




## Задание по программированию: Управление временем

 Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 14 дек. г., 13:59 +06  
Вы все еще можете выполнить задание до завершения курса.

### Инструкции

Моя работа

Обсуждения

В этом задании необходимо реализовать дополнительные функции для управления временем.

```
1  var date = require('./index.js');
2
3  var time = date('2017-05-16 13:45')
4
5  .add(24, 'hours')
6
7  .subtract(1, 'months')
8
9  .add(3, 'days')
10
11 .add(15, 'minutes');
12
13 console.info(time.value);
14
15 // "2017-04-20 14:00"
```

### Заготовка

## Условия

- В функцию всегда передается строка в правильном и полном формате. Дополнительных проверок не требуется.
- Формат даты — "YYYY-MM-DD HH:SS", где YYYY — год, MM — месяц, DD — день, HH — час, SS — минуты
- В функции add/subtract всегда передается целое число
- Гарантируется, что после всех манипуляций получится корректная дата, которая будет не ранее 1 января 2000 года

## Функции add/subtract

Каждая функция принимает первым аргументом количество единиц, на которое нужно изменить дату, а вторым — единицу измерения.

Можно менять следующие значения: years (годы), months (месяцы), days (дни), hours (часы), minutes (минуты)

## Обработка ошибок

Если первый аргумент отрицательный, либо второй содержит неизвестную единицу измерения, функции должны выбросить исключение `TypeError`.

## Подсказки

Решение задания не требует использования регулярных выражений, но с ними реализация может быть лаконичнее.

## Даты

В реализации советуем использовать объект [Date](#). В нём реализован правильный переход через месяц, год, час:

```
1 var date = new Date(2016, 8, 26, 15, 0);
2
3 console.info(date);
4
5 // "2016-09-26T10:00:00.000Z"
6
```

```
6
7   date.setMinutes(60 * 24 * 365)
8
9   console.info(date);
10
11  // "2017-09-26T10:00:00.000Z" (сменился год)
```

Обратите внимание, что нумерация месяцев идет с "0"

## Регулярные выражения

При использовании регулярных выражений полезно прочитать [документацию](#). Обратите внимание на символьный класс "\d", сопоставляемый с числом и на квантификаторы "{}", через которые удобно указать точное количество символов

Для разбора строки, может пригодиться строковый метод [match](#).

## Реализация интерфейса

Для реализации рекомендуемого интерфейса необходимо из функции date вернуть объект с методами. Чтобы эти методы можно было вызывать цепочкой, из каждого метода нужно возвращать сам объект.

Пример:

```
1  function createObject() {
2      return {
3          first: function () {
4              return this;
5          },
6          second: function () {
7              return this;
8          }
9      }
10 }
11
12 createObject().first().second().second();
```

**Важно.** В рамках курса мы следуем стандарту языка EcmaScript 5. Подробнее о EcmaScript вы можете узнать в [википедии](#).

## Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

