Lesson:

Date & Eval in JavaScript

Topics

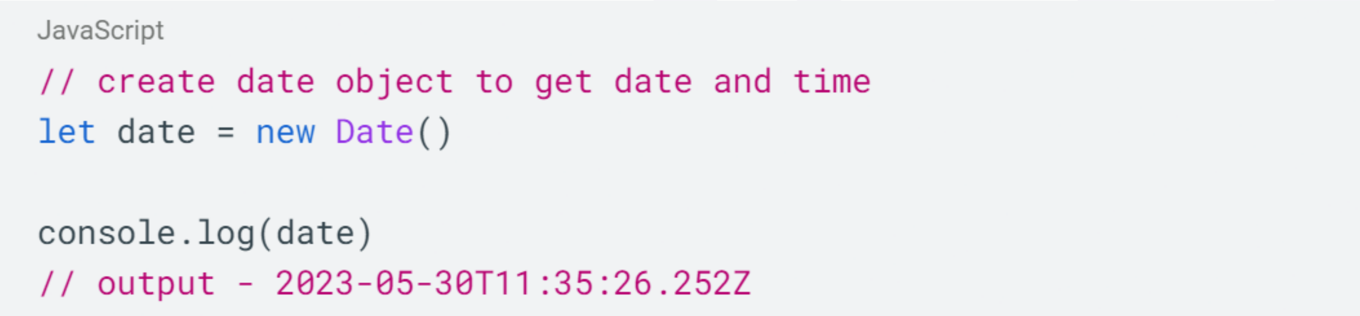
- Introduction to date object
- Getting the date and time
- Date methods
- Eval in JavaScript

Introduction to date object -

In JavaScript, date is an object that works with dates and time. It Provides methods for creating, manipulating, and formatting dates. It is fundamentally specified as the time in milliseconds that has elapsed since the epoch, which is defined as midnight at the beginning of January 1, 1970, UTC.

Getting the date and time -

In JavaScript, the new Date() constructor can be used to get the date and time. This will create a new date object with the current date and time.
Example -



JavaScript

```
// create date object to get date and time
let date = new Date()

console.log(date)
// output - 2023-05-30T11:35:26.252Z
```

