* 1. **Question 1. What Is Javascript Objects?**

**Answer :**

JavaScript is an Object Oriented Programming (OOP) language. A programming language can be called object-oriented if it provides four basic capabilities to developers −

* + - Encapsulation − the capability to store related information, whether data or methods, together in an object.
    - Aggregation − the capability to store one object inside another object.
    - Inheritance − the capability of a class to rely upon another class (or number of classes) for some of its properties and methods.
    - Polymorphism − the capability to write one function or method that works in a variety of different ways.

Objects are composed of attributes. If an attribute contains a function, it is considered to be a method of the object, otherwise the attribute is considered a property.

* 1. **Question 2. What Are Object Properties?**

**Answer :**

Object properties can be any of the three primitive data types, or any of the abstract data types, such as another object. Object properties are usually variables that are used internally in the object's methods, but can also be globally visible variables that are used.

The syntax for adding a property to an object is:

objectName.objectProperty = propertyValue;

For example − The following code gets the document title using the "title" property of the document object.

var str = document.title;

* 1. **Question 3. What Is Object Methods?**

**Answer :**

Methods are the functions that let the object do something or let something be done to it. There is a small difference between a function and a method – at a function is a standalone unit of statements and a method is attached to an object and can be referenced by the this keyword.

Methods are useful for everything from displaying the contents of the object to the screen to performing complex mathematical operations on a group of local properties and parameters.

For example − Following is a simple example to show how to use the write() method of document object to write any content on the document.

document.write("This is test");

* 1. **Question 4. What Is User-defined Objects?**

**Answer :**

**User-Defined Objects:**All user-defined objects and built-in objects are descendants of an object called Object.

**The new Operator:**The new operator is used to create an instance of an object. To create an object, the new operator is followed by the constructor method.

In the following example, the constructor methods are Object(), Array(), and Date(). These constructors are built-in JavaScript functions.

var employee = new Object();  
var books = new Array("C++", "Perl", "Java");  
var day = new Date("August 15, 1947");

**The Object() Constructor:**A constructor is a function that creates and initializes an object. JavaScript provides a special constructor function called Object() to build the object. The return value of the Object() constructor is assigned to a variable.

The variable contains a reference to the new object. The properties assigned to the object are not variables and are not defined with the var keyword.

Example 1: Try the following example; it demonstrates how to create an Object.

<html>  
   <head>  
      <title>User-defined objects</title>  
        
      <script type="text/javascript">  
         var book = new Object();   // Create the object  
         book.subject = "Perl"; // Assign properties to the object  
         book.author  = "Mohtashim";  
      </script>  
        
   </head>  
     
   <body>  
     
      <script type="text/javascript">  
         document.write("Book name is : " + book.subject + "<br>");  
         document.write("Book author is : " + book.author + "<br>");  
      </script>  
     
   </body>  
</html>

* 1. **Question 5. Defining Methods For An Object?**

**Answer :**

The previous examples demonstrate how the constructor creates the object and assigns properties. But we need to complete the definition of an object by assigning methods to it.

Example: Try the following example; it shows how to add a function along with an object.

<html>  
   <head>  
   <title>User-defined objects</title>  
     
      <script type="text/javascript">  
         // Define a function which will work as a method  
         function addPrice(amount){  
            this.price = amount;   
         }  
           
         function book(title, author){  
            this.title = title;  
            this.author  = author;  
            this.addPrice = addPrice; // Assign that method as property.  
         }  
      </script>  
        
   </head>  
   <body>  
     
      <script type="text/javascript">  
         var myBook = new book("Perl", "Mohtashim");  
         myBook.addPrice(100);  
           
         document.write("Book title is : " + myBook.title + "<br>");  
         document.write("Book author is : " + myBook.author + "<br>");  
         document.write("Book price is : " + myBook.price + "<br>");  
      </script>  
        
   </body>  
</html>

* 1. **Question 6. What Is The 'with' Keyword?**

**Answer :**

The ‘with’ keyword is used as a kind of shorthand for referencing an object's properties or methods.

The object specified as an argument to with becomes the default object for the duration of the block that follows. The properties and methods for the object can be used without naming the object.

Syntax: The syntax for with object is as follows --

with (object){  
   properties used without the object name and dot  
}

* 1. **Question 7. What Are The Javascript Native Objects?**

**Answer :**

JavaScript has several built-in or native objects. These objects are accessible anywhere in your program and will work the same way in any browser running in any operating system.

