

Date

Вбудований об'єкт Date дозволяє здійснювати маніпуляції з датами і часом

Загалом дата зберігається як ***timestamp*** - кількість мілісекунд, які пройшли починаючи з 1 січня 1970 року (початок епохи Unix)

<https://262.ecma-international.org/5.1/#sec-15.9.1.15>

<https://itwiki.dev/front-end/js-reference/javascript/jsref-obj-date-aspc>

https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Date

<https://uk.javascript.info/date>

Формати дати і часу

Формат	Особливості
GMT - " <u>Greenwich mean time</u> " – "Середній час за Грінвічським меридіану"	<ul style="list-style-type: none">• засновано на русі Землі відносно Сонця• початок доби опівдні
UTC - " <u>Coordinated universal time</u> " - всесвітній координований час (найбільш використовуваний)	<ul style="list-style-type: none">• визначається атомним годинником• початок доби опівночі

Створення об'єкта Date з використанням конструктора

Параметри конструктора	Приклад
Без параметрів – дозволяє отримати поточну дату <code>new Date()</code>	<pre>let currentDate = new Date() //Thu Apr 27 2023 09:50:33 GMT+0300 (Eastern European Summer Time)</pre>
кількість мілісекунд (<i>одне значення</i> timestamp – мітка часу), які пройшли з 1 січня 1970р. – отримаємо відповідну даній кількості мілісекунд дату <code>new Date(timestamp)</code>	<pre>let date = new Date(1254210000000) console.log(date) //Tue Sep 29 2009 10:40:00 GMT+0300 (Eastern European Summer Time) //--- роки до 1.01.1070 мають від'ємне значення timestamp // 31 грудня 1969 року let Dec31_1969 = new Date(-24 * 3600 * 1000);</pre>
Окремі значення (<i>не менше двох</i>) (не обов'язково вказувати усі наступні значення після номера місяця, тоді їх значення дорівнює 0): <code>year</code> (4 цифри) (<code>0-99</code> == <code>1900-1999</code>), <code>month</code> (0-11), <code>day</code> (1, 2, ...), <code>hour</code> (0-11), <code>minute</code> (0-59), <code>second</code> (0-59), <code>millisecond</code> (0-999)	<pre>//new Date(year, month[, day[, hour[, minute[, second[, millisecond]]]]) var birthday = new Date(1995, 11, 17) var birthday = new Date(1995, 11, 17, 3, 24, 0) // якщо вказувати значення, що виходять за межі допустимого діапазону, то ці значення будуть автоіматично відкореговані let date = new Date(2013, 0, 32); // 32 січня 2013 ?!? alert(date); // ...це 1 лютого 2013!</pre>

Створення об'єкта Date з використанням конструктора і рядкового представлення

Параметри конструктора	Приклад
Рядкове представлення дати – отримаємо відповідний даному представлені об'єкт <code>new Date(dateString)</code>	<pre>// IETF RFC: https://datatracker.ietf.org/doc/html/rfc2822#page-14 date = new Date("15 April 2023") //Sat Apr 15 2023 00:00:00 GMT+0300 (Eastern European Summer Time) const date1 = new Date('December 17, 1995 03:24:00') // ISO8601: https://262.ecma-international.org/5.1/#sec-15.9.1.15 Format: YYYY-MM-DDTHH:mm:ss.sss Z //Z- зміщення таймзони +-hh:mm (Одинична буква Z буде означати UTC+0) Format: YYYY-MM-DDTHH:mm:ss ... Format: YYYY-MM-DDTHH ... const date2 = new Date('1995-12-17T03:24:00');</pre>

Створення об'єкта Date з використанням конструктора і рядкового представлення

При створенні дат можуть бути використані різні стандарти рядкового подання (наприклад: **IETF RFC**, **ISO8601**)

Параметри конструктора	Приклад
Рядкове представлення дати – отримаємо відповідний даному представлені об'єкт new Date(dateString)	<pre>// IETF RFC: https://datatracker.ietf.org/doc/html/rfc2822#page-14 date = new Date("15 April 2023") //Sat Apr 15 2023 00:00:00 GMT+0300 (Eastern European Summer Time) const date1 = new Date('December 17, 1995 03:24:00') //Sun Dec 17 1995 03:24:00 GMT+0200 (Eastern European Standard Time) //===== // ISO8601: https://262.ecma-international.org/5.1/#sec-15.9.1.15 Format: YYYY-MM-DDTHH:mm:ss.sss Z //Z- зміщення таймзони +-hh:mm (Одинична буква Z буде означати UTC+0) // Можуть використовуватись і скорочені формати // (не вказані значення будуть дорівнювати 0) Format: YYYY-MM-DDTHH:mm:ss Format: YYYY-MM-DDTHH const date2 = new Date('1995-12-17T03:24:00');</pre>