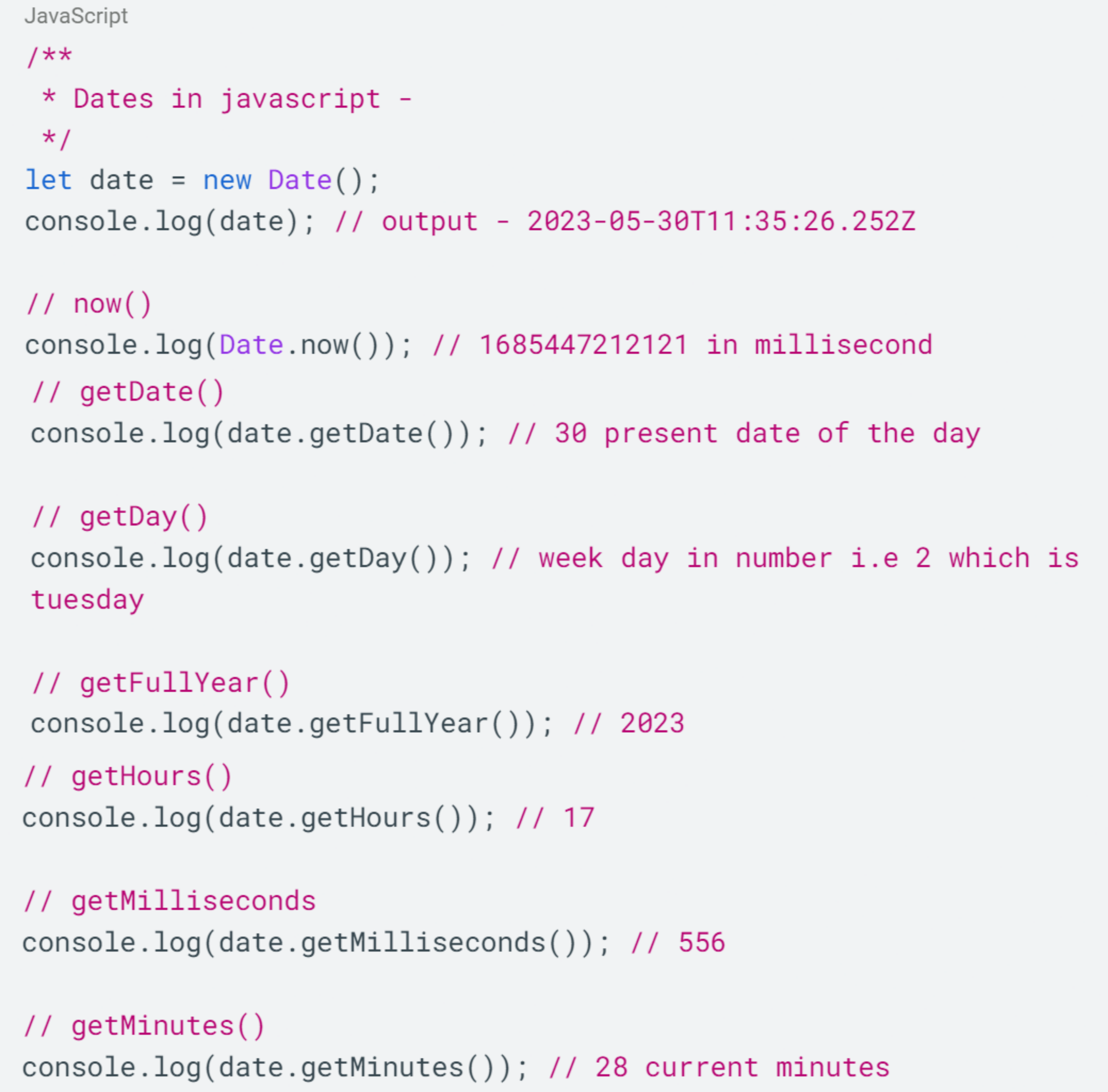
Date methods -

some of the important date methods in javascript are as follows -

now() - method returns the current date and time in milliseconds
getDate() - method of Date instances returns the day of the month for this date according to local time.
getDay() - method of Date instances returns the day of the week for this date according to local time, where 0 represents Sunday.
getFullYear() - method of Date instances returns the year for this date according to local time.
getHours() - method of Date instances returns the hours for this date according to local time.
getSeconds() - method of Date instances returns the seconds for this date according to local time.
getMilliseconds() - method of Date instances returns the milliseconds for this date according to local time.
getMinutes() - method of Date instances returns the minutes for this date according to local time.
getMonth() - method of Date instances returns the month for this date according to local time, as a zero-based value

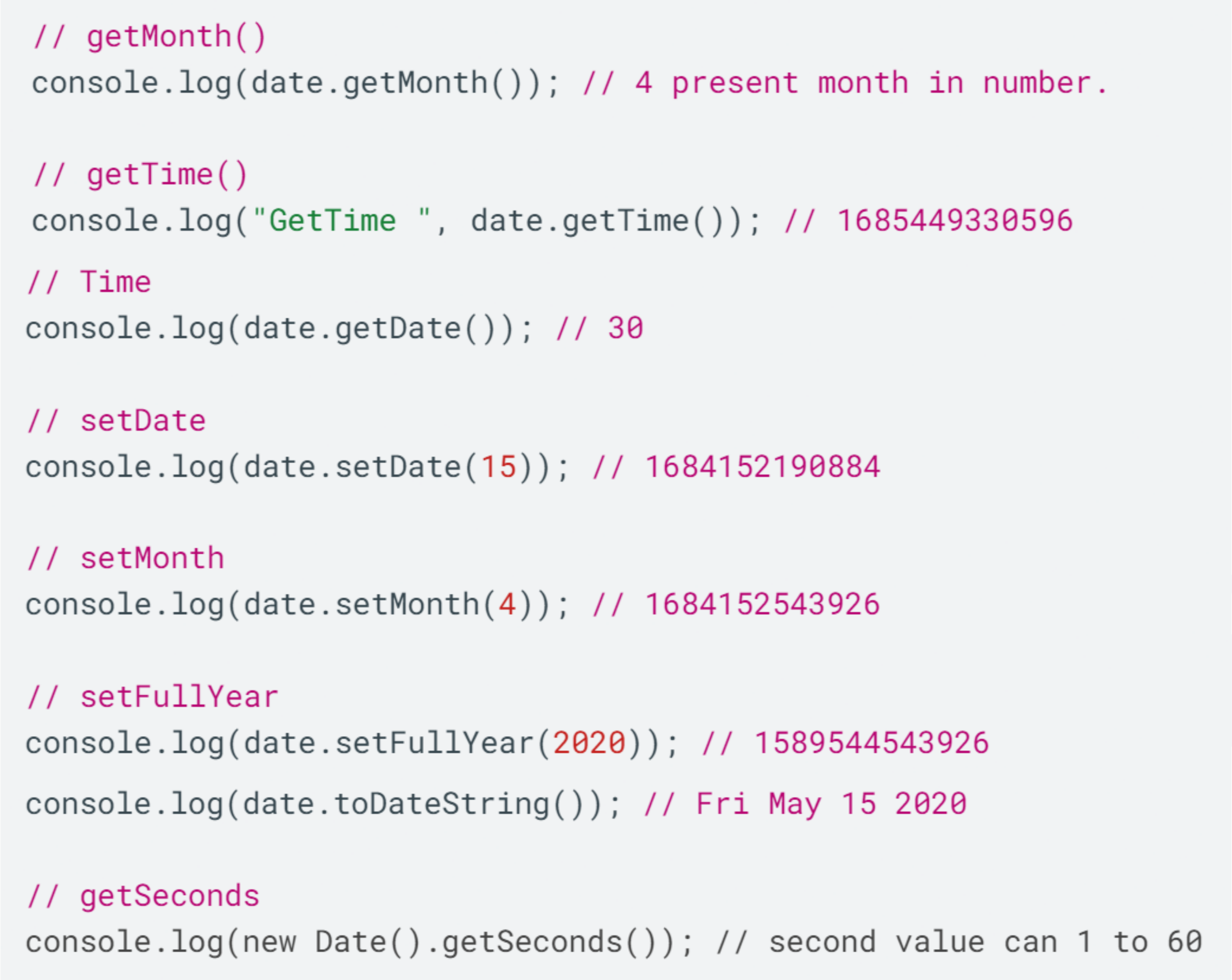
getTime() - method of Date instances returns the number of milliseconds for this date since January 1, 1970, UTC.
setDate() - method of Date instances changes the day of the month for this date according to local time.
setMonth() - method of Date instances changes the month for this date according to local time.
setFullyear() - method of Date instances changes the year, for this date according to local time.
toDateString() - method returns the date portion of a Date object interpreted in the local timezone in English.

