






Frontender[1.0] JavaScript - ДАТЫ и ВРЕМЯ. Date JS. Методы, особенности, UTC vs ISO

 YouTube	https://youtu.be/hpnWgdvznpl
 Telegram	https://t.me/Dmitry_Kolotilshikov
 Github	https://github.com/DmitryKolotilshikov/
 Boosty	https://boosty.to/dmitry_ko
# Номер урока	14



Задачи к этому уроку тут https://boosty.to/dmitry_ko

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="script.js"></script>
  <title>Data & Time</title>
</head>
<body>
</body>
</html>
```

```
// Date & Time
// объект Date - содержит дату и время, также методы управления ими.

const log = console.log;

let d = new Date();

d = new Date(0);
/*
объект Date с временем, равным количеству миллисекунд (тысячная доля секунды), прошедших с 1
января 1970 года UTC+0
Целое число, представляющее собой количество миллисекунд, прошедших с начала 1970 года, назыв
ается таймстамп (англ. timestamp)
*/
```

```

d = new Date(2024, 7, 14, 9, 30, 0);

d = new Date("2024-08-14");
d = new Date("2024-08-14 09:30");

d = new Date("08-14-2024");
d = new Date("2024-08-14T00:00:00");

// log(d);

// Текущий таймстамп
d = Date.now();

d = new Date("2023-08-14T00:00:00");
d = d.getTime(); // 1691953200000

d = new Date(1691953200000);

d = Math.floor(Date.now() / 1000 / 60 / 60 / 24 / 365) // прошло лет с 1970

// log(d);

// ----- Методы -----

d = Date.parse("2024-08-14T00:00:00");

/*
Следующие методы позволяют установить 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 дату в виде целого количества миллисекунд, прошедших с 0
1.01.1970 UTC)
*/

d = new Date();

// СЕТТЕРЫ (установочники)

// d.setDate(1);
// d.setDate(31);
// d.setDate(0);
// d.setDate(-1);

// d.setHours(0);
// d.setMinutes(0);
// d.setSeconds(0);
// d.setFullYear(2025);
// d.setFullYear(2025, 0, 10);

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

let d1 = new Date();

```

```

// log(d1, " ", d);
// log(d1.getTime())
// log(d.getTime())
// log((d.getTime() - d1.getTime()) / 1000 / 60 / 60 / 24);

// ГЕТТЕРЫ (получатели)

log(d.getDate());
log(d.valueOf()); // таймстамп
log(d.getFullYear());
log(d.getMonth()); // 7 - это август
log(d.getMonth() + 1);
log(d.getDay()); // день недели
log(d.getHours());

// представление даты
// https://www.w3schools.com/jsref/jsref_tolocalestring.asp

let date = Intl.DateTimeFormat("default").format(d);
date = Intl.DateTimeFormat("en-GB").format(d);

date = Intl.DateTimeFormat("ru-RU", { day: "numeric", month: "long", year: "numeric" }).format(d);

date = d.toLocaleString("ru-RU", {
    year: "numeric",
    month: "long",
    weekday: "long",
    day: "numeric",
    hour: "numeric",
    minute: "numeric",
    // second: "numeric",
    // timeZone: "Australia/Sydney",
    // timeZone: "Europe/Berlin",
    // timeZone: "Europe/Moscow",
    timeZone: "Europe/Minsk",
});

log(date);

// -----

// UTC и ISO
// UTC (Coordinated Universal Time)
/*
UTC (Координированное универсальное время) — это стандарт времени, который используется по всему миру.

--> UTC: Подходит для отображения даты и времени в текстовом виде, когда важна читаемость.
*/

/*
ISO (International Organization for Standardization)
ISO 8601 — это международный стандарт для представления дат и времени.

--> ISO: Предпочтительно использовать для хранения данных и передачи между системами, так как формат строгий и однозначный.

```

```
*/
```

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

```
console.group("UTC & ISO");
```

```
log(now.toUTCString());
```

```
log(now.toISOString());
```

```
console.groupEnd();
```