

# РАБОТА С ДАТОЙ И ВРЕМЕНЕМ

JavaScript предлагает встроенный **объект Date** для работы с датой и временем. Его можно использовать для хранения времени создания/изменения, для измерения времени или просто для вывода текущей даты. Вот несколько основных операций и функций для работы с датой и временем в JavaScript:

## Создание объекта Date

```
let now = new Date(); // Создание объекта Date с текущей датой и временем
let specificDate = new Date(2021, 0, 1); // Создание объекта Date с конкретной датой (год, месяц, день)
```

## Получение компонентов даты и времени

```
let date = now.getDate(); // Получение дня месяца (от 1 до 31)
let month = now.getMonth(); // Получение месяца (от 0 до 11, 0 - январь)
let year = now.getFullYear(); // Получение года (четырёхзначное число)
let hours = now.getHours(); // Получение часов (от 0 до 23)
let minutes = now.getMinutes(); // Получение минут (от 0 до 59)
let seconds = now.getSeconds(); // Получение секунд (от 0 до 59)
let milliseconds = now.getMilliseconds(); // Получение миллисекунд (от 0 до 999)
let dayOfWeek = now.getDay(); // Получение дня недели (от 0 до 6, 0 - воскресенье)
```

## Форматирование даты и времени

```
let formattedDate = now.toLocaleDateString(); // Форматирование даты в формате локали браузера
let formattedTime = now.toLocaleTimeString(); // Форматирование времени в формате локали браузера
let formattedDateTime = now.toLocaleString(); // Форматирование даты и времени в формате локали браузера
```

## Установка даты и времени:

```
now.setFullYear(2022); // Установка года
now.setMonth(11); // Установка месяца (11 - декабрь)
now.setDate(25); // Установка дня месяца
now.setHours(10); // Установка часов
now.setMinutes(30); // Установка минут
now.setSeconds(0); // Установка секунд
now.setMilliseconds(500); // Установка миллисекунд
```

## Преобразование к числу, разность дат

Если объект Date преобразовать в число, то получим таймстамп по аналогии с date.getTime()

```
let date = new Date();
alert(+date); // количество миллисекунд, то же самое,
что date.getTime()
```

Если нужно просто измерить время, объект Date нам не нужен. Существует особый **метод Date.now()**, возвращающий текущую метку времени.

Семантически он эквивалентен new Date().getTime(), однако метод не создаёт промежуточный объект Date. Так что этот способ работает быстрее и не нагружает сборщик мусора.

В таком случае, предыдущий пример лучше переписать так:

```
let start = Date.now(); // количество миллисекунд с 1 января
1970 года

// выполняем некоторые действия
for (let i = 0; i < 100000; i++) {
  let doSomething = i * i * i;
}

let end = Date.now(); // заканчиваем отсчёт времени

alert( `Цикл отработал за ${end - start} миллисекунд` ); //
вычитаются числа, а не даты
```

## Бенчмаркинг

Вычисления, измеряющие производительность, также называют «бенчмарками» (**benchmark**).

В основе бенчмаркинга лежит сбор данных о времени выполнения, использовании ресурсов или других показателях производительности для различных сценариев или нагрузок.

## Преобразование даты в строку

```
let dateString = now.toDateString(); // Преобразование даты в строку формата "Fri Dec 31 2021"
```

```
let isoString = now.toISOString(); // Преобразование даты в строку формата ISO "2021-12-31T23:59:59.999Z"
```

```
let utcString = now.toUTCString(); // Преобразование даты в строку формата UTC "Fri, 31 Dec 2021 23:59:59 GMT"
```

## Манипуляции с датой и временем

```
now.setDate(now.getDate() + 7); // Прибавление 7 дней к текущей дате
```

```
now.setHours(now.getHours() + 1); // Прибавление 1 часа к текущему времени
```