Отримання Date без прямого використання конструктора

Метод	Приклад
Виклик як функції – дозволяє отримати поточну дату <code>Date()</code>	<code>Date()</code> <code>// 'Thu Apr 27 2023 10:08:10 GMT+0300 (Eastern European Summer Time)'</code>
<code>Date.now()</code> – поточна кількість мілісекунд	<code>Date.now()</code> <code>// 1682579683785</code>
<code>Date.parse(dateString)</code> – дозволяє отримати значення кількості мілісекунд з рядкового представлення (неяквний виклик <code>new Date(dateString)</code>)	<code>Date.parse('Wed, 09 Aug 1995 00:00:00')</code> // GMT – поточні налаштування <code>// 807915600000</code> <code>Date.parse('Thu, 01 Jan 1970 00:00:00 GMT-0')</code> <code>// 0</code> <code>Date.parse('Thu, 01 Jan 1970 00:00:00 GMT-0400')</code> <code>// 14400000</code>
<code>Date.UTC()</code> – дозволяє отримати дату шляхом задання значень <code>year</code> (4 цифри), <code>month</code> (0-11), <code>day</code> (1-31), <code>hour</code> (0-11), <code>minute</code> (0-59), <code>second</code> (0-59), <code>millisecond</code> (0-999)	<code>Date.UTC(year[, month[, day[, hour[, minute[, second[, millisecond]]]]])</code> <code>const utcDate1 = new Date(Date.UTC(96, 1, 2, 3, 4, 5));</code> <code>console.log(utcDate1.toUTCString());</code> <code>// Expected output: "Fri, 02 Feb 1996 03:04:05 GMT"</code> <code>const utcDate2 = new Date(Date.UTC(0, 0, 0, 0, 0, 0));</code> <code>console.log(utcDate2.toUTCString());</code> <code>// Expected output: "Sun, 31 Dec 1899 00:00:00 GMT"</code>

Отримання значень з об'єкта Date

Метод/властивість та аналоги (за часовим поясом UTC+0)	Значення	Приклад
getFullYear() getUTCFullYear()	рік (4 цифри)	let date = new Date() date.getFullYear() //2023
getMonth() getUTCMonth()	місяць (0 - 11)	let date = new Date() date.getMonth() //3
getDate() getUTCDay()	день місяця (1 – 31)	let date = new Date() date.getDate() //27
getHours() getUTCHours()	години (0-23)	let date = new Date() date.getHours() //10
getMinutes() getUTCMinutes()	хвилини (0-59)	let date = new Date() date.getMinutes() //33
getSeconds() getUTCSeconds()	секунди (0-59)	let date = new Date() date.getSeconds() //54
getMilliseconds() getUTCMilliseconds()	мілісекунди (0-999)	let date = new Date() date.getMilliseconds() //973
getDay() getUTCDay()	день тижня, від 0 (неділя) до 6 (субота)	let date = new Date() date.getDay() //4
getTime()	Повертає timestamp для дати – кількість мілісекунд, що пройшли з 1 січня 1970 UTC+0	date.getTime() //1682587512801
getTimezoneOffset()	зміщення часової зони (різницю між UTC та місцевим часовим поясом, у хвилинах)	let date = new Date() date.getTimezoneOffset() //-180

Встановлення значень Date (якщо значення виходять за межі, то вони будуть відкореговані)

Метод-встановлення значення	Значення	Приклад
setFullYear(year, [month], [date])	рік (4 цифри)	<pre>let date = new Date() date.getFullYear() //2023 date.setFullYear(2027) date.getFullYear() //2027</pre>
setMonth(month, [date])	місяць (0 - 11)	<pre>let date = new Date() date.getMonth() //3 date.setMonth(14) //(буде змінено і рік!) date.getMonth() //2 date Mon Mar 27 2024 13:16:47 GMT+0300 (Eastern European Summer Time)</pre>
setDate(date)	день місяця (1 – 31)	<pre>let date = new Date() date.getDate() //27 date.setDate(50) //(буде змінено і місяць!) date.getDate() // 20 date //Sat May 20 2023 13:20:18 GMT+0300 (Eastern European Summer Time)</pre>

