Javascript Objects

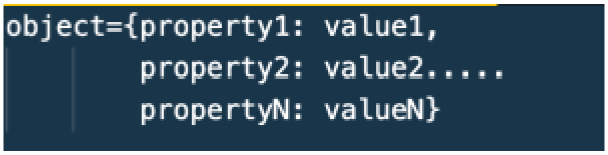
Object

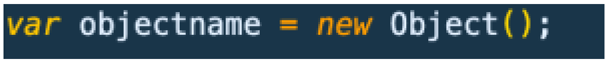
A javaScript object is a state-and-behavior-containing entity (properties and method). Examples include a car, pen, bicycle, chair, glass, keyboard, and monitor. JavaScript relies on templates rather than classes. To obtain the object in this case, no class is created. But, we direct create objects.

Creating Objects in Javascript

There are three ways to create objects -

S. No	Types	Description or Syntax

1	By object literal	 `object={property1: value1,`
		`      property2: value2.....`
		`      propertyN: valueN}`

2	By creating instance of Object directly (using new keyword)	 `var objectname = new Object();`
3	By using an object constructor (using new keyword)	Here, we have create function with arguments. Each argument value can be assigned in the current object by using this keyword.

Javascript Object Methods

Below are the various Javascripts Object Methods -

S. No	Methods	Description
1	Object.assign()	This method is used to copy enumerable and own properties from a source object to a target object
2	Object.create()	This method is used to create a new object with the specified prototype object and properties.
3	Object.defineProperty()	This method is used to describe some behavioral attributes of the property.
4	Object.defineProperties()	This method is used to create or configure multiple object properties.
5	Object.entries()	This method returns an array with arrays of the key, value pairs.
6	Object.freeze()	This method prevents existing properties from being removed.
7	Object. getOwnPropertyDescriptor()	This method returns a property descriptor for the specified property of the specified object.
8	Object. getOwnPropertyDescriptors()	This method returns all own property descriptors of a given object.
9	Object.getOwnPropertyNames()	This method returns an array of all properties (enumerable or not) found.
10	Object.getOwnPropertySymbols()	This method returns an array of all own symbol key properties.
11	Object.getPrototypeOf()	This method returns the prototype of the specified object.
12	Object.is()	This method determines whether two values are the same value.
13	Object.isExtensible()	This method determines if an object is extensible
14	Object.isFrozen()	This method determines if an object was frozen.
15	Object.isSealed()	This method determines if an object is sealed.
16	Object.keys()	This method returns an array of a given object's own property names.
17	Object.preventExtensions()	This method is used to prevent any extensions of an object.
18	Object.seal()	This method prevents new properties from being added and marks all existing properties as non-configurable.

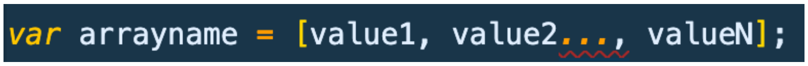
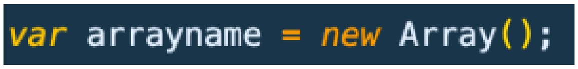
19	Object.setPrototypeOf()	This method sets the prototype of a specified object to another object.
20	Object.values()	This method returns an array of values.

Array

JavaScript array is an object that represents a collection of similar type of elements.

Creating Arrays in Javascript

There are three ways to create arrays -

Types	Description or Syntax
By array literal	 `var arrayname = [value1, value2..., valueN];`
By creating instance of Array directly (using new keyword)	 `var arrayname = new Array();`
By using an Array constructor (using new keyword)	Here, we must make an instance of the array by passing arguments to the function Object() { [native code] } so that we don't need to explicitly supply the value.

Javascript Array Methods

Below are the few array methods -

S.No	Methods	Description
1	concat()	It returns a new array object that contains two or more merged arrays.
2	copywithin()	It copies the part of the given array with its own elements and returns the modified array.
3	entries()	It creates an iterator object and a loop that iterates over each key/value pair.
4	every()	It determines whether all the elements of an array are satisfying the provided function conditions.
5	flat()	It creates a new array carrying sub-array elements concatenated recursively till the specified depth.
6	flatMap()	It maps all array elements via mapping function, then flattens the result into a new array.
7	fill()	It fills elements into an array with static values.
8	from()	It creates a new array carrying the exact copy of another array element.
9	filter()	It returns the new array containing the elements that pass the provided function conditions.
10	find()	It returns the value of the first element in the given array that satisfies the specified condition.
11	findIndex()	It returns the index value of the first element in the given array that satisfies the specified condition.
12	forEach()	It invokes the provided function once for each element of an array.
13	includes()	It checks whether the given array contains the specified element.
14	indexOf()	It searches the specified element in the given array and returns the index of the first match.
15	isArray()	It tests if the passed value ia an array.
16	join()	It joins the elements of an array as a string.
17	keys()	It creates an iterator object that contains only the keys of the array, then loops through these keys.
18	lastIndexOf()	It searches the specified element in the given array and returns the index of the last match.
19	map()	It calls the specified function for every array element and returns the new array
20	of()	It creates a new array from a variable number of arguments, holding any type of argument.
21	pop()	It removes and returns the last element of an array.
22	push()	It adds one or more elements to the end of an array.

