the operating system. Focusin and onblur methods are used for using these events.

8. The element that can also register handlers for load and error events is _________
a) html
b) img
c) body
d) form

Answer: b
Explanation: Individual document elements, such as img elements, can also register handlers for load and error events.onload and onerror methods are used for handling such events.

9. The events that are emulated by the jQuery library are _________
a) focusarea and focusover
b) focusall and focusnone
c) focusdown and focusup
d) focusin and focusout

Answer: b
Explanation: The jQuery library emulates focusin and focusout events for browsers that do not support them. The focusin and focusout events bubble, the focus and blur events doesn't. That means that you can use the focusin and focusout on the parent element of a form field.

10. Which event is triggered sooner when the document and its elements are ready to manipulate?
a) DOMContentLoaded
b) readystatechange
c) Both DOMContentLoaded & readystatechange
d) Statechange

Answer: c

*Explanation: **DOMContentLoaded** and **readystatechange** are alternatives to the load event: they are triggered sooner, when the document and its elements are ready to manipulate, but before external resources are fully loaded.*

1. DOM Level 3 Events standardizes which of the following events?
a) focusarea and focusover
b) focusall and focusnone
c) focusdown and focusup
d) focusin and focusout

Answer: d
Explanation: The DOM Level 3 Events specification standardizes the focusin and focusout events as bubbling alternatives to the focus and blur events. The focusin and focusout events bubble, the focus and blur events doesn't. That means that you can use the focusin and focusout on the parent element of a form field.

2. Which of the following are the necessary events currently?
a) DOMActivate
b) DOMFocusIn
c) DOMNodeInserted
d) Onclick

Answer: d
Explanation: Browsers are still allowed to generate events like DOMActivate, DOMFocusIn, and DOMNodeInserted, but these are no longer required. For example DOM activate occurs when any element becomes activate.

3. Which object is passed as the argument to handlers for keydown, keyup, and keypress events?
a) KeyboardEvent
b) Key Event
c) Mouse Event
d) Alphabet Event

Answer: a
Explanation: What is new in the DOM Level 3 Events specification is standardized support for two dimensional mouse wheels via the wheel event and better support for text input events with a new KeyboardEvent object that is passed as the argument to handlers for keydown, keyup, and keypress events.

4. Which among these is a property that reports rotation of mouse wheel axes?
a) ctrlKey
b) alterX
c) alterY
d) deltaX

Answer: d
*Explanation: A handler for a wheel event receives an event object with all the usual mouse event properties, and also **deltaX**, **deltaY**, and **deltaZ** properties that report rotation around three different mouse wheel axes.*

5. Which of the following property specifies the string of text that was entered?
a) message
b) data
c) string
d) text

Answer: b
*Explanation: A textinput event handler has a **data** property that specifies the string of text that was entered. This data property is used for manipulating the data which is entered by the user.*

6. Which of the following is defined by the specification?
a) dataMethod
b) input

c) inputMethod
d) inputdataMethod

Answer: c
*Explanation: The specification defines an **inputMethod** property on the event object and a set of constants representing different kinds of text input (keyboard, paste or drop, handwriting or voice recognition, and so on). prompt function is a kind of input method.*

7. Which two events will have the generated text for key events?
a) key and char
b) char and text
c) text and key
d) key and value

Answer: a
*Explanation: For key events that generate printable characters, **key** and **char** will be equal to the generated text. Both key and char are onkeypress event which occurs when the user presses a key (on the keyboard).*

8. Which of the following are the drag and drop events?
a) drop
b) dragstart
c) both drop and dragstart
d) dropover

Answer: c
Explanation: HTML Drag and Drop interfaces enable applications to use drag and drop features in Firefox and other browsers. The API defines the following seven event types :

1. *dragstart*
2. *drag*
3. *dragend*
4. *dragenter*
5. *dragover*
6. *dragleave*
7. *drop*

9. Which property holds a DataTransfer object that contains information about the data being transferred and the formats in which it is available?
a) dataTransfer
b) transferData
c) dataExchange
d) exchangeData

Answer: a
*Explanation: The property, **dataTransfer**, holds a DataTransfer object that contains information about the data being transferred and the formats in which it is available. The DataTransfer object is used to hold the data that is being dragged during a drag and drop operation.*

10. Which API allows scripts in a document from one server to exchange messages with scripts?
a) Cross-Document Messaging API
b) Web application API
c) Both Cross-Document Messaging API & Web application API
d) Cross-linking API

Answer: a
Explanation: The Cross-Document Messaging API allows scripts in a document from one server to exchange messages with scripts in a document from another server. This works around the limitations of the same-origin policy in a secure way. Each message that is sent triggers a message event on the Window of the receiving document.

1. When are the keyboard events fired?
a) When the user manually calls the button
b) When the user clicks a key
c) When the user calls the modifier
d) When the user right clicks the mouse

Answer: b
Explanation: The keydown and keyup are the keyboard events are fired when the user presses or releases a key on the keyboard. They are generated for modifier keys, function keys, and alphanumeric keys.

2. How does a user generate multiple keydown events?
a) Repeating the same process
b) Pressing multiple keys
c) Pressing the key longer than usual
d) Pressing the key multiple times

Answer: c
Explanation: If the user holds the key down long enough for it to begin repeating, there will be multiple keydown events before the keyup event arrives. Pressing the key for long time results in multiple calls to the function onkeypress.

3. Which property is used to specify the key type when pressed?
a) keyCode
b) keyType
c) keyName
d) keyProperty

Answer: a
Explanation: The keyCode property returns the Unicode character code of the key that triggered the onkeypress event, or the Unicode key code of the key that triggered the onkeydown or onkeyup event. The event object associated with these events has a numeric keyCode property that specifies which key was pressed.

4. For what value does the keyCode property persists even when a Shift key is pressed for adding punctuation character.
a) Special characters
b) Alphabets
c) Alphanumeric
d) Digits

Answer: d
*Explanation: The number keys always generate **keyCode** values for the digit that appears on the key, even if you are holding down Shift in order to type a punctuation character.*

5. Which of the following are not key event properties?
a) Code key
b) Alt Key
c) Ctrl Key
d) Shift Key

Answer: a
*Explanation: **altKey**, **ctrlKeY**, **shiftKey**, and **metaKey** are key event object's properties, which are set to true if the corresponding modifier key is held down when the event occurs. The keyCode values of ShiftKey, ctrlKey, altKey are respectively 16, 17 and 18.*

6. Which of the following key property holds the key name as a string?
a) keyName
b) key
c) keyName(string)
d) Nameofkey(string)

Answer: b
Explanation: The DOM Level 3 Events defines a new key property that contains the key name as a string. Object.keys() returns an array whose elements are strings corresponding to the enumerable properties found directly upon object.

7. Which of the following is not the value the key property will hold if the key is a function key?
a) F8
b) F2
c) F9
d) Left

Answer: d
Explanation: If the key is a function key, the key property will be a value like "F2", "F8" or "F9"etc. The Keys from F1-F12 are known as function keys.

8. Which method is used to add a binding?
a) binding()
b) add_bind()
c) bind()
d) addbind()

Answer: c
Explanation: 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.

9. Which method is used to remove a binding?
a) Unbind()
b) removebind()
c) remove(Bind b)
d) unbind()

Answer: d
Explanation: The ***unbind()*** *method removes event handlers from selected elements. This method can remove all or selected event handlers, or stop specified functions from running when the event occurs.*

10. Which of the following are the parameters passed to the keymap after installation?
a) Key property
b) Key name
c) Keydown event's object
d) Key stroke

Answer: c
Explanation: After the keymap is installed, the following parameters are passed:

1. *The event object for the keydown event.*
2. *The key identifier of the key that was pressed.*

Identifier key contains the information of the key that is pressed and the object information about the function to be performed when the key is pressed.

1. How many node types are there in total?
a) 11
b) 12
c) 13
d) 14

Answer: b
Explanation: There are total of 12 node types. The nodeType property returns the node type, as a number, of the specified node.

2. What is the purpose of the Node object property ownerDocument?
a) Returns the root element
b) Returns the last element
c) Returns the parent node
d) Returns the immediate node

Answer: a
*Explanation: The **ownerDocument** property returns the owner document of a node, as a Document object. It returns the root element for a node.*

3. Which of the following Node object property returns the local part of the name of a node?
a) lastName
b) localName
c) firstName
d) objectname

Answer: b
*Explanation: The Node object property **localName** Returns the local part of the name of a node. If the selected node is not an element or attribute, this property returns NULL.*

4. What is the property textContent?
a) Sets the textual content of a node
b) Returns the textual content of a node
c) Sets & Returns the textual content of a node
d) Modifies texual content

Answer: c
*Explanation: The property **textContent** sets or returns the textual content of a node and its descendants. If you set the textContent property, any child nodes are removed and replaced by a single Text node containing the specified string.*

5. How many Node object methods are available?
a) 18
b) 19
c) 20
d) 21

Answer: a
Explanation: The Node object represents a single node in the document tree. There are totally 18 node object methods.

6. Which of the following Node object property returns the node immediately before a node?
a) previousSibling
b) textContent
c) index
d) localName

Answer: a
*Explanation: The node object property **previousSibling** returns the node immediately before a node. The returned node is returned as a Node object.*

7. What is the purpose of the method getUserData(key)?
a) Returns the associated object
b) Gets the user data
c) Returns the user data
d) Gets the user key

Answer: a
*Explanation: The method **getUserData(key)** returns the object associated to a key on a this node. The object must first have been set to this node by calling setUserData with the same key.*

8. How to test if two nodes are equal?
a) isEqualNode()
b) equal()
c) ==
d) equalto()

Answer: a
*Explanation: The method **isEqualNode()** is used to test if two nodes are equal. Two nodes are equal when they have the same type, defining characteristics (for elements, this would be their ID, number of children, and so forth), its attributes match, and so on.*

9. How to associate an object to a key on a node?
a) getUserData()
b) cloneNode()
c) setUserData(key,data,handler)
d) clonedata()

Answer: c
Explanation: The Node.setUserData() method allows a user to attach (or remove) data to an element, without needing to modify the DOM. This method is used to associate an object to a key on a node.

10. Which method is used to compare the placement of two nodes in the DOM hierarchy (document)?
a) compareDocumentPosition()
b) cloneNode()
c) getUserData()
d) getFeature()

Answer: a
Explanation: The compareDocumentPosition() method is used to compare the placement of two nodes in the DOM hierarchy (document). The return value is an integer value whose bits represent the calling Node's relationship to otherNode within the Document.

1. What does a Node object represent?
a) Single node
b) Set of nodes
c) Sequence of nodes
d) Node array

Answer: a
Explanation: The Node object represents a single node in the document tree. A node can be an element node, an attribute node, a text node, or any other of the node types explained in the Node Types chapter.

2. What does the nodeName of the nodeType Document return?
a) doctype name
b) target
c) #comment
d) #document

Answer: d
*Explanation: The nodeName of the nodeType **Document** returns **#document**. If the node is an element node, the nodeName property will return the tag name. If the node is an attribute node, the nodeName property will return the name of the attribute.*

3. What is the purpose of the method item()?
a) Returns node after the specified index
b) Returns node before the specified index
c) Returns node at specified index
d) Returns the node following the specified node

Answer: c
*Explanation: The method **item()** returns the node at the specified index in a node list. The nodes are sorted as they appear in the source code, and the index starts at 0.A Node object's collection of child nodes is an example of a NodeList object.*

4. How can the nodes in the node list be accessed?
a) Key
b) Index number
c) Looping
d) Value

Answer: b
Explanation: The nodes in the node list can be accessed through their index number. The nodes are sorted as they appear in the source code, and the index starts at 0.

5. Which of the following is the child(s) of the node type EntityReference?
a) Element
b) Text
c) Both Element and Text
d) Entity

Answer: c
*Explanation: The **createEntityReference()** method creates the specified EntityReference Object. The children of the node type **EntityReference** are **Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference**.*

6. Which node type represents the root-node of the DOM tree?
a) Document
b) DocumentFragment
c) DocumentType
d) Entity

Answer: a
Explanation: Everything inside an html document is classified in the form of different nodes. The node type Document represents the root-node of the DOM tree, the entire document.

7. What is the purpose of the DocumentFragment node type?
a) To hold a portion of a document
b) To split the document into fragments
c) To hold the entire document
d) To hold the fragments

Answer: a
*Explanation: The **DocumentFragment** node type represents a "lightweight" Document object that has no parent. It is used as a lightweight version of Document that stores a segment of a document structure comprised of nodes just like a standard document.*

8. How many nodetype – named constants are available?
a) 13
b) 11
c) 12
d) 10

Answer: c
Explanation: DOM nodes can be of various types, such as element nodes and text nodes, and each node has a nodeType property giving its type. There are totally 12 nodetype – named constants available.

9. Which of the following Node types have a node value equal to null?
a) Document

b) DocumentFragment
c) DocumentType
d) All of the mentioned

Answer: d
Explanation: All the three node types namely, ***Document****,* ***DocumentFragment****,* ***DocumentType*** *have a node value equal to null. If the node is an element node, the nodeType property will return 1. If the node is an attribute node, the nodeType property will return 2.*

10. How many node object properties are there?
a) 12
b) 14
c) 16
d) 17

Answer: c
Explanation: Node interface is the primary datatype for the entire Document Object Model. The node is used to represent a single XML element in the entire document tree. There are totally 16 node object properties.

1. Cookies were originally designed for __________
a) Client-side programming
b) Server-side programming
c) Both Client-side & Server-side programming
d) Web programming

Answer: b
Explanation: Cookies are data, stored in small text files, on your computer. Cookies were originally designed for server-side programming, and at the lowest level, they are implemented as an extension to the HTTP protocol.

2. The Cookie manipulation is done using which property?
a) cookie
b) cookies
c) manipulate
d) modify

Answer: a
Explanation: The cookie property sets or returns all name/value pairs of cookies in the current document. There are no methods involved: cookies are queried, set, and deleted by reading and writing the cookie property of the Document object using specially formatted strings.

3. Which of the following explains Cookies nature?
a) Non Volatile
b) Volatile
c) Intransient
d) Transient

Answer: d
Explanation: Cookies are transient by default; the values they store last for the duration of the web browser session but are lost when the user exits the browser. When a web server has sent a web page to a browser, the connection is shut down, and the server forgets everything about the user.

4. Which attribute is used to extend the lifetime of a cookie?
a) higher-age
b) increase-age
c) max-age
d) lifetime

Answer: c

Explanation: If you want a cookie to last beyond a single browsing session, you must tell the browser how long (in seconds) you would like it to retain the cookie by specifying a max-age attribute. A number of seconds until the cookie expires. A zero or negative number will expire the cookie immediately.

5. Which of the following defines the Cookie visibility?
a) domain Path
b) local & session storage
c) server storage
d) transient Path

Answer: b
*Explanation: sessionStorage, localStorage and Cookies all are used to store data on the client-side. Each one has its own storage and expiration limit. Cookie visibility is scoped by document origin as **localStorage** and **sessionStorage** are, and also by document path.*

6. Which of the following can be used to configure the scope of the Cookie visibility?
a) path
b) domain
c) both path and domain
d) server

Answer: c
Explanation: The Cookie visibility scope is configurable through cookie attributes path and domain. Domain attribute in the cookie is used to specify the domain for which the cookie is sent.path includes the Path attribute in the cookie to specify the path for which this cookie is sent.

7. How can you set a Cookie visibility scope to localStorage?
a) /
b) %
c) *
d) //

Answer: a
Explanation: Setting the path of a cookie to "/" gives scoping like that of localStorage and also specifies that the browser must transmit the cookie name and value to the server whenever it requests any web page on the site.

8. Which of the following is a boolean cookie attribute?
a) bool
b) secure
c) lookup
d) domain

Answer: b
Explanation: The final cookie attribute is a boolean attribute named secure that specifies how cookie values are transmitted over the network. By default, cookies are insecure, which means that they are transmitted over a normal, insecure HTTP connection. If a cookie is marked secure, however, it is transmitted only when the browser and server are connected via HTTPS or another secure protocol.

9. Which of the following function is used as a consequence of not including semicolons, commas or whitespace in the Cookie value?
a) encodeURIComponent()
b) encodeURI()
c) encodeComponent()
d) encode()

Answer: a
*Explanation: Cookie values cannot include semicolons, commas, or whitespace. For this reason, you may want to use the core JavaScript global function **encodeURIComponent()** to encode the value before storing it in the cookie.*

10. What is the constraint on the data per cookie?
a) 2 KB
b) 1 KB
c) 4 KB
d) 3 KB

Answer: c
Explanation: Each cookie can hold up to only 4 KB. In practice, browsers allow many more than 300 cookies total, but the 4 KB size limit may still be enforced by some. Whereas storage of session is around a minimum of 5mb.

1. Which property helps to initiate the HTTP requests?
a) request
b) location
c) send
d) write

Answer: b
*Explanation: It is possible for JavaScript code to script HTTP, however. HTTP requests are initiated when a script sets the **location** property of a window object or calls the **submit()** method of a form object. The location object is part of the window object and is accessed through the window.location property.*

2. Which method is an alternative of the property location of a window object?
a) submit()
b) locate()
c) load()
d) write()

Answer: a
*Explanation: HTTP requests are initiated when a script sets the **location** property of a window object or calls the **submit()** method of a form object. In both cases, the browser loads a new page.*

3. Which of the following uses scripted HTTP?
a) XML
b) HTML
c) Ajax
d) CSS

Answer: c
Explanation: AJAX stands for Asynchronous JavaScript And XML. The key feature of an Ajax application is that it uses scripted HTTP to initiate data exchange with a web server without causing pages to reload.

4. Which of the below is a liberal reverse of Ajax?
a) HTTP
b) HTML
c) XML
d) Comet

Answer: d
Explanation: Comet is the reverse of Ajax: in Comet, it is the web server that initiates the communication, asynchronously sending messages to the client. The big advantage of Comet is that each client always has a communication link open to the server.

5. The other name for Comet is _____________
a) Server Push
b) Ajax Push
c) HTTP Streaming
d) All of the mentioned

Answer: d
Explanation: Comet is a web application model where a request is sent to the server and kept alive for a long time, until a time-out or a server event occurs. Other names for Comet include “Server Push”, “Ajax Push”, “HTTP Streaming”.

6. Which is the element that has a *src* property to initiate HTTP GET request?
a) img
b) iframe
c) script
d) both img and script

Answer: d
*Explanation: Both **img** and **script** contains the **src** property that can be set to initiate an HTTP GET request. The src property sets or returns the value of the src attribute of an image.The required src attribute specifies the URL of an image.*

7. XMLHttpRequest is a ____________
a) Object
b) Class
c) Both Object and Class
d) Array

Answer: c
*Explanation: **XMLHttpRequest** is both an object and a class. The XMLHttpRequest object can be used to request data from a web server.*

8. Which of the following are the features of an HTTP request?
a) URL being requested
b) Optional request body
c) Optional set of request headers
d) All of the mentioned

Answer: d
Explanation: An HTTP request consists of four parts :

1. *the HTTP request method or “verb”*
2. *the URL being requested*
3. *an optional set of request headers, which may include authentication information*
4. *an optional request body*

9. Which of the following is a feature of the HTTP response?
a) Mandatory response body
b) Optional response body
c) URL being released
d) Optional set of response headers

Answer: a
Explanation: The HTTP response sent by a server has three parts :

1. *a numeric and textual status code that indicates the success or failure of the request*
2. *a set of response headers*
3. *the response body*

10. Which is the appropriate code to begin a HTTP GET request?
a) request.open(“GET”,”data”);
b) request.open(GET,”data.csv”);
c) request.open(“GET”,”data.csv”);
d) request.open(“GET”);

Answer: c
Explanation: The code that begins a HTTP GET request for the contents of the specified URL is

```
request.open("GET","data.csv");
```

To send a request to a server open() and send() methods of the XMLHttpRequest object are used.

1. Which of the following is not the feature of jQuery?
a) Efficient query method for finding the set of document elements
b) Expressive syntax for referring to elements in the document
c) Useful set of methods for manipulating selected elements
d) Powerful functional programming techniques is not used for operating on sets of elements as a group

Answer: d
Explanation: These features are at the heart of jQuery's power and utility:

1. *An expressive syntax (CSS selectors) for referring to elements in the document*
2. *An efficient query method for finding the set of document elements that match a CSS selector*
3. *A useful set of methods for manipulating selected elements*
4. *Powerful functional programming techniques for operating on sets of elements as a group, rather than one at a time*
5. *A succinct idiom (method chaining) for expressing sequences of operations.*

2. Which of the following is a single global function defined in the jQuery library?
a) jQuery()
b) $()
c) Queryanalysis()
d) global()

Answer: a
Explanation: The jQuery library defines a single global function named jQuery(). This function is so frequently used that the library also defines the global symbol $ as a shortcut for it. The $ sign it's just an alias to jQuery(), then an alias to a function which is used as a selector element.

3. Which of the following is a factory function?
a) $()
b) jQuery()
c) Queryanalysis()
d) onclick()

Answer: b
*Explanation: **jQuery()** is a factory function rather than a constructor: it returns a newly created object but is not used with the **new** keyword. jQuery objects define many methods for operating on the sets of elements they represent.*

b)

```
var divs = $(div);
```

c)

```
var divs = jQuery("div");
```

d)

```
var divs = $("div");
```

5. Which is the method that operates on the return value of $()?
a) show()
b) css()
c) click()

d) done()

Answer: b
Explanation: The css() method sets or returns one or more style properties for the selected elements. The css() method operates on the jQuery object returned by $(), and returns that same object, so that the show() method can be invoked next in a compact "method chain."

6. What does the min mean in the following JavaScript code?

```
var divs = #("div");
```

a) Minimised version
b) Miniature
c) Minimised parameters
d) Minimum value

Answer: a
*Explanation: The **min** means the minimised version of the library, with unnecessary comments and whitespace removed, and internal identifiers replaced with shorter ones. The file size of minfile is smaller than the original file hence it makes it easier to load.*

7. Which of the following is a heavily overloaded function?
a) jQuery()
b) $()
c) script()
d) Both jQuery() and $()

Answer: d
Explanation: $() is just an alias function of jquery(). The jQuery() function (a.k.a) $()) is the most important one in the jQuery library. It is heavily overloaded, however, and there are four different ways you can invoke it.

