

ASSIGNMENT

Build-in functions and objects.

Java script has several "top-level" build-in functions.

JavaScript also has four built-in objects: Array, Date, Math and String, objects.

- Object methods User-defined objects

Methods are

All user-defined objects and build-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 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, 1997");
```

The object() Constructor

A constructor is a function that creates and initializes an object. JavaScript provides a special constructor function named object(). This function is used to create objects.

return value of the `Object()` constructor is assigned for a variable.

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

The 'with' keyword: 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 method for the object can be used without naming the object.

`with (Object) {`

 properties used without the object name and
 det-

`}`

javascript Native Objects

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

~~Here is the~~

(1) javascript Number Object

The `Number` object represents numerical data, either integer 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)

Number Properties :

Properties and Description

1 MAX_VALUE

The largest possible value a number in JavaScript can have is $1.7976931348623157 \times 10^308$

2 MIN_VALUE

The smallest possible value a number in JavaScript can have is $-1.7976931348623157 \times 10^{-308}$

3 NaN - equal to a value that is not number.

4 NEGATIVE_INFINITY - A value that is less than MIN_VALUE

5 POSITIVE_INFINITY - A value that is greater than MAX_VALUE

6 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.

7 Constructor -> Returns the function that created the object instance. By default this is the Number object.

Number methods

- Number object contains only the default methods that are a part of every objects definition.

Method and Description

- 1 toExponential() - Forces a number to display in exponential notation, even if the number is in the range in which JavaScript normally uses standard notation.
- 2 toFixed() → formats a number with a specific number of digits to the right of the decimal.
- 3 toLocaleString() → Returns a string value version of the current number in a format that may vary according to a browser's local settings.
- 4 toPrecision() → Define how many total digits (including digits to the left and right of the decimal) to display of a number.
- 5 toString() → Returns the string representation of the number's value
- 6 valueOf() → Returns the number value

② The Boolean Object

The Boolean object has value, 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.

as javascript automatically converts between string primitive and string objects you can call any of the help methods of the string object on a object primitive

Syntax var val = new String(string)

The string parameter is a series of characters that has been properly enclosed.

String Properties

Sl No	Properties & Description
1	<u>constructor</u> - Return a reference to the string function that created the object.
2	<u>length</u> - Return the length of the string.
3	<u>prototype</u> - The prototype property allows you to add properties and methods to an object.

String methods

Sl No	Method & Description
1	<u>charAt(index)</u> - Returns the character at the specified index
2	<u>charCodeAt(index)</u> - Returns a number indicating the Unicode value of the character at the given index.
3	<u>concat()</u> - Combines the text of two strings and returns a new string.
4	<u>indexOf()</u> - Returns the index within the calling string object of the first occurrence of the specified value, or -1 if not found.

- 5 lastIndexOf() → Returns the index within the calling string object of the last occurrence of the specified value or -1 if not found
- 6 localCompare() → Returns a number indicating whether a reference string comes before or after or is the same as the given string in sort order.
- 7 match()
used to match a regular expression against a string
- 8 replace(): Used to find a match between a regular expression and a string, and to replace the matched substring with a new substring.
- 9 search() → Creates the search for a match between
- 10 ~~slice()~~ → Extracts a section of a string and returns a new string.

String HTML wrapper

Sr No	Method & Description
1	<u>anchor()</u> → Creates an HTML anchor tag it is used as a hyperlink target.
2.	<u>big()</u> → Creates a string to be displayed in a big font as if it were in a <big> tag</big>
3	<u>blink()</u> → Creates a string to blink as if it were in a <blink> tag.</blink>

- 4 `bold()` → creates a string to be displayed as bold as if it were in a `` tag.
- 5 `fixed()` → causes a string to be displayed in fixed-pitch font as if it were in a `<tt>` tag.
6. `font-color()` → causes a string to be displayed in the specified color as if it were in a `` tag.
- 7 `fontsize()` → causes a string to be displayed in the specified font size if it were in a `` tag.
- 8 `italics()` → causes a string to be italic, as if it were in an `<i>` tag.
- 9 `link()` → creates an HTML hyperlink link that requests another URL.
- 10 `small()` → causes a string to be displayed in a small font, as if it were in a `<small>` tag.

④ The Array Objects

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 collections of data, but it is often more useful to think of an array as a collection of variables of the same type.

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

Array parameter is 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 1,294,967,295.

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

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

fruits[0] is the first element

fruits[1] is the 2nd element

fruits[2] is the 3rd element

~~fruits[3] is the 4th~~

Array Properties

No.

Property & Description

- 1 constructor → Refers to a reference to the array function that created the object.
- 2 index → The property represents the zero-based index of the match in the string.
- 3 input → This property is only present in arrays created by regular expression matches.
- 4 length → Reflects the number of elements in an array.
- 5 prototype → prototype property allows you to add properties and methods to an object.

Array Methods

- 1) concat() → Returns a new array composed of this array joined with other arrays and/or values.
- 2) every() → Returns true if every element in the array satisfies the provided testing function.
- 3) filter() → Creates a new array with all of the elements of this array for which the provided filtering function returns true.