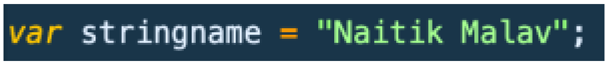
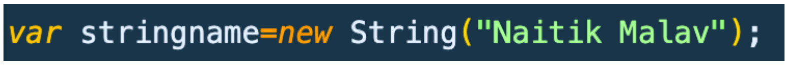
23	reverse()	It reverses the elements of given array.
24	reduce(function, initial)	It executes a provided function for each value from left to right and reduces the array to a single value.
25	reduceRight()	It executes a provided function for each value from right to left and reduces the array to a single value.
26	some()	It determines if any element of the array passes the test of the implemented function.
27	shift()	It removes and returns the first element of an array.
28	slice()	It returns a new array containing the copy of the part of the given array.
29	sort()	It returns the element of the given array in a sorted order.
30	splice()	It add/remove elements to/from the given array.
31	toLocaleString()	It returns a string containing all the elements of a specified array.
32	toString()	It converts the elements of a specified array into string form, without affecting the original array.
33	unshift()	It adds one or more elements in the beginning of the given array.
34	values()	It creates a new iterator object carrying values for each index in the array.

String

The JavaScript string is an object that represents a sequence of characters.

Creating Strings in Javascript

There are two ways to create Javascript strings -

S. No	Types	Description
1	By string literal	 `var stringname = "Naitik Malav";`
2	By string object (using new keyword)	 `var stringname=new String("Naitik Malav");`

Javascript String Methods

Below are the few string methods -

S.No	Methods	Description
1	charAt()	It provides the char value present at the specified index.
2	charCodeAt()	It provides the Unicode value of a character present at the specified index.
3	concat()	It provides a combination of two or more strings.
4	indexOf()	It provides the position of a char value present in the given string.
5	lastIndexOf()	It provides the position of a char value present in the given string by searching a character from the last position.
6	search()	It searches a specified regular expression in a given string and returns its position if a match occurs.
7	match()	It searches a specified regular expression in a given string and returns that regular expression if a match occurs.
8	replace()	It replaces a given string with the specified replacement.
9	substr()	It is used to fetch the part of the given string on the basis of the specified starting position and length.
10	substring()	It is used to fetch the part of the given string on the basis of the specified index.
11	slice()	It is used to fetch the part of the given string. It allows us to assign positive as well negative index.
12	toLowerCase()	It converts the given string into lowercase letter.

13	toLocaleLowerCase()	It converts the given string into lowercase letter on the basis of host?s current locale.
14	toUpperCase()	It converts the given string into uppercase letter.
15	toLocaleUpperCase()	It converts the given string into uppercase letter on the basis of host?s current locale.
16	toString()	It provides a string representing the particular object.
17	valueOf()	It provides the primitive value of string object.
18	split()	It splits a string into substring array, then returns that newly created array.
19	trim()	It trims the white space from the left and right side of the string.

Date

The year, month, and day can be obtained using the JavaScript date object. A timer can be shown on a webpage with the use of a JavaScript date object. To generate a date object, you can utilise various Date constructors. It offers instructions for obtaining and setting the day, month, year, hour, minute, and second. We can create date object in 4 different ways -

1. Date()
2. Date(milliseconds)
3. Date(dateString)
4. Date(year, month, day, hours, minutes, seconds, milliseconds)

Javascript Date Methods

Below are the few date methods -

S. No	Methods	Description
1	getDate()	It returns the integer value between 1 and 31 that represents the day for the specified date on the basis of local time.
2	getDay()	It returns the integer value between 0 and 6 that represents the day of the week on the basis of local time.
3	getFullYears()	It returns the integer value that represents the year on the basis of local time.
4	getHours()	It returns the integer value between 0 and 23 that represents the hours on the basis of local time.
5	getMilliseconds()	It returns the integer value between 0 and 999 that represents the milliseconds on the basis of local time.
6	getMinutes()	It returns the integer value between 0 and 59 that represents the minutes on the basis of local time.
7	getMonth()	It returns the integer value between 0 and 11 that represents the month on the basis of local time.
8	getSeconds()	It returns the integer value between 0 and 60 that represents the seconds on the basis of local time.
9	getUTCDate()	It returns the integer value between 1 and 31 that represents the day for the specified date on the basis of universal time.
10	getUTCDay()	It returns the integer value between 0 and 6 that represents the day of the week on the basis of universal time.
11	getUTCFullYears()	It returns the integer value that represents the year on the basis of universal time.
12	getUTCHours()	It returns the integer value between 0 and 23 that represents the hours on the basis of universal time.
13	getUTCMinutes()	It returns the integer value between 0 and 59 that represents the minutes on the basis of universal time.
14	getUTCMonth()	It returns the integer value between 0 and 11 that represents the month on the basis of universal time.
15	getUTCSeconds()	It returns the integer value between 0 and 60 that represents the seconds on the basis of universal time.
16	setDate()	It sets the day value for the specified date on the basis of local time.
17	setDay()	It sets the particular day of the week on the basis of local time.
18	setFullYears()	It sets the year value for the specified date on the basis of local time.
19	setHours()	It sets the hour value for the specified date on the basis of local time.
20	setMilliseconds()	It sets the millisecond value for the specified date on the basis of local time.
21	setMinutes()	It sets the minute value for the specified date on the basis of local time.
22	setMonth()	It sets the month value for the specified date on the basis of local time.