8. Which of the following is an equivalent replacement of $(document).ready(f)?
a) jQuery(f)
b) $(f)
c) #(f)
d) read(f)

Answer: b
*Explanation: The equivalent replacement of **$(document).ready(f)** is **$(f)**. Writing $(document) performs the function of selecting the whole document which is the same as writing $() only.*

9. Which of the following is a utility function in jQuery?
a) jQuery.each()
b) jQuery.parseJSON()
c) jQuery.noConflict()
d) jQuery.conflict()

Answer: c
*Explanation: **jQuery.noConflict()** is the utility function in jQuery. The noConflict() method releases the hold on the $ shortcut identifier, so that other scripts can use it.*

10. Which of the following is used for parsing JSON text?
a) jQuery.each()
b) jQuery.parseJSON()
c) jQuery.noConflict()
d) jQuery.conflict()

Answer: b

*Explanation: **jQuery.parseJSON()** is used for parsing JSON text. The function converts json to javascript object.*

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

```
<script src="jquery-1.4.2.min.js"></script>
```

a) Tables
b) tables
c) Undefined
d) Error

Asnwer: a
Explanation: The i modifier is used to perform case-insensitive matching. It is found in the regex library of Javascript.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "Tables";
   var patt1 = /tables/i;
   var result = str.match(patt1);
   document.getElementById("demo").innerHTML = result;
}
</script>
```

a) 8
b) 17
c) 14
d) 10

Answer: b
Explanation: The lastIndex property specifies the index at which to start the next match. This property only works if the "g" modifier is set.

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

```
<script>
var str = "The rain in Spain";
var patt1 = /ain/g;
document.write(patt1.lastIndex);
</script>
```

a) IS
b) is
c) Error
d) Undefined

Answer: b
Explanation: The m modifier is used to perform a multiline match. The m modifier treat beginning (^) and end ($) characters to match the beginning or end of each line of a string (delimited by \n or \r), rather than just the beginning or end of the string.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "\nIs th\nis it?";
   var patt1 = /^is/m;
   var result = str.match(patt1);
```

```
    document.getElementById("demo").innerHTML = result;
}
</script>
```

a) 4
b) 3
c) 1
d) 0

Answer: a
Explanation: The o+ quantifier matches any string that contains at least one o. It concatenates the total number of o in a string.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
    var str = "Hellooo World!";
    var patt1 = /o+/g;
    var result = str.match(patt1);
    var count=0;
    While(result[i]!=NULL)
    {
        If(result[I]==o)
        count++;
    }
    document.getElementById("demo").innerHTML = count;
}
</script>
```

a) true
b) false
c) error
d) undefined

Answer: a
Explanation: The n$ quantifier matches any string with n at the end of it. The above string has words ending with is therefore the output will be true.

1. Which is not a form of client-side storage?
a) Web Databases
b) FileSystem API
c) Offline Web Applications
d) Online Web Applications

Answer: d
Explanation: Client-side storage allows the creater to store data on the users system for faster loading of the website. The various forms of client-side storage are web databases, filesystem API, Offline web applications and cookies.

2. Which is the storage that allows the caching of web pages and their associated resources?
a) Web Databases
b) FileSystem API
c) Offline Web Applications
d) Cookies

Answer: c
Explanation: HTML5 defines an "Offline Web Applications" API that allows the caching of web pages and their associated resources (scripts, CSS files, images, and so on). This is client-side storage for web applications themselves rather than just their data, and it allows web apps to install themselves so that they are available even when there is no

connection to the Internet.

3. Which is Microsoft's own proprietary client-side storage?
a) IE User Data
b) Offline Web Applications
c) Cookies
d) Offline Apis

Answer: a
Explanation: Microsoft implements its own proprietary client-side storage mechanism, known as "userData," in IE5 and later. userData enables the storage of medium amounts of string data and can be used as an alternative to Web Storage in versions of IE before IE8. This makes loading of programs and software faster.

4. Which object supports Filesystem API?
a) Element
b) File
c) Window
d) DOM

Answer: b
Explanation: These objects can be obtained from the filesystem property on any file system entry. Some browsers offer additional APIs to create and manage file systems, such as Chrome's requestFileSystem() method.

5. Which is the most appropriate database for developers requiring a huge amount of data?
a) Database
b) Datawarehouse
c) Web databases
d) Access

Answer: c
Explanation: Developers who need to work with really huge amounts of data like to use databases, and the most recent browsers have started to integrate client-side database functionality into their browsers. Client data base helps in making the website faster and handling the data easier.

6. The localStorage and sessionStorage belongs to ___________
a) Window object
b) Element object
c) Hash object
d) DOM object

Answer: a
*Explanation: Browsers that implement the "Web Storage" draft specification define two properties on the Window object: **localStorage** and **sessionStorage**. Local storage and Session storage are the web storage objects. Session storage is destroyed once the user closes the browser whereas, Local storage stores data with no expiration date.*

7. What is the main difference between localStorage and sessionStorage?
a) Lifetime
b) Scope
c) Both Lifetime and Scope
d) Storage Location

Answer: c
*Explanation: The difference between **localStorage** and **sessionStorage** has to do with lifetime and scope: how long the data is saved for and who the data is accessible to. Session storage is destroyed once the user closes the browser whereas, Local storage stores data with no expiration date.*

8. What is the lifetime of the data stored through localStorage?
a) Permanent

b) Temporary
c) Both Permanent and Temporary at times
d) Cannot store

Answer: a
*Explanation: Data stored through **localStorage** is permanent. it does not expire and remains stored on the user's computer until a web app deletes it or the user asks the browser (through some browser-specific UI) to delete it. This data is stored on the client side server and is used for faster access of data.*

9. Which is the function used to retrieve a value?
a) get()
b) retrieve()
c) getItem()
d) retrieveItem()

Answer: c
*Explanation: To retrieve a value, pass the name to **getItem()**. The getItem() method of the Storage interface, when passed a key name, will return that key's value, or null if the key does not exist, in the given Storage object.*

10. Which is the function used to store a value?
a) setItem()
b) set()
c) storeItem()
d) store()

Answer: a
*Explanation: To store a value, pass the name and value to **setItem()**. The setItem() method of the Storage interface, when passed a key name and value, will add that key to the given Storage object, or update that key's value if it already exists.*

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

```
<p id="demo"></p>
<script>
var carName = "Volvo";
var carName;
document.getElementById("demo").innerHTML = carName;
</script>
```

a) Error
b) Undefined
c) Volvo
d) Garbage value

Answer: c
Explanation: A variable does not lose its value if it is re-declared. The Javascript variable will store the value and the output will be Volvo.

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

```
<p>The result of adding "5" + 2 + 3:</p>
<p id="demo"></p>
<script>
x = "5" + 2 + 3;
document.getElementById("demo").innerHTML = x;
</script>
```

a) 523
b) 10
c) 5
d) Error

Answer: a
Explanation: When a string and integer is added in Javascript then the resulting output is string. Javascript will type caste integer to string and would then concatenate to produce the output.

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

```
<p id="demo">code</p>
<script>
function myFunction()
{
   var text = document.getElementById("demo").innerHTML;
   document.getElementById("demo").innerHTML = text.toUpperCase();
}
</script>
```

a) Code
b) CODE
c) code
d) Error

Answer: b
Explanation: toUpperCase function is present in the string Library of Javascript. It converts the string to uppercase.

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

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Number(true);
</script>
```

a) 1
b) 0
c) true
d) undefined

Answer: a
Explanation: The Number() method converts variables to numbers. True value is converted to 1 and a false value is converted to 0.

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

```
<p id="demo2"></p>
<script>
Var arr = ["one", "two", "three",];
arr.shift();
document.getElementById("demo2").innerHTML = arr;
</script>
```

a) two three
b) one two
c) one three
d) error

Answer: a
Explanation: The shift() method removes the first element of an array. It "shifts" all other elements to the left.

1. The main purpose of JavaScript in web browser is to ___________
a) Creating animations and other visual effects
b) User Interface
c) Visual effects
d) User experience

Answer: a
Explanation: JavaScript can help to facilitate that experience, for example by:

1. *Creating animations and other visual effects to subtly guide a user and help with page navigation*
2. *Sorting the columns of a table to make it easier for a user to find what she needs*

2. A JavaScript program can traverse and manipulate document content through __________
a) Element Object
b) Document Object
c) Both Element and Document Object
d) Data object

Answer: c
Explanation: A JavaScript program can traverse and manipulate document content through the Document object and the Element objects it contains. It can alter the presentation of that content by scripting CSS styles and classes. The Element object represents an HTML element, like P, DIV, A, TABLE, or any other HTML element.

3. The behaviour of the document elements can be defined by __________
a) Using document object
b) Registering appropriate event handlers
c) Using element object
d) Using data element

Answer: b
Explanation: The JavaScript program can define the behavior of document elements by registering appropriate event handlers. A JavaScript program can traverse and manipulate document content through the Document object and the Element objects it contains.

4. The service(s) that enables networking through scripted HTTP requests is __________
a) XMLHttpResponse
b) XMLRequest
c) XMLHttpRequest
d) XMLHttps

Answer: c
Explanation: The best known advanced services is the XMLHttpRequest object, which enables networking through scripted HTTP requests. The XMLHttpRequest object can be used to request data from a web server.

5. The HTML5 specification does not includes __________
a) Data storage
b) Graphics APIs
c) Other APIs for web apps
d) Networking

Answer: d
Explanation: The HTML5 specification (which, at the time of this writing, is still in draft form) and related specifications are defining a number of other important APIs for web apps. These include data storage and graphics APIs.The data storage api can store data locally within the user's browser.

6. Which of the following is not an advanced services?
a) Data storage
b) Networking
c) XMLHttpRequest object
d) Graphics APIs

Answer: d
Explanation: Data storage is used to store data locally on user's computer and networking is used for connecting between different platforms.

7. JavaScript code between a pair of "script" tags are called __________
a) Non-inline
b) External
c) Referenced
d) Inline

Answer: d
Explanation: The <script> tag is used to define a client-side script (JavaScript). The <script> element either contains scripting statements, or it points to an external script file through the src attribute. Inline code are those that are written between a pair of "script" tags.

8. Client-side JavaScript code is embedded within HTML documents in __________
a) A URL that uses the special javascript:encoding
b) A URL that uses the special javascript:stack
c) A URL that uses the special javascript:protocol
d) A URL that uses the special javascript:code

Answer: c
Explanation: The Client-side JavaScript code is embedded within HTML documents in four ways :

1. *Inline, between a pair of "script" tags*
2. *From an external file specified by the src attribute of a "script" tag*
3. *In an HTML event handler attribute, such as onclick or onmouseover*
4. *In a URL that uses the special javascript: protocol.*

9. What is the programming philosophy that argues that content and behaviour should as much as possible be kept separate?
a) Unobtrusive JavaScript
b) Obtrusive JavaScript
c) Inherited JavaScript
d) Modular JavaScript

Answer: a
Explanation: A programming philosophy known as unobtrusive JavaScript argues that content (HTML) and behavior (JavaScript code) should as much as possible be kept separate. According to this programming philosophy, JavaScript is best embedded in HTML documents using "script" elements with src attributes.

10. Which of the following communicates with server-side CGI scripts through HTML form submissions and can be written without the use of JavaScript?
a) Static Web Pages
b) Interactive Web Pages
c) Conditional Web Pages
d) All web pages

Answer: b
Explanation: An interactive web page can dynamically vary its content based on user preferences. Interactive web pages that communicate with server-side CGI scripts through HTML form submissions were the original "web application" and can be written without the use of JavaScript.

1. What is the advantage of the code produced graphics being smaller than the images themselves?
a) Bandwidth saving
b) Increase in bandwidth
c) Dynamic advantages
d) Static advantage

Answer: a
Explanation: The code used to produce graphics on the client side is typically much smaller than the images themselves, creating substantial bandwidth savings.

2. Which of the following uses a lot of CPU cycles?
a) GUI
b) Statically generated graphics
c) Dynamically generated graphics
d) Images

Answer: c
Explanation: Dynamic graphics for data, means simulating motion or movement using the computer. It may also be thought of as multiple plots linked by time. Dynamically generating graphics from real-time data uses a lot of CPU cycles.

3. Which HTML element is used to include images?
a) image
b) img
c) src
d) sourcing

Answer: b
Explanation: Web pages include images using the HTML img element. src tag is used to include the image link.

4. What is the purpose of image replacement?
a) To replace an image
b) To implement special effects
c) Removal of image rollovers
d) Implementation of image rollovers

Answer: d
Explanation: Image replacement is a technique developed to allow designers to use image-based typesetting while meeting accessibility requirements. One common use for image replacement is to implement image rollovers, in which an image changes when the mouse pointer moves over it.

5. When is JavaScript called *obtrusive*?
a) JavaScript code is medium sized
b) JavaScript code is small
c) JavaScript code is so large
d) JavaScript code is Very small

Answer: c
Explanation: When the amount of JavaScript code is so large that it effectively obscures the HTML, we call JavaScript as obtrusive. On the other hand unobtrusive JavaScript is a best practice methodology for attaching JavaScript to the front-end of a website.

6. Which is a possible way of finding all the img elements in the document?
a) document(images)
b) document.images[]
c) document(img)
d) doc(img)

Answer: b
Explanation: The best suited option is document.image[] to find all img elements in the document.[index] is used to specify the index of which img tag is to be selected.

7. Which of the following elements are used to include audio?
a) audio
b) video
c) svg
d) aud

Answer: a
Explanation: The audio tag is used to include audio in the HTML document. The audio tag includes method like play(), pause() etc.

8. Which of the following attributes are common to both *audio* and *video*?
a) enter
b) control
c) controls
d) add

Answer: c
*Explanation: Both audio and video support a **controls** attribute. When present, it specifies that audio controls should be displayed. Both audio and video support a controls attribute.*

9. Which of the following is not the property of the *video* tag?
a) width
b) height
c) breadth
d) area

Answer: c
*Explanation: The video tag does not contain a **breadth** property. The width and height property specifies the width and height of the video.*

10. Which of the following is the parameter used to invoke the Audio() constructor?
a) File type
b) Music type
c) Both File and Music
d) Video type

Answer: c
*Explanation: The HTMLAudioElement interface provides access to the properties of <audio> elements, as well as methods to manipulate them. It derives from the HTMLMediaElement interface. The parameter type of the **Audio()** constructor is any file type that contains audio to be played.*

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var num = 098;
   var n = num.valueOf()
   document.getElementById("demo").innerHTML = n;
}
</script>
```

a) 098
b) 98
c) Error
d) Undefined

Answer: b
Explanation: The valueOf() method returns the primitive value of a number. The value of num in the above case would be 98.

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

```
<p id="demo"></p>
<script>
```

```
function myFunction()
{
   var num = 3+2;
   var n = num.valueOf()
   document.getElementById("demo").innerHTML = n;
}
</script>
```

a) 5
b) 3+2
c) Error
d) Undefined

Answer: a
Explanation: The valueOf() method returns the primitive value of a number. It performs the calculations in the number and then displays the result.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var num = 13.3714;
   document.getElementById("demo").innerHTML = num.toPrecision(3);
}
</script>
```

a) 13
b) 13.3714
c) 13.3
d) 13.4

Answer: d
Explanation: The toPrecision() method formats a number to a specified length. A decimal point and nulls are added (if needed), to create the specified length.

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

```
<script>
var num = new Number(1000000).toLocaleString("fi-FI");
document.write(num);
</script>
```

a) 1 000 000
b) 1 0 00000
c) 100 000 0
d) Undefined

Answer: a
Explanation: This method formats a number into a string, using language specific format. In this example we use the "fi-FI" value to specify the locale number format in FINLAND.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   document.getElementById("demo").innerHTML = Number.POSITIVE_INFINITY;
}
</script>
```

a) 10000
b) -infinity
c) infinity
d) error

Answer: c
Explanation: The POSITIVE_INFINITY property represents positive infinity. Positive infinity can be explained as something that is higher than any other number.

1. What is the purpose of the *canvas* element?
a) Creates drawing surface
b) Exposes powerful drawing API to client-side JavaScript
c) Creates drawing surface & Exposes powerful drawing API to client-side JavaScript
d) Creates a rectangular box

Answer: c
Explanation: The HTML canvas element is used to draw graphics, on the fly, via JavaScript. The canvas element is only a container for graphics. The canvas element has no appearance of its own but creates a drawing surface within the document and exposes a powerful drawing API to client-side JavaScript.

2. From which version of IE is *canvas* supported?
a) 7
b) 8
c) 9
d) 6

Answer: c
Explanation: The canvas element is not supported by IE before IE9, but it can be reasonably well emulated in IE6, 7, and 8. It is supported from version 4 in chrome and 2 in firefox.

3. Which method is used to obtain the "drawing context" object?
a) getContext()
b) getObject()
c) get()
d) getDrawing()

Answer: a
Explanation: The HTMLCanvasElement.getContext() method returns a drawing context on the canvas, or null if the context identifier is not supported. Most of the Canvas drawing API is defined not on the canvas element itself, but instead on a "drawing context" object obtained with the getContext() method of the canvas.

4. What is the returning value of the getContext() method?
a) Drawing model
b) CanvasRenderingContext2D object
c) Context2D object
d) Container

Answer: b
Explanation: Call getContext() with the argument "2d" to obtain a CanvasRenderingContext2D object that you can use to draw two-dimensional graphics into the canvas. It is important to understand that the canvas element and its context object are two very different objects.

5. How does SVG describe complex shapes?
a) Path of lines
b) Path of curves
c) Path of lines and curves
d) Planes

Answer: c
Explanation: SVG stands for 'Scalable Vector Graphics' and it is used to define graphics for the Web. SVG is mostly used for vector type diagrams like Two-dimensional graphs in an X, Y coordinate system, Pie charts etc. SVG describes complex shapes as a "path" of lines and curves that can be drawn or filled.

6. Which is the method invoked to begin a path?
a) begin()
b) path()
c) createPath()
d) beginPath()

Answer: d
*Explanation: The **beginPath()** method begins a path, or resets the current path. It is called before creating any path or curve.*

7. Which is the method invoked to connect the last vertex back to the first?
a) closePath()
b) close()
c) connectlast(first)
d) connect()

Answer: a
*Explanation: The **closePath()** method connects the last vertex back to the first, thereby creating a path. It connects the last point to the first point thereby creating a closed figure.*

8. Which of the following are not the properties of a canvas object?
a) fillStyle
b) strokeStyle
c) lineWidth
d) lineSize

Answer: d
*Explanation: The fillStyle property sets or returns the color, gradient, or pattern used to fill the drawing. The strokeStyle property sets or returns the color, gradient, or pattern used for strokes. There is no property called **lineSize** associated with the canvas object.*

9. Which of the following is a property used to check how crisp or fuzzy shadows are?
a) shadowColor
b) shadowBlur
c) strokeStyle
d) stroke

Answer: b
*Explanation: **shadowBlur** is used to check how crisp or fuzzy shadows are. The shadowBlur property sets or returns the blur level for shadows.*

10. How do you restore a saved coordinate system?
a) restore()
b) getback()
c) set()
d) back()

Answer: a
*Explanation: The saved coordinate system is restored by calling the method **restore()** associated with the canvas object. The restore() method reset the canvas by "popping" the last state saved to the stack.*

1. What is the purpose of a Rendering Engine?
a) Parsing objects in page

b) Drawing all objects in page
c) Both Parsing & Drawing all objects in page
d) Rendering object

Answer: c
Explanation: It's responsible for displaying the web page. The rendering engine parses the HTML and the CSS and displays the parsed content on the screen. A Rendering Engine is generally used for parsing and drawing all of the objects in the page.

2. What is the purpose of the JavaScript Engine?
a) Compiling the JavaScript
b) Interpreting the JavaScript
c) Both Compiling & Interpreting the JavaScript
d) Parsing the javascript

Answer: b
Explanation: The JavaScript Engine is generally used for interpreting the JavaScript. It is used to interpret the javascript and execute the javscript on the web page.

3. Which layer is used to handle the HTTP requests?
a) Network Layer
b) Transport Layer
c) Application Layer
d) Presentation Layer

Answer: a
Explanation: HTTP is a protocol which allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. The network layer is used to handle the HTTP requests.

4. The User Interface is taken care of which layer?
a) Transport Layer
b) Network Layer
c) UI Layer
d) Presentation Layer

Answer: c
Explanation: The user interface layer represents the front end of the Web Client, and contains the actual GUI elements that users view and click. The UI Layer takes care of the User Interface.

5. Which of the following browsers use Webkit?
a) Chrome
b) Internet Explorer
c) Safari
d) Both Chrome and Safari

Answer: d
Explanation: WebKit is a browser engine used in Apple's Safari browser and other products. Webkit is what Chrome and Safari use, and is your target for most mobile web development since it is used as the layout or rendering engine for Android devices as well as mobile Safari for iOS devices and the Silk browser on Kindle Fires.

6. Which of the following first developed Gecko?
a) Safari
b) Netscape
c) Opera
d) Internet Explorer

Answer: b

Explanation: Gecko is the name of the layout engine developed by the Mozilla Project. Gecko was first developed at Netscape, before the Mozilla Project spun out as its own entity, as the successor to the original Netscape rendering engine, back in 1997.

7. Which of the following render HTML?
a) Browsers
b) Email Clients
c) Web Components
d) All of the mentioned

Answer: d
Explanation: Rendering Engine takes HTML code and interprets it into what you see visually. More tools than just browsers render HTML, including email clients and web components in other applications.

8. SpiderMonkey was developed by _____________
a) Firefox
b) Internet Explorer
c) Safari
d) Opera

Answer: a
Explanation: SpiderMonkey is Mozilla's JavaScript engine written in C and C++. It is used in various Mozilla products, including Firefox, and is available under the MPL2. SpiderMonkey is the JavaScript engine made by Mozilla that is used in Firefox.

9. Carakan is used by which of the following browsers?
a) Firefox
b) Internet Explorer
c) Safari
d) Opera

Answer: d
Explanation: Opera uses ***Carakan****, which was introduced in 2010. Mozilla uses spidermonkey and safari uses nitro javascript engine.*

10. Which is the alternate name for JavaScriptCore that is used by Safari?
a) Nitro
b) SpiderMoney
c) Carakan
d) V8

Answer: a
Explanation: Safari uses ***JavaScriptCore****, sometimes called Nitro. Opera uses Carakan and Mozilla uses SpiderMonkey.*

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "have best";
   var patt = new RegExp("es");
   var res = patt.test(str);
   document.getElementById("demo").innerHTML = res;
}
</script>
```

a) true
b) false
c) error

d) undefined

Answer: a
Explanation: The test() method tests for a match in a string. This method returns true if it finds a match, otherwise it returns false.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var patt = new RegExp("World", "g");
   var res = patt.toString();
   document.getElementById("demo").innerHTML = res;
}
</script>
```

a) /World/g
b) World
c) World/g
d) Undefined

Answer: a
Explanation: The toString() method returns the string value of the regular expression. It is found in the regular expression library of Javascript.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "Give 100%!";
   var patt1 = /\d/g;
   var result = str.match(patt1);
   if(result)
        result=true;
   else
        result=false;
   document.getElementById("demo").innerHTML = result;
}
</script>
```

a) True
b) False
c) Error
d) Undefined

Answer: a
Explanation: The \d metacharacter is used to find a digit from 0-9. The above code results true if a digit is present in the string.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "Is this all there is?";
   var patt1 = /\s/g;
   var result = str.match(patt1);
   if(result)
        result=true;
   else
        result=false;
   document.getElementById("demo").innerHTML = result;
}
</script>
```

a) true
b) false
c) error
d) undefined

Answer: a
Explanation: The \s metacharacter is used to find a whitespace character. A whitespace character can be a space, a tab or a new line character.

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