Встановлення значень Date (якщо значення виходять за межі, то вони будуть відкореговані)

Метод-встановлення значення	Значення	Приклад
setHours(hour, [min], [sec], [ms])	години (0-23)	<pre>let date = new Date() date.getHours() //12 date.setHours(10) date.getHours() //10</pre>
setMinutes(min, [sec], [ms])	хвилини (0-59)	<pre>let date = new Date() date.getMinutes() //23 date.setMinutes(15) date.getMinutes() //15</pre>
setSeconds(sec, [ms])	секунди (0-59)	<pre>let date = new Date() date.getSeconds() //26 date.setSeconds(35) date.getSeconds() //35</pre>
setMilliseconds(ms)	мілісекунди (0-999)	<pre>let date = new Date() date.getMilliseconds() //701 date.setMilliseconds(801) date.getMilliseconds() //801</pre>
setTime(milliseconds)	встановлює всю дату в мілісекундах з 01.01.1970 UTC	<pre>let date = new Date() date.getTime() 1682587511801 date.setTime(1682587512801) date.getTime() 1682587512801</pre>

Зміщення дат (додавання/віднімання shift годин, хвилин, секунд, ...)

Метод-встановлення, який потрібно використати для зміни значення	Значення	Приклад (до поточного значення додаємо/віднімаємо зміщення)
setFullYear(year + shift)	Рік	let date = new Date() date.getFullYear() //2023 date.setFullYear(date.getFullYear()+2) date.getFullYear() //2025
setMonth(month + shift)	Місяць	let date = new Date() date.getMonth() //3 date.setMonth(date.getMonth()+10) //(буде змінено і рік!) date.getMonth() //1 date //Tue Feb 27 <u>2024</u> 10:33:54 GMT+0200 (Eastern European Standard Time)
setDate(date+ shift)	день місяця	let date = new Date() date.getDate() //27 date.setDate(date.getDate()+7) //(буде змінено і місяць!) date.getDate() //4 date Thu <u>May 04</u> 2023 12:19:46 GMT+0300 (Eastern European Summer Time)

Зміщення дат (додавання/віднімання shift годин, хвилин, секунд, ...)

Метод-встановлення, який потрібно використати для зміни значення	Значення	Приклад (до поточного значення додаємо/віднімаємо зміщення)
setHours(hour+ shift)	години	<pre>let date = new Date() date.getHours() //12 date.setHours(date.getHours()+5) date.getHours() //17</pre>
setMinutes(min+ shift)	хвилини	<pre>let date = new Date() date.getMinutes() //23 date.setMinutes(date.getMinutes()+10) date.getMinutes() //33</pre>
setSeconds(sec+ shift)	секунди	<pre>let date = new Date() date.getSeconds() //26 date.setSeconds(date.getSeconds()+10) date.getSeconds() //36</pre>
setMilliseconds(ms+ shift)	мілісекунди	<pre>let date = new Date() date.getMilliseconds() //701 date.setMilliseconds(date.getMilliseconds()+100) date.getMilliseconds() //801</pre>
setTime(milliseconds+ shift)	встановлює всю дату в мілісекундах з 01.01.1970 UTC	<pre>let date = new Date() date.getTime() 1682587511801 date.setTime(date.getTime()+1000) date.getTime() 1682587512801</pre>

Перетворення у рядок

Метод/властивість та аналоги ()	Значення	Приклад
toString()	Рядкове представлення	<pre>let date = new Date() date.toString() 'Thu Apr 27 2023 10:33:54 GMT+0300 (Eastern European Summer Time)'</pre>
toISOString()	Рядкове представлення (Часовий пояс UTC (з суфіксом "Z"))	<pre>date.toISOString() '2023-04-27T07:33:54.973Z'</pre>
toGMTString()	Рядкове представлення (зручне для людини)	<pre>date.toGMTString() 'Thu, 27 Apr 2023 07:33:54 GMT'</pre>
toUTCString()	Рядкове представлення (зручне для людини)	<pre>date.toUTCString() 'Thu, 27 Apr 2023 07:33:54 GMT'</pre>
toJSON()	Представлення у JSON	<pre>date.toJSON() '2023-04-27T07:33:54.973Z'</pre>
toDateString()	рядкове представлення тільки дати	<pre>let date = new Date() date.toDateString() // 'Thu Apr 27 2023'</pre>
toTimeString()	Отримати рядкове представлення тільки часу	<pre>let date = new Date() date.toTimeString() // '10:33:54 GMT+0300 (Eastern European Summer Time)'</pre>