Example -



JavaScript

```
/**
 * Dates in javascript -
 */
let date = new Date();
console.log(date); // output - 2023-05-30T11:35:26.252Z

// now()
console.log(Date.now()); // 1685447212121 in millisecond
// getDate()
console.log(date.getDate()); // 30 present date of the day

// getDay()
console.log(date.getDay()); // week day in number i.e 2 which is
tuesday

// getFullYear()
console.log(date.getFullYear()); // 2023
// getHours()
console.log(date.getHours()); // 17

// getMilliseconds
console.log(date.getMilliseconds()); // 556

// getMinutes()
console.log(date.getMinutes()); // 28 current minutes
```



```
// getMonth()
console.log(date.getMonth()); // 4 present month in number.

// getTime()
console.log("GetTime ", date.getTime()); // 1685449330596
// Time
console.log(date.getDate()); // 30

// setDate
console.log(date.setDate(15)); // 1684152190884

// setMonth
console.log(date.setMonth(4)); // 1684152543926

// setFullYear
console.log(date.setFullYear(2020)); // 1589544543926
console.log(date.toDateString()); // Fri May 15 2020

// getSeconds
console.log(new Date().getSeconds()); // second value can 1 to 60
```

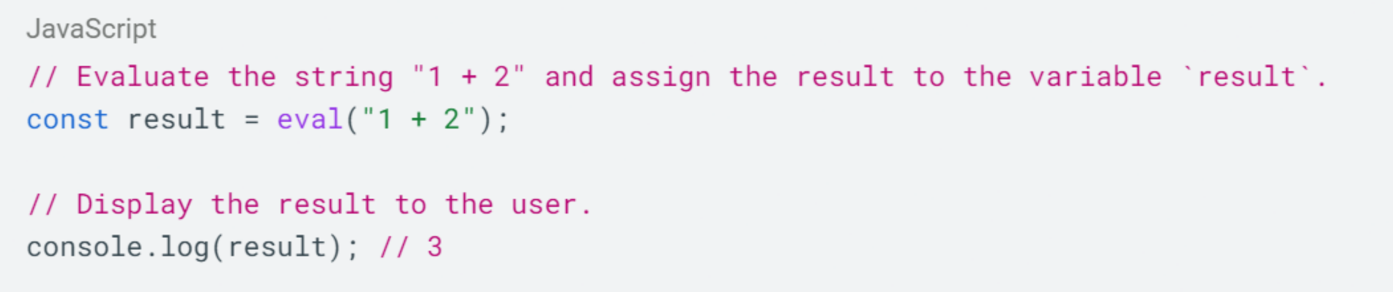
Eval in JavaScript -

The eval() function in JavaScript is a global function that evaluates a string as JavaScript code and executes it. This means that you can use eval() to dynamically execute code at run time.

Syntax -

```
Unset
eval(script)
// Note - script is a string representing a JavaScript expression, statement or
// sequence of statements.
```

Example



JavaScript

```javascript
// Evaluate the string "1 + 2" and assign the result to the variable `result`.
const result = eval("1 + 2");

// Display the result to the user.
console.log(result); // 3
```

It is generally recommended to avoid using **eval()** as it is a dangerous function. This is because it can be used to execute malicious code, such as code that steals your personal information or damages your computer.

We will be discussing more about eval() in the next chapter.