```
<p id="demo"></p>
<script>
var str = "HELLO, LOOK AT YOU!";
var patt1 = /\bLO/;
var result = str.search(patt1);
document.getElementById("demo").innerHTML = result;
</script>
```

a) 4
b) 7
c) 3
d) 8

Answer: b
Explanation: The \b metacharacter is used to find a match at the beginning or end of a word. The above code returns the position of the occurrence of the pattern at the starting of the word.

1. What are the features of an interpreter?
a) Shared with other properties
b) Embedded in other tools
c) Shared with other properties & Embedded in other tools
d) Used with html

Answer: c
Explanation: A JavaScript engine is a program or an interpreter which executes JavaScript code. The interpreter can be shared with other properties, or embedded in other tools.

2. Why is the total size of the page important?
a) Time taken to download
b) Size of IP packet should be less than 65500
c) Size of IP packet should be less than 65535
d) Size of IP packet should be greater than 65500

Answer: c
Explanation: The total size of the page is important, not just because of the time it takes to download, but because the maximum size of an IP packet is 65535 octets for IPv4 and IPv6.

3. How will the HTTP GET request be sent from the browser?
a) Remote server
b) Local server
c) By itself
d) Standby Server

Answer: a
Explanation: The two most common methods of http are get and post. Once the TCP/IP connection has been established, the browser sends an HTTP GET request over the connection to the remote server. The remote server finds the resource and returns it in an HTTP Response, the status of which is 200 to indicate a good response.

4. What is the return type of the remote server?
a) HTTP Response
b) HTTP Request
c) Get Request
d) Post request

Answer: a
Explanation: The remote server finds the resource and returns it in an HTTP Response, the status of which is 200 to indicate a good response. 400 status code is used to indicate a bad request.

5. Which layer is used to control the communication between the hardware in the network?
a) Network Access Layer
b) Internet Layer
c) Transport Layer
d) Presentation Layer

Answer: a
*Explanation: The Network Access Layer is the lowest layer of the TCP/IP protocol hierarchy. The **Network Access layer** controls the communication between the hardware in the network.*

6. Which layer is used to handle the network addressing and routing?
a) Network Access Layer
b) Internet Layer
c) Transport Layer
d) Presentation Layer

Answer: b
*Explanation: The internet layer is a group of internetworking methods, protocols, and specifications in the Internet protocol suite that are used to transport network packets from the originating host across network boundaries. The **Internet layer** handles network addressing and routing, getting IP and MAC addresses.*

7. The layer in which the TCP (or UDP) communication takes place is _____________
a) Network Access Layer
b) Internet Layer
c) Transport Layer
d) Presentation Layer

Answer: c
*Explanation: The transport layer is the layer in the open system interconnection (OSI) model responsible for end-to-end communication over a network. The **Transport layer** is where our TCP (or UDP) communication takes place.*

8. Which layer handles top-level communication?
a) Network Access Layer
b) Internet Layer
c) Transport Layer
d) Application Layer

Answer: d
*Explanation: An application layer is an abstraction layer that specifies the shared communications protocols and interface methods used by hosts in a communications network. The **Application layer** handles the top-level communication that the client and servers use, like HTTP and SMTP for email clients.*

9. What are the various possessions in the three-way handshake by the TCP?

a) Synchronize
b) Synchronize-Acknowledge
c) Acknowledge message
d) All of the mentioned

Answer: d
Explanation: TCP stands for transmission control protocol which controls the transmission of data over the internet. This handshake consists of a ***Synchronize****,* ***Synchronize-Acknowledge****, and* ***Acknowledge message*** *to be passed between the browser and the remote server. This handshake allows the client to attempt communication, the server to acknowledge and accept the attempt, and the client to acknowledge that the attempt has been accepted.*

10. Arrange the TCP/IP model layers in the order from farthest to closest to the end user.
a) Network Access Layer, Internet Layer, Application Layer, Transport Layer
b) Network Access Layer, Transport Layer, Internet Layer, Application Layer
c) Network Access Layer, Internet Layer, Transport Layer, Application Layer
d) Network Access Layer, Application Layer, Internet Layer, Transport Layer

Answer: c
Explanation: TCP stands for Transmission Control Protocol. The TCP/IP model is a concise version of the OSI model. The four layers in the TCP/IP model are, in order from farthest to closest to the end user, ***the Network Access layer****,* ***the Internet layer****,* ***the Transport layer****, and* ***the Application layer****.*

1. How do we define the term Thread?
a) Device that controls input
b) Variable that controls movement
c) Controlled execution of applications
d) Device that controls input & Variable that controls movement

Answer: c
Explanation: Threads are sequential units of controlled execution for applications. A thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system.

2. What does the browser do to set up a TCP/IP connection?
a) TCP one-way handshake
b) TCP two-way handshake
c) TCP three-way handshake
d) TCP four-way handshake

Answer: c
Explanation: A three-way handshake is a method used in a TCP/IP network to create a connection between a local host/client and server. It is a three-step method that requires both the client and server to exchange SYN and ACK (acknowledgment) packets before actual data communication begins.

3. What does the handshake pass between the browser and the remote server?
a) Synchronize
b) Accept
c) Reject
d) Decline

Answer: a
Explanation: A three-way handshake is a method used in a TCP/IP network to create a connection between a local host/client and server. The handshake consists of a Synchronize, Synchronize-Acknowledge, and Acknowledge message to be passed between the browser and the remote server.

4. How does the handshake help the server?
a) Acknowledge
b) Accept the attempt

c) Both Acknowledge and Accept the attempt
d) Asynchronize

Answer: c
Explanation: The handshake allows the server to acknowledge and accept the attempt. The handshake consists of a Synchronize, Synchronize-Acknowledge, and Acknowledge message to be passed between the browser and the remote server.

5. What is the purpose of the transport layer?
a) TCP Communication takes place
b) UDP Communication takes place
c) Both TCP and UDP Communication takes place
d) IP communication

Answer: c
Explanation: The transport layer is the layer in the open system interconnection (OSI) model responsible for end-to-end communication over a network. The ***Transport layer*** *is where our TCP (or UDP) communication takes place.*

6. What does the status code 200 indicate?
a) Error in request
b) Error in response
c) Error in server
d) Successful

Answer: d
Explanation: The status code ***200*** *indicates a successful response from the server. A 400 status code represents a bad request.*

7. What does the status code 500 indicate?
a) Error in request
b) Error in response
c) Error in server
d) Successful

Answer: a
Explanation: The 500 status code, or Internal Server Error, means that server cannot process the request for an unknown reason. Sometimes this code will appear when more specific 5xx errors are more appropriate.

8. What does the application layer handle?
a) Top-level communication
b) Bottom-level communication
c) Both Top-level and Bottom-level communication
d) Middle-level communication

Answer: a
Explanation: An application layer is an abstraction layer that specifies the shared communications protocols and interface methods used by hosts in a communications network. The ***Application layer*** *handles the top-level communication that the client and servers use, like HTTP and SMTP for email clients.*

9. Which status code indicates that the server could not find the resource requested?
a) 200
b) 404
c) 500
d) 566

Answer: b
Explanation: The HTTP 404, 404 Not Found, and 404 error message is a Hypertext Transfer Protocol (HTTP) standard response code, in computer network communications, to indicate that the client was able to communicate with a given

server, but the server could not find what was requested.

10. What is the maximum size of an IP packet for IPv4 or IPv6?
a) 65540
b) 65535
c) 65577
d) 67544

Answer: b
Explanation: Internet Protocol version 4 (IPv4) is the fourth version of the Internet Protocol (IP). IPv6 is the most recent version of Internet Protocol. The maximum size of an IP packet for IPv4 or IPv6 is ***65535****.*

1. How can we define the term Performance?
a) Speed of the input takes in
b) Speed of the output display
c) Speed at which application functions
d) Speed of data transmission

Answer: c
Explanation: ***Performance*** *refers to the speed at which an application functions. It is a multifaceted aspect of quality. Better performance results in better user experience.*

2. When is an application said to show a web performance?
a) Time to respond
b) Time to load
c) Time to send a request
d) Time to receive data

Answer: b
Explanation: Web performance refers to the speed in which web pages are downloaded and displayed on the user's web browser. When we're talking about web applications, the time it takes your application to be presented to your users is what we will call web performance.

3. When is an application said to show a runtime performance?
a) Speed of response to user
b) Speed of user request
c) Speed of sending data
d) Speed of receiving data

Answer: a
Explanation: The speed at which your application responds to your users' interactions is what we'll call runtime performance. Better runtime performance results in better user experience.

4. What does the span of time waiting for the page to be useful depending on?
a) Runtime performance
b) Web performance
c) Speed
d) Runtime performance & Speed

Answer: b
Explanation: Web performance refers to the speed in which web pages are downloaded and displayed on the user's web browser. The span of time that you are waiting for the page to be usable depends on web performance.

5. What is the advantage of the code produced graphics being smaller than the images themselves?
a) Bandwidth saving
b) Increase in bandwidth
c) Dynamic advantages
d) Static advantages

Answer: a
Explanation: The code used to produce graphics on the client side is typically much smaller than the images themselves, creating substantial bandwidth savings. This also helps in increasing web performance.

6. In order to skip or seek to the desired location in a sound or video, which property becomes helpful?
a) audioSkip
b) currentTime
c) videoSkip
d) skiptoTime

Answer: b
Explanation: The currentTime property sets or returns the current position (in seconds) of the audio/video playback. In addition to starting and stopping sound and video, you can skip (or "seek") to the desired location within the media by setting the currentTime property.

7. Which of the following shows a better runtime performance for coalescing functionality, using functions, and using objects?
a) Firefox unwoundfun
b) Firefox UsingFunct
c) Firefox UsingObject
d) Firefox UsingStruct

Answer: b
Explanation: When we compare the runtime performance for coalescing functionality, using functions, and using objects, the Firefox UsingFunct shows better performance. UsingFunct improves runtime performance and inturn the user experience.

8. Which of the following shows a poorer runtime performance for coalescing functionality, using functions, and using objects?
a) Firefox unwoundfun
b) Firefox UsingFunct
c) Firefox UsingObject
d) Firefox UsingStruct

Answer: a
*Explanation: When we compare the runtime performance for coalescing functionality, using functions, and using objects, the **Firefox UsingFunct** shows a poorer performance. This results in poor user experience.*

9. In how many modes can the Closure compiler be run?
a) 2
b) 3
c) 4
d) 5

Answer: a
Explanation: Closure Compiler can be run in either of two modes:

- *In Simple mode it mostly performs like most other minifiers, removing whitespace, line breaks, and comments*
- *In Advanced mode it rewrites the JavaScript by renaming variables and functions from longer descriptive names to single letters to save file size, and it inlines functions, coalescing them into single functions wherever it determines that it can*

10. What is the purpose of the advanced mode in the Closure compiler?
a) Removing the variables and other parameters
b) Renaming the variables and other parameters
c) Slight alteration to improve the runtime performance
d) Calls the function

Answer: b
Explanation: In Advanced mode, the Closure Compiler rewrites the JavaScript by renaming variables and functions from longer descriptive names to single letters to save file size, and it inlines functions, coalescing them into single functions wherever it determines that it can.

1. Firebug is an extension of which browser?
a) Mozilla
b) Chrome
c) IE
d) Opera

Answer: a
Explanation: Firebug is a discontinued free and open-source web browser extension for Mozilla Firefox that facilitated the live debugging, editing, and monitoring of any website's CSS, HTML, DOM, XHR, and JavaScript.

2. Firebug can be used to inspect ___________
a) HTML
b) CSS
c) DOM
d) All of the mentioned

Answer: d
Explanation: Firebug is an extension for the Mozilla Firefox browser that allows you to debug and inspect HTML, CSS, the Document Object Model (DOM) and JavaScript. In addition to debugging web pages, Firebug was used for web security testing and web page performance analysis.

3. Why do Web Developers use Firebug?
a) Track cookies
b) Track sessions
c) Both Track cookies and sessions
d) Track data

Answer: c
Explanation: Web developers use Firebug for the following reasons:

- *Inspect the behavior of HTML/CSS, and modify style & layout with true WYSIWYG*
- *Debug JavaScript*
- *Detect performance of website*
- *Track Cookies & Sessions*
- *Web security analysis*

4. What are the goals for using Firebug?
a) Performance
b) Adaptability
c) Complexity
d) Reliability

Answer: a
Explanation: The goals for using Firebug is:

- *Performance*
- *Modularity*
- *Shared code*
- *Compatibility*
- *Web security analysis*

5. Which of the following is not a feature of the User Interface?
a) Skinnable Interface

b) Resizable Side Panel
c) Fixed Side Panel
d) Menu options

Answer: c
Explanation: The features present in the User Interface are:

- *Port of Firebug's Visual Object Representation (aka Reps)*
- *Recreation of Firebug 1.3 User Interface with pixel precision*
- *Menu options*
- *Resizable Side Panel*
- *Skinnable Interface*

6. What is the default value of the property overrideConsole?
a) 1
b) true
c) 0
d) false

Answer: b
Explanation: The default value of the property ***overrideConsole*** *is true.*

7. Which of the following property(s) has a default value as false?
a) disableWhenFirebugActive
b) showIconWhenHidden
c) disableXHRListener
d) both disableWhenFirebugActive & showIconWhenHidden

Answer: c
Explanation: Only ***disableXHRListener*** *property has a default value false. The properties* ***disableWhenFirebugActive*** *and* ***showIconWhenHidden*** *has a default value of true.*

8. Which of the following action is possible in Firebug when used as a JavaScript Debugger and Profiler?
a) Pause execution in any line
b) Find Scripts easily
c) Find Scripts easily & also Pause execution in any line
d) Find text easily

Answer: c
Explanation: When Firebug is used as a ***JavaScript Debugger and Profiler****, it can be used to find scripts easily and also pause the execution in any desired line. Thus it helps in debugging javascript much easier.*

9. What will be the output or type of error if p is not defined in the following JavaScript code?

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

a) Zero
b) Null
c) ReferenceError
d) ValueNotFoundError

Answer: c
Explanation: The above code snippet, p is not defined. Hence, it gives a ReferenceError.

10. The let keyword can be used ___________
a) in a for or for/in loop, as a substitute for var
b) as a block statement, to define new variables
c) to define variables that are scoped to a single expression

d) all of the mentioned

Answer: d
*Explanation: The **let** keyword can be used in four ways:*

1. *as a variable declaration like var;*
2. *in a for or for/in loop, as a substitute for var;*
3. *as a block statement, to define new variables and explicitly delimit their scope; and*
4. *to define variables that are scoped to a single expression.*

1. How does PhantonJS use YSlow?
a) Queries
b) Statements
c) Command Line Prompt
d) Command Line Script

Answer: d
*Explanation: YSlow for PhantomJS is a **command line script** that allows page performance analysis from live URLs, unlike YSlow for Command Line (HAR) where a pre-generated HAR file is needed in order to analyze page performance.*

2. What are the two output formats YSlow uses?
a) TAP, JUnit
b) JIT, TRD
c) JKP, RFD
d) TIP, KIT

Answer: a
Explanation: YSlow for PhantomJS is a command line script that allows page performance analysis from live URLs. YSlow for PhantomJS introduces new output formats for automated test frameworks: TAP (Test Anything Protocol) and JUnit.

3. Initially, YSlow was an extension of which browser?
a) Chrome
b) Firefox
c) IE
d) Opera

Answer: b
*Explanation: Initially, YSlow was an extension of **Firefox**. YSlow for Firefox needs Firebug to run.*

4. Which of the following can be used for a deeper analysis of the web page's performance?
a) WebPageTest
b) FireBug
c) YSlow
d) WebPageTest & FireBug

Answer: c
Explanation: YSlow analyzes web page performance by examining all the components on the page, including components dynamically created by using JavaScript. It measures the page's performance and offers suggestions for improvement.

5. How many rules are there in the YSlow version 2.0?
a) 23
b) 21
c) 27
d) 33

Answer: a

Explanation: There are totally 23 rules in the ruleset of YSlow (V2) namely :

1. *Minimize HTTP Requests*
2. *Use a Content Delivery Network*
3. *Avoid empty src or href*
4. *Add an Expires or a Cache-Control Header*
5. *Gzip Components*
6. *Put StyleSheets at the Top*
7. *Put Scripts at the Bottom*
8. *Avoid CSS Expressions*
9. *Make JavaScript and CSS External*
10. *Reduce DNS Lookups*
11. *Minify JavaScript and CSS*
12. *Avoid Redirects*
13. *Remove Duplicate Scripts*
14. *Configure ETags*
15. *Make AJAX Cacheable*
16. *Use GET for AJAX Requests*
17. *Reduce the Number of DOM Elements*
18. *No 404s*
19. *Reduce Cookie Size*
20. *Use Cookie-Free Domains for Components*
21. *Avoid Filters*
22. *Do Not Scale Images in HTML*
23. *Make favicon.ico Small and Cacheable*

6. Which of the following is mandatory to run before running YSlow?
a) WebPageTest
b) FireBug
c) Both WebPageTest and FireBug
d) Mozilla

Answer: b
Explanation: YSlow analyzes web page performance by examining all the components on the page, including components dynamically created by using JavaScript. It is mandatory to run ***FireBug*** *before running YSlow.*

7. How does the YSlow for Mobile work as?
a) Bookwise
b) Booklet
c) Bookmarklet
d) Bookmark

Answer: c
Explanation: The YSlow for Mobile works as bookmaklet. Bookmarklet is a JavaScript code stored as the URL of a bookmark in a web browser.

8. Which network allows you to distribute static assets like images, etc?
a) Content Delivery Network
b) Content Receiving Network
c) System Area Network
d) Local area network

Answer: a
Explanation: A content delivery network (CDN) allows you to distribute your static assets like images, JavaScript files and stylesheets to geographically distributed servers. This gets the content of your page to your user's browser faster.

9. 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. Firstly the variable is created then it's first tested then updated.

10. What convenience does the following JavaScript code?

```
let succ = function(x) x+1, yes = function() true, no = function() false;
```

What convenience does the above code snippet provide?
a) Functional behaviour
b) Modular behaviour
c) No convenience
d) Shorthand expression

Answer: a
Explanation: The functions defined in this way behave exactly like functions defined with curly braces and the ***return*** *keyword. The functions can be defined in a shorthand expression.*

1. When does WebPageTest become necessary?
a) To run any common website
b) To run private websites
c) To run QA testing
d) To run media testing

Answer: b
Explanation: If you want to run tests on web sites that are not publicly available—like a QA or development environment, or if you can only have your test results stored on your own servers because of legal or other reasons, then installing your own private instance of WebPagetest is the way to go.

2. What is the purpose of the Auth tab in the testing platform?
a) To specify credentials
b) To authorize a page
c) To run the test
d) To run user access

Answer: a
Explanation: In the Auth tab you can specify credentials to use if the web site uses HTTP authentication for access.

3. What is the purpose of the Script tab?
a) To edit the coding
b) To run on multiple servers
c) To run more complex tests
d) To run media

Answer: c
Explanation: The ***Script tab*** *can be used to run more complex tests that involve multiple steps including navigate to multiple URLs, etc.*

4. Which is the command that is used to spoof the client user agent?
a) setUserAgent
b) spoofAgent
c) spoofClientUserAgent
d) spoofUser

Answer: a
*Explanation: The **setUserAgent** spoofs the client user agent.*

5. What is the purpose of the Block tab?
a) To block the response
b) To block the request
c) To block the mouse pointer
d) To block the cursor

Answer: b
*Explanation: The **Block tab** allows us to block content coming in our request.*

6. What is the purpose of the Video tab?
a) To take video of the page
b) To capture screenshots of the test only
c) To take video of the test
d) To capture screenshots of the page

Answer: d
*Explanation: The **Video tab** allows you to capture screenshots of your page as it loads and views them as a video.*

7. What is the task of the Advanced Panel?
a) Test stop running
b) Loading the pages
c) Skewing the results
d) Loading the server

Answer: a
Explanation: In the Advanced panel, you can have the test stop running at document completion. That will tell us when the document.onload event is fired, instead of when all assets on the page are loaded.

8. What is the vulnerability of XHR communications?
a) There is no vulnerability
b) Skewing the results
c) Registration and Skewing the results
d) Registering the test results

Answer: c
Explanation: XMLHttpRequest (XHR) is an API in the form of an object whose methods transfer data between a web browser and a web server. XHR communications that may happen after page load could register as new activity and skew the test results.

9. Which of the following methods adds and connects the point to the cubic bezier curve?
a) bezierConnect()
b) bezierCurveTo()
c) Connectbezier()
d) bezierTo()

Answer: b
Explanation: A cubic bezier curve requires three points. The first two points are control points that are used in the cubic Bézier calculation and the last point is the ending point for the curve. This method adds a new point P to the subpath and connects it to the current point with a cubic Bezier curve.

10. Which of the following methods adds an arc to the current subpath?
a) bezierCurveTo()
b) arcTo()
c) arc()
d) Curve()

Answer: c
*Explanation: The **arc()** method adds an arc to the current subpath. The arc() function is fined in the canvas method.*

1. What is the purpose of minifying the JavaScript?
a) To streamline the visits
b) To save the visits
c) To save data
d) To increase the loading time

Answer: a
Explanation: To ensure that your first-time visits are as streamlined as possible, we need to minify our JavaScript. Minifying JavaScript results in decreasing the loading time and hence helps in creating a better user experience.

2. From which did Minification concept originate?
a) JavaScript code efficiency
b) JavaScript interpreter
c) JavaScript Compiler
d) JavaScript writer

Answer: b
Explanation: Minification is originally based on the idea that the JavaScript interpreter ignores white space, line breaks, and of course comments, so we can save on total file size of our .js files if we remove those unneeded characters.

3. What is the purpose of the product Minify?
a) Storing the data
b) Streamlining the data
c) Proxies the JavaScript file
d) Loading the data

Answer: c
Explanation: Minify proxies the JavaScript file; the script tag on the page points to Minify, which is a PHP file.

4. What does the Minify set the encoding HTTP header?
a) deflate
b) gzip
c) both deflate and gzip
d) inflate

Answer: c
*Explanation: Minification refers to the process of removing unnecessary or redundant data without affecting how the resource is processed by the browser. Minify reads the JavaScript file in, minifies it and when it responds it sets the accept encoding HTTP header to **gzip, deflate**.*

5. What is in-built in the Minify?
a) Dynamic compression
b) Static compression
c) Static content
d) Dynamic content

Answer: b
Explanation: Minification is the process of minimizing code and markup in your web pages and script files. Effectively Minify has built in HTTP static compression. This is especially useful if your web host doesn't allow the gzipping of static content.

6. Which folder contains the Minify control panel?
a) /min/builder/
b) /builder/
c) /minify/build

d) /minify/builder

Answer: a
Explanation: The minified file version provides the same functionality while reducing the bandwidth of network requests. To navigate to the Minify control panel, it is located in the /min/builder/.

7. Which of the following is the order of Minify process?
a) Remove extraneous characters, gzip the response, Read
b) Remove extraneous characters, Read, gzip the response
c) Read, Remove extraneous characters, gzip the response
d) Read, gzip, extract, remove

Answer: c
Explanation: To minify JS, CSS and HTML files, comments and extra spaces need to be removed, as well as crunch variable names so as to minimize code and reduce file size. Minify reads in the content, decorates it by way of removing extraneous characters, and gzips the response.

8. YUI Compressor is analogous to ___________
a) Minify
b) JavaScript
c) Both Minify and JavaScript
d) CSS

Answer: a
*Explanation: The YUI Compressor is JavaScript minifier designed to be 100% safe and yield a higher compression ratio than most other tools. Just like Minify, **YUI Compressor** strips out all of the unnecessary characters from your JavaScript, including spaces, line breaks, and comments.*

9. What type of file is YUI Compressor?
a) Binary file
b) JAR file
c) Text file
d) Assembly file

Answer: b
Explanation: The YUI Compressor is JavaScript minifier designed to be 100% safe and yield a higher compression ratio than most other tools. YUI Compressor is a jar file and runs from the command line. Because of this, it is easily integrated into a build process. It looks like this:
java -jar yuicompressor-[version].jar [options] [file name]

10. What is the function of a Closure Compiler?
a) Originates the JavaScript
b) Compiles the JavaScript
c) Rewrites JavaScript
d) Links the JavaScript

Answer: c
Explanation: The Closure Compiler is a tool for making JavaScript download and runs faster. Instead of compiling from a source language to machine code, it compiles from JavaScript to better JavaScript. Closure Compiler runs through a number of "scorched-earth" optimizations—it unfurls functions, rewrites variable names, and removes functions that are never called (as far as it can tell).

1. What is scorched-earth optimizations?
a) They optimize based on certain constraints
b) They strip out everything including best practices
c) Based on certain constraints
d) Striping out additional things

Answer: b
Explanation: The Closure Compiler is a tool for making JavaScript download and runs faster. Closure Compiler runs through a number of "scorched-earth" optimizations—it unfurls functions, rewrites variable names, and removes functions that are never called (as far as it can tell).

2. The Closure Compiler was introduced by ___________
a) Microsoft
b) Apple
c) Google
d) Yahoo

Answer: c
*Explanation: The Closure Compiler is a tool for making JavaScript download and runs faster. Instead of compiling from a source language to machine code, it compiles from JavaScript to better JavaScript. The Closure Compiler was introduced by **Google**.*

3. What does the Minify return back to the *Script tag*?
a) File I/O
b) Minify JavaScript
c) HTTP Request
d) Gzip encoded

Answer: d
*Explanation: Minification refers to the process of removing unnecessary or redundant data without affecting how the resource is processed by the browser. The Minify sends back the **gzip encoded** back to the **Script tag**.*

4. What is the return type of the script tag?
a) File I/O
b) Minify JavaScript
c) HTTP Request
d) Gzip encoded

Answer: c
*Explanation: The **script tag** sends the **HTTP Request** to the **Minify**. Minify reads the JavaScript file in, minifies it and when it responds it sets the accept encoding HTTP header to gzip, deflate.*

5. What does the JS File return?
a) File I/O
b) Minify JavaScript
c) HTTP Request
d) Gzip encoded

Answer: b
*Explanation: Minification is the process of minimizing code and markup in your web pages and script files. Effectively Minify has built in HTTP static compression. The JS File returns the **Minify JavaScript**.*

6. What is the outcome of using R along with minification?
a) Sheer file size reduction
b) File size increase
c) More efficient
d) Compatibility

Answer: a
Explanation: Minification refers to the process of removing unnecessary or redundant data without affecting how the resource is processed by the browser. Sheer file size reduction is the only one aspect of the overall determination got from the R along with the minification.

7. Which one of the following is more efficient in terms of the file size reduction?

a) YUI
b) Closure Compiler (advanced)
c) Minify
d) Closure Compiler (simple)

Answer: b
Explanation: The Closure Compiler is a tool for making JavaScript download and runs faster. Instead of compiling from a source language to machine code, it compiles from JavaScript to better JavaScript. The Closure Compiler (advanced) has a more number percent of file size reduction by product.

8. Which of the following are JavaScript optimizers?
a) JSMin
b) Packer
c) Minify
d) Both JSMin and Packer

Answer: d
Explanation: JSMin is a filter which removes comments and unnecessary whitespace from JavaScript files. It typically reduces filesize by half, resulting in faster downloads. JavaScript optimizers such as JSMin and Packer are specially designed for modern web programming techniques and are able to understand and preserve conditional comments, and similar.

9. Which of the following is the approach used to compress HTML in web servers and modern web browsers?
a) Content encoding
b) Content decoding
c) Compression algorithm – DEFLATE
d) Content minification

Answer: a
Explanation: Minification refers to the process of removing unnecessary or redundant data without affecting how the resource is processed by the browser. Content encoding is an approach taken by compatible web servers and modern web browsers to compress HTML and related textual content, often in the gzip format.

10. Which is the game that requires extremely minified source in the Perl culture?
a) Perl pool
b) Perl golf
c) Both Perl pool and golf
d) Perl bat

Answer: b
Explanation: Perl golf is a game where one attempts to write the shortest Perl program to accomplish some goal. In Perl culture, aiming at extremely minified source code is the purpose of the Perl golf game.

1. Which of the following is a stateless protocol?
a) HTML
b) XHTML
c) HTTP
d) XML

Answer: c
Explanation: A stateless protocol does not require the server to retain information or status about each user for the duration of multiple requests. HTTP is a stateless protocol, which means that the connection between the browser and the server is lost once the transaction ends.

2. What does the value 2 of the WebSocket attribute Socket.readyState indicate?
a) Closed connection
b) Handshake connection
c) Unestablished connection

d) Established connection and communication is possible

Answer: b
Explanation: The WebSocket object provides the API for creating and managing a WebSocket connection to a server, as well as for sending and receiving data on the connection. The readonly attribute readyState represents the state of the connection. It can have the following values:

1. *A value of 0 indicates that the connection has not yet been established.*
2. *A value of 1 indicates that the connection is established and communication is possible.*
3. *A value of 2 indicates that the connection is going through the closing handshake.*
4. *A value of 3 indicates that the connection has been closed or could not be opened.*

3. How many WebSocket events are available?
a) 2
b) 3
c) 4
d) 5

Answer: c
Explanation: Web sockets are defined as a two-way communication between the servers and the clients, which mean both the parties, communicate and exchange data at the same time. There are fourWebSocket events namely:

1. *open*
2. *close*
3. *message*
4. *error*

4. Which method is used to close the WebSocket?
a) Socket.flush()
b) Socket.close()
c) Socket.Close()
d) Socket.dispose()

Answer: b
*Explanation: The **Socket.close()** is used to close the WebSocket. The Close method closes the remote host connection and releases all managed and unmanaged resources associated with the Socket. WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection.*

5. How does the client and the server communicate following the rules defined by the WebSocket protocol?
a) Long-lived TCP Socket
b) Short-lived TCP Socket
c) UDP Socket
d) HTTP Socket

Answer: a
Explanation: The client and server communicate over a long-lived TCP socket following rules defined by the WebSocket protocol.

6. Which of the following is not a socket property?
a) onopen
b) readyState
c) onmessage
d) ready

Answer: d
*Explanation: There is no Socket property called **ready**. Various Socket properties include onopen, readystate, ready, binaryTree, onclose, onerror etc.*

7. What does the following JavaScript code snippet do?