Перетворення у рядок (врахування регіональних стандартів)

Метод/властивість та аналоги ()	Значення	Приклад
<code>toLocaleString([locales[, options]])</code>	Дата у національному стандарті <i>locales</i> – мова(країна), (en-US, en-GB, uk-UA,...) <i>options</i> – параметри відображення значень	<code>date.toLocaleString()</code> '4/27/2023, 10:33:54 AM' <code>var options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' }; console.log(date.toLocaleString('de-DE', options));</code>
<code>toLocaleDateString([locales[, options]])</code>	Дата у національному стандарті <i>locales</i> – мова(країна), (en-US, en-GB, uk-UA,...) <i>options</i> – параметри відображення значень	<code>date.toLocaleDateString('en-GB')</code> '27/04/2023' <code>date.toLocaleDateString('uk-UA')</code> '27.04.2023' <code>var options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' }; console.log(date.toLocaleDateString('de-DE', options)); // → "Donnerstag, 20. Dezember 2012" options.timeZone = 'UTC'; options.timeZoneName = 'short'; console.log(date.toLocaleDateString('en-US', options)); // → "Thursday, December 20, 2012, GMT"</code>
<code>toLocaleTimeString([locales[, options]])</code>	Дата у національному стандарті <i>locales</i> – мова(країна), (en-US, en-GB, uk-UA,...) <i>options</i> – параметри відображення значень	<code>console.log(date.toLocaleTimeString('en-US'))</code> // → "7:00:00 PM" <code>console.log(date.toLocaleTimeString('en-GB'))</code> // → "03:00:00" <code>const options = { timeZone: 'UTC', timeZoneName: 'short' }; console.log(date.toLocaleTimeString('en-US', options)) // → "3:00:00 AM GMT"</code>

Перетворення у число

Метод/властивість	Значення	Приклад
valueOf()	Повертає відповідну кількість мілісекунд (починаючи з 1 січня 1970р. UTC)	let date = new Date() date.valueOf() //1682580834973
Неявне приведення до Number +date 1*date • • • •	Повертає відповідну кількість мілісекунд (починаючи з 1 січня 1970р. UTC)	+date //1682580834973 -date //-1682580834973 1*date //1682580834973

Різниця між датами

Результатом віднімання дат є різниця у мілісекундах

<code>date1</code> - <code>date2</code>	<pre>//різниця у мілісекундах let date1 = new Date('Thu Apr 27 2023 12:07:17 GMT+0300') let date2 = new Date('Thu Apr 27 2023 12:07:15 GMT+0300') date1-date2 //2000</pre>

Фіксування часу виконання фрагментів коду

<pre>let start = new Date(); //початковий час виконання //----- досліджуваний фрагмент коду //----- let end = new Date(); //кінцевий час виконання //----- //кількість мілісекунд, протягом яких виконувався код millisecondsCount = end - start</pre>	<pre>let start = new Date(); //початковий час виконання //----- // код, який вимагає деякого часу for (let i = 0; i < 9000000; i++) { let someValue = Math.sin (i * i * i); } //----- let end = new Date(); //кінцевий час виконання //----- //кількість мілісекунд, протягом яких виконувався код millisecondsCount = end - start alert(`Використано \${ millisecondsCount } мс`);</pre>

Задача. Визначити привітання у залежності від часу доби (Доброго дня, Доброго ранку, Доброго вечора)

Задача. Вивести дату у форматі

місяць – рік – вихідний/робочий

Задача. Визначити поточну пору року

Задача. Вивести назву дня

Задача. Користувач вводить дату народження. Визначити, скільки йому повних років.

Задача. Користувач іде у відпустку на 56 днів. Визначити дату на момент його повернення.

Задача. Користувач бере авто на вказану кількість годин. Визначити кінцевий час повернення.

Задача. Задаємо користувачу задачу з додавання двох чисел. Визначити скільки секунд було використано для розв'язання задачі.