

# setInterval()

---

- setInterval() is a method of the window object (browser) that executes a function or a string of JavaScript code at specified intervals (in milliseconds).
- It will continue to execute the function or the JavaScript code until clearInterval() is called on the interval ID.

## Example

```
function myAlert() {  
    alert("Hello World!");  
}  
setInterval(myAlert, 1000);
```

- This will call the function myAlert() every second.
- Here 1000 milliseconds is 1 second.
- Note : Here the myAlert is passed without the parenthesis in the setInterval() method.

## Quick Quiz

- Fill in the blanks to call the calc() every 3 seconds:

```
function calc() {  
    console.log(Math.random() * 10);  
}  
  
setInterval(___, ___);
```

## The Date object

- The Date object is a part of the JavaScript language.
- It is used to represent a date and time.
- It is used to get the current date and time.
- It consists of methods to get the current date, time, year and so on.
- Using the new Date(), we can create a new Date object with the current date and time.

## Creating a new Date object

```
var today = new Date();
```

- The other way to create a new Date object is to pass the date and time as arguments to the new Date() method.

```
var today = new Date(year, month, day, hours, minutes, seconds, milliseconds);
```

- Js Dates are calculated from 01 January, 1970. 00:00:00 (UTC)
- One day contains 86,400,000 milliseconds.
- One hour contains 3,600,000 milliseconds.
- One minute contains 60,000 milliseconds.
- One second contains 1,000 milliseconds.

## Example

```
// Fri Jan 02 1970 00:00:00
var d1 = new Date(86400000);

// Sat Jan 03 1970 00:00:00
var d2 = new Date(86400000 * 2);
```

- JavaScript counts months from 0 to 11 (January to December).
- The computer time is ticking but the date object does not.

## Quick Quiz

- Q. What information results from creating a Date object with the new Date() method?

- A. The current date and time.
- B. The current date.
- C. The current time.
- D. The current year.

## Date Methods

- When a Date object is created, a number of methods make it possible to get information about the date and time.

Method	Description
getDate()	Returns the day of the month (from 1 to 31)
getDay()	Returns the day of the week (from 0 to 6)
getFullYear()	Returns the year (four digits)
getHours()	Returns the hour (from 0 to 23)
getMilliseconds()	Returns the milliseconds (from 0 to 999)
getMinutes()	Returns the minutes (from 0 to 59)
getMonth()	Returns the month (from 0 to 11)

Method	Description
<code>getSeconds()</code>	Returns the seconds (from 0 to 59)
<code>getTime()</code>	Returns the number of milliseconds since January 1, 1970
<code>getTimezoneOffset()</code>	Returns the time-zone offset from UTC
<code>getUTCDate()</code>	Returns the day of the month in universal time
<code>getUTCDay()</code>	Returns the day of the week in universal time
<code>getUTCFullYear()</code>	Returns the year in universal time
<code>getUTCHours()</code>	Returns the hour in universal time
<code>getUTCMilliseconds()</code>	Returns the milliseconds in universal time
<code>getUTCMinutes()</code>	Returns the minutes in universal time
<code>getUTCMonth()</code>	Returns the month in universal time
<code>getUTCSeconds()</code>	Returns the seconds in universal time
<code>setDate()</code>	Sets the day of the month (from 1 to 31)
<code>setFullYear()</code>	Sets the year (four digits)
<code>setHours()</code>	Sets the hour (from 0 to 23)
<code>setMilliseconds()</code>	Sets the milliseconds (from 0 to 999)
<code>setMinutes()</code>	Sets the minutes (from 0 to 59)
<code>setMonth()</code>	Sets the month (from 0 to 11)
<code>setSeconds()</code>	Sets the seconds (from 0 to 59)
<code>setTime()</code>	Sets the number of milliseconds since January 1, 1970
<code>setUTCDate()</code>	Sets the day of the month in universal time
<code>setUTCFullYear()</code>	Sets the year in universal time
<code>setUTCHours()</code>	Sets the hour in universal time
<code>setUTCMilliseconds()</code>	Sets the milliseconds in universal time
<code>setUTCMinutes()</code>	Sets the minutes in universal time
<code>setUTCMonth()</code>	Sets the month in universal time
<code>setUTCSeconds()</code>	Sets the seconds in universal time
<code>toDatestring()</code>	Returns the date as a string
<code>toISOString()</code>	Returns the date as a string in ISO format
<code>toJSON()</code>	Returns the date as a string in JSON format
<code>toLocaleDateString()</code>	Returns the date as a string in the current locale

Method	Description
toLocaleString()	Returns the date as a string in the current locale
toLocaleTimeString()	Returns the time as a string in the current locale
toString()	Returns the date as a string
toTimeString()	Returns the time as a string
toUTCString()	Returns the date as a string in UTC

... and many more.

## For Example

```
var today = new Date();
today.getDate();
today.getDay();
today.getFullYear();
today.getHours();
```

- Lets say we want to develop a function printTime() that prints the current time to browser every second.

```
function printTime() {
  var d = new Date();
  var hours = d.getHours();
  var minutes = d.getMinutes();
  var seconds = d.getSeconds();
  document.body.innerHTML = hours + ":" + minutes + ":" + seconds;
}
setInterval(printTime, 1000);
```

- We declared a function printTime() that prints the current time to browser every second by using setInterval() method.
- The innerHTML property of the document.body object is used to write the content of the page.