```
var httpserver = new http.Server();
```

a) Create an HTTP Server
b) Create HTTP Connection between Client and Server
c) HTTP Server & Connection
d) Create a connection between two servers

Answer: a
Explanation: HTTP defines what actions Web servers and browsers should take in response to various commands. The above code creates an HTTP Server.

8. How can we check the subprotocol being used by the client?
a) subprotocol property
b) protocol property
c) clientprotocol property
d) client property

Answer: b
Explanation: A WebSocket is a standard bidirectional TCP socket between the client and the server. The socket starts out as an HTTP connection and then "Upgrades" to a TCP socket after an HTTP handshake. Once the connection is established, the client can determine which subprotocol is in use by checking the ***protocol*** *property of the socket.*

9. How will you transmit data using the connection?
a) send(data)
b) Socket.send("data")
c) Socket.send(data)
d) Socket(data)

Answer: c
Explanation: The ***Socket.send(data)*** *method transmits data using the connection. This Socket method sends data to connected socket.*

10. Which of the following is not a WebSocket event?
a) open
b) close
c) error
d) deny

Answer: d
Explanation: There is no WebSocket event named ***deny****. The four WebSocket events are*

1. *open*
2. *close*
3. *message*
4. *error*

Listen to these events using addEventListener() or by assigning an event listener to the oneventname property of this interface.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "hello to \r world.";
   var patt1 = /\v/;
   var result = str.search(patt1);
```

```
    document.getElementById("demo").innerHTML = result;
}
</script>
```

a) 8
b) 9
c) 3
d) -1

Answer: b
Explanation: The \v metacharacter is used to find a vertical tab character. \v returns the position where the vertical tab character was found. If no match is found, it returns -1.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
    var str = "Hello World!";
    var patt1 = /\127/g;
    var result = str.match(patt1);
    if(result)
         result=true;
    else
         result=false;
    document.getElementById("demo").innerHTML = result;
}
</script>
```

a) true
b) false
c) error
d) null

Answer: a
Explanation: The \xxx character is used to find the Latin character specified by an octal number xxx. If no match is found, it returns null.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
    var str = "Hello World!";
    var patt1 = /\x57/g;
    var result = str.match(patt1);
    if(result)
         result=true;
    else
         result=false;

    document.getElementById("demo").innerHTML = result;
}
</script>
```

a) error
b) false
c) true
d) undefined

Answer: c
Explanation: The \xdd character is used to find the Latin character specified by a hexadecimal number dd. The

character 'W' is represented as x57 in hexadecimal.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "Visit W3Schools. Hello World!";
   var patt1 = /\u0057/g;
   var result = str.match(patt1);
   var result = str.match(patt1);
   if(result)
        result=true;
   else
        result=false;
   document.getElementById("demo").innerHTML = result;
}
</script>
```

a) true
b) false
c) error
d) undefined

Answer: a
Explanation: The \udddd character is used to find the Unicode character specified by a hexadecimal number dddd. If no match is found, it returns null.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "100%!";
   var patt1 = /\W/g;
   var result = str.match(patt1);
   var result = str.match(patt1);
   if(result)
        result=true;
   else
        result=false;
   document.getElementById("demo").innerHTML = result;
}
</script>
```

a) false
b) true
c) error
d) undefined

Answer: b
Explanation: The \W metacharacter is used to find a non-word character. A word character is a character from a-z, A-Z, 0-9, including the _ (underscore) character.

1. What does a History object contain?
a) URL
b) Parameters
c) Attribute values
d) Property

Answer: a
*Explanation: The **history object** contains the URLs visited by the user. By using history object, you can load previous,*

forward or any particular page using various methods.

2. The history object is a part of which object?
a) Property
b) Window
c) Location
d) Screen

Answer: b
Explanation: The window object represents an open window in a browser. The history object belongs to the Window object.

3. How many methods are there in the History object?
a) 3
b) 4
c) 5
d) 6

Answer: a
Explanation: There are three methods belonging to the History object namely :

1. *back()*
2. *forward()*
3. *go()*

back() loads the previous URL, forward loads the next URL and go() loads a specific URL in the history list.

4. What is the purpose of the method forward()?
a) Loads any random URL in the history list
b) Loads the previous URL in the history list
c) Loads a specific URL from the history list
d) Loads the next URL in the history list

Answer: d
Explanation: The forward() method is found in the history object. The ***forward()*** *method loads the next URL in the history list.*

5. How will you update the URL displayed in the location bar?
a) location
b) location.URL
c) location.hash
d) url

Answer: c
Explanation: The hash property sets or returns the anchor part of a URL, including the hash sign (#). The property ***location.hash*** *needs to be updated to display the updated URL in the location bar.*

6. How do you add a particular state to the browsing history?
a) pushState()
b) replaceState()
c) state()
d) addstate()

Answer: a
Explanation: The ***pushState()*** *method adds a particular state to the browsing history. It pushes the given data onto the session history stack with the specified title and, if provided, URL.*

7. What does the pushState() method do?

a) Removes the state
b) Adds new state
c) Replaces the state
d) Change the state

Answer: b
*Explanation: When a web app enters a new state, it calls **history.pushState()** to add that state to the browsing history.*

8. Which of the following method is used to replace the current history state instead of adding a new state to the browsing history?
a) replaceState()
b) replace(state)
c) replace()
d) change()

Answer: a
*Explanation: The **replaceState()** method is used to replace the current history state instead of adding a new state to the browsing history. It updates the most recent entry on the history stack to have the specified data, title, and, if provided, URL.*

9. How many parameters does the replaceState() method take?
a) 2
b) 3
c) 4
d) 5

Answer: b
*Explanation: The replaceState() updates the most recent entry on the history stack to have the specified data, title, and, if provided, URL. **window.history.replaceState(stateObj, title, url)** : This is just like window.history.pushState, except that the current browser state is removed from the history, so you cannot hit "back" to return to it.*

10. What is the purpose of the event window.onpopstate?
a) When a state object is replaced
b) When a state object is added
c) When a state object is removed
d) When a state object is changed

Answer: c
*Explanation: The **window.onpopstate** event is fired whenever a state object is removed from the browser history, which occurs on browser "back" or "forward". The object passed into a call to pushState or replaceState is provided as the state property on the event object in the "popstate" event.*

1. R is an extension of which of the following language?
a) C
b) C++
c) S
d) C#

Answer: c
Explanation: The R language is widely used among statisticians and data miners for developing statistical software and data analysis. R is an extension of and successor to the S language, which was itself a statistical language created in 1976 by John Chambers while at Bell Labs.

2. Which of the following is/are statistical languages?
a) R
b) S
c) P
d) Both R and S

Answer: c
Explanation: Both R and S are statistical languages. The R language is widely used among statisticians and data miners for developing statistical software and data analysis.

3. Which of the following is a successor to the S language?
a) C++
b) R
c) S
d) Java

Answer: b
Explanation: R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. R is an extension of and successor to the S language, which was itself a statistical language.

4. What are the purposes R can be used for?
a) Suck in data
b) Parse data
c) Process data
d) All of the mentioned

Answer: d
Explanation: The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Generally, R can be used to suck in data, parse it, process it, and then visualize it for reporting purposes.

5. What type of language is generally used to collect the data?
a) Glue language
b) Statistical language
c) Both Glue and Statistical language
d) Operational language

Answer: a
Explanation: A glue language is generally used to collect the data, that is written out as a comma-separated file, and read it into R. It enables interconnecting, support and the integration of software programs and components created using different programming languages and platforms.

6. What are the processes that take place within R?
a) Splitting of data
b) Aggregating of data
c) Overlaying two or more data
d) All of the mentioned

Answer: d
Explanation: R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Within R, processing the data, splitting it, averaging it, aggregating it, overlaying two or more data sets, and then from with R, a chart that depicts the data out is done.

7. Which platform is R imported to after charting as a PDF?
a) Adobe Illustrator
b) Adobe Photoshop
c) Both Adobe Illustrator and Adobe Photoshop
d) Adobe Indesign

Answer: a
Explanation: Adobe Illustrator is used to produce professionally looking at statistical graphics. From R, the chart is imported into the Adobe Illustrator, or any other such program, where it can clean up things like font consistency and make sure axis labels with long names are visible.

8. What is the purpose of a glue language?
a) Format data
b) Product data
c) Collect data
d) Both Format and Collect

Answer: d
Explanation: Glue language refers to a programming language that is designed specifically to write and manage program and code, which connects together different software components. A glue language is generally used to collect and format the data.

9. Why do we use Adobe Illustrator along with R?
a) Collect the relevant data
b) Format the chart and to correct the errors
c) Tighten and format the chart
d) Ingest and process the chart

Answer: c
Explanation: Adobe Illustrator is used to produce professionally looking at statistical graphics. The Adobe Illustrator is used along with R to tighten and format the chart.

10. What does the R language do?
a) Tighten and format the chart
b) Ingest and process the chart
c) Format and Ingest the chart
d) Create the chart

Answer: b
Explanation: R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible. The R language is generally used to collect data that has been collected by the glue language and then ingest and process and chart.

1. What kind of data can be run in R?
a) Binary
b) Text
c) Decimals
d) All kinds

Answer: d
Explanation: The R language is widely used among statisticians and data miners for developing statistical software and data analysis. All kinds of data can be run in R.

2. Which of the following is/are not the features of R?
a) Small
b) Self-contained
c) Extensible
d) Large

Answer: d
Explanation: R and its libraries implement a wide variety of statistical and graphical techniques, including linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, and others. As a language, R is small, self-contained and extensible.

3. What is the advantage for Linux users using R?
a) They get a command sequence to install the particular Linux flavor
b) They get a compiled installer
c) Easily compatible
d) Runtime is less

Answer: a
Explanation: Instead of a compiled installer, Linux users get the command sequence to install for their particular Linux flavor.

4. Which of the following is the base install for R?
a) Contrib
b) Base
c) Rtools
d) RBase

Answer: b
Explanation: Base is the binaries for base distribution. The PC installer comes in three flavors: Base is the base install, Contrib comes with compiled third-party packages, and Rtools comes with tools to build your own R packages.

5. Which of the following comes with compiled third-party packages?
a) Contrib
b) Base
c) Rtools
d) RBase

Answer: a
Explanation: Contrib are the binaries of contributed CRAN packages. The PC installer comes in three flavors: Base is the base install, Contrib comes with compiled third-party packages, and Rtools comes with tools to build your own R packages.

6. Which of the following comes with tools to build your own R packages?
a) Contrib
b) Base
c) Rtools
d) RBase

Answer: c
Explanation: Rtools provide the tools to build R and R packages. The PC installer comes in three flavors: Base is the base install, Contrib comes with compiled third-party packages, and Rtools comes with tools to build your own R packages.

7. The code of the R language has an extension?
a) .Rl
b) .R
c) .lR
d) .RR

Answer: b
Explanation: R is a programming language designed for statistical computing and graphics purposes. R is a file extension for a script written in R.

8. Which of the following is used to run ad hoc R commands?
a) R Console
b) R Primer
c) Both R Console and R Primer
d) R timer

Answer: a
Explanation: Interactive data analysis usually occurs on the R console that executes commands as you type them. R Console is a command-line environment for running ad hoc R commands.

9. Which is the keyword used to open the Help window?
a) ()

b) #
c) ?
d) =

Answer: c
Explanation: The help() function and ? help operator in R provides access to the documentation pages for R functions, data sets, and other objects, both for packages in the standard R distribution and for contributed packages. At any time, you can type ? (keyword) to open the help window for a particular subject.

10. For a more extensive search, which keyword needs to be used?
a) ??
b) ?
c) ?=
d) =?

Answer: a
Explanation: The help.search() or the ?? function scans the documentation for packages installed in your library. For a more extensive search, just type ?? (keyword).

1. What is the usage of the keyword ?? in R?
a) Help window
b) Extensive search
c) Error correction
d) Error detection

Answer: b
Explanation: The ?? function scans the documentation for packages installed in your library. ?? (keyword) is generally used for a more extensive search when compared to the keyword ?.

2. Which type of comment is not supported in R?
a) Single-line comments
b) Multi-line comments
c) Both Single-line & Multi-line comments
d) Multiple Comments

Answer: b
Explanation: R supports single-line comments, but not multiline comments. # is used for making a single line comment.

3. What is the usage of the keyword ? in R?
a) Help window
b) Extensive search
c) Error correction
d) Error detection

Answer: a
Explanation: ? is generally used to open the help window at any time. The help() function and ? help operator in R provides access to the documentation pages for R functions, data sets, and other objects, both for packages in the standard R distribution and for contributed packages.

4. Which symbol is used to start a comment?
a) /
b) $
c) #
d) ?

Answer: c
Explanation: '#' is used to comment a line. A single line comment can be made in R.

5. Which of the following is the assignment operator?
a) <-
b) ->
c) =
d) ==

Answer: a
Explanation: The operators <- and = can be used, almost interchangeably, to assign to a variable in the same environment. The assignment operator is a left-pointing arrow, so creating and declaring variables looks like this:
foo <- bar

6. Which of the following is a loosely-typed language?
a) R
b) T
c) S
d) Both R and S

Answer: c
Explanation: R is an implementation of the S programming language combined with lexical scoping semantics, inspired by Scheme. [15] S was created by John Chambers in 1976, while at Bell Labs. R and S are loosely-typed languages and it supports all of the scalar data types you would expect: string, numbers, and booleans.

7. What is the limit to matrices in R?
a) One dimensional
b) Two dimensional
c) Three dimensional
d) No limit

Answer: b
*Explanation: Matrices are like strictly typed two-dimensional arrays. You create a matrix using the matrix function, which accepts five parameters: a vector to use as the content, the number of rows to shape the content into, the number of columns to shape the content into, an optional boolean value to indicate whether the content should be shaped by row or by column (the default is **FALSE** for by column), and a list that contains vectors for row names and column names: matrix([content vector], nrow=[number of rows], ncol=[number of columns], byrow=[how to sort], dimnames= [vector of row names, vector of column names]).*

8. Which is the function used to add the vectors?
a) c()
b) add(vectors)
c) c(vectors)
d) vectors.add

Answer: a
*Explanation: The **c()** is a generic function which combines its arguments. It is used to add the vectors.*

9. Which of the following list contains multiple data types?
a) Vectors
b) Data frames
c) Matrices
d) Arrays

Answer: b
Explanation: Data frames are multidimensional lists that can contain multiple data types.

10. Which function is used to create data frames?
a) data.frames()
b) frame.data()
c) data.frame()

d) frame(data)

Answer: c
Explanation: Data frames are multidimensional lists that can contain multiple data types. A data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

1. What is the purpose of the method plot()?
a) Displays symbols
b) Displays charts
c) Displays symbols and charts
d) Display characters

Answer: b
Explanation: The most used plotting function in R programming is the plot() function. It is a generic function, meaning, it has many methods which are called according to the type of object passed to plot(). The plot() function will display a different type of chart depending on the arguments that you pass into it.

2. How many parameters does the method plot() accept?
a) 6
b) 7
c) 8
d) 9

Answer: d
Explanation: Plot() is a generic function, meaning, it has many methods which are called according to the type of object passed to plot(). The method plot() accepts a total of 9 parameters.

3. What is the need for bubble charts?
a) Represent 2D data
b) Represent 3D data
c) Represent 2D and 3D data
d) Represents meta data

Answer: b
Explanation: A bubble plot is a scatterplot where a third dimension is added the value of an additional variable is represented through the size of the dots. They are used to represent three-dimensional data.

4. Which of the following is the initial function used to create a bubble chart natively in R?
a) init()
b) chart(bubble)
c) symbols()
d) bchart()

Answer: c
Explanation: symbols() function draws symbols on a plot. One of six symbols; circles, squares, rectangles, stars, thermometers, and boxplots, can be plotted at a specified set of x and y coordinates. The method ***symbols()*** *is used to create a bubble chart natively in R.*

5. What is the purpose of the method symbols in R?
a) Draw symbols
b) Draw other shapes
c) Draw symbols and other shapes
d) Plotting symbols

Answer: b
Explanation: The symbols() function can be used to draw other shapes on a plot; for more information about this type ? ***symbols*** *at the R console. One of six symbols; circles, squares, rectangles, stars, thermometers, and boxplots, can be*

plotted at a specified set of x and y coordinates.

6. How to save chart as a Window metafile?
a) metafile()
b) win.metafile()
c) file()
d) metawin()

Answer: b
Explanation: Metafile is a piece of graphical information stored in a format that can be exchanged between different systems or software. The method ***win.metafile([filename])*** *is used to save chart as a Window metafile.*

7. Which is the method used to save chart as a ps file?
a) ps()
b) postscript()
c) script()
d) post(script)

Answer: b
Explanation: postscript starts the graphics device driver for producing PostScript graphics. The syntax for the method to save chart as a ps file is : ***postscript([filename])****.*

8. Which of the following are methods not used to save charts?
a) pdf()
b) jpeg()
c) bmp()
d) pmb()

Answer: d
Explanation: Since R runs on so many different operating systems, and supports so many different graphics formats, it's not surprising that there are a variety of ways of saving your plots, depending on what operating system you are using, what you plan to do with the graph, and whether you're connecting locally or remotely.

9. Which is the method used to draw a bar plot?
a) bar_plot()
b) plot(bar)
c) barplot()
d) plotbar()

Answer: c
Explanation: A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable. R uses the function barplot() to create bar charts.

10. What is the purpose of the method par()?
a) Makes the text vertical
b) Makes the text horizontal
c) Makes the text diagonal
d) Makes the text small

Answer: b
Explanation: The par(mfrow) function is handy for creating a simple multi-paneled plot, while layout should be used for customized panel plots of varying sizes. With the method ***par()****, you can make the text horizontal.*

1. How many parameters does the WebPageTest API accept?
a) 5
b) 6
c) 7
d) 8

Answer: d
Explanation: WebPageTest helps you run a free website speed test from multiple locations around the globe using real browsers (IE and Chrome) and at real consumer connection speeds. The WebPageTest API accepts totally 8 parameters, namely url, location, runs, fvonly, private, block, f, k.

2. What is the purpose of getting the parameter block?
a) Permission to block
b) Space separated block list
c) Comma separated block list
d) Underscore separated block list

Answer: c
Explanation: A ParameterBlock encapsulates all the information about sources and parameters (Objects) required by a RenderableImageOp, or other classes that process images. This parameter allows you to set a comma separated list of block options.

3. What will happen if you set the private flag as 1?
a) Test will be run only by the administrator
b) Test will not be run
c) Test is public
d) Test is not displayed in public

Answer: d
Explanation: The WebPageTest results provide rich diagnostic information including resource loading waterfall charts, Page Speed optimization checks and suggestions for improvements which can be kept public or private. Setting the ***private flag*** *to 1 will make sure that the test is not displayed in the public list of tests.*

4. What is the purpose of getting the parameter fvonly?
a) To repeat the view test
b) To get the first view
c) To debug the code
d) To compile the code

Answer: b
Explanation: fvonly is an optional parameter which is set to 1 to skip the Repeat View test. If you set fvonly to 1, you get results only for the first view and do not run the repeat view test.

5. What is the purpose of the file_get_contents()?
a) To get the errors and exceptions
b) To get the client's response
c) To get the server's response
d) To get the data

Answer: c
Explanation: The ***file_get_contents()*** *is a PHP's native function used to hit the URL and read the server's response into a variable $wpt_response:*

```
$wpt_response = file_get_contents($wpt_url . $urls_to_benchmark[$x]);
```

This function is similar to file(), except that file_get_contents() returns the file in a string, starting at the specified offset up to maxlen bytes. On failure, file_get_contents() will return FALSE.

6. What will the file_get_contents() return?
a) Server's response
b) Errors
c) Exception
d) Client's response

Answer: a
Explanation: The ***file_get_contents()*** *is a PHP's native function used to hit the URL and read the server's response into a variable $wpt_response:*

```
$wpt_response = file_get_contents($wpt_url . $urls_to_benchmark[$x]);
```

This function is similar to file(), except that file_get_contents() returns the file in a string, starting at the specified offset up to maxlen bytes. On failure, file_get_contents() will return FALSE.

7. How will you convert the returned API into an XML object?
a) SimpleElement()
b) SimpleXMLElement()
c) XMLElement()
d) CovertXML()

Answer: b
Explanation: file_get_contents() is the preferred way to read the contents of a file into a string. It will use memory mapping techniques if supported by your OS to enhance performance. The API returned by the method ***file_get_contents()*** *can be converted into an XML object using the method* ***SimpleXMLElement()****.*

8. Which tag can handle mouse events in Netscape?
a) img
b) a
c) br
d) src

Answer: a
Explanation: Netscape is a brand name associated with the development of the Netscape web browser. The img element can handle mouse events in Netscape.

9. What is the tainted property of the window object?
a) Pathname
b) Protocol
c) Default status
d) Host

Answer: c
Explanation: The ***Defaultstatus*** *is the tainted property of the window object. The defaultStatus property sets or returns the default text in the status bar at the bottom of the browser (the text will be displayed when the page loads).*

10. Which environment variable must the user enable in order to enable data tainting?
a) ENABLE_TAINT
b) MS_ENABLE_TAINT
c) NS_ENABLE_TAINT
d) ENABLE_TAINT_NS

Answer: c
Explanation: Data Tainting (or Taint Checking) is a language feature wherein user-input data is flagged as tainted, a flag that propagates to all data derived from this input. The environment variable ***NS_ENABLE_TAINT*** *must be enabled in order to enable data tainting.*

1. What is the necessity to create a separate file after having an API key?
a) To hold configuration information
b) To hold key details
c) To hold URL details
d) To hold the speed of the process

Answer: a

Explanation: An application programming interface key (API key) is a code passed in by computer programs calling an application programming interface (API) to identify the calling program, its developer, or its user to the Web site. Once you have an API key you should create a separate file to hold all of the configuration information that you will need to share between processes.

2. How many parameters does the API accept?
a) 5
b) 6
c) 7
d) 8

Answer: d
Explanation: Parameters are options you can pass with the endpoint (such as specifying the response format or the amount returned) to influence the response. The API accepts a total of 8 parameters namely:

1. *url*
2. *location*
3. *runs*
4. *fvonly*
5. *private*
6. *block*
7. *f*
8. *k*

.

3. What is not the purpose of the parameter *location*?
a) Specifies agent location
b) Specifies speed
c) Specifies browser
d) Specifies block

Answer: d
*Explanation: The window.location object can be used to get the current page address (URL) and to redirect the browser to a new page. The parameter **location** specifies the agent location, speed and browser to use for the test, formatted as location.browser:location.*

4. What will happen if the *fvonly* parameter is set to 1?
a) Results got for the first view
b) Can run the repeat view test
c) Running the test again
d) Ending the test

Answer: a
*Explanation: fvonly is an optional parameter which is set to 1 to skip the Repeat View test. If you set **fvonly** to 1, you get results only for the first view, and do not run the repeat view test.*

5. How many reserved words are there in JavaScript?
a) 63
b) 54
c) 68
d) 90

Answer: a
*Explanation: Keywords are reserved words in JavaScript which you cannot use to name the variables labels or function names. There are a total of **63** reserved words in JavaScript.*

6. What is the purpose of the window.location object in JavaScript?

a) Get the URL and redirect
b) Get the location of the cursor
c) Get the path to the next page
d) Get the location & path of the next page

Answer: a
Explanation: The ***window.location*** *object can be used to get the current page address (URL) and to redirect the browser to a new page. The parameter location specifies the agent location, speed and browser to use for the test, formatted as location.browser:location.*

7. Which of the following method loads a new document?
a) location.new()
b) loadnew()
c) location.load()
d) location.assign()

Answer: d
Explanation: The window.location object can be written without the window prefix.
Some examples are window.location.href returns the href (URL) of the current page
window.location.hostname returns the domain name of the web host.

8. Which of the following method will wait for certain milliseconds to execute a specified method?
a) setInterval()
b) setTimeout()
c) setmilli()
d) setseconds()

Answer: a
Explanation: The setInterval() method calls a function or evaluates an expression at specified intervals (in milliseconds). The ***setInterval()*** *method will wait a specified number of milliseconds, and then execute a specified function, and it will continue to execute the function, once at every given time-interval.*

9. What is the method used to stop an execution of a method?
a) clearInterval()
b) clearTimeout()
c) both clearInterval() and clearTimeout()
d) clearmethod()

Answer: c
Explanation: The ***clearInterval()*** *method is used to stop further executions of the function specified in the setInterval() method) The* ***clearTimeout()*** *method is used to stop the execution of the function specified in the setTimeout() method.*

10. What is the meaning of JavaScript Hoisting?
a) Moving declarations to bottom
b) Moving declarations to top
c) Hosting variables by itself
d) Moving declarations to specified location

Answer: b
Explanation: Hoisting is JavaScript's default behavior of moving all declarations to the top of the current scope (to the top of the current script or the current function). In JavaScript, a variable can be declared after it has been used. In other words; a variable can be used before it has been declared.

1. What is the function of the XML parser?
a) Converts XML document to XML DOM object
b) Converts XML DOM object to XML document
c) Converts XML DOM object to a comment
d) Compiles the html document

Answer: a
*Explanation: An **XML parser** converts an XML document into an XML DOM object – which can then be manipulated with JavaScript. All major browsers have a built-in XML parser to access and manipulate XML.*

2. What is the purpose of the method ActiveXObject()?
a) Used to call automation object
b) Used to reference automation object
c) Used to instantiate automation object
d) Used to call & reference automation object

Answer: c
*Explanation: The ActiveXObject.prototype object allows adding properties and methods to the ActiveXObject object that can be used with instances of the ActiveXObject object like any predefined property or method. The **ActiveXObject()** object is used only to instantiate Automation objects, and has no members.*

3. What is the purpose of the url json?
a) Belongs to JSON object
b) Reference JSON formatted data
c) Belongs to JSON
d) Compiles json data

Answer: b
*Explanation: The URL **JSON** is assumed to reference a file of JSON-formatted data. The value passed to the callback is the object obtained by parsing the URL contents with **jQuery.parseJSON()**. jQuery.getJSON() uses this type. If the type is "json" and the URL or data string contains "=?", the type is converted to "jsonp".*

4. What is the parameter of the method Date.parse()?
a) date
b) string
c) datestring
d) string

Answer: c
*Explanation: The parse method is defined as **Date.parse(datestring)**. The parse() method parses a date string and returns the number of milliseconds between the date string.*

5. Which is the function in JavaScript that will print the current page in JavaScript?
a) print()
b) printcurrent()
c) print(now)
d) print(this)

Answer: a
Explanation: The print() method prints the contents of the current window. The print() method opens the Print Dialog Box, which lets the user to select preferred printing options.