- 4) forEach() → calls a function for each element in the array.
- 5) indexOf() → Returns the first (least) index of an element within the array equal to the specified value, or -1 if none is found.
- 6) join() → Joins all elements of an array into a string.
- 7) lastIndexOf() → Returns the last (greatest) index of an element within the array equal to the specified value, or -1 if none is found.
- 8) map() → Creates a new array with the results of calling a provided function on every element in this array.
- 9) pop() → Removes the last element from an array and returns that element.
- 10) push() → Adds one or more elements to the end of an array and returns the new length of the array.

⑤ The Date Object

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

Syntax

new Date()

new Date(milliseconds)

new Date(dateString)

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

Note - parameters in the brackets are always optional.

No Argument - with no argument, the Date() constructor creates a Date object set to the current date and time.

milliseconds - when one numeric argument is passed, it is taken as the internal numeric representations of the date in milliseconds, as returned by the getTime() method. For example, passing the argument 5000 creates a date that represents five seconds past midnight on 1/1/0.

DateString - when one string argument is passed, it a date string representation of a date, in the format accepted by the Date.parse() method.

3 arguments: To use the last form of the constructor shown above. Here is description of each argument.

year: integer value representing the year. For compatibility, you should always specify the year in full; use 1998, rather than 98.

month: integer value representing the month,

beginning with 0 for January to 11 for December
~~the day of the month~~
~~the hour of the day~~

date → integer value representing the hour of the
hour → integer value representing the hour of the
day (24-hour scale)

minute → integer value representing the minute segment
of a time reading.

second → integer value representing the second segment
of a time reading.

millisecond → integer value representing the millisecond
segment of a time reading

Date Properties

Sr. No	Property and Description
1	constructor → specifies the function that creates an object prototype.
2	prototype → The prototype property allows you to add properties and methods to an object.

Date Methods

Sr. No	Method and Description
1	getDate() → Returns todays date and time
2	getDay() → Returns the day of the month for the specified date according to local time.
3	getDay() → Returns the day of the week for the specified date according to local time.

- 4 `getFullYear()` — Returns year of the specified date according to local time.
- 5 `getHours()` — Returns the hours in the specified date according to local time.
- 6 `getMilliseconds()` — Returns the milliseconds in the specified date according to local time.
- 7 `getMinutes()` — Returns the minutes in the specified date according to local time.
- 8 `getMonth()` — Returns the month in the specified date according to local time.
- 9 `getSeconds()` — Returns the ~~nan~~ seconds in the specified date according to local time.
- 10 `getTime()` — Returns the numeric value of the specified date as the number of milliseconds since January 1, 1970, 00:00 UTC

Date static method

Method and Description

- 1 `Date.parse()` — Parses a string representation of a date and time and returns an internal millisecond representation of that date
- 2 `Date.UTC()` — Returns the millisecond representation of the specified UTC date and time.

⑥ The math object

The math object provides you to properties and methods for mathematical constant 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 sine function as `math.sin(x)`, where x is the method argument.

Syntax

`var pi-val = math.PI;`

`var sine-val = math.sin(30);`

Math Properties

Sr. No	Property and Description
1	E — Euler's constant and the base of natural logarithms, approximately 2.718.
2	LN2 — Natural logarithm of 2, approximately 0.693
3	LN10 — Natural logarithm of 10, approximately 2.302
4	LOG2E — Base 2 logarithm of E, approximately 1.442
5	LOG10E — Base 10 logarithm of E, approximately 0.434
6	PI — Ratio of the circumference of a circle to its diameter, approximately 3.14159
7	SQRT1_2 — Square root of 1/2; equivalently twice the square root of 2, approximately 0.707

8 $\text{SQRT2} \dots$

square root of 2, approximately 1.414

math methods

No.	Method and Description
1	<code>abs()</code> — Returns the absolute value for the number.
2	<code>acos()</code> — Returns the arccosine (in radians) of a NO:
3	<code>asin()</code> — Returns the arcsine (in radians) of a number
4	<code>atan()</code> — Returns the arctangent (in radians) of a NO:
5	<code>atan2()</code> — Returns the arctangent of the quotient of its arguments.
6	<code>ceil()</code> — Returns the smallest integer greater than or equal to a number.
7	<code>cos()</code> — Returns the cosine of a number.
8	<code>exp()</code> — Returns e^N , where N is the argument, and e is Euler's constant, the base of the natural logarithms.
9	<code>floor()</code> — Returns the largest integer less than or equal to a number
10	<code>log()</code> — Returns the natural logarithm (base e) of a NO:

(7) RegExp Object

A regular expression is an object that describes a pattern of characters.

The javascript `RegExp` class represents regular expressions, both `String` and `RegExp` define methods that use regular expressions to perform powerful pattern-matching and search-and-replace functions on text.

Syntax

var pattern = new RegExp(pattern, attributes);

or simply

var pattern = /pattern|attributes/;

Here the description of parameter

pattern — A string 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 multi-line matches, respectively.

RegExp Properties

Sr No	Properties and description
1	constructor — Specifies the function that creates an object's prototype.
2	global — Specifies if the "g" modifier is set
3	ignoreCase — Specifies if the "i" modifier is set
4	lastIndex — The index at which to start the next match
5	multiline — Specifies if the "m" modifier is set
6	source — The text of the pattern

RegExp methods

Sr No	method and Description
1	exec() — Executes a search for a match in its string parameter.

2 `test()` — Tests for a match in its string parameter

3 `toSource()` — Returns an object representing the specified object; you can use this value to create a new object.

4 `toString()` — Returns a string representing the specified object.