Here is the list of all important JavaScript Native Objects −

* + - JavaScript Number Object
    - JavaScript Boolean Object
    - JavaScript String Object
    - JavaScript Array Object
    - JavaScript Date Object
    - JavaScript Math Object
    - JavaScript RegExp Object
  1. **Question 8. What Is Javascript Number Object?**

**Answer :**

The Number object represents numerical date, either integers or floating-point numbers. In general, you do not need to worry about Number objects because the browser automatically converts number literals to instances of the number class.

Syntax: The syntax for creating a number object is as follows:

var val = new Number(number);

In the place of number, if you provide any non-number argument, then the argument cannot be converted into a number, it returns NaN (Not-a-Number).

* 1. **Question 9. What Are The Number Properties?**

**Answer :**

Here is a list of each property and their description.

* + - MAX\_VALUE: The largest possible value a number in JavaScript can have 1.7976931348623157E+308
    - MIN\_VALUE: The smallest possible value a number in JavaScript can have 5E-324
    - NaN: Equal to a value that is not a number.
    - NEGATIVE\_INFINITY: A value that is less than MIN\_VALUE.
    - POSITIVE\_INFINITY: A value that is greater than MAX\_VALUE
    - prototype: A static property of the Number object. Use the prototype property to assign new properties and methods to the Number object in the current document
    - constructor: Returns the function that created this object's instance. By default this is the Number object.
  1. **Question 10. What Are The Number Methods?**

**Answer :**

The Number object contains only the default methods that are a part of every object's definition.

* + - toExponential(): Forces a number to display in exponential notation, even if the number is in the range in which JavaScript normally uses standard notation.
    - toFixed(): Formats a number with a specific number of digits to the right of the decimal.
    - toLocaleString(): Returns a string value version of the current number in a format that may vary according to a browser's local settings.
    - toPrecision(): Defines how many total digits (including digits to the left and right of the decimal) to display of a number.
    - toString(): Returns the string representation of the number's value.
    - valueOf(): Returns the number's value.
  1. **Question 11. What Is Javascript Boolean Object?**

**Answer :**

The Boolean object represents two values, either "true" or "false". If value parameter is omitted or is 0, -0, null, false, NaN, undefined, or the empty string (""), the object has an initial value of false.

Syntax: Use the following syntax to create a boolean object.

var val = new Boolean(value);

* 1. **Question 12. What Are The Boolean Properties?**

**Answer :**

Here is a list of the properties of Boolean object:

* + - constructor: Returns a reference to the Boolean function that created the object.
    - prototype: The prototype property allows you to add properties and methods to an object.
  1. **Question 13. What Are Boolean Methods?**

**Answer :**

Here is a list of the methods of Boolean object and their description.

* + - toSource(): Returns a string containing the source of the Boolean object; you can use this string to create an equivalent object.
    - toString(): Returns a string of either "true" or "false" depending upon the value of the object.
    - valueOf(): Returns the primitive value of the Boolean object.
  1. **Question 14. What Is Javascript Strings Object?**

**Answer :**

The String object lets you work with a series of characters; it wraps Javascript's string primitive data type with a number of helper methods.

As JavaScript automatically converts between string primitives and String objects, you can call any of the helper methods of the String object on a string primitive.

Syntax: Use the following syntax to create a String object −

var val = new String(string);

The String parameter is a series of characters that has been properly encoded.

* 1. **Question 15. What Are String Properties?**

**Answer :**

Here is a list of the properties of String object and their description.

* + - constructor: Returns a reference to the String function that created the object.
    - length: Returns the length of the string.
    - prototype: The prototype property allows you to add properties and methods to an object.
  1. **Question 16. What Are String Methods?**

**Answer :**

Here is a list of the methods available in String object along with their description.

* + - charAt(): Returns the character at the specified index.
    - charCodeAt(): Returns a number indicating the Unicode value of the character at the given index.
    - concat(): Combines the text of two strings and returns a new string.
    - indexOf(): Returns the index within the calling String object of the first occurrence of the specified value, or -1 if not found.
    - lastIndexOf(): Returns the index within the calling String object of the last occurrence of the specified value, or -1 if not found.
    - localeCompare(): Returns a number indicating whether a reference string comes before or after or is the same as the given string in sort order.
    - match(): Used to match a regular expression against a string.
    - replace(): Used to find a match between a regular expression and a string, and to replace the matched substring with a new substring.
    - search(): Executes the search for a match between a regular expression and a specified string.
    - slice(): Extracts a section of a string and returns a new string.
    - split(): Splits a String object into an array of strings by separating the string into substrings.
    - substr(): Returns the characters in a string beginning at the specified location through the specified number of characters.
    - substring(): Returns the characters in a string between two indexes into the string.
    - toLocaleLowerCase(): The characters within a string are converted to lower case while respecting the current locale.
    - toLocaleUpperCase(): The characters within a string are converted to upper case while respecting the current locale.
    - toLowerCase(): Returns the calling string value converted to lower case.
    - toString(): Returns a string representing the specified object.
    - toUpperCase(): Returns the calling string value converted to uppercase.
    - valueOf(): Returns the primitive value of the specified object.
  1. **Question 17. What Are String Html Wrappers?**

**Answer :**

Here is a list of the methods that return a copy of the string wrapped inside an appropriate HTML tag.