6. To which of the following object does the print() method belong to?
a) window
b) document
c) hash
d) string

Answer: a
*Explanation: The method **print()** belongs to the **window** object. The print() method opens the Print Dialog Box, which lets the user to select preferred printing options.*

7. What will happen if the radix parameter of the parseInt() function is omitted?
a) Runs in assumption

b) Throws exception
c) Aborts
d) Taken as 0

Answer: a
Explanation: The print() method opens the Print Dialog Box, which lets the user to select preferred printing options.

- *If the string begins with "0x", the radix is 16 (hexadecimal)*
- *If the string begins with "0", the radix is 8 (octal). This feature is deprecated*
- *If the string begins with any other value, the radix is 10 (decimal)*

8. What will be the radix value if the string begins with 0x?
a) 13
b) 14
c) 15
d) 16

Answer: d
Explanation: If the string begins with ***0x****, then the radix value will be* ***16****. If the string begins with "0", the radix is 8 (octal). If the string begins with any other value, the radix is 10 (decimal).*

9. What is the function of the parseInt() method?
a) Parses a data type and stores in an integer
b) Parses a string and returns an integer
c) Parses an integer and returns a string
d) Parses a string and return an object

Answer: b
Explanation: The function ***parseInt()*** *method parses a string and returns an integer. If the string begins with 0x, then the radix value will be 16. If the string begins with "0", the radix is 8 (octal).*

10. What does it indicate when the radix value is 16?
a) String begins with 0x
b) String begins with 0
c) String begins with 0P
d) String begins with FF

Answer: a
Explanation: If the string begins with "0x", the radix is 16 (hexadecimal). If the string begins with "0", the radix is 8 (octal).

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

```
<p id="demo"></p>
<script>
var str = "HELLO, LOOK AT YOU!";
var patt1 = /\BLO/;
var result = str.search(patt1);
document.getElementById("demo").innerHTML = result;
</script>
```

a) 4
b) 7
c) 3
d) 1

Answer: c
Explanation: The \B metacharacter is used to find a match, but where it is NOT at the beginning/end of a word. It is found in the regex library.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "welcome to.\0JavaScript test.";
   var patt1 = /\0/;
   var result = str.search(patt1);
   document.getElementById("demo").innerHTML = result;
}
</script>
```

a) 12
b) 13
c) 10
d) 11

Answer: d
Explanation: The \0 metacharacter is used to find NUL character. \0 returns the position where the NUL character was found. If no match is found, it returns -1.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "hello\f world.";
   var patt1 = /\f/;
   var result = str.search(patt1);
   document.getElementById("demo").innerHTML = result;
}
</script>
```

a) 6
b) 5
c) 8
d) -1

Answer: b
Explanation: The \f metacharacter is used to find a form feed character. \f returns the position where the form feed character was found. If no match is found, it returns -1.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "hello \r world.";
   var patt1 = /\r/;
   var result = str.search(patt1);
   document.getElementById("demo").innerHTML = result;
}
</script>
```

a) 6
b) 5
c) 3
d) -1

Answer: a
Explanation: The \r metacharacter is used to find a carriage return character. \r returns the position where the carriage

return character was found. If no match is found, it returns -1.

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

```
<p id="demo"></p>
<script>
function myFunction()
{
   var str = "hello to \r world.";
   var patt1 = /\t/;
   var result = str.search(patt1);
   document.getElementById("demo").innerHTML = result;
}
</script>
```

a) 6
b) 8
c) 9
d) -1

Answer: c
Explanation: The \t metacharacter is used to find a tab character. \t returns the position where the tab character was found. If no match is found, it returns -1.

1. What is the purpose of the radix parameter in the parseInt() method?
a) Numeral system not to be used
b) Numeral system to be used
c) Conversion mode
d) Parsing mode

Answer: b
Explanation: function parseInt() method parses a string and returns an integer. The radix parameter is used to specify which numeral system to be used, for example, a radix of 16 (hexadecimal) indicates that the number in the string should be parsed from a hexadecimal number to a decimal number.

2. What will be the radix value of the parseInt() method when the string begins with 0?
a) 6
b) 7
c) 8
d) 9

Answer: c
Explanation: If the string begins with 0x, then the radix value will be 16. If the string begins with "0", the radix is 8 (octal). This feature is deprecated.

3. What is the purpose of the method JSON.*parse()*?
a) Parses a string to integer
b) Parses a string to JSON
c) Parses a string from JSON to JSON2
d) Parses integer to string

Answer: b
*Explanation: The **JSON.parse()** method parses a string as JSON, optionally transforming the value produced by parsing. The function parseInt() method parses a string and returns an integer.*

4. What is the return type of the method parseInt()?
a) String
b) Float
c) Integer
d) Date

Answer: c
*Explanation: The function parseInt() method parses a string and returns an integer. The method **parseInt()** returns an integer.*

5. What are the parameters of the method JSON.*parse()*?
a) text
b) reviver
c) both text and reviver
d) object

Answer: c
Explanation: The JSON.parse() method can optionally transform the result with a function. The parameters of the method JSON.parse() are :

- ***text** : The string to parse as JSON.*
- ***reviver** : If a function, prescribes how the value originally produced by parsing is transformed, before being returned.*

6. What will be the radix value of the parseInt() method when the string begins with any other value other than 0x and 0?
a) 8
b) 9
c) 10
d) 11

Answer: c
Explanation: The radix value will be 10 when the string of the method parseInt() begins with any other value other than 0x and 0. If the string begins with 0x, then the radix value will be 16.

7. What kind of an exception will be thrown if the string to parse is not valid JSON?
a) SyntaxError
b) ArrayOutOfBoundException
c) Both SyntaxError and ArrayOutOfBoundException
d) Compilation error

Answer: a
*Explanation: The JSON.parse() method parses a string and returns a JavaScript object. The method **JSON.parse()**throws a SyntaxError exception if the string to parse is not valid JSON.*

8. Which of the following is a JavaScript Compressor?
a) Esprima
b) UgilifyJS
c) Acron
d) Compressify

Answer: b
*Explanation: **UgilifyJS** is a JavaScript compressor/minifier written in JavaScript. It also contains tools that allow one to automate working with JavaScript code.*

9. What is the purpose of the UgilifyJS?
a) Exposes a simple API
b) Minification
c) Both Exposes a simple API and Minification
d) Compilation

Answer: c
*Explanation: UglifyJS is a JavaScript compressor/minifier written in JavaScript. **UglifyJS2** is an excellent tool to help you minify your JavaScript! It's a tried and tested tool, used by libraries such as jQuery.*

10. Which of the following is the fastest JavaScript parser?
a) JSLint
b) Esprima
c) Acron
d) Traceur

Answer: c
Explanation: Acorn is a JavaScript parser written in JavaScript. : Acron is the fastest JavaScript parser written in JavaScript which takes only 96.9ms when compared to all the JavaScript parsers available. ***Acron*** *is the fastest JavaScript parser written in JavaScript which takes only 96.9ms when compared to all the JavaScript parsers available.*

1. How many default number methods are available in JavaScript?
a) 5
b) 6
c) 7
d) 8

Answer: c
Explanation: There are a total of 7 default number methods in JavaScript namely:

1. *constructor()*
2. *toExponential()*
3. *toFixed()*
4. *toLocaleString()*
5. *toPrecision()*
6. *toString()*
7. *valueOf()*

2. What is the observer effect?
a) Observing influences outcome
b) Observing never influences outcome
c) Observing and the outcome are independent
d) Observing and the outcome are not related

Answer: a
Explanation: The ***observer effect*** *says that simply observing an act influences its outcome. This is often the result of instruments that, by necessity, alter the state of what they measure in some manner.*

3. Which of the following are ways to benchmark code?
a) Timing the code
b) Calculating the number of operations performed
c) Timing the code & Calculating the number of operations performed
d) Calculating the number of lines

Answer: c
Explanation: There are a couple of ways to benchmark code—either by timing it or by calculating the number of operations performed during execution. Both of them help in calculating how efficient and modular the code.

4. How to calculate the run time of a code?
a) Start time – End time
b) End time – Start time
c) Start time/ End time
d) Start time * End time

Answer: b
Explanation: Run-time statements use the static resources created by compile-time statements, but can also create, use, and destroy dynamic resources at run time. The run time of code can be calculated as:
run time = end time – start time

5. According to the workflow of a runtime logging, what happens after calculating the run time?
a) Display to screen
b) Log to server
c) Either Display to screen or Log to server
d) Server to log

Answer: c
Explanation: According to the workflow of a runtime logging, after calculating the run time, decision boxes are kept that will check to either display to the debug screen or log to the server.

6. Which of the following object is used to post the data to an external process, say savePerfData?
a) XML
b) XBT
c) XHR
d) XTR

Answer: c
*Explanation: XHR(XMLHttpRequest) is an API in the form of an object whose methods transfer data between a web browser and a web server. The benchmarking process will call the external process. We use **XHR object** to post the data to the external process.*

7. Where does the external process save the result of the test?
a) Normal file
b) Flat file
c) Folder
d) .exe file

Answer: b
*Explanation: A flat file consists of a single table of data. It allows the user to specify data attributes, such as columns and data types table by table, and stores those attributes separate from applications. The external process, say **savePerfData** should in turn save the results of the test in a **flat file**.*

8. Which function is used to start the time logging?
a) startTimeLogging()
b) start()
c) Loggingstart()
d) startLogging()

Answer: a
*Explanation: The function **startTimeLogging()** is used to start the time logging. The startTimeLogging() method captures the timing data for ad hoc, etc for referencing an uncached document.location.*

9. Which of the following attribute of form tag is not set by using document object in JavaScript?
a) Target
b) Enctype
c) Action
d) Logging

Answer: d
Explanation: Target, enctype & action can be set by using the document object in JavaScript. The Document object has various properties that refer to other objects which allow access to and modification of document content.

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