23	setSeconds()	It sets the second value for the specified date on the basis of local time.
24	setUTCDate()	It sets the day value for the specified date on the basis of universal time.
25	setUTCDay()	It sets the particular day of the week on the basis of universal time.
26	setUTCFullYears()	It sets the year value for the specified date on the basis of universal time.
27	setUTCHours()	It sets the hour value for the specified date on the basis of universal time.
28	setUTCMilliseconds()	It sets the millisecond value for the specified date on the basis of universal time.
29	setUTCMinutes()	It sets the minute value for the specified date on the basis of universal time.
30	setUTCMonth()	It sets the month value for the specified date on the basis of universal time.
31	setUTCSeconds()	It sets the second value for the specified date on the basis of universal time.
32	toDateString()	It returns the date portion of a Date object.
33	toISOString()	It returns the date in the form ISO format string.
34	toJSON()	It returns a string representing the Date object. It also serializes the Date object during JSON serialization.
35	toString()	It returns the date in the form of string.
36	toTimeString()	It returns the time portion of a Date object.
37	toUTCString()	It converts the specified date in the form of string using UTC time zone.
38	valueOf()	It returns the primitive value of a Date object.

Math

The math object in JavaScript offers a number of constants and methods for carrying out mathematical operations. Unlike date object, it doesn't have constructors.

Javascript Math Methods

Below are few math methods in javascript -

S.No	Methods	Description
1	abs()	It returns the absolute value of the given number.
2	acos()	It returns the arccosine of the given number in radians.
3	asin()	It returns the arcsine of the given number in radians.
4	atan()	It returns the arc-tangent of the given number in radians.
5	cbrt()	It returns the cube root of the given number.
6	ceil()	It returns a smallest integer value, greater than or equal to the given number.
7	cos()	It returns the cosine of the given number.
8	cosh()	It returns the hyperbolic cosine of the given number.
9	exp()	It returns the exponential form of the given number.
10	floor()	It returns largest integer value, lower than or equal to the given number.
11	hypot()	It returns square root of sum of the squares of given numbers.
12	log()	It returns natural logarithm of a number.
13	max()	It returns maximum value of the given numbers.
14	min()	It returns minimum value of the given numbers.
15	pow()	It returns value of base to the power of exponent.
16	random()	It returns random number between 0 (inclusive) and 1 (exclusive).

17	round()	It returns closest integer value of the given number.
18	sign()	It returns the sign of the given number
19	sin()	It returns the sine of the given number.
20	sinh()	It returns the hyperbolic sine of the given number.
21	sqrt()	It returns the square root of the given number
22	tan()	It returns the tangent of the given number.
23	tanh()	It returns the hyperbolic tangent of the given number.
24	trunc()	It returns an integer part of the given number.

Number

You can display a numeric value using the JavaScript number object. It could be floating-point or integer. The IEEE standard is followed by the JavaScript number object when representing floating-point numbers.

Javascript Number Constants

S. No	Constant	Description
1	MIN_VALUE	returns the largest minimum value.
2	MAX_VALUE	returns the largest maximum value.
3	POSITIVE_INFINITY	returns positive infinity, overflow value.
4	NEGATIVE_INFINITY	returns negative infinity, overflow value.
5	NaN	represents "Not a Number" value.

Javascript Number Methods

Let's see the list of JavaScript number methods with their description.

S.No	Methods	Description
1	isFinite()	It determines whether the given value is a finite number.
2	isInteger()	It determines whether the given value is an integer.
3	parseFloat()	It converts the given string into a floating point number.
4	parseInt()	It converts the given string into an integer number.
5	toExponential()	It returns the string that represents exponential notation of the given number.
6	toFixed()	It returns the string that represents a number with exact digits after a decimal point.
7	toPrecision()	It returns the string representing a number of specified precision.
8	toString()	It returns the given number in the form of string.

Boolean

JavaScript Boolean is an object that represents value in two states: true or false. You can create the JavaScript Boolean object by Boolean() constructor as given below.

Syntax: Boolean b = new Boolean(value);

Javascript Boolean Methods

S. No	Method	Description
1	toSource()	returns the source of Boolean object as a string.
2	toString()	converts Boolean into String.
3	valueOf()	converts other type into Boolean.