

Дата и время. JSON.



# Дата и время

объект: Date. Он содержит дату и время, а также предоставляет методы управления ими.

Даты хранятся в нем как количество миллисекунд, прошедших с полуночи 1 января 1970 г. согласно универсальному временному коду (Universal Time Code, UTC).

# Дата и время

Чтобы создать объект даты, используйте оператор `new` с конструктором `Date`:

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

# Дата и время

Если конструктор `Date` вызывается без аргументов, создается объект с текущими значениями даты и времени. Чтобы создать объект `Date` с другой датой или временем, нужно передать в конструктор значение даты в миллисекундах, прошедших с полуночи 1 января 1970 г. согласно UTC.

# Дата и время

```
let Jan02_1970 = new Date(24 * 3600 * 1000);
```

```
let date = new Date("2017-01-26");
```

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

```
new Date(2011, 0, 1);
```

# Получение значений

- `getFullYear()` - Получить год (4 цифры)
- `getMonth()` - Получить месяц, от 0 до 11.
- `getDate()` - Получить день месяца, от 1 до 31
- `getHours()`, `getMinutes()`, `getSeconds()`, `getMilliseconds()`
- `getDay()` - Получить день недели от 0 (воскресенье) до 6 (суббота).

# Установка значений

- `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)`

# Date.now()

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



# Разбор строки с датой

Метод `Date.parse(str)` считывает дату из строки.

Формат строки должен быть следующим: `YYYY-MM-DDTHH:mm:ss.sssZ`, где:

- `YYYY-MM-DD` – это дата: год-месяц-день.
- Символ "T" используется в качестве разделителя.
- `HH:mm:ss.sss` – время: часы, минуты, секунды и миллисекунды.
- Необязательная часть 'Z' обозначает часовой пояс в формате `+hh:mm`. Если указать просто букву Z, то получим UTC+0.

Возможны и более короткие варианты, например, `YYYY-MM-DD` или `YYYY-MM`, или даже `YYYY`.

# Задание 1

Создать дату 18 декабря 2019 года, 12 часов, 20 минут

## Задание 2

Вывести какой день недели было 4 марта 2000 года

# Задание 3

Вывести дату в формате dd.mm.yyyy hh:mm из количества миллисекунд:  
1590749435756

# JSON

JSON - это формат данных

JSON поддерживает значения трех типов:

- Простые значения - строки, числа, логические значения и значения null можно представлять в JSON, используя тот же синтаксис, что и в JavaScript. Специальное значение undefined не поддерживается.
- Объекты - первый сложный тип данных, служащий для хранения упорядоченных пар ключей и значений. Каждое значение может быть примитивным ИЛИ СЛОЖНЫМ ТИПОМ.
- Массивы - второй сложный тип данных, который представляет упорядоченный список значений, доступных по числовому индексу. Значениями массивов могут быть данные любого типа, в том числе простые значения, объекты и даже другие массивы.

# JSON

JSON легко использовать для обмена данными

У объекта JSON есть два метода: `stringify()` и `parse()`. В простых сценариях они просто сериализуют объект JavaScript в строку JSON и преобразуют ее в значение JavaScript соответственно.

- `JSON.stringify` для преобразования объектов в JSON.
- `JSON.parse` для преобразования JSON обратно в объект.

# JSON

Пример:

```
let book = {  
  
  title: "Professional JavaScript",  
  
  authors : [ "Nicholas C. Zakas" ],  
  
  edition : 3, year: 2011  
  
  let jsonText = JSON.stringify(book);
```

```
{"title":"Professional  
JavaScript","authors":["Nicholas C.  
Zakas"], "edition" : 3, "year" : 2011}
```

# Задание 1

Преобразуйте user в JSON, затем прочитайте этот JSON в другую переменную.

```
let user = {  
  
  name: "User photo",  
  
  size: "200 x 200",  
  
};
```