```
<script type="text/javascript">
var name1 = "Sanfoundry";
function DisplayName () {
var name2 = " The Best";
document.write(name1+name2);
}
</script>
```

a) SanfoundryThe Best
b) Sanfoundry The Best
c) Object required error
d) Javascript Error

Answer: b
Explanation: The write() method writes HTML expressions or JavaScript code to a document.The write() method is mostly used for testing: If it is used after an HTML document is fully loaded, it will delete all existing HTML. The output for the above code is ***Sanfoundry The Best****.*

1. What type of scope is present in JavaScript?
a) Lexical
b) Literal
c) Both Lexical and Literal
d) Local

Answer: a
Explanation: Lexical scoping (sometimes known as static scoping) is a convention used with many programming languages that sets the scope (range of functionality) of a variable so that it may only be called (referenced) from within the block of code in which it is defined.

2. Which function is used to stop the time logging?
a) stopTimeLogging()
b) stop()
c) finish()
d) abort()

Answer: a
Explanation: The function stopTimeLogging() is used to stop the time logging. The stopTimeLogging() stops the timing data for ad hoc, etc for referencing an uncached document.location.

3. The lexical scooping in JavaScript is based on which of the following?
a) Segments
b) Blocks
c) Functions
d) Entire code

Answer: c
Explanation: Lexical scoping (sometimes known as static scoping) is a convention used with many programming languages that sets the scope (range of functionality) of a variable so that it may only be called (referenced) from within the block of code in which it is defined. JavaScript has lexical scoping based on functions but not blocks.

4. What is the return data type of the property accept that belongs to the Input element?
a) integer
b) string
c) boolean
d) float

Answer: b
Explanation: The accept attribute is used to define the types of files that the control can select. When type is "file", this property is a comma-separated list of MIME types that specify the types of files that may be selected. The strings "audio/", "video/*", and "image/*" are also legal.*

5. What is the purpose of the property width belonging to the ImageData element?
a) Number of data

b) Number of pixels
c) Number of pixels per row of data
d) Number of columns

Answer: c
Explanation: Image data element indicates the intrinsic width of the image, in CSS pixels. The ***width*** *property is used to retrieve or store the number of pixels per row of data.*

6. What is the purpose of the adoptNode() method?
a) Removes node
b) Makes it ready for insertion
c) Removes node and Makes it ready for insertion
d) Changes the node

Answer: c
Explanation: This method removes the node from whatever document it is currently part of and changes its ownerDocument property to this document, making it ready for insertion into this document. The adoptNode() method adopts a node from another document. The adopted node can be of all node types.

7. What is the function of the method importNode()?
a) Copies without removing
b) Copies and removes
c) Only copies
d) Only removes

Answer: a
Explanation: The importNode() method imports a node from another document. The imported node can be of all node types. The method ***importNode()*** *copies a node from another document without removing it.*

8. How can you create a node for comment in JavaScript?
a) comment()
b) createComment()
c) comm()
d) create()

Answer: b
Explanation: The comment node can be created using the method Comment ***createComment(string data)****. The createComment() method creates a Comment node with the specified text.*

9. How can you dispatch a synthetic event object?
a) dispatchEvent()
b) dispatch()
c) dispatch(Event)
d) eventdispatch()

Answer: a
Explanation: When you have created and initialized a synthetic event object, you can dispatch it using the ***dispatchEvent()*** *method of EventTarget. The dispatchEvent method throws UNSPECIFIED_EVENT_TYPE_ERR if the event's type was not specified by initializing the event before the method was called, or if the event's type is null or an empty string.*

10. How do you specifically execute a command in JavaScript?
a) execcommand()
b) exec(command)
c) execCommand()
d) exec(command id)

Answer: c

*Explanation: The command can be executed using boolean **execCommand(string commandId, [boolean showUI, [string value]])**. The execCommand() method executes the specified command for the selected part of an editable section.*

1. What is necessary when we need to create a new field in craft?
a) Type of the input
b) Type of the output
c) Type of the field
d) Type of argument

Answer: c
Explanation: Whenever someone creates a new field in Craft, they must specify what type of field it is. They are organized into Field Groups for convenience, but Field Groups have very little relevance anywhere else in the system.

2. What does the getInputHtml() return?
a) Input
b) Fieldtype's input HTML
c) Array
d) Value

Answer: b
*Explanation: The method **getInputHtml()** returns a fieldtype's input HTML. It accepts two arguments: $name and $value.*

3. How many arguments does the getInputHtml() accept?
a) 1
b) 2
c) 3
d) 4

Answer: b
*Explanation: The method getInputHtml() returns a fieldtype's input HTML. The method **getInputHtml(0** accepts two arguments: $name and $value. **$name** is the name you should assign your HTML input name= attribute, and **$value** is the field's current value (either from the DB, or the POST data if there was a validation error).*

4. Which is the method used to process on the input post data before it is saved to the database?
a) prep()
b) settings()
c) defineSettings()
d) prepSettings()

Answer: d
*Explanation: defineSettings() method returns an array whose keys define the setting names, and values define the parameters (the type of value, etc). If you need to do any processing on your settings' post data before they're saved to the database's content table, you can do it with the **prepSettings()** method.*

5. What is the purpose of the parameter $name ?
a) Document Name
b) Input Name
c) Output Name
d) ID

Answer: b
*Explanation: getinputhtml() accepts two arguments: $name and $value. The parameter **$name** is the name you should assign your HTML input name= attribute.*

6. What does the method defineSettings() return?
a) Array of settings name

b) Array of hash key
c) Array of strings
d) Array of objects

Answer: a
*Explanation: The **defineSettings()** method returns an array whose keys define the setting names, and values define the parameters (the type of value, etc). If you need to do any processing on your settings' post data before they're saved to the database's content table, you can do it with the prepSettings() method.*

7. What is the purpose of the parameter $value?
a) Field's expected value
b) Field's previous value
c) Field's current value
d) Field's probability value

Answer: c
*Explanation: getinputhtml() accepts two arguments: $name and $value. The parameter **$value** is the field's current value (either from the DB, or the POST data if there was a validation error).*

8. When does the defineContentAttribute() method return a false?
a) Data stored in different table
b) Data stored in its own table
c) Data is not stored at all
d) Data is not obtained

Answer: b
*Explanation: If your fieldtype is storing data in its own table, and doesn't have any use for a column within the main content table, you may also set **defineContentAttribute()** to return false. By default, BaseFieldType sets the column to VARCHAR(255), but you can override that with defineContentAttribute().*

9. How many events does the BaseFieldType provide?
a) 1
b) 2
c) 3
d) 4

Answer: c
*Explanation: **BaseFieldType** provides three events that you can latch code onto:*

- ***onBeforeSave()** : Called right before a field is saved.*
- ***onAfterSave()** : Called right after a field is saved, and **$this->model->id** is set.*
- ***onAfterElementSave()** : Called right after an element is saved, and **$this->element->id** is set.*

10. Which method is called right before a field is saved?
a) onBeforeSave()
b) BeforeSave()
c) SaveBefore()
d) onSave()

Answer: a
*Explanation: BaseFieldType provides three events that you can latch code onto:onBeforeSave(), onAfterSave() and onAfterElementSave(). The method **onBeforeSave()** is called right before a field is saved.*

1. Which of the following are JavaScript logging library?
a) log
b) loglevel
c) leveljavascript
d) jslogjava

Answer: b
*Explanation: **loglevel** is a JavaScript logging library that is reliable and extremely easy-to-use over the various console.log() methods. loglevel is a lightweight, minimal and convenient JavaScript logging library which provides a reliable and extremely easy-to-use layer over the various console.log() methods that are often available, with none of their downsides.*

2. What is the significance of the JavaScript logging library loglevel?
a) Lightweight
b) Unreliable
c) Minimal usage
d) Inconvenient

Answer: a
*Explanation: **loglevel** is a lightweight, minimal and convenient JavaScript logging library which provides a reliable and extremely easy-to-use layer over the various console.log() methods that are often available, with none of their downsides. loglevel makes these methods safe to use by gracefully handling the cases where they don't exist and also lets you filter the output of these, to only show messages at warn level or above.*

3. Which of the following browsers support the usage of the JavaScript logging library log4javascript?
a) IE
b) Safari
c) Opera
d) All of the mentioned

Answer: d
Explanation: Apache Log4j is a Java-based logging utility. It was originally written by Ceki Gülcü and is part of the Apache Logging Services project of the Apache Software Foundation. Since a major use of JavaScript logging is in browser testing, log4javascript is tested and works across all recent major browsers, including:

- *Internet Explorer 5+ for Windows*
- *Mozilla, Firefox, Netscape*
- *Google Chrome*
- *Safari 1.3+*
- *Opera 7.5+*
- *WebKit*
- *Konqueror 3.4+*

4. What is the significance of the JavaScript logging library log4javascript?
a) Fully featured
b) Easy to use
c) Fully featured and easy to use
d) Easily accessible

Answer: c
Explanation: Apache Log4j is a Java-based logging utility.log4javascript is a fully-featured, easy-to-use JavaScript logging framework based on the Java logging framework log4j. Its purpose is to provide JavaScript developers with a familiar, robust and flexible logging framework with which to debug JavaScript applications.

5. Which of the following JavaScript logging framework lets one to toggle the plane?
a) Lumberjack
b) Log Hound
c) jsTracer
d) fvlogger

Answer: a
*Explanation: **Lumberjack** is a JavaScript utility that hijacks the browser console, makes it more beautiful and enhances it so logs can be split into more manageable chunks. Lumberjack also saves everything that is logged (so you can review*

later) and colorizes your logs (where supported) to increase legibility.

6. Which of the following JavaScript logging tool is also a debugging tool?
a) Blackbird
b) Log Hound
c) jsTracer
d) fvlogger

Answer: c
*Explanation: **jsTracer** is a simple yet feature rich JavaScript logging and debugging tool. It is especially useful for debugging what has been coined "Web 2.0" code. It helps in dynamic tracing for JavaScript, written in JavaScript, providing you insight into your live nodejs applications, at the process, machine, or cluster level.*

7. Which of the following JavaScript logging tool is standalone?
a) Blackbird
b) Log Hound
c) jsTracer
d) fvlogger

Answer: b
*Explanation: **Log Hound** is a tool that was designed for finding frequent patterns from event log data sets with the help of a breadth-first frequent itemset mining algorithm. LogHound can be employed for mining frequent line patterns from raw event logs, e.g. Log Hound is a standalone JavaScript logging utility that allows you to log messages during execution of JavaScript code.*

8. Which of the following is not an appenders?
a) DummyAppender
b) ConsoleAppender
c) FileAppender
d) DeleteAppender

Answer: d
Explanation: There are several ways to log using "appender"s. Some of the current available Appenders are:

- ***DummyAppender**: log nothing.*
- ***ConsoleAppender**: open a new window in the browser or an inline div element and insert log messages in real time.*
- ***WindowsEventAppender**: send log messages in the MS Windows event manager (Internet Explorer only).*
- ***FileAppender**: write log messages in a local file on the client (IE and Mozilla).*
- ***AjaxAppender**: allow to send log messages to a remote server with asynchronous HTTP request.*
- ***MetatagAppender**: add log messages as meta data.*
- ***JavaScript Console Appenders** for Opera, Mozilla and Safari.*

9. Which of the following reads the textual contents of a URL and returns as a string?
a) spawn(f);
b) load(filename,…);
c) readFile(file);
d) readUrl(url);

Answer: d
Explanation: readUrl() opens an input connection to the given string url, read all its bytes and convert them to a string using the specified character coding or default character coding if an explicit coding argument is not given.

10. When does JSNLog have server-side extensions?
a) Use of .NET
b) Use of Java
c) Both .NET and Java
d) Use of C#

Answer: a
Explanation: If you use .Net, ***JSNLog*** *has server side extensions that receive log data from the browser and have it logged on the server. And it lets you configure your loggers in your web.config. It includes a large class library named as Framework Class Library and provides language interoperability across several programming languages.*

1. Which of the following property gives access to the JavaScript memory usage data?
a) performance.memory
b) memory(performance)
c) performance(memory)
d) performance()

Answer: a
Explanation: The property ***performance.memory*** *gives access to the JavaScript memory usage data. It returns an object of type ObjectObject.*

2. What is the purpose of the timing property in the window.*performance* object?
a) Time of navigation event
b) Time of page load event
c) Time of navigation and page load event
d) Time of scrolling

Answer: c
Explanation: Each ***performance.timing*** *attribute shows the time of a navigation event (such as when the page was requested) or page load event (such as when the DOM began loading), measured in milliseconds. The legacy Performance.timing read-only property returns a PerformanceTiming object containing latency-related performance information.*

3. Which of the following property is associated with the Response event?
a) responseStart
b) responseEnd
c) both responseStart and responseEnd
d) responsiveStart

Answer: c
Explanation: PerformanceTiming.responseStart read-only property returns an unsigned long long representing the moment in time (in milliseconds since the UNIX epoch) when the browser received the first byte of the response from the server, cache, or local resource. The properties associated with the ***Response*** *event are:*

- *responseStart*
- *responseEnd.*

4. Which of the following computation is correct to calculate the time taken for page load once the page is received from the server?
a) responseEnd-loadEventEnd
b) loadEventEnd-responseEnd
c) loadEventEnd/responseEnd
d) responseEnd/loadEventEnd

Answer: b
Explanation: The legacy PerformanceTiming.loadEventEnd read-only property returns an unsigned long representing the moment, in milliseconds since the UNIX epoch, when the load event handler terminated, that is when the load event is completed. The time taken for page load once the page is received from the server: ***loadEventEnd-responseEnd****.*

5. Which of the following property is associated with the Processing event?
a) domComplete
b) domContentLoaded
c) domInteractive
d) domload

Answer: d
*Explanation: PerformanceTiming.domComplete read-only property returns an unsigned long long representing the moment, in miliseconds since the UNIX epoch, when the parser finished its work on the main document. The following properties are associated with the **Processing** event:*

- *domComplete*
- *domContentLoaded*
- *domInteractive*
- *domLoading*
- *unLoadEnd.*

6. What does it indicate when the type attribute of the *navigation* object is set to 2?
a) Navigation by moving back through history
b) Navigation by moving forward through history
c) Navigation by moving back & forward through history
d) Navigation by moving in favorites

Answer: c
*Explanation: The navigator object contains information about the browser. When the **type** attribute of the **navigation** object is set to 2, it means that the navigation is done by moving back or forward through history.*

7. What does the method Performance.*now()* return?
a) DOMTimeStamp
b) DOMHighResTimeStamp
c) DOM\Stamp
d) TimeStamp

Answer: b
*Explanation: The **Performance.now()** method returns a **DOMHighResTimeStamp**, measured in milliseconds, accurate to one thousandth of a millisecond equal to the number of milliseconds since the **PerformanceTiming.navigationStart** property and the call to the method. The returned value represents the time elapsed since the time origin.*

8. Which of the following is a read-only property?
a) PerformanceTiming.navigationStart
b) PerformanceTiming.fetchStart
c) PerformanceTiming.navigationStart & PerformanceTiming.fetchStart
d) PerformanceTiming.responseStart

Answer: c
*Explanation: PerformanceTiming.navigationStart read-only property returns an unsigned long long representing the moment, in miliseconds since the UNIX epoch, right after the prompt for unload terminates on the previous document in the same browsing context. If there is no previous document, this value will be the same as PerformanceTiming.fetchStart. Both **PerformanceTiming.navigationStart** and also the **PerformanceTiming.fetchStart** are a read-only properties.*

9. Which of the following is an interface?
a) Time
b) Timing
c) Performance
d) PerformanceTiming

Answer: d
*Explanation: Performance.timing read-only property returns a PerformanceTiming object containing latency-related performance information. **PerformanceTiming** is an interface in JavaScript.*

10. How many properties are associated with the Response event?
a) 1
b) 2

c) 3
d) 4

Answer: b
*Explanation: There are a total of 2 properties associated with the **Response** event namely:*

- *PerformanceTiming.responseEnd*
- *PerformanceTiming.responseStart.*

PerformanceTiming.responseStart read-only property returns an unsigned long long representing the moment in time (in milliseconds since the UNIX epoch) when the browser received the first byte of the response from the server, cache, or local resource.

1. How many properties does a prototype object have?
a) 6
b) 7
c) 8
d) 9

Answer: b
*Explanation: The prototype is an object that is associated with every functions and objects by default in JavaScript, where function's prototype property is accessible and modifiable and object's prototype property (aka attribute) is not visible. There are a total of 7 properties in the **prototype** object namely:*

1. *perceivedTime*
2. *redirectTime*
3. *cacheTime*
4. *dnsLookupTime*
5. *tcpConnectionTime*
6. *roundTripTime*
7. *pageRenderTime.*

2. Which of the following does not serialize the undefined values or objects within an object?
a) JSON.string
b) JSON
c) JSON.stringify
d) JSON.change()

Answer: c
*Explanation: **JSON.stringify** does not serialize undefined values or functions within an object. JSON.stringify() converts a JavaScript object into a string.*

3. How many properties are there in the interface PerformanceTiming?
a) 21
b) 22
c) 23
d) 24

Answer: a
*Explanation: Performance.timing read-only property returns a PerformanceTiming object containing latency-related performance information. There are a total of 23 properties associated with the interface **PerformanceTiming**.*

4. How many properties are there in window.*performance* object?
a) 1
b) 4
c) 2
d) 3

Answer: c
Explanation: The Window interface's performance property returns a Performance object, which can be used to gather performance information about the current document. There are totally 2 properties associated with the ***window.performance*** *and they are:*

- *navigation*
- *type.*

5. What is the purpose of the navigation property in the window.*performance* object?
a) To which page the user navigated
b) How the user navigated
c) Information about the page
d) Information of the curser

Answer: b
Explanation: The ***navigation*** *tells how the user navigated to the page.*

6. What is the purpose of the property PerformanceTiming.*navigationStart*?
a) Ready to end the navigation
b) Ready to jump the navigation
c) Ready for navigation
d) Ready to changing the navigation

Answer: c
Explanation: The ***PerformanceTiming.navigationStart*** *read-only property returns an unsigned long long representing the moment, in milliseconds since the UNIX epoch, right after the prompt for unload terminates on the previous document in the same browsing context. If there is no previous document, this value will be the same as:* ***PerformanceTiming.fetchStart****.*

7. Which of the following does JSON.*stringify* not serialize?
a) Undefined values
b) Functions within an object
c) Both Undefined values and Functions within an object
d) Functions outside the object

Answer: c
Explanation: ***JSON.stringify*** *does not serialize undefined values or functions within an object. JSON.stringify() converts a JavaScript object into a string.*

8. What is the purpose of the property PerformanceTiming.*fetchStart*?
a) Browser ready to fetch input
b) Browser ready to fetch document
c) Browser ready to fetch summary
d) Browser ready to fetch output

Answer: b
Explanation: The ***PerformanceTiming.fetchStart*** *read-only property returns an unsigned long long representing the moment, in milliseconds since the UNIX epoch, the browser is ready to fetch the document using an HTTP request. If there is no previous document, this value will be the same as PerformanceTiming.fetchStart.*

9. Which of the following property is associated with the Request event?
a) requestStart
b) requestEnd
c) both requestStart and requestEnd
d) requestchange

Answer: a
Explanation: The ***Request*** *event has only one property:* ***requestStart****. The Request event has only one property:*

requestStart.

10. Which of the following API can be used to get the timing without affecting the page loading process?
a) Navigation API
b) Timing API
c) Navigation Timing API
d) Navigate API

Answer: c
Explanation: The Navigation Timing API provides data that can be used to measure the performance of a web site. The timing code is on the page, so it affects how the page loads and the time that takes. The ***Navigation Timing*** *API can be used to get the timing without affecting the page loading process.*

1. How many read-only attributes are present in the navigator object?
a) 1
b) 2
c) 3
d) 4

Answer: b
Explanation: The navigator object contains information about the browser. There are a total of 2 read-only attributes present in the navigator object namely:

1. *redirectCount*
2. *type.*

2. Why are HTTP redirects significant?
a) TCP connection available
b) Complete roundtrip absent
c) Complete roundtrip present
d) TCP connection not available

Answer: b
Explanation: HTTP redirects are significant because they cause a complete roundtrip for each redirect. The original request is returned from the web server as either a 301 or a 302 with the path to the new location.

3. Where does the DNS reply go to in a single HTTP redirect?
a) Browser
b) Client
c) Server
d) DNS Server

Answer: a
Explanation: DNS is a query/response protocol. The client queries information (for example the IP address corresponding to www.google.com) in a single UDP request. The DNS reply goes to the Browser in a single HTTP redirect.

4. How can one access the *redirectCount* property?
a) navigation.redirectCount
b) performance.navigation.redirectCount
c) performance.redirectCount
d) redirectCount

Answer: b
Explanation: The redirectCount property can be accessed as ***performance.navigation.redirectCount****. The PerformanceNavigation.redirectCount read-only property returns an unsigned short representing the number of REDIRECTs done before reaching the page.*

5. How many constant values can the property type be represented?

a) 2
b) 3
c) 4
d) 5

Answer: c
Explanation: The type property sets or returns the value of the type attribute of an <object> element. The type attribute specifies the Internet media type (formerly known as MIME type) of the object. Totally 4 constant values can be represented by the property type.

6. Which of the following constants has the value 255?
a) TYPE_NAVIGATE
b) TYPE_RELOAD
c) TYPE_BACK_FORWARD
d) TYPE_RESERVED

Answer: d
Explanation: TYPE_BACK_FORWARD performs navigation through a history traversal operation. TYPE_RESERVED Has the value of 255 and is a catch-all for any navigation type not defined above.

7. What is the purpose of the constant TYPE_RELOAD?
a) Reload performed
b) Reload not performed
c) Reload must be performed
d) Reload may be performed

Answer: a
*Explanation: **TYPE_RELOAD**: Has the value of 1, indicating that the current page was arrived at via a reload operation. TYPE_RELOAD performs navigation through the reload operation or the location.reload() method.*

8. What does the constant TYPE_BACK_FORWRD indicate?
a) Navigation via browser history
b) Navigation via user request
c) Navigation via URL
d) Navigation via load operation

Answer: a
Explanation: TYPE_BACK_FORWARD performs navigation through a history traversal operation.
***TYPE_BACK_FORWARD** Has the value of 2, indicating that the page was navigated to via the browser history, either using the back or forward buttons or programmatically through the browser's history object.*

9. Which of the following constants hold the value 2?
a) TYPE_NAVIGATE
b) TYPE_RELOAD
c) TYPE_BACK_FORWARD
d) TYPE_RESERVED

Answer: c
Explanation: TYPE_BACK_FORWARD performs navigation through a history traversal operation.
***TYPE_BACK_FORWARD**: Has the value of 2, indicating that the page was navigated to via the browser history, either using the back or forward buttons or programmatically through the browser's history object.*

10. Where does the DNS Lookup direct to ?
a) Browser
b) Client
c) Server
d) DNS Server

Answer: d
Explanation: DNS is a query/response protocol. The client queries an information (for example the IP address corresponding to www.google.com) in a single UDP request. The ***DNS Lookup*** *directs to the* ***DNS Server****.*

1. How many properties are available in a memory object?
a) 1
b) 2
c) 3
d) 4

Answer: c
Explanation: Advanced JavaScript application features such as caches and undo buffers need to know the memory consumption of objects in order to function effectively. There are a total of 3 memory objects namely:

- *jsHeapSizeLimit*
- *totalJsHeapSize*
- *usedJsHeapSize.*

2. What is a heap in JavaScript?
a) Collection of Java objects
b) Collection of JavaScript objects
c) Collection of memory usage values
d) Collection of data stored in memory

Answer: b
Explanation: A heap is a tree-like data structure where each node must be ordered with respect to the value of its children. The heap is the collection of JavaScript objects that the interpreter keeps in resident memory.

3. What is the function of the *memory* object?
a) Gets unused memory details
b) Shows optimization
c) Gets memory usage
d) Memory optimization

Answer: c
Explanation: The JavaScript Memory column represents the JS heap. This column contains two values. The value you're interested in is the live number (the number in parentheses). The memory object is a feature of Chrome that allows us to see the memory usage that Chrome is taking up while running our page.

4. What is the command to access the Heap size limit?
a) performance.memory. SizeLimit
b) performance.memory. jsHeapSizeLimit
c) jsHeapSizeLimit
d) performance.jsHeapSizeLimit

Answer: b
Explanation: Memory contains an object created with MemoryInfo constructor, containing jsHeapSizeLimit, totalJSHeapSize and usedJSHeapSize properties with numerical values. The command performance.memory. jsHeapSizeLimit is used to access the Heap size limit.

5. What is the purpose of *garbage collection*?
a) Removes object with many reference
b) Removes object with reference
c) Removes object with invalid reference
d) Removes object with no reference

Answer: d
Explanation: When the interpreter sees an object in the heap with no object references, it removes that object from the

*heap. This is called **garbage collection**. An object is considered garbage collectible if there are zero references pointing at this object.*

6. What does the *usedJsHeapSize* property indicate?
a) Amount of memory used
b) Amount of memory unused
c) Amount of memory used & unused
d) Amount of memory required

Answer: a
*Explanation: usedJsHeapSize returns an object of type ObjectObject. An object created with MemoryInfo constructor, containing jsHeapSizeLimit, totalJSHeapSize and usedJSHeapSize properties with numerical values. The **usedJsHeapSize** property is the amount of memory that all of the current objects in the heap are using.*

7. Which of the following property indicate the total size of the heap?
a) heapSize
b) totalHeapSize
c) totalJsHeapSize
d) totalHeap

Answer: c
Explanation: usedJsHeapSize stores an object created with MemoryInfo constructor, containing jsHeapSizeLimit, totalJSHeapSize and usedJSHeapSize properties with numerical values.

8. What can be done to monitor the memory usage?
a) Profiling
b) Sequencing
c) Serializing
d) Serializing & Sequencing

Answer: a
*Explanation: Garbage collected languages help developers manage memory by periodically checking which previously allocated pieces of memory can still be "reached" from other parts of the application. **Profiling** allows us to monitor our memory usage.*

9. Which of the following gives the high level breakdown of memory usage?
a) about:memory
b) memory
c) about-memory
d) about::memory

Answer: a
*Explanation: The Memory column represents native memory. DOM nodes are stored in native memory. If this value is increasing, DOM nodes are getting created. Typing **about:memory** into the location bar brings up a page that gives a high-level breakdown of memory usage.*

10. Which keyword must be used to get a more granular insight into the memory usage?
a) verb
b) verbose
c) granule
d) gran

Answer: b
*Explanation: Memory column represents native memory. To get a more granular insight into the memory usage, we must type **about:memory?verbose**.*

1. When does the browser stop rendering the HTML?
a) Inline JavaScript block

b) External JavaScript file
c) Both Inline JavaScript block & External JavaScript file
d) External HTML file

Answer: c
Explanation: When the browser parses the HTML markup, it stops rendering the HTML when it encounters an inline JavaScript block or external JavaScript file. At this point, the user experiences rendering delays.

2. Which of the following handles painting the content on to the screen?
a) Rendering engine
b) JavaScript Interpreter
c) UI Layer
d) Network Layer

Answer: a
Explanation: A rendering engine is a software that draws text and images on the screen. The rendering engine handles painting the content to the screen. When it encounters JavaScript, it hands it off to the JavaScript interpreter.

3. What does the rendering engine do when it encounters JavaScript?
a) Skips the code
b) Continues painting
c) Switches to Javascript Interpreter
d) Restructures the code

Answer: c
Explanation: A rendering engine is a software that draws text and images on the screen. The rendering engine handles painting the content to the screen. When it encounters JavaScript, it hands it off to the JavaScript interpreter.

4. Which of the following runs the JavaScript code?
a) Just In Time compiler
b) JavaScript Interpreter
c) Both Just In Time compiler and JavaScript Interpreter
d) Javascript compiler

Answer: b
Explanation: A JavaScript engine is a computer program that executes JavaScript (JS) code. The first JS engines were mere interpreters, but all relevant modern engines utilize just-in-time compilation for improved performance. The JavaScript Interpreter runs the JavaScript code.

5. Which of the following layer retrieves the content from the network?
a) Transport Layer
b) Application Layer
c) Network Layer
d) Physical Layer

Answer: c
Explanation: The network layer is the third level of the Open Systems Interconnection Model (OSI Model) and the layer that provides data routing paths for network communication. The network layer retrieves the content from the network.

6. Which of the following gets converted to DOM elements by the rendering engine?
a) Tokens
b) Strings
c) Address
d) Characters

Answer: a
Explanation: The string returned by the network layer gets tokenized into meaningful chunks. The rendering engine then takes the tokens and converts them to DOM elements. There are five categories of tokens: 1) constants, 2) identifiers, 3)

operators, 4) separators, and 5) reserved words.

7. Which of the below does not belong to the Render Engine workflow?
a) Paint DOM elements
b) Parse Content
c) Build DOM nodes in render tree
d) Parse identifiers

Answer: d
Explanation: A rendering engine is software that draws text and images on the screen. All of the mentioned belongs to the Render Engine workflow. The Render engine workflow contains:

1. *Parse Content*
2. *Build DOM nodes in render tree*
3. *Layout positioning of DOM elements*
4. *Paint DOM elements.*

8. Which is the next step after retrieving the content in chunks?
a) Paint DOM elements
b) Parse Content
c) Build DOM nodes in render tree
d) Layout positioning of DOM elements

Answer: b
Explanation: After retrieving the content in chunks, the contents will be parsed. The Render engine workflow contains:

1. *Parse Content*
2. *Build DOM nodes in render tree*
3. *Layout positioning of DOM elements*
4. *Paint DOM elements.*

9. What will happen after executing the script?
a) Execute script
b) Layout positioning of DOM elements
c) Paint DOM elements
d) Build DOM nodes in render tree

Answer: d
Explanation: After executing the script, the DOM nodes will be built in the render tree. The Render engine workflow contains:

1. *Parse Content*
2. *Build DOM nodes in render tree*
3. *Layout positioning of DOM elements*
4. *Paint DOM elements.*

10. What would happen if there were no script tags?
a) Build DOM nodes in render tree
b) Layout positioning of DOM elements
c) Paint DOM elements
d) Execute script

Answer: a
Explanation: If there were no script tags, the DOM nodes will be built in the render tree. After executing the script, the DOM nodes will be built in the render tree.

1. What is the purpose of script loading?
a) Load Scripts programmatically

b) Load JavaScript files manually
c) Load JavaScript files programmatically
d) Load Scripts programmatically & manually

Answer: c
Explanation: The script loading loads remote JavaScript files programmatically and allow us to trick the rendering engine. The async attribute is a boolean attribute. When present, it specifies that the script will be executed asynchronously as soon as it is available.

2. What will happen if the browser encounters a script tag without an src attribute?
a) Throws an error
b) Throws an exception
c) Sends it to the compiler
d) Sends it to the interpreter

Answer: d
Explanation: If the browser encounters a script tag without an src attribute, the rendering engine simply passes the code to the JavaScript Interpreter for execution. The src attribute specifies the location (URL) of the external resource.

3. What is the solution to the absence of a script tag without an src attribute?
a) Resend the scripts
b) Create inline JavaScript
c) Attach a javascript file
d) Include a file

Answer: b
*Explanation: The solution to the absence of a script tag without an **src** attribute is to create inline JavaScript to append script tags to the document dynamically, for example:*

```
<script>
var script = window.document.createElement('SCRIPT');
script.src = src;
window.document.getElementsByTagName('HEAD')[0].appendChild(script);
</script>
```

4. How to get a particular value using the tagged name?
a) getElementbyID()
b) getElementsbyName()
c) getElementsbyTagName()
d) getTagName()

Answer: c
*Explanation: The getElementsByTagName() method returns a collection of all elements in the document with the specified tag name, as a NodeList object. The method **getElementsbyTagName()** can be used to get a particular value using the tagged name associated with the document.*

5. What should be the type of script_url?
a) Object
b) String
c) Array
d) Any of the mentioned

Answer: d
Explanation: The type of script_url can be anything that will be compared with the typeof keyword's result. The src attribute specifies the location (URL) of the external resource.

6. What is the purpose of using the *async* attribute in the script tag?
a) Load the script asynchronously
b) Load the script synchronously

c) Load the page asynchronously
d) Load the page synchronously

Answer: a
Explanation: The async option is a native attribute that will tell the browser to load the script asynchronously. The async attribute is a boolean attribute. When present, it specifies that the script will be executed asynchronously as soon as it is available.

7. Why do we need to use an onload event in the script tag after using the async attribute?
a) Invoke code during page loading
b) Invoke code during script loading
c) Invoke code during downloading
d) Invoke code during reloading

Answer: c
Explanation: When using async you don't know when the file will be downloaded, so you can attach an onload event handler to the script tag. This will allow you to invoke or instantiate any code that will need to be run when the file is downloaded:

```
<script src="[URL"] async onload="init();]"></script>
```

8. What is the purpose of the startTimeLogging() method?
a) Start the timer
b) Capture time logging
c) Capture timing data for referencing
d) All of the mentioned

Answer: d
*Explanation: The **startTimeLogging()** method captures the timing data for ad hoc, etc for referencing an uncached document.location. Once the startTimeLogging() method is called, run the code to test.*

9. What is the type of datatype the async attribute optionally accepts?
a) Integer
b) String
c) Boolean
d) Decimal

Answer: c
*Explanation: The async attribute is a boolean attribute. When present, it specifies that the script will be executed asynchronously as soon as it is available. The **async** attribute optionally accepts the boolean datatype of default value as **true**.*

10. What is the method used to create an element in the HTML DOM?
a) createDOMelement()
b) createElement()
c) createDOMElement()
d) create()

Answer: b
*Explanation: The **createElement()** can be used to create an element in the HTML DOM. After the element is created, use the element.appendChild() or element.insertBefore() method to insert it to the document.*

1. What is the purpose of the method createDocumentFragment()?
a) Creates a fragment object
b) Creates a document fragment
c) Creates imaginary node object
d) Create a node fragment

Answer: c
*Explanation: The **createDocumentFragment()** method creates a imaginary Node object, with all the properties and methods of the Node object. DocumentFragments are DOM Nodes. They are never part of the main DOM tree.*

2. What is the default value of the asyc attribute?
a) 0
b) 1
c) False
d) True

Answer: d
*Explanation: The async attribute is a boolean attribute. When present, it specifies that the script will be executed asynchronously as soon as it is available. The async attribute optionally accepts a boolean value and by default holds the value **true**.*

3. What is the method to create a data frame?
a) frame(data)
b) frameData()
c) data.frame()
d) frame.Data()

Answer: c
Explanation: A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

4. What is the purpose of creating a data frame?
a) Hold the page render time
b) Hold the load time
c) Hold the page render time & load time
d) Hold the reload time

Answer: c
Explanation: A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. You can create a data frame to hold the mean page render times for each test URL, and a data frame to hold the mean load time for each test URL.

5. Which of the following navigator object properties is the same in both Netscape and IE?
a) navigator.appCodeName
b) navigator.appName
c) navigator.appVersion
d) appcode.navigator()

Answer: a
*Explanation: The property **navigator.appCodeName** is the same in both Netscape and IE. The appCodeName property returns the code name of the browser.*

6. Which best explains getSelection()?
a) Returns the VALUE of a selected OPTION
b) Returns document.URL of the window in focus
c) Returns the value of cursor-selected text
d) Returns the VALUE of a checked radio input

Answer: c
*Explanation: The Window.getSelection() method returns a Selection object representing the range of text selected by the user or the current position of the caret. The **getSelection()** method returns the value of the cursor-selected text.*

7. Which of the following are client-side JavaScript object?
a) Database

b) Cursor
c) Client
d) FileUpLoad

Answer: d
Explanation: The Input FileUpload object represents an HTML <input> element with type="file". The FileUpLoad is a client-side JavaScript object.

8. What is the purpose of the method localeCompare()?
a) If the reference string comes before or after another string
b) If the reference string is validated
c) If the string is a reference string
d) If the reference string comes first

Answer: a
*Explanation: The localeCompare() method compares two strings in the current locale. The locale is based on the language settings of the browser. The method **localeCompare()** returns a number indicating whether a reference string comes before or after or is the same as the given string in sort order.*

9. When the "end" event fires on EOF when no more data will arrive, which function is called?
a) s.on("data",f);
b) s.on("end",f);
c) s.on("error",f);
d) s.on("default",f);

Answer: b
Explanation: In computing, end-of-file (commonly abbreviated EOF) is a condition in a computer operating system where no more data can be read from a data source. The above code snippet gets "end" event fired on EOF when no more data will arrive.

10. To define each of the set classes as a property of the sets object (namespace) for the module, the statement is ___________
a) sets = sets.AbstractEnumerableSet.extend();
b) sets.SingletonSet = sets.AbstractEnumerableSet.extend(…);
c) sets.SingletonSet = sets.extend(…);
d) sets = sets.extend(…);

Answer: b
Explanation: The Set object lets you store unique values of any type, whether primitive values or object references. The sets object is the namespace for the module, and we define each of the set classes as a property of this object.

1. What is the purpose of lazy loading?
a) Immediate loading is necessary
b) Loading under command
c) Immediate loading is not necessary
d) Loading after a definite interval

Answer: c
*Explanation: Lazy loading is a design pattern commonly used in computer programming to defer initialization of an object until the point at which it is needed. It can contribute to efficiency in the program's operation if properly and appropriately used. **Lazy loading** is a kind of loading in which we don't need our JavaScript code to be available as soon as the page loads.*

2. Where is the external JavaScript placed in the case of lazy loading?
a) After window.onload event
b) Before window.onload event
c) In the header tag
d) In the HTML tag

Answer: a
Explanation: Lazy loading is a kind of loading in which we don't need our JavaScript code to be available as soon as the page loads. When we don't need our JavaScript code to be available as soon as the page loads, we can script-load our external JavaScript after the ***window.onload*** *event.*

3. What are the parameters of the attachEvent function?
a) Function
b) Function, Event
c) Event, Function
d) Event

Answer: c
Explanation: The addEventListener() method attaches an event handler to an element without overwriting existing event handlers. The ***attachEvent*** *function accepts two parameters: the event to attach to, and the function to invoke when the event occurs.*

4. What is the result when the showPerformanceMetrics() is called before loading the remote script?
a) Throws an exception
b) Throws an error
c) It will load by itself
d) Throws an exception and It will load by itself

Answer: b
Explanation: perflogger.showPerformanceMetrics() tells the user about the performance parameters of the script loading. If you try to make the call to ***perfLogger.showPerformanceMetrics()*** *and the script hasn't just loaded but also executed, then you will get an error.*

5. Which of the following is an attribute to the script object?
a) onclick
b) onload
c) onshow
d) onhover

Answer: b
Explanation: The onload event occurs when an object has been loaded. The onload event occurs when an object has been loaded.

6. Which is the method used to add an event listener?
a) addEventListener()
b) addListener()
c) addEvent(Listener)
d) addListener(Event)

Answer: a
Explanation: The addEventListener() method attaches an event handler to the specified element. The ***addEventListener()*** *method is used to add an event listener to the script.*

7. What is the purpose of the domLoading attribute?
a) Document exists
b) Document can load
c) Document has loaded
d) Document begins to load

Answer: d
Explanation: domLoading is the time immediately before the user agent sets the current document readiness to 'loading'. The ***domLoading*** *attribute is used when the document begins to load.*

8. How is the render time calculated?

a) Date.Now()
b) Date.Now() – performance.timing.domLoading
c) performance.domLoading
d) Date.Now() – performance.domLoading

Answer: b
Explanation: The rendering engine parses the HTML and the CSS and displays the parsed content on the screen. The render time is calculated as ***Date.Now() – performance.timing.domLoading****.*

9. The object whose properties are inherited by all instances of the class, and properties whose values are functions behaving like instance methods of the class, is _____________
a) Instance object
b) Constructor object
c) Destructor object
d) Prototype object

Answer: d
Explanation: A prototype is an object that is associated with every functions and object by default in JavaScript, where function's prototype property is accessible and modifiable and object's prototype property (aka attribute) is not visible. The properties of the prototype object are inherited by all instances of the class, and properties whose values are functions behave like instance methods of the class.

10. You can refresh the webpage in JavaScript by using ___________
a) window.reload
b) location.reload
c) window.refresh
d) page.refresh

Answer: b
Explanation: The reload() method is used to reload the current document.The reload() method does the same as the reload button in your browser. One can refresh the webpage in JavaScript by using ***location.reload****.*

1. What is the initial step to set up a CSS Lazy Loading?
a) Fetching data
b) Loading the script
c) Loading the page
d) Adding the event listener

Answer: a
Explanation: The loading attribute allows a browser to defer loading offscreen images and iframes until users scroll near them. The CSS Lazy Loading is begun with fetching the JavaScript and the CSS files.

2. What is being done in the following JavaScript code?