* + - anchor(): Creates an HTML anchor that is used as a hypertext target.
    - big(): Creates a string to be displayed in a big font as if it were in a tag.
    - blink(): Creates a string to blink as if it were in a tag.
    - bold(): Creates a string to be displayed as bold as if it were in a **tag.**
    - fixed(): Causes a string to be displayed in fixed-pitch font as if it were in a tag
    - fontcolor(): Causes a string to be displayed in the specified color as if it were in a tag.
    - fontsize(): Causes a string to be displayed in the specified font size as if it were in a tag.
    - italics(): Causes a string to be italic, as if it were in an *tag.*
    - link(): Creates an HTML hypertext link that requests another URL.
    - small(): Causes a string to be displayed in a small font, as if it were in a tag.
    - strike(): Causes a string to be displayed as struck-out text, as if it were in a ~~tag.~~
    - sub(): Causes a string to be displayed as a subscript, as if it were in a tag
    - sup(): Causes a string to be displayed as a superscript, as if it were in a tag
  1. **Question 18. What Is Javascript Arrays Object?**

**Answer :**

The Array object lets you store multiple values in a single variable. It stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Syntax: Use the following syntax to create an Array object:

var fruits = new Array( "apple", "orange", "mango" );

The Array parameter is a list of strings or integers. When you specify a single numeric parameter with the Array constructor, you specify the initial length of the array. The maximum length allowed for an array is 4,294,967,295.

You can create array by simply assigning values as follows:

var fruits = [ "apple", "orange", "mango" ];

You will use ordinal numbers to access and to set values inside an array as follows.

fruits[0] is the first element  
fruits[1] is the second element  
fruits[2] is the third element

* 1. **Question 19. What Are Array Properties?**

**Answer :**

Here is a list of the properties of the Array object along with their description.

* + - constructor: Returns a reference to the array function that created the object.
    - index: The property represents the zero-based index of the match in the string
    - input: This property is only present in arrays created by regular expression matches.
    - length: Reflects the number of elements in an array.
    - prototype: The prototype property allows you to add properties and methods to an object.
  1. **Question 20. What Are Array Methods?**

**Answer :**

Here is a list of the methods of the Array object along with their description.

* + - concat(): Returns a new array comprised of this array joined with other array(s) and/or value(s).
    - every(): Returns true if every element in this array satisfies the provided testing function.
    - filter(): Creates a new array with all of the elements of this array for which the provided filtering function returns true.
    - forEach(): Calls a function for each element in the array.
    - indexOf(): Returns the first (least) index of an element within the array equal to the specified value, or -1 if none is found.
    - join(): Joins all elements of an array into a string.
    - lastIndexOf(): Returns the last (greatest) index of an element within the array equal to the specified value, or -1 if none is found.
    - map(): Creates a new array with the results of calling a provided function on every element in this array.
    - pop(): Removes the last element from an array and returns that element.
    - push(): Adds one or more elements to the end of an array and returns the new length of the array.
    - reduce(): Apply a function simultaneously against two values of the array (from left-to-right) as to reduce it to a single value.
    - reduceRight(): Apply a function simultaneously against two values of the array (from right-to-left) as to reduce it to a single value.
    - reverse(): Reverses the order of the elements of an array -- the first becomes the last, and the last becomes the first.
    - shift(): Removes the first element from an array and returns that element.
    - slice(): Extracts a section of an array and returns a new array.
    - some(): Returns true if at least one element in this array satisfies the provided testing function.
    - toSource(): Represents the source code of an object
    - sort(): Sorts the elements of an array
    - splice(): Adds and/or removes elements from an array.
    - toString(): Returns a string representing the array and its elements.
    - unshift(): Adds one or more elements to the front of an array and returns the new length of the array.
  1. **Question 21. What Is Javascript Date Object?**

**Answer :**

The Date object is a datatype built into the JavaScript language. Date objects are created with the new Date( ) as shown below.

