

Раздел

[Регулярные выражения](#)

Навигация по уроку

Обратные символьные
классы

Точка – это любой символ

Итого

Комментарии

Поделиться

[Редактировать на GitHub](#) → [Регулярные выражения](#) 10-го октября 2019

Символьные классы

Рассмотрим практическую задачу – у нас есть номер телефона вида "+7(903)-123-45-67", и нам нужно превратить его в строку только из чисел: 79035419441.

Для этого мы можем найти и удалить все, что не является числом. С этим нам помогут символьные классы.

Символьный класс – это специальное обозначение, которое соответствует любому символу из определённого набора.

Для начала давайте рассмотрим класс «цифра». Он обозначается как `\d` и в регулярном выражении соответствует «любой одной цифре».

Например, давайте найдём первую цифру в номере телефона:

```
1 let str = "+7(903)-123-45-67";
2
3 let regexp = /\d/;
4
5 alert( str.match(regexp) ); // 7
```

Без флага `g` регулярное выражение ищет только первое совпадение, то есть первую цифру `\d`.

Давайте добавим флаг `g`, чтобы найти все цифры:

```
1 let str = "+7(903)-123-45-67";
2
3 let regexp = /\d/g;
4
5 alert( str.match(regexp) ); // массив совпадений: 7,9,0
6
7 // и можно сделать из них уже чисто цифровой номер теле
8 alert( str.match(regexp).join('') ); // 79035419441
```

Это был символьный класс для цифр. Есть и другие символьные классы.

Наиболее используемые:

`\d` («`d`» от английского «`digit`» означает «цифра»)

Цифра: символ от 0 до 9.

`\s` («`s`»: от английского «`space`» – «пробел»)

Пробельные символы: включает в себя символ пробела, табуляции `\t`, перевода строки `\n` и некоторые другие редкие пробельные символы, обозначаемые как `\v`, `\f` и `\r`.

`\w` («`w`»: от английского «`word`» – «слово»)

Символ «слова», а точнее – буква латинского алфавита или цифра или подчёркивание `_`. Нелатинские буквы не являются частью класса `\w`, то есть буква русского алфавита не подходит.

Для примера, `\d\s\w` обозначает «цифру», за которой идёт пробельный символ, а затем символ слова, например `1 a`.

Регулярное выражение может содержать как обычные символы, так и символьные классы.

Например, `CSS\d` соответствует строке `CSS` с цифрой после неё:

```
1 let str = "Есть ли стандарт CSS4?";
```

Раздел

Регулярные выражения

Навигация по уроку

Обратные символьные классы

Точка – это любой символ

Итого

Комментарии

Поделиться



Редактировать на GitHub



```
2 let regexp = /CSS\d/
3
4 alert( str.match(regexp) ); // CSS4
```

Также мы можем использовать несколько символьных классов:

```
1 alert( "I love HTML5!".match(/s\w\w\w\w\d/) ); // I love HTML5
```

Соответствие (каждому символьному классу соответствует один символ результата):

I love HTML5

Обратные символьные классы

Для каждого символьного класса существует «обратный класс», обозначаемый той же буквой, но в верхнем регистре.

«Обратный» означает, что он соответствует всем другим символам, например:

\D

Не цифра: любой символ, кроме \d, например буква.

\S

Не пробел: любой символ, кроме \s, например буква.

\W

Любой символ, кроме \w, то есть не буквы из латиницы, не знак подчёркивания и не цифра. В частности, русские буквы принадлежат этому классу.



Мы уже видели, как сделать чисто цифровой номер из строки вида +7(903)-123-45-67: найти все цифры и соединить их.



```
1 let str = "+7(903)-123-45-67";
2
3 alert( str.match(/\d/g).join('') ); // 79031234567
```

Альтернативный, более короткий путь – найти нецифровые символы \D и удалить их из строки:

```
1 let str = "+7(903)-123-45-67";
2
3 alert( str.replace(/\D/g, "") ); // 79031234567
```

Точка – это любой символ

Точка . – это специальный символьный класс, который соответствует «любому символу, кроме новой строки».

Для примера:

```
1 alert( "Ю".match(/./) ); // Ю
```

Или в середине регулярного выражения:

```
1 let regexp = /CS.4/;
2
3 alert( "CSS4".match(regexp) ); // CSS4
4 alert( "CS-4".match(regexp) ); // CS-4
5 alert( "CS 4".match(regexp) ); // CS 4 (пробел тоже явл.
```

Обратите внимание, что точка означает «любой символ», но не «отсутствие символа». Там должен быть какой-либо символ, чтобы соответствовать условию поиска:

```
1 alert( "CS4".match(/CS.4/) ); // null, нет совпадений
```

Точка как буквально любой символ, с флагом «s»

Обычно точка не соответствует символу новой строки `\n`.

То есть, регулярное выражение `A.B` будет искать символ `A` и затем `B`, с любым символом между ними, кроме перевода строки `\n`:

```
1 alert( "A\nB".match(/A.B/) ); // null (нет совпадения)
```

Но во многих ситуациях точкой мы хотим обозначить действительно «любой символ», включая перевод строки.

Как раз для этого нужен флаг `s`. Если регулярное выражение имеет его, то точка `.` соответствует буквально любому символу:

```
1 alert( "A\nB".match(/A.B/s) ); // A\nB (совпадение!)
```

⚠ Внимание, пробелы!

Обычно мы уделяем мало внимания пробелам. Для нас строки `1-5` и `1 - 5` практически идентичны.

Но если регулярное выражение не учитывает пробелы, оно может не сработать.

Давайте попробуем найти цифры, разделённые дефисом:

```
1 alert( "1 - 5".match(/\d-\d/) ); // null, нет совпадения
```

Исправим это, добавив пробелы в регулярное выражение `\d - \d`:

```
1 alert( "1 - 5".match(/\d - \d/) ); // 1 - 5, теперь  
2 // или можно использовать класс \s:  
3 alert( "1 - 5".match(/\d\s-\s\d/) ); // 1 - 5, тоже
```

Пробел – это символ. Такой же важный, как любой другой.

Нельзя просто добавить или удалить пробелы из регулярного выражения, и ожидать, что оно будет также работать.

Другими словами, в регулярном выражении все символы имеют значение, даже пробелы.

Итого

Существуют следующие символьные классы:

- `\d` – цифры.
- `\D` – не цифры.
- `\s` – пробельные символы, табы, новые строки.
- `\S` – все, кроме `\s`.
- `\w` – латиница, цифры, подчёркивание `'_'`.
- `\W` – все, кроме `\w`.
- `.` – любой символ, если с флагом регулярного выражения `s`, в противном случае любой символ, кроме перевода строки `\n`.

Раздел

Регулярные выражения

Навигация по уроку

Обратные символьные классы

Точка – это любой символ

Итого

Комментарии

Поделиться



Редактировать на GitHub

Раздел

[Регулярные выражения](#)

Навигация по уроку

Обратные символьные
классы

Точка – это любой символ

Итого

Комментарии

Поделиться



[Редактировать на GitHub](#)



...Но это не всё!

В кодировке Юникод, которую JavaScript использует для строк, каждому символу соответствует ряд свойств, например – какого языка это буква (если буква), является ли символ знаком пунктуации, и т.п.

Можно искать, в том числе, и по этим свойствам. Для этого нужен флаг u, который мы рассмотрим в следующей главе.

Проводим [курсы по JavaScript и фреймворкам](#).



Комментарии

перед тем как писать...