```
<script>
if (window.attachEvent)
window.attachEvent('onload', fetch);
else
window.addEventListener('load', fetch, false);
</script>
```

a) Event and EventListener is created according to the if-else
b) The values are updated
c) The value is called
d) The values are stored

Answer: a
Explanation: In the above code, the event is attached with onload and the fetch function is called. Also, the event listener is created and added if the "if" fails and is stored as load and the fetch function is called.

3. What is the purpose of holding whatever tag you create in the attribute type?
a) To have more information
b) To identify the scripting language
c) To store data
d) To store variable name

Answer: b
Explanation: A variable is created to hold whatever tag you create, and then branch the logic based on the value of type, which identifies it's for JavaScript or for CSS. Hence, the attribute acts as an identifier to the tag.

4. What does the appendChild() method perform?
a) Appends a node in the middle of the index taken as the parameter
b) Appends a node as the first child
c) Appends a node as the last child
d) Replaces and appends at the last node

Answer: c
*Explanation: The **appendChild()** method appends a node as the last child of a node. You can also use this method to move an element from one element to another.*

5. What is being done in the following JavaScript code?

```
<script>
function fetch()
{
   remoteLoader.loadJS("/lab/perfLogger.js", init)
   remoteLoader.loadCSS(["/style/base.css",
   "http://fonts.googleapis.com/css?family=Metrophobi
   c&amp;v2"])
}
</script>
```

a) JS and CSS files are deleted
b) CSS is replaced with JS
c) JS is replaced with CSS
d) JS and CSS files are loaded

Answer: d
Explanation: In the above function, the JavaScript and the CSS files are loaded remotely. The loadURLs method returns a promise which will receive the temporary folder as an argument.

6. What should be the value of the type attribute of a variable if the type of file is CSS?
a) text/js/css
b) text/js
c) text/css
d) text

Answer: c
*Explanation: Since the type of file is CSS, the **type** attribute should hold the value text/css. For javascript file type attribute would hold the vale text/js.*

7. How to lazy load images?
a) Remove the rel attribute
b) Remove the src attribute
c) Make rel = src
d) Make src = rel

Answer: b
Explanation: The way we would lazy load images would be to alter the HTML of the page to remove the contents of the src attribute of each image. We could just move the contents of the src attribute to an attribute in the image tag of our

own design, maybe the rel attribute.

```
<img src="#" rel="[path to image]" />
```

8. What does the rel attribute of a variable have when the type of file is CSS?
a) css
b) stylesheet
c) text/css
d) plainsheet

Answer: b
Explanation: A web style sheet is a form of separation of presentation and content for web design in which the markup (i.e., HTML or XHTML) of a webpage contains the page's semantic content and structure, but does not define its visual layout (style). The ***rel*** *attribute must hold the value* ***rel = 'stylesheet'****.*

9. What is the parameter of the method getElementsbyTagName() if we need to get an image?
a) image
b) src
c) img
d) imageurl

Answer: c
Explanation: To get the source attribute the parameter is src. The method must look like ***getElementsbyTageName("img")*** *if we need to get an image.*

10. How do we stop blocking of loading and executing the perfLogger, a logging type data?
a) Inlining the perfLogger
b) Removing the perfLogger
c) Placing the perfLogger before the "script" tag
d) Placing the perflogger after the "script" tag

Answer: a
Explanation: A performance logging tool that provides file logging with timestamp and memory usage statistics. On calling End() a csv file of logged items will be generated. By inlining the perfLogger, we no longer block the loading and executing of it.

1. What does the interpreter do when you reference variables in other scopes?
a) Traverses the queue
b) Traverses the stack
c) Finds the bugs
d) Traverse the array

Answer: b
Explanation: The interpreter executes the javascript code. Normally when you reference variables in other scopes at the global level, in other namespaces, and so on—the interpreter needs to traverse the stack to get to the variable.

2. The attribute location belongs to which element?
a) document
b) html
c) image
d) pre

Answer: a
Explanation: window.location.href returns the href (URL) of the current page. The attribute ***location*** *belongs to the* ***document*** *element.*

3. What will happen if you reference document.location from within an object?
a) Traverses the queue

b) Finds the bugs
c) Traverses the stack
d) Traverses the array

Answer: c
Explanation: window.location.href returns the href (URL) of the current page. If you reference document.location from within an object, the interpreter will need to go from the function that references the variable, up out of the namespace to the global window scope, down to the document scope, and gets the location value.

4. Why do we need to create locally scoped variables to hold value?
a) To optimize the testing process
b) To increase the speed
c) To minimize memory usage
d) To cache the reference document.location

Answer: d
Explanation: window.location.assign loads a new document. The locally scoped variables are created to cache the reference to document.location.

5. What is the next step after calling the startTimeLogging()?
a) Interpret the code
b) Compile the code
c) Run the code
d) Debug the code

Answer: c
*Explanation: The startTimeLogging() method captures the timing data for ad hoc, etc for referencing an uncached document.location. Once the **startTimeLogging()** method is called, run the code to test.*

6. During the traversing through the stack, where does it go after it goes to the namespace?
a) Window
b) Function
c) Document
d) Location

Answer: a
*Explanation: Namespace in JavaScript is nothing but a single global object which will contain all our functions, methods, variables and all that. During the traversal in the stack manner, after going to the namespace, it goes to the **Window**.*

7. During the traversing through the stack, where does it go after it goes to the window?
a) Namespace
b) Function
c) Document
d) Nowhere

Answer: d
*Explanation: window.location.assign loads a new document. During the traversal in the stack manner, after going to the window, it goes **nowhere**.*

8. During the traversing of the stack when you create a locally scoped variable, where does it go after it goes to the namespace?
a) Window
b) Function
c) Document
d) Location

Answer: b

Explanation: Namespace in JavaScript is nothing but a single global object which will contain all our functions, methods, variables and all that. When you create a locally scoped variable for caching the reference, the traversal reverses in the forward direction and thus, after going to the namespace, it goes to the ***function****.*

9. What is the function used to stop capturing the ad hoc timing ?
a) stopadhoc()
b) stopTimer()
c) stopTimeLogging()
d) stophoc()

Answer: c
Explanation: The startTimeLogging() method captures the timing data for ad hoc, etc for referencing an uncached document.location. In order to stop capturing the ad hoc timing for referencing uncached ***document.location****, we call the method* ***stopTimeLogging()****.*

10. During the traversing of the stack when you create a locally scoped variable, where does it go after it goes to the location?
a) Window
b) Function
c) Document
d) Nowhere

Answer: d
Explanation: window.location.assign loads a new document. When you create a locally scoped variable for caching the reference, the traversal reverses in the forward direction and thus, after going to the location, it goes ***nowhere****.*

1. In how many modes can the closure compiler be run?
a) 1
b) 2
c) 3
d) 4

Answer: b
Explanation: The Closure Compiler is a tool for making JavaScript download and run faster. There are totally 2 modes in which the closure compiler can be run namely:

1. *Simple mode*
2. *Advanced mode.*

2. What is the purpose of the simple mode?
a) Removes whitespaces
b) Does not remove white spaces
c) Removes the unwanted words
d) Removes characters

Answer: a
Explanation: The javascript minifier compresses the javascript code. In ***Simple mode*** *it mostly performs like most other minifiers, removing whitespace, line breaks, and comments.*

3. What is a closure compiler UI?
a) Run time application
b) Web application
c) Standalone application
d) Changes structures

Answer: b
Explanation: The Closure Compiler is a tool for making JavaScript download and runs faster. Instead of compiling from a source language to machine code, it compiles from JavaScript to better JavaScript. A closure compiler UI is a ***web***

application.

4. What are the benefits of closure compiler?
a) Efficiency
b) Code checking
c) Both Efficiency and Code checking
d) Modularity

Answer: d
Explanation: The Closure Compiler is a tool for making JavaScript download and runs faster. Instead of compiling from a source language to machine code, it compiles from JavaScript to better JavaScript. The closure compiler is highly beneficial in terms of:

- ***Efficiency**: The Closure Compiler reduces the size of your JavaScript files and makes them more efficient, helping your application to load faster and reducing your bandwidth needs.*
- ***Code checking**: The Closure Compiler provides warnings for illegal JavaScript and warnings for potentially dangerous operations, helping you to produce JavaScript that is less buggy and easier to maintain.*

5. In what way is the closure compiler efficient?
a) Increases the size of the JavaScript files
b) Reduces the size of the JavaScript files
c) Reduces the execution time
d) Reduces the speed

Answer: b
Explanation: The Closure Compiler is a tool for making JavaScript download and runs faster. The Closure Compiler reduces the size of your JavaScript files and makes them more efficient, helping your application to load faster and reducing your bandwidth needs.

6. In which way can the closure compiler can be used?
a) Open source
b) Run time application
c) Web application
d) Only Open source and Web application

Answer: d
Explanation: Closure compiler parses your JavaScript, analyzes it, removes dead code and rewrites and minimizes what's left. You can use the Closure Compiler as:

- *An open source Java application that you can run from the command line.*
- *A simple web application.*
- *A RESTful API.*

7. What is the purpose of the advanced mode in the closure compiler?
a) Renames variables
b) Renames function
c) Both Renames variables and function
d) Changes structures

Answer: c
*Explanation: In **Advanced mode** it rewrites the JavaScript by renaming variables and functions from longer descriptive names to single letters to save file size, and it inlines functions, coalescing them into single functions wherever it determines that it can.*

8. Why is a closure template used?
a) Statically updating in JavaScript
b) To increase the efficiency and convenience
c) Dynamically generating HTML in Java and JavaScript

d) Changing code

Answer: c
Explanation: Instead of compiling from a source language to machine code, closure templates compiles from JavaScript to better JavaScript. Closure compiler provides with the basic facilities of removing spaces and dead code. Closure Templates are a templating system for dynamically generating HTML in both Java and JavaScript. Because the language was apparently referred to as "Soy" internal to Google, and "Soy" remains in some of the documentation and classes, sometimes Closure Templates are referred to as "Soy Templates".

9. In what way does the closure compiler help in checking the code?
a) Warnings
b) Suddenly aborts
c) Rejects malicious inputs
d) Terminates the execution

Answer: a
Explanation: Closure compiler parses your JavaScript, analyzes it, removes dead code and rewrites and minimizes what's left. The Closure Compiler provides warnings for illegal JavaScript and warnings for potentially dangerous operations, helping you to produce JavaScript that is less buggy and easier to maintain.

10. What is the function of the closure compiler?
a) Download faster
b) Run faster
c) Both Download faster and Run faster
d) Changes the code

Answer: c
Explanation: The Closure Compiler is a tool for making JavaScript download and runs faster. It parses your JavaScript, analyzes it, removes dead code and rewrites and minimizes what's left. It also checks syntax, variable references, and types, and warns about common JavaScript pitfalls.

1. Which of the following is one of the fundamental features of JavaScript?
a) Single-threaded
b) Multi-threaded
c) Both Single-threaded and Multi-threaded
d) Simple-threaded

Answer: a
Explanation: In computer programming, single-threading is the processing of one command at a time. One of the fundamental features of client-side JavaScript is that it is single-threaded: a browser will never run two event handlers at the same time, and it will never trigger a timer while an event handler is running.

2. Which of the following functions are synchronous?
a) load()
b) require()
c) both load() and require()
d) create()

Answer: c
Explanation: JavaScript has two synchronous ***load()*** *and* ***require()****. The load() method loads data from a server and puts the returned data into the selected element.*

3. Why shouldn't JavaScript functions not be too long?
a) User friendliness
b) Tie up event loops
c) Browser becomes unresponsive
d) All of the mentioned

Answer: d
Explanation: When JavaScript code runs for longer than a predefined amount of time than the browser shows a message of unresponsive script. The client-side JavaScript functions must not run too long: otherwise, they will tie up the event loop and the web browser will become unresponsive to user input.

4. The object that looks to the thread that creates it is _______________
a) Window
b) Worker
c) Element
d) Hash

Answer: b
Explanation: A web worker is a JavaScript running in the background, without affecting the performance of the page. The Worker object: this is what a worker looks like from the outside to the thread that creates it.

5. Which of the following is a global object for a new worker?
a) WorkerGlobalScope
b) Worker
c) WorkerScope
d) Window

Answer: a
*Explanation: A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. The **WorkerGlobalScope** is the global object for a new worker, and it is what a worker thread looks like, on the inside, to itself.*

6. Which will be invoked to create a new worker?
a) Function
b) Destructor
c) Constructor
d) Interface

Answer: c
*Explanation: When executing scripts in an HTML page, the page becomes unresponsive until the script is finished. To create a new worker, just use the **Worker()** constructor, passing a URL that specifies the JavaScript code that the worker is to run :*

```
var loader = new Worker("utils/loader.js");
```

7. What will happen if you specify an absolute URL in the Worker constructor?
a) Resolves itself
b) Must have the same origin
c) Must not have the same origin
d) Specify the address

Answer: b
Explanation: If you specify an absolute URL, it must have the same origin (same protocol, host, and port) as that containing document. In Firefox, if you want to use workers in extensions and would like to have access to js-ctypes, you should use the ChromeWorker object instead.

8. How can you send data using a Worker object?
a) postMessage()
b) sendMessage()
c) Message()
d) post()

Answer: a
Explanation: Once you have a Worker object, you can send data to it with postMessage(). The value you pass to

postMessage() will be cloned, and the resulting copy will be delivered to the worker via a message event. postMessage() sends a message — which can consist of any JavaScript object — to the worker's inner scope.

```
loader.postMessage("file.txt");
```

9. Which property is used to manage multiple event handlers?
a) onmessage
b) onerror
c) both onmessage and onerror
d) postmessage

Answer: c
*Explanation: message event is fired when the worker's parent receives a message from its worker which is also available via the onmessage property. You can use **onmessage** and **onerror** properties if you want to manage multiple event handlers.*

10. Which is the function that allows a worker to terminate itself?
a) close()
b) exit()
c) terminate()
d) halt()

Answer: a
*Explanation: Worker.terminate() Immediately terminates the worker. This does not offer the worker an opportunity to finish its operations. The **close()** function allows a worker to terminate itself, and it is similar in effect to the **terminate()** method of a Worker object.*

1. BLOB stands for ______________
a) Binary Little Object
b) Binary Large Object
c) Broken Large Object
d) Binary Small object

Answer: b
Explanation: A Blob is an opaque reference to, or handle for, a chunk of data. The name comes from SQL databases, where it means "Binary Large Object." In JavaScript, Blobs often represent binary data, and they can be large, but neither is required: a Blob could also represent the contents of a small text file.

2. The size of blobs are generally calculated in ____________
a) Meters
b) Kilometers
c) Bytes
d) Pixels

Answer: c
Explanation: A Blob object represents a file-like object of immutable, raw data. Blobs represent data that isn't necessarily in a JavaScript-native format. Blobs are opaque all you can do with them directly is determine their size in bytes, ask for their MIME type, and chop them up into smaller Blobs.

3. The blobs are generally stored in ____________
a) Memory
b) Disk
c) Both Memory and Disk
d) Temporary storage

Answer: c
Explanation: A Blob object represents a file-like object of immutable, raw data. The web browser can store Blobs in memory or on disk, and Blobs can represent really enormous chunks of data (such as video files) that are too large to fit

in main memory without first being broken into smaller pieces with slice().

4. The blobs are broken into smaller pieces using which of the following functions?
a) partition()
b) cut()
c) sliceall()
d) slice()

Answer: d
Explanation: The File interface is based on Blob, inheriting blob functionality and expanding it to support files on the user's system. The web browser can store Blobs in memory or on disk, and Blobs can represent really enormous chunks of data (such as video files) that are too large to fit in main memory without first being broken into smaller pieces with slice().

5. Which algorithm supports blobs?
a) Structured clone algorithm
b) Double buffer algorithm
c) Chen's algorithm
d) Retrieval algorithm

Answer: a
Explanation: Blobs represent data that isn't necessarily in a JavaScript-native format. Blobs are supported by the structured clone algorithm, which means that you can obtain one from another window or thread via the message event.

6. Which database can be used to retrieve blobs?
a) Server-side databases
b) Client-side databases
c) Both Server-side and Client-side databases
d) Temporary databases

Answer: b
Explanation: A Blob object represents a file-like object of immutable, raw data. Blobs represent data that isn't necessarily in a JavaScript-native format. Blobs can be retrieved from client-side databases.

7. Which object can be used to create your own blobs?
a) Creator
b) BlobCreator
c) BlobBuilder
d) BuilderBlob

Answer: c
Explanation: Blobs represent data that isn't necessarily in a JavaScript-native format. You can create your own blobs, using a BlobBuilder object to build them out of strings, ArrayBuffer objects, and other Blobs.

8. Which of the following is a subtype of Blob?
a) Elemental Object
b) Create Object
c) Data Object
d) File Object

Answer: d
Explanation: The File interface is based on Blob, inheriting blob functionality and expanding it to support files on the user's system. The client-side JavaScript File object is a subtype of Blob: a File is just a Blob of data with a name and a modification date.

9. Which method facilitates in uploading a Blob to a server?
a) send()
b) pass()

c) upload()
d) store()

Answer: a
Explanation: The Blob() constructor allows one to create blobs from other objects. You can upload a Blob to a server by passing it to the send() method of an XMLHttpRequest object.

10. A Blob URL can be created using which of the following function?
a) createURL()
b) createObjectURL()
c) designURL()
d) URLCreation()

Answer: b
Explanation: The Blob() constructor allows one to create blobs from other objects. Create a Blob URL with the function createObjectURL(). At the time of this writing, the draft specification and Firefox 4 put this function in a global object named URL, and Chrome and Webkit prefix that new global, calling it webkitURL.

1. Where are memory leaks found?
a) Client side objects
b) Server side objects
c) Both Client side and Server side objects
d) User side objects

Answer: a
Explanation: Memory leaks happen when your code needs to consume memory in your application, which should be released after a given task is completed but isn't. Memory leaks occur when we are developing client-side reusable scripting objects.

2. Which handler is triggered when the content of the document in the window is stable and ready for manipulation?
a) onload
b) manipulate
c) create
d) oncreate

Answer: a
Explanation: One of the most important event handlers is the onload handler of the Window object. It is triggered when the content of the document displayed in the window is stable and ready to be manipulated. JavaScript code is commonly wrapped within an onload event handler.

3. What is the central concept of JavaScript memory management?
a) Reliability
b) Reachability
c) Efficiency
d) Transparency

Answer: b
Explanation: The central concept of JavaScript memory management is a concept of reachability. The main cause for leaks in garbage collected languages are unwanted references.

1. *A distinguished set of objects are assumed to be reachable: these are known as the roots. Typically, these include all the objects referenced from anywhere in the call stack (that is, all local variables and parameters in the functions currently being invoked), and any global variables.*
2. *Objects are kept in memory while they are accessible from roots through a reference or a chain of references.*

4. When does a memory leak happen?
a) Browser doesn't release memory from objects unnecessary
b) Browser releases too many memories

c) Browser releases memory iteratively
d) Browser releases memory quickly

Answer: a
Explanation: Memory leaks happen when your code needs to consume memory in your application, which should be released after a given task is complete but isn't. Memory leak happens when the browser for some reason doesn't release memory from objects which are not needed any more.

5. What will happen when the data of the jQuery.cache is read from an element?
a) Unique number is retrieved as elem[jQuery.expando]
b) Data is read from jQuery.cache[id]
c) Unique number is retrieved as elem[jQuery.expando] & Data is read from jQuery.cache[id]
d) Data is cleared from jQuery.cache[id]

Answer: c
Explanation: jQuery.cache[id] is used to associate handlers and other data with elements. When the data is read from an element:

1. *The element unique number is retrieved from **id = elem[jQuery.expando]**.*
2. *The data is read from **jQuery.cache[id]**.*

6. The style property belongs to which of the following object?
a) Element
b) Window
c) Location
d) Navigation

Answer: a
*Explanation: Each Element object has **style** and **className** properties that allow scripts to specify CSS styles for a document element or to alter the CSS class names that apply to the element.*

7. Which of the following functions are referenced internally?
a) setTimeout
b) setInterval
c) both setTimeout and setInterval
d) clearInterval

Answer: c
*Explanation: setTimeout(function, milliseconds) executes a function, after waiting a specified number of milliseconds. Functions used in **setTimeout/setInterval** are referenced internally and tracked until complete, then cleaned up.*

8. What is the purpose of destroying the functions and objects?
a) Consume unnecessary CPU cycles
b) Prevent the dropping of reference count to 0
c) Centralize the responsibility to clean up
d) All of the mentioned

Answer: d
Explanation: The primary purpose of a destroy function is to centralize the responsibility for cleaning up anything that the object has done that will:

- *Prevent its reference count from dropping to 0 (for example, removing problematic event listeners and callbacks and unregistering from any services).*
- *Consume unnecessary CPU cycles, such as intervals or animations.*

9. When does a cycle occur during memory leak?
a) No reference occurs
b) Two objects reference

c) One object gets referenced
d) Three object gets referenced

Answer: b
Explanation: Old versions of Internet Explorer could not detect cyclic references between DOM nodes and JavaScript code. A cycle happens when two objects reference each other in such a way that both objects retain each other.

10. Which of the following is a way to retain an object in memory?
a) Console Log
b) Closures
c) Destroy objects
d) Clear object

Answer: a
Explanation: Any object inside the timer will hold a reference in order to run that piece of code somewhere in the future without any problems. One particularly obscure way to retain an object in memory is to log it to the console.

1. In which format does JavaScript support external JavaScript?
a) .js
b) .php
c) .js/php
d) .jss

Answer: a
Explanation: JavaScript supports external JavaScript, in the form of ***.js*** *file. This extension can be captured by a number of applications including: Windows Script Host, Dreamweaver MX, Notepad, Netscape Navigator, PavScrip, UltraEdit. The files are generally text files.*

2. What are the two parts of JavaScript libraries?
a) “script” tag and “body” tag
b) External JavaScript and the “script” tag
c) “html” tag and “body” tag
d) ”html” and “style” tag

Answer: b
Explanation: All JavaScript libraries consists of two parts:

- *The external JavaScript itself, which is simply a text file with the containing JavaScript code, saved as a .js file.*
- *A “script” tag referencing the external JavaScript file and defined on the page(s) that uses the library.*

The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

3. Which of the following is added to prefs.js when the console is automatically opened during JavaScript error?
a) user_pref(“javascript.console.open_on_error”, true);
b) user_pref(“javascript.console.open_error “, true);
c) user_pref(“javascript.console.open_error “, false);
d) user_pref(” javascript.console.open_on_error”, false);

Answer: a
Explanation: The prefs.js file, located in the profile folder, is used by Firefox and other Mozilla-based applications to store settings. For instance, when you create a new e-mail account in Thunderbird, the account name and server settings will be stored in the prefs.js file in your Thunderbird profile folder.
The code :

```
user_pref("javascript.console.open_on_error", true);
```

is added to prefs.js when the console is automatically opened during JavaScript error.

4. Which of the following is possible to be referenced in external JavaScript?

a) CPP
b) Cs
c) PHP
d) Python

Answer: c
*Explanation: PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. One of the lesser known sides of external JavaScript is the ability to reference a **PHP file instead of the familiar .js file**.*

5. Which of the following attribute takes the source of the PHP file?
a) img
b) src
c) source
d) psrc

Answer: b
Explanation: Php file is used an scripting language for backend development. The syntax to referencing a PHP file using external JavaScript is consistent enough with what we already know:

```
<script type="text/javascript" src="myscript.php"></script>
```

6. What kind of path can the PHP file be?
a) Absolute
b) Relative
c) Either Absolute or Relative
d) Both Absolute and Relative

Answer: c
Explanation: The PHP code is enclosed in special start and end processing instructions <?php and ?> that allows you to jump into and out of "PHP mode". The PHP file is either an absolute or relative path to a PHP script instead of the usual .js file.

7. Which of the following global variables is used to get parameters?
a) $HTTP_GET_VAR[]
b) $HTTP_GET_VARS()
c) $HTTP_GET_VARS
d) $HTTP_GET_VARS[]

Answer: d
*Explanation: The global variable **$HTTP_GET_VARS[]** is used to get parameters. $HTTP_GET_VARS contains the same initial information, but is not a superglobal.*

8. What is the purpose of the RegExp method test()?
a) Tests for a match in its float parameter
b) Tests for a match in its string parameter
c) Tests for a match in its integer parameter
d) Test for a match in node

Answer: b
Explanation: RegExp stands for regular expression which is an object that describes a pattern of characters.

9. What is the purpose of the function parameter filetype?
a) File type to be expected
b) File type previously got
c) File type that should not be got
d) File type available

Answer: a
*Explanation: The File.type property is an inbuilt function of File WebAPI which gives the media type (MIME) of the file represented by a file object. The function parameter **"filetype"** lets you tell the script what file type to expect before loading.*

10. Which of the following is the method used to add an element to the desired location?
a) add.element()
b) element.add()
c) element.appendChild()
d) addelement()

Answer: c
*Explanation: The method **element.appendChild()** is used to add the element to the desired location within the document tree. The appendChild() method appends a node as the last child of a node.*

1. Which program code doesn't need preprocessing before being run?
a) Text
b) Script
c) Both Text and Script
d) Comment

Answer: b
*Explanation: A scripting or script language is a programming language for a special run-time environment that automates the execution of tasks. A **script** is program code that doesn't need pre-processing (e.g. compiling) before being run.*

2. What is the significance of scripting?
a) Convenient
b) Dynamic
c) Reachable
d) Modular

Answer: b
Explanation: Scripting languages are often interpreted (rather than compiled). Scripting makes Web pages more dynamic) For example, without reloading a new version of a page it may allow modifications to the content of that page, or allow content to be added to or sent from that page. The former has been called DHTML (Dynamic HTML), and the latter AJAX (Asynchronous JavaScript and XML).

3. What is the purpose of XMLHttpRequest?
a) Multiple loading
b) Load content by loading new document
c) Load content without loading new document
d) Repetitive loading

Answer: c
Explanation: The XMLHttpRequest object can be used to request data from a web server. XMLHttpRequest makes it possible to load additional content from the Web without loading a new document, a core component of AJAX.

4. Which API makes the user's current location available to browser-based application?
a) Java API
b) SDL API
c) Object API
d) Geolocation API

Answer: d
Explanation: The Geolocation API allows the user to provide their location to web applications if they so desire. For privacy reasons, the user is asked for permission to report location information.

5. Which of the following ensures additional interactivity mechanism?
a) WAI ARIA
b) Geolocation API
c) Object API
d) SDL API

Answer: a
Explanation: Web Accessibility Initiative – Accessible Rich Internet Applications (WAI-ARIA) is a technical specification published by the World Wide Web Consortium (W3C) that specifies how to increase the accessibility of web pages, in particular, dynamic content, and user interface components developed with Ajax, HTML, JavaScript, and related technologies. ***WAI ARIA*** *offers mechanisms to ensure that this additional interactivity remains usable independent of devices and disabilities.*

6. What is the necessity to have API?
a) Guide to performing activities
b) Describe particular task
c) Both performing activities and Describe particular task
d) Rearrangement of tasks

Answer: b
Explanation: In computer programming, an application programming interface is a set of subroutine definitions, communication protocols, and tools for building software. An ***API*** *may describe the ways in which a particular task is performed.*

7. What is the purpose of the event onAirEvent?
a) Content is played
b) Content is transferred
c) Both Content is played and transferred
d) Content is changed

Answer: a
Explanation: The event ***onAirEvent*** *is fired when the player starts playing content being broadcasted on the channel. The method is called at the time of the creation of adspace.*

8. What is the purpose of the event disconnectionEvent?
a) Player demands for disconnection
b) Player disconnects from the channel
c) There is no user interaction
d) Player demands for reconnection

Answer: b
Explanation: navigator.onLine is a property that maintains a true/false value (true for online, false for offline). This property is updated whenever the user switches into "Offline Mode". The event ***disconnectionEvent*** *is fired when the player disconnects from the channel, whether a result of user interaction or not.*

9. When does one use the event ready?
a) Before loading
b) During loading
c) After loading
d) During reloading

Answer: c
Explanation: The ready event occurs when the DOM (document object model) has been loaded. It is safe to interact with it after this event fired.

10. When does one use the method startOverlays()?
a) Edit ad request
b) Delete ad request

c) Create ad request
d) Modify ad request

Answer: c
*Explanation: A new ad request is made every time when the method **startOverlays()** gets called. It is used at the time of creation of the ad.*

1. What is the reason for avoiding the attributes property in the HTML DOM?
a) Found unnecessary
b) Attributes don't have attributes
c) Attributes have attributes
d) Considered irrelevant

Answer: b
Explanation: When a web page is loaded, the browser creates a Document Object Model of the page. The reason for avoiding the attributes property in the HTML DOM is because Attributes don't have attributes.

2. What is the purpose of the method nodeMap.setNamedItem()?
a) Sets ID
b) Sets attribute node
c) Sets element name
d) Sets element type

Answer: b
*Explanation: The setNamedItem() method adds the specified node to the NamedNodeMap. The method **nodeMap.setNamedItem()** sets the specified attribute node (by name).*

3. How is everything treated in HTML DOM?
a) Node
b) Attributes
c) Elements
d) Arrays

Answer: a
*Explanation: The HTML DOM model is constructed as a tree of Objects. In the HTML DOM (Document Object Model), everything is a **node:***

- *The document itself is a document node.*
- *All HTML elements are element nodes.*
- *All HTML attributes are attribute nodes.*
- *Text inside HTML elements are text nodes.*
- *Comments are comment nodes.*

4. What does the NamedNodeMap object represent in the HTML DOM?
a) Unordered collection of elements
b) Unordered collection of attributes
c) Unordered collection of nodes
d) Unordered collection of arrays

Answer: d
*Explanation: In the HTML DOM, the **NamedNodeMap object** represents an unordered collection of an elements attribute nodes. The nodes in the NamedNodeMap can be accessed through their name.*

5. What is the purpose of the Attr object in the HTML DOM?
a) Used to focus on a particular part of the HTML page
b) HTML Attribute
c) Used to arrange elements
d) CSS attribute

Answer: b
Explanation: When a web page is loaded, the browser creates a Document Object Model of the page. In the HTML DOM, the ***Attr object*** *represents an HTML attribute.*

6. What is the work of the form control elements in the HTML DOM?
a) User Interface elements
b) All the possible elements
c) Debugging elements
d) Collecting elements

Answer: a
Explanation: ***Form control elements:*** *The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes. The elements collection returns a collection of all elements in a form.*

7. How are the objects organized in the HTML DOM?
a) Class-wise
b) Queue
c) Hierarchy
d) Stack

Answer: c
Explanation: The HTML DOM model is constructed as a tree of Objects. The objects are organized in the hierarchy format in the HTML DOM.

8. Which of the following is a type of HTML DOM?
a) Legacy DOM
b) W3C DOM
c) IE4 DOM
d) All of the mentioned

Answer: d
Explanation: IE4 document object model was introduced in Version 4 of Microsoft's Internet Explorer browser. IE 5 and later versions include support for most basic W3C DOM features. All of the above mentioned are types of HTML DOM.

9. What is the purpose of the Legacy DOM?
a) Makes the scripting easier
b) Allows access to few keys and elements
c) Modify the nodes
d) Making the script modular

Answer: b
Explanation: ***The Legacy DOM:*** *This is the model which was introduced in early versions of JavaScript language. It is well supported by all browsers but allows access only to certain key portions of documents, such as forms, form elements, and images.*

10. Which object is the top of the hierarchy?
a) Window Object
b) Document Object
c) Form Object
d) Form Control Elements

Answer: a
Explanation: The DOM is arranged in the form of tree with every node as an object. Window object is the top of the hierarchy. It is the outmost element of the object hierarchy.

1. Which of the following can be implemented using animation?
a) Fireworks
b) Fade Effect

c) Roll-in or Roll-out
d) All of the mentioned

Answer: d
Explanation: JavaScript animations are done by programming gradual changes in an element's style. You can use JavaScript to create a complex animation which includes but not limited to:

- *Fireworks*
- *Fade Effect*
- *Roll-in or Roll-out*
- *Page-in or Page-out*
- *Object movements.*

2. Which is the function that calls another function after a time interval?
a) setTimeout()
b) setTime()
c) callafter()
d) timeSet()

Answer: a
*Explanation: The **setTimeout(function, duration)** calls function after duration milliseconds from now. setInterval(function, milliseconds) is same as setTimeout(), but repeats the execution of the function continuously.*

3. Which function is used to clear the timer value?
a) clearTimervalue()
b) clearTimeout()
c) clear()
d) flush(timer)

Answer: b
Explanation: The clearTimeout() method clears a timer set with the setTimeout() method. The ID value returned by setTimeout() is used as the parameter for the clearTimeout() method.

4. Which is the property used to position the object in the left of the screen?
a) object.position = left
b) object = position.left
c) object.style.left
d) object.shiftleft

Answer: c
*Explanation: The property **object.style.left = distance in pixels or points** sets distance from left edge of the screen. This property specifies the left position of the element including padding, scrollbar, border and margin. Whereas the setTimeout(function, duration) calls function after duration milliseconds from now.*

5. Which is the function used to call a function in every time duration?
a) callafter()
b) setInterval()
c) setTimeout()
d) setTime()

Answer: b
*Explanation: The **setInterval(function, duration)** calls function after every duration milliseconds.*

6. How do we get the DOM object in JavaScript?
a) getElementbyId()
b) getObject()
c) getElement()
d) getNodeobject()

Answer: a
*Explanation: The **getElementbyId()** is used to get the DOM object in JavaScript by simply calling that function associated with the HTML document. The object in this method gets referenced by the id name.*

7. How to assign the image source in JavaScript?
a) image = "url"
b) source("url")
c) image.src = "url"
d) img.src="url"

Answer: c
*Explanation: For accesing any image in webpage its address or url is specified. The image source is defined as **image.src = "/images/html.gif"**.*

8. How do we create and preload an image object in JavaScript?
a) Use new keyword
b) Call Image()
c) Both Use new keyword and Call Image()
d) Set image()

Answer: c
*Explanation: The **Image()** constructor creates and preloads a new image object. The url is specified and the image is preloaded.*

9. Which event handler is triggered when the user's mouse moves onto a link?
a) onMouseOver
b) onMouseOut
c) onMouse
d) onMouseOnto

Answer: a
Explanation: The onmouseover attribute fires when the mouse pointer moves over an element. The onmouseover attribute is often used together with the onmouseout attribute.

10. Which event handler is triggered when the user's mouse moves away from a link?
a) onMouseOver
b) onMouseOut
c) onMouse
d) onMouseOnto

Answer: b
*Explanation: The **onMouseOut** event handler is triggered when the user's mouse moves away from the link. It is often used together with onMouseOver attribute.*

1. In which part does the form validation occur?
a) Client
b) Server
c) Both Client and Server
d) User side

Answer: b
Explanation: The data information from the client side is first sent to the server side. Form validation used to occur at the server after the client had entered all necessary data and then pressed the Submit button.

2. What would happen if the data in the client had been wrong?
a) Sends back the data
b) Waits for correction
c) Sends back the data and Waits for correction

d) Returns the data instantly

Answer: c
Explanation: The only way to obtain data is from the server side which is used to perform further operations on the data. If some of the data that had been entered by the client had been in the wrong form or was simply missing, the server would have to send all the data back to the client and request that the form is resubmitted with correct information.

3. What is the purpose of the basic validation?
a) Data correctness
b) Mere data existence
c) Both Data correctness and Mere data existence
d) Data modification

Answer: b
Explanation: The data entered through the server side is used for validation. First of all, the form must be checked to make sure data was entered into each form field that required it. This would need just loop through each field in the form and check for data.

4. What is the purpose of data format validation?
a) Data correctness
b) Mere data existence
c) Both Data correctness and Mere data existence
d) Data modification

Answer: a
Explanation: The data entered through the server side is used for validation. The data that is entered must be checked for correct form and value. This would need to put more logic to test the correctness of data.

5. Which is the function that is called to validate a data?
a) validate()
b) valid()
c) validation()
d) no predefined function for data validation

Answer: d
Explanation: There is no such function to validate a data but, you can write a function with any name to validate the data. Hence there is no predefined function for data validation.

6. How to find the index of a particular string?
a) position()
b) index()
c) indexOf()
d) positionof()

Answer: c
*Explanation: The **indexOf()** function can be used to find out the index of a particular character or a string. This method return an integer telling the address of the particular character.*

7. How do you focus a particular part of the HTML page in JavaScript?
a) hover()
b) focus()
c) on()
d) focuson()

Answer: b
*Explanation: The **focus()** function can be used to focus a particular part of the HTML page in JavaScript. It sets the element as the active element in the current document. It can be applied to one html element at a single time in a current document.*

8. Which of the following is the child object of the JavaScript navigator?
a) Navicat
b) Plugins
c) NetRight
d) Plugs

Answer: b
Explanation: The JavaScript navigator object is used for browser detection. The JavaScript ***navigator*** *object includes a child object called* ***plugins****.*

9. Which of the following is not the properties of a plug-in entry?
a) name
b) filename
c) mimeTypes
d) value

Answer: d
Explanation: Plugins are reusable portions of code which help you write even less Javascript to achieve specific features on the client side. Each plug-in has an entry in the array. Each entry has the following properties:

- *name – is the name of the plug-in.*
- *filename – is the executable file that was loaded to install the plug-in.*
- *description – is a description of the plug-in, supplied by the developer.*
- *mimeTypes – is an array with one entry for each MIME type supported by the plug-in.*

10. What is the purpose of the mimeTypes property of a plug-in entry?
a) Contains MIME properties
b) Contains MIME sizes
c) Contains MIME types
d) Contains MIME methods

Answer: c
Explanation: MIME stands for Multi-purpose Internet Mail Extensions. ***mimeTypes*** *is an array with one entry for each MIME type supported by the plug-in.*

1. Which side of the image map can be created using JavaScript?
a) Server side
b) Client side
c) Both Server and Client side
d) User side

Answer: b
Explanation: Javascript is used for the creation of client side data. You can use JavaScript to create client side image map.

2. Which is the attribute used to enable the Client-side image map?
a) map
b) area
c) usemap
d) areamap

Answer: c
Explanation: The usemap attribute is associated with a <map> element's name or id attribute, and creates a relationship between the <img> and the <map>.

3. Which are the special tags used for image mapping?
a) map and area
b) map and usemap

c) only map
d) only usemap

Answer: a
Explanation: The usemap attribute specifies an image (or an object) as an image-map (an image-map is an image with clickable areas). The special tags used for image mapping are ***map*** *and* ***area****.*

4. Which is the element that follows the use of "img"?
a) area
b) usemap
c) map
d) any element can follow the use of "img"

Answer: c
Explanation: The usemap attribute is associated with a <map> element's name or id attribute, and creates a relationship between the <img> and the <map>. The "map" element actually creates the map for the image and usually follows directly after the "img " element.

5. What is the purpose of the area element?
a) Area of the text
b) Shape and coordinates of the hotspot
c) Shape and area of the hotspot
d) Coordinates and area

Answer: b
Explanation: The <area> tag defines an area inside an image-map (an image-map is an image with clickable areas). The <area> element is always nested inside a <map> tag. The ***area*** *element specifies the shape and the coordinates that define the boundaries of each clickable hotspot.*

6. Which of the following is not a navigator property?
a) platform[]
b) plugin[]
c) userAgent[]
d) browser[]

Answer: d
Explanation: The navigator object contains information about the browser. All of the above mentioned are the properties of a navigator.

7. What is the purpose of the platform[] property in a navigator?
a) Platform of the script
b) Platform where the image map was designed
c) Platform where the browser was compiled
d) Platform where the plugin was designed

Answer: c
Explanation: The ***platform[]*** *property is a string that contains the platform for which the browser was compiled. "Win32" for 32-bit Windows operating systems.*

8. What is the purpose of the preference method in a navigator?
a) Set Browser preference
b) Set Netscape preference
c) Both Set Browser & Netscape preference
d) Sets user preference

Answer: b
Explanation: The ***preference(name,value)*** *method allows a signed script to get and set some Netscape preferences. If the second parameter is omitted, this method will return the value of the specified preference; otherwise, it sets the value*

Netscape only.

9. Which of the following is not a navigator method?
a) postEnabled
b) reference
c) preference
d) postreference

Answer: c
*Explanation: The **preference(name, value)** method allows a signed script to get and set some Netscape preferences. If the second parameter is omitted, this method will return the value of the specified preference; otherwise, it sets the value Netscape only.*

10. What is the purpose of the userAgent property?
a) Identifying the data
b) Identifying the client
c) Both Identifying the data and client
d) Identifying the dataset

Answer: b
*Explanation: The **userAgent[]** property is a string that contains the code name and version of the browser. This value is sent to the originating server to identify the client. The userAgent property returns the value of the user-agent header sent by the browser to the server.*

1. What is it called when we make a mistake in the script?
a) Error
b) Bug
c) Mistake
d) Debug

Answer: b
Explanation: A bug in a programming language refers to a set of code which results in an error in compilation. A mistake in a script is referred to as a bug.

2. Which of the following is the definition for debugging?
a) Finding bugs
b) Fixing bugs
c) Both Finding & Fixing bugs
d) Clearing bugs

Answer: c
Explanation: Debugging is the process of finding and resolving defects or problems within a computer program that prevent correct operation of computer software or a system. The process of finding and fixing bugs is called debugging and is a normal part of the development process.

3. Where is the error icon shown in the Internet Explorer?
a) Taskbar
b) Status bar
c) Both Taskbar and Status bar
d) Bookmarks bar

Answer: b
Explanation: The error icon option can be enabled in the Tools. To view the console, select Tools –> Error Consol or Web Development.

4. Where is the error icon option available?
a) Tools
b) Help

c) File
d) Edit

Answer: a
*Explanation: The error icon option can be enabled in the **Tools**. To view the console, select Tools –> Error Consol or Web Development.*

5. Which of the following is the window that the Firefox sends the error messages to?
a) Bug Window
b) Error Issues
c) Error Window
d) Error Console

Answer: d
Explanation: The browsers like Firefox, Netscape and Mozilla send error messages to a special window called the JavaScript Console or Error Console. The Error Console is deprecated in Firefox and is now only made available if you set the devtools.errorconsole.enabled preference to true.

6. What is the procedure to view the console in the Firefox?
a) Tools -> Error Console
b) Tools -> Error Window
c) Help -> Error Console
d) Tools -> Bug window

Answer: a
Explanation: The error icon option can be enabled in the Tools. To view the console, select Tools –> Error Console or Web Development.

7. What is the other way of calling the Error Console in Firefox?
a) Error Window
b) JavaScript Console
c) JavaScript Window
d) Error or JavaScript Window

Answer: b
Explanation: The browsers like Firefox, Netscape and Mozilla send error messages to a special window called the JavaScript Console or Error Console. The Error Console is also termed as the JavaScript Console.

8. What kind of error notifications are shown in the console window?
a) Syntax error
b) Runtime error
c) Both Syntax error and Runtime error
d) Compilation error

Answer: c
Explanation: The browsers like Firefox, Netscape and Mozilla send error messages to a special window called the JavaScript Console or Error Console. Error notifications that show up on Console or through Internet Explorer dialog boxes are the result of both syntax and runtime errors. These error notification include the line number at which the error occurred.

9. How do we debug a script?
a) Use of JavaScript Validator
b) Use of JavaScript Debugger
c) Use of JavaScript Validator & Debugger
d) Use of javascript interpreter

Answer: c
*Explanation: Both the **JavaScript Validator** and the **JavaScript Debugger** can be used to debug a script. The debugger*

statement stops the execution of JavaScript, and calls (if available) the debugging function.

10. What is the purpose of a JavaScript debugger?
a) Correction of errors
b) Placing script execution under control
c) Correction of errors & Placing script execution under control
d) Compilation

Answer: b
Explanation: A debugger is an application that places all aspects of script execution under the control of the programmer. Using the debugger statement has the same function as setting a breakpoint in the code. Debuggers provide fine-grained control over the state of the script through an interface that allows you to examine and set values as well as control the flow of execution.

1. What is the framework?
a) User time efficiency
b) Author time efficiency
c) Both User time and Author time efficiency
d) Client time efficiency

Answer: b
Explanation: A framework, or software framework, is a platform for developing software applications. Frameworks are an author-time efficiency, meaning that they make coding tasks much simpler by abstracting the real work that goes into doing those tasks.

2. Which is the function used to loop in an array to view all the values?
a) all()
b) loop()
c) each()
d) every()

Answer: c
Explanation: for.each() function is used for traversing through each values of the array. The method ***jQuery.each(array, function)*** *loops through the array.*

3. Which of the following has a greater benchmark time for looping and JQuery vs core JavaScript in milliseconds?
a) Chrome JQuery
b) Chrome JavaScript
c) Firefox JQuery
d) Firefox JavaScript

Answer: a
Explanation: jQuery Injector allows you to inject jQuery into every frame on a page so that you can use jQuery in the chrome dev console. When we try to compare the average benchmark time for looping and JQuery vs core JavaScript in milliseconds, the ***Chrome JQuery*** *has a greater benchmark time.*

4. Which class provides an interface for invoking JavaScript methods and examining JavaScript properties?
a) ScriptObject
b) JSObject
c) JavaObject
d) Jobject

Answer: b
Explanation: JSObject is the type of JavaScript objects in the JSAPI. When a JavaScript object is passed or returned to Java code, it is wrapped in an instance of JSObject. When a JSObject instance is passed to the JavaScript engine, it is unwrapped back to its original JavaScript object. The JSObject class provides a way to invoke JavaScript methods and examine JavaScript properties.

5. What is the purpose of the function eval?
a) Executes the string as an integer
b) Gets the value of the string
c) Executing string as JavaScript
d) Executing string as an object

Answer: c
Explanation: ***eval*** *is a JavaScript native function that accepts a string and executes the string as JavaScript. The argument of the eval() function is a string. It basically fires up the interpreter and allows the passed-in string to be parsed and interpreted at the time of invocation.*

6. Which of the following has a lesser benchmark time for using JQuery to access DOM versus pure JavaScript in milliseconds?
a) Chrome JQueryDOM_
b) Chrome JSDOM_benc
c) Firefox JQueryDOM_
d) Firefox JSDOM_benc

Answer: d
Explanation: jsdom is a pure-JavaScript implementation of many web standards, notably the WHATWG DOM and HTML Standards, for use with Node.js. When we try to compare the average benchmark time for using JQuery to access DOM versus pure JavaScript in milliseconds, the ***Firefox JSDOM_benc*** *has a lesser benchmark time.*

7. Which is a wrapped Java array, accessed from within JavaScript code?
a) JavaArray
b) JavaClass
c) JavaObject
d) JavaPackage

Answer: a
Explanation: Java array is an object which contains elements of a similar data type. ***JavaArray*** *is accessed from within JavaScript code.*

8. What is the syntax of close method for document object?
a) Close(object)
b) Close(doc)
c) Close(val)
d) Close()

Answer: d
Explanation: In order to close a document object, we need to call Close(). In order to close a document object, we need to call ***Close()****.*

9. How do you find the number with the highest value of x and y?
a) ceil(x,y)
b) top(x,y)
c) Math.ceil(x,y)
d) Math.max(x,y)

Answer: d
Explanation: max function is used for comapring and finding maximum values. It is found in the math library. ***Math.max(x,y)*** *is used to find the highest value of x and y.*

10. How can you find a client's browser name?
a) browser.name
b) navigator.appName
c) client.navName
d) client.name

Answer: b
*Explanation: The client's browser name can be found by using **navigator.appName**. It is found inside the navigator object.*

1. Which of the following is not a JavaScript framework?
a) Rico
b) Prototype
c) Joco
d) DoJo

Answer: d
Explanation: Dojo Toolkit is an open source modular JavaScript library (or more specifically JavaScript toolkit) designed to ease the rapid development of cross-platform, JavaScript/Ajax-based applications and web sites. Rico was an open-source JavaScript library for developing rich internet applications with Ajax.

2. What is the purpose of the Math method toSource()?
a) Returns the string "Math"
b) Sends the source to the Math Library
c) Returns the value of the object
d) Returns an integer value

Answer: a
Explanation: The method Math.toSource() returns the string "Math". But this method does not work with many browsers like IE. The toSource() method returns a string representing the source code of the object.

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

```
var o = new F();
o.constructor === F
```

a) false
b) true
c) 0
d) 1

Answer: b
Explanation: '===' sign is used for comparing the type of values. The result is true if the constructor property specifies the class.

4. How many static methods does a Date object have?
a) 3
b) 5
c) 4
d) 2

Answer: d
Explanation: Date objects are created with the new Date() constructor. The Date object defines two static methods namely Date.parse() and Date.UTC().

5. Which of the following are static methods in JavaScript?
a) Date.parse()
b) Date.UTC()
c) Both Date.parse() and Date.UTC()
d) Date.clear()

Answer: c
*Explanation: Date objects are created with the new Date() constructor. **Date.parse()** parses a string representation of a date and time and returns the internal millisecond representation of that date. **Date.UTC()** Returns the millisecond*

representation of the specified UTC date and time.

6. What will be the work of the getAvg in the following JavaScript function?

```
<script>
function getAvg(){
var avg = 0;
for(var x = 0; x < 200; x++){
avg += x;
}
return(avg/200);
}
```

a) Multiples values from 0 to 200
b) Adds values from 0 to 200
c) Simply traverses with no operation
d) Find the average of 199 numbers

Answer: d
Explanation: For loop is used in the above code for adding the numbers. The above code performs the average calculation of numbers from 0 to 199.

7. If we have an object r and want to know if it is a Range object, we can write ______________
a) r typeof Range
b) r is Range
c) r equals Range
d) r instanceof Range

Answer: d
Explanation: To know the range object r instanceof range is used. The r instanceof Range returns true if r inherits from Range.prototype. The instanceof operator does not actually check whether r was initialized by the Range constructor.

8. What is the property to access the first child of a node?
a) timestamp.Child1
b) timestamp.Child(1)
c) timestamp.Child(0)
d) timestamp.firstChild

Answer: d
*Explanation: The firstChild property returns the first child node of the specified node, as a Node object. The first child of a node can be accessed using the **firstChild** property.*

9. Which of the following is not an object?
a) Element
b) Location
c) Position
d) Window

Answer: c
Explanation: The window object represents an open window in a browser. There is no object called Position.

10. What is the code snippet to change the class and let the stylesheet specify the details?
a) timestamp.className = “highlight”;
b) timestamp.className = “change”;
c) timestamp.className = “specify”;
d) timestamp.className = “move”;

Answer: a
Explanation: “typename” is a keyword in the C++ programming language used when writing templates. The above code snippet changes the class and lets the stylesheet specify the details.