Once a Date object is created, a number of methods allow you to operate on it. Most methods simply allow you to get and set the year, month, day, hour, minute, second, and millisecond fields of the object, using either local time or UTC (universal, or GMT) time.

The ECMAScript standard requires the Date object to be able to represent any date and time, to millisecond precision, within 100 million days before or after 1/1/1970. This is a range of plus or minus 273,785 years, so JavaScript can represent date and time till the year 275755.

Syntax: You can use any of the following syntaxes to create a Date object using Date() constructor.

new Date( )  
new Date(milliseconds)  
new Date(datestring)  
new Date(year,month,date[,hour,minute,second,millisecond ])

* 1. **Question 22. What Is Date Properties?**

**Answer :**

Here is a list of the properties of the Date object along with their description.

* + - constructor: Specifies the function that creates an object's prototype.
    - prototype: The prototype property allows you to add properties and methods to an object.
  1. **Question 23. What Are Date Methods?**

**Answer :**

Here is a list of the methods used with Date and their description.

* + - Date(): Returns today's date and time
    - getDate(): Returns the day of the month for the specified date according to local time.
    - getDay(): Returns the day of the week for the specified date according to local time.
    - getFullYear(): Returns the year of the specified date according to local time.
    - getHours(): Returns the hour in the specified date according to local time.
    - getMilliseconds(): Returns the milliseconds in the specified date according to local time.
    - getMinutes(): Returns the minutes in the specified date according to local time.
    - getMonth(): Returns the month in the specified date according to local time.
    - getSeconds():Returns the seconds in the specified date  according to local time.
    - getTime(): Returns the numeric value of the specified date as the number of milliseconds since January 1, 1970, 00:00:00 UTC.
    - getTimezoneOffset(): Returns the time-zone offset in minutes for the current locale.
    - getUTCDate(): Returns the day (date) of the month in the specified date according to universal time.
    - getUTCDay(): Returns the day of the week in the specified date according to universal time.
    - getUTCFullYear(): Returns the year in the specified date according to universal time.
    - getUTCHours(): Returns the hours in the specified date according to universal time.
    - getUTCMilliseconds(): Returns the milliseconds in the specified date according to universal time.
    - getUTCMinutes(): Returns the minutes in the specified date according to universal time.
    - getUTCMonth(): Returns the month in the specified date according to universal time.
    - getUTCSeconds(): Returns the seconds in the specified date according to universal time.
    - getYear(): Deprecated - Returns the year in the specified date according to local time. Use getFullYear instead.
    - setDate(): Sets the day of the month for a specified date according to local time.
    - setFullYear(): Sets the full year for a specified date according to local time.
    - setHours(): Sets the hours for a specified date according to local time.
    - setMilliseconds(): Sets the milliseconds for a specified date according to local time.
    - setMinutes(): Sets the minutes for a specified date according to local time.
    - setMonth(): Sets the month for a specified date according to local time.
    - setSeconds(): Sets the seconds for a specified date according to local time.
    - setTime(): Sets the Date object to the time represented by a number of milliseconds since January 1, 1970, 00:00:00 UTC.
    - setUTCDate(): Sets the day of the month for a specified date according to universal time.
    - setUTCFullYear(): Sets the full year for a specified date according to universal time.
    - setUTCHours(): Sets the hour for a specified date according to universal time.
    - setUTCMilliseconds(): Sets the milliseconds for a specified date according to universal time.
    - setUTCMinutes(): Sets the minutes for a specified date according to universal time.
    - setUTCMonth(): Sets the month for a specified date according to universal time.
    - setUTCSeconds(): Sets the seconds for a specified date according to universal time.
    - setYear(): Deprecated - Sets the year for a specified date according to local time. Use setFullYear instead.
    - toDateString(): Returns the "date" portion of the Date as a human-readable string.
    - toGMTString(): Deprecated - Converts a date to a string, using the Internet GMT conventions. Use toUTCString instead.
    - toLocaleDateString(): Returns the "date" portion of the Date as a string, using the current locale's conventions.
    - toLocaleFormat(): Converts a date to a string, using a format string.
    - toLocaleString(): Converts a date to a string, using the current locale's conventions.
    - toLocaleTimeString(): Returns the "time" portion of the Date as a string, using the current locale's conventions.
    - toSource(): Returns a string representing the source for an equivalent Date object; you can use this value to create a new object.
    - toString(): Returns a string representing the specified Date object.
    - toTimeString(): Returns the "time" portion of the Date as a human-readable string.
    - toUTCString(): Converts a date to a string, using the universal time convention.
    - valueOf(): Returns the primitive value of a Date object.
  1. **Question 24. What Are Date Static Methods?**

**Answer :**

In addition to the many instance methods listed previously, the Date object also defines two static methods. These methods are invoked through the Date() constructor itself.

