

# Front-end

---

JavaScript



# JS. Numbers

---

- Date



# JS. Numbers

---

## Date

Date — класс для работы со временем. В основном используется для отображения и вычисления.

Время отображается в двух форматах:

- UTC — время по Гринвичу;
- В текущей часовой зоне (например, UTC+3 для Москвы).

Поэтому часть методов для взаимодействия со временем имеют два формата — для UTC и для текущей часовой зоны.

Часовая зона или часовой пояс — правильный перевод английского timezone. В русскоязычной разработке также часто говорят таймзона.



# JS. Numbers

---

## Создание текущей даты и времени.

Получить текущую дату и время в JavaScript осуществляется посредством создания экземпляра объекта Date без указания параметров:

Текущая дата (дата и время, которая была в момент создания экземпляра объекта Date на локальном компьютере пользователя)

```
let now = new Date();
```

```
console.log(now);
```

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



# JS. Numbers

---

Если нужно получить только сегодняшнюю дату в формате строки, то можно воспользоваться методом `toLocaleDateString`:

```
let now = new Date().toLocaleDateString(); // => 19.12.2019
```

Текущее время пользователя можно получить так:

```
let now = new Date().toLocaleTimeString(); // => 11:02:48
```

```
let now = new Date().toLocaleTimeString().slice(0,-3); // => 11:02
```

Дату и время в формате строки можно получить следующим образом:

```
let now = new Date().toLocaleString(); // => 19.12.2019, 11:02:48
```



# JS. Numbers

---

Получение отдельных компонентов даты и времени

В JavaScript для получения отдельных компонентов даты и времени предназначены следующие методы:

- **getFullYear()** — возвращает год;
- **getMonth()** — возвращает месяц с 0 до 11;
- **getDate()** — возвращает день месяца с 1 до 31;
- **getDay()** — возвращает порядковый номер дня недели с 0 до 6;
- **getHours()** — возвращает часы с 0 до 23;
- **getMinutes()** - возвращает минуты от 0 до 59;
- **getSeconds()** - возвращает секунды от 0 до 59;
- **getMilliseconds()** - возвращает миллисекунды от 0 до 999.

Все эти методы возвращают отдельные компоненты даты и времени в соответствии с часовым поясом установленном на локальном устройстве пользователя.



# JS. Numbers

---

Все вышеперечисленные методы возвращают значения для текущей часовой зоны. Если необходимо вернуть время по Гринвичу, то нужно в метод добавить UTC: `getUTCFullYear()`, `getUTCMonth()`, `getUTCDate()`, `getUTCDay()`, `getUTCHours()`, `getUTCMinutes()`, `getUTCSeconds()`, `getUTCMilliseconds()`.

**Есть два метода, которые не привязаны к часовой зоне:**

- `getTime()` возвращает значение в миллисекундах, прошедших с 1 января 1970 года, соответствующее указанной дате по UTC.
- `getTimezoneOffset()` возвращает смещение в минутах между текущей часовой зоной и UTC.



# JS. Numbers

---

Дата 1 января 1970 года досталась нам в наследство от UNIX. Именно в 1970 году началась разработка этой системы. Это время называется POSIX временем, Unix временем или Unix timestamp.



# JS. Numbers

---

Получить номер дня недели в JavaScript можно с помощью метода `getDay()`.

Данный метод возвращает число от 0 до 6 (0 – воскресенье, 1 – понедельник, ..., 6 – суббота).

Пример, в котором переведем день недели из числового в строковое представление:

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

Получим текущую дату

```
let now = new Date();
```

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



# JS. Numbers

---

## Установка значений

Также, как и с получением значений, у многих методов есть присваивание значений в локальной часовой зоне и UTC. Для локальной:

- `setFullYear(год, месяц, день)` устанавливает год, значения месяца и дня необязательны.
- `setMonth(месяц, день)` устанавливает месяц, передавать день необязательно.
- `setDate(день)` устанавливает день месяца.
- `setHours(часы, минуты, секунды, миллисекунды)` устанавливает часы. Значения минут, секунд, миллисекунд необязательны.
- `setMinutes(минуты, секунды, миллисекунды)` - устанавливает минуты. Секунды и миллисекунды необязательны.
- `setSeconds(секунды, миллисекунды)` устанавливает секунды. Миллисекунды передавать необязательно.
- `setMilliseconds(миллисекунды)` - устанавливает миллисекунды.