**new Date(milliseconds) -** Для создания нового объекта типа Date

**new Date(datestring) –**

**new Date(year, month, date, hours, minutes, seconds, ms) –**

new Date(2011, 0, 1, 0, 0, 0, 0); // // 1 января 2011, 00:00:00

new Date(2011, 0, 1); // то же самое, часы/секунды по умолчанию равны 0

**getFullYear() -** **[Получение компонентов даты](https://learn.javascript.ru/datetime" \l "poluchenie-komponentov-daty)**

**getMonth()**

**getDate()**

**getHours(), getMinutes(), getSeconds(), getMilliseconds()**

**getDay() -** Дополнительно можно получить день недели: Неделя в JavaScript начинается с воскресенья, так что результат будет числом **от 0(воскресенье) до 6(суббота)**.

**getTime() -** Возвращает число миллисекунд, прошедших с 1 января 1970 года GMT+0, то есть того же вида, который используется в конструкторе new Date(milliseconds).

**getTimezoneOffset() -** Возвращает разницу между местным и UTC-временем, в минутах.

Следующие методы позволяют устанавливать компоненты даты и времени:

* setFullYear(year [, month, date])
* setMonth(month [, date])
* setDate(date)
* setHours(hour [, min, sec, ms])
* setMinutes(min [, sec, ms])
* setSeconds(sec [, ms])
* setMilliseconds(ms)
* setTime(milliseconds) (устанавливает всю дату по миллисекундам с 01.01.1970 UTC)
* setFullYear(year [, month, date]) – установка года (дополнительно можно задать ещё месяц и число);
* setMonth(month [, date]) – установка месяца от 0 до 11 (0 – январь, 1 – февраль, 2 – март, ..., 11 – декабрь); дополнительно этот метод позволяет ещё установить число;
* setDate(date) – установка числа;
* setHours(hour [, min, sec, ms]) – устанавливает час от 0 до 23 (дополнительно можно ещё установить минуты, секунды и миллисекунды);
* setMinutes(min [, [sec, ms]) – устаналивает минуты от 0 до 59 (дополнительно можно установить ещё секунды и миллисекунды);
* setSeconds(sec, [ms]) – устанавливает секунды от 0 до 59 (дополнительно можно установить ещё миллисекунды);
* setMilliseconds(ms) – устанавливает миллисекунды (от 0 до 999).
* getFullYear() – возвращает год, состоящий из 4 чисел;
* getMonth() – возвращает месяц в формате числа от 0 до 11 (0 – январь, 1 – февраль, 2 – март, ..., 11 – декабрь);
* getDate() – возвращает число месяца от 1 до 31;
* getHours() – возвращает количество часов от 0 до 23;
* getMinutes() – возвращает количество минут от 0 до 59;
* getSeconds() – возвращает количество секунд от 0 до 59;
* getMilliseconds() – возвращает количество миллисекунд от 0 до 999.
* newDate.getFullYear(); //2019
* newDate.getMonth(); // 10
* newDate.getDate(); // 11
* newDate.getHours(); // 10
* newDate.getMinutes(); // 0
* newDate.getSeconds(); // 0
* newDate.getMilliseconds(); // 0

var d = new Date(2011, 1, 28);

d.setDate(d.getDate() + 2);

var d = new Date();

d.setSeconds(d.getSeconds() + 70);

var d = new Date;

d.setDate(1); // поставить первое число месяца

alert( d );

d.setDate(0); // нулевого числа нет, будет последнее число предыдущего месяца

alert( d );

var d = new Date;

d.setDate(-1); // предпоследнее число предыдущего месяца

alert( d );

alert(+new Date) // +date то же самое, что: +date.valueOf()

Для измерения с одновременным выводом результатов в консоли есть методы:

* console.time(метка) – включить внутренний хронометр браузера с меткой.
* console.timeEnd(метка) – выключить внутренний хронометр браузера с меткой и вывести результат.

date.toLocaleString(локаль, опции)

var d = new Date();

alert( d.toString() ); // вывод, похожий на 'Wed Jan 26 2011 16:40:50 GMT+0300'

toUTCString() То же самое, что toString(), но дата в зоне UTC.

var d = new Date();

alert( d.toISOString() ); // вывод, похожий на '2011-01-26T13:51:50.417Z'

var msUTC = Date.parse('2012-01-26T13:51:50.417Z'); // зона UTC

alert( msUTC ); // 1327571510417 (число миллисекунд)

var ms = Date.parse('2012-01-26T13:51:50.417-07:00');

alert( ms ); // 1327611110417 (число миллисекунд)

Метод Date.now() возвращает дату сразу в виде миллисекунд. Технически, он аналогичен вызову +new Date(), но в отличие от него не создаёт промежуточный объект даты, а поэтому – во много раз быстрее.

* Отсчёт месяцев начинается с нуля.
* Отсчёт дней недели (для getDay()) тоже начинается с нуля (и это воскресенье).
* Объект Date удобен тем, что автокорректируется. Благодаря этому легко сдвигать даты.
* При преобразовании к числу объект Date даёт количество миллисекунд, прошедших с 1 января 1970 UTC. Побочное следствие – даты можно вычитать, результатом будет разница в миллисекундах.
* Для получения текущей даты в миллисекундах лучше использовать Date.now(), чтобы не создавать лишний объект Date (кроме IE8-)
* Для бенчмаркинга лучше использовать performance.now() (кроме IE9-), он в 1000 раз точнее.
* // создадим дату 01.01.2015 (неуказанные параметры по умолчанию равны: число - 01, часы - 00, минуты - 00, секунды - 00, миллисекунды – 000).
* var date1 = new Date(2015,01);
* // создадим дату 24.01.2015, 21:23
* var date2 = new Date(2015,01,24,21,23);
* // создание даты на основе строки в формате DD.MM.YY
* var date1 = new Date('05.11.19');
* // создадим дату на основе строки в формате YYYY-MM-DDThh:mm:ss.sss (символ T используется для разделения даты и времени)
* var date2 = new Date('2015-02-24T21:23');
* // создадим дату на основе строки с указанием часового пояса (формат YYYY-MM-DDThh:mm:ss.sss±hh:mm):
* var date3 = new Date('2015-02-24T22:02+03:00');

var days =["Воскресенье", "Понедельник", "Вторник", "Среда", "Четверг", "Пятница", "Суббота"];

*// получим текущую дату*

var now = new **Date**();

*// выведем в консоль день недели*

console.**log**(now);

console.**log**(now.**getDay**());

console.**log**('Сегодня ' + days[now.**getDay**()]);

// создадим дату 15.04.2019 18:43:59 (в часовом поясе пользователя)

var newDate = new Date(2019, 03, 15, 18, 43, 59);

console.log(newDate.toString()); // Mon Apr 15 2019 18:43:59 GMT+1000 (Владивосток, стандартное время)

console.log(newDate.toDateString()); // Mon Apr 15 2019

console.log(newDate.toTimeString()); // 18:43:59 GMT+1000 (Владивосток, стандартное время)

console.log(newDate.toLocaleString()); // 15.04.2019, 18:43:59

console.log(newDate.toLocaleDateString()); // 15.04.2019

console.log(newDate.toLocaleTimeString()); // 18:43:59

console.log(newDate.toUTCString()); // Mon, 15 Apr 2019 08:43:59 GMT

console.log(newDate.toISOString()); // 2019-04-15T08:43:59.000Z