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.

* 1. **Question 25. What Is Javascript Math Object?**

**Answer :**

The math object provides you properties and methods for mathematical constants and functions. Unlike other global objects, Math is not a constructor. All the properties and methods of Math are static and can be called by using Math as an object without creating it.

Thus, you refer to the constant pi as Math.PI and you call the sine function as Math.sin(x), where x is the method's argument.

Syntax: The syntax to call the properties and methods of Math are as follows

var pi\_val = Math.PI;  
var sine\_val = Math.sin(30);

* 1. **Question 26. What Are The Math Properties?**

**Answer :**

Here is a list of all the properties of Math and their description.

* + - E : Euler's constant and the base of natural logarithms, approximately 2.718.
    - LN2: Natural logarithm of 2, approximately 0.693.
    - LN10: Natural logarithm of 10, approximately 2.302.
    - LOG2E: Base 2 logarithm of E, approximately 1.442.
    - LOG10E: Base 10 logarithm of E, approximately 0.434.
    - PI: Ratio of the circumference of a circle to its diameter, approximately 3.14159.
    - SQRT1\_2: Square root of 1/2; equivalently, 1 over the square root of 2, approximately 0.707.
    - SQRT2: Square root of 2, approximately 1.414.
  1. **Question 27. What Are The Math Methods?**

**Answer :**

Here is a list of the methods associated with Math object and their description

* + - abs(): Returns the absolute value of a number.
    - acos(): Returns the arccosine (in radians) of a number.
    - asin(): Returns the arcsine (in radians) of a number.
    - atan(): Returns the arctangent (in radians) of a number.
    - atan2(): Returns the arctangent of the quotient of its arguments.
    - ceil(): Returns the smallest integer greater than or equal to a number.
    - cos(): Returns the cosine of a number.
    - exp(): Returns EN, where N is the argument, and E is Euler's constant, the base of the natural logarithm.
    - floor(): Returns the largest integer less than or equal to a number.
    - log(): Returns the natural logarithm (base E) of a number.
    - max(): Returns the largest of zero or more numbers.
    - min(): Returns the smallest of zero or more numbers.
    - pow(): Returns base to the exponent power, that is, base exponent.
    - random(): Returns a pseudo-random number between 0 and 1.
    - round(): Returns the value of a number rounded to the nearest integer.
    - sin(): Returns the sine of a number.
    - sqrt(): Returns the square root of a number.
    - tan(): Returns the tangent of a number.
    - toSource(): Returns the string "Math".
  1. **Question 28. What Are Regular Expressions And Regexp Object?**

**Answer :**

A 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 to perform powerful pattern-matching and search-and-replace functions on text.

Syntax: A regular expression could be defined with the RegExp () constructor, as follows −

var pattern = new RegExp(pattern, attributes);  
or simply  
var pattern = /pattern/attributes;

Here is the description of the parameters:

* + - pattern − A string that specifies the pattern of the regular expression or another regular expression.
    - attributes − An optional string containing any of the "g", "i", and "m" attributes that specify global, case-insensitive, and multiline matches, respectively.
  1. **Question 29. What Are Brackets?**

**Answer :**

Brackets ([]) have a special meaning when used in the context of regular expressions. They are used to find a range of characters.

* + - [...]: Any one character between the brackets.
    - [^...]: Any one character not between the brackets.
    - [0-9]: It matches any decimal digit from 0 through 9.
    - [a-z]: It matches any character from lowercase a through lowercase z.
    - [A-Z]: It matches any character from uppercase A through uppercase Z.
    - [a-Z]: It matches any character from lowercase a through uppercase Z.

The ranges shown above are general; you could also use the range [0-3] to match any decimal digit ranging from 0 through 3, or the range [b-v] to match any lowercase character ranging from b through v.

* 1. **Question 30. What Are Quantifiers?**

**Answer :**

The frequency or position of bracketed character sequences and single characters can be denoted by a special character. Each special character has a specific connotation. The +, \*, ?, and $ flags all follow a character sequence.

* + - p+: It matches any string containing one or more p's.
    - p\*: It matches any string containing zero or more p's.
    - p?: It matches any string containing at most one p.
    - p{N}: It matches any string containing a sequence of N p's
    - p{2,3}: It matches any string containing a sequence of two or three p's.
    - p{2, }: It matches any string containing a sequence of at least two p's.
    - p$: It matches any string with p at the end of it.
    - ^p: It matches any string with p at the beginning of it.