

Раздел

[Регулярные выражения](#)

Навигация по уроку

Поиск в начале строки ^

Поиск в конце строки \$

Ищем \n вместо ^ \$

Комментарии

Поделиться

[Редактировать на GitHub](#)[🏠 → Регулярные выражения](#)

4-го сентября 2019

Многострочный режим якорей ^ \$, флаг "m"

Многострочный режим включается флагом m.

Он влияет только на поведение ^ и \$.

В многострочном режиме они означают не только начало/конец текста, но и начало/конец каждой строки в тексте.

Поиск в начале строки ^

В примере ниже текст состоит из нескольких строк. Шаблон /^d/gm берёт цифру с начала каждой строки:

```
1 let str = `1е место: Винни
2 2е место: Пятачок
3 3е место: Слонопотам`;
4
5 alert( str.match(/^d/gm) ); // 1, 2, 3
```

Обратим внимание – без флага m было бы найдено только первое число:

```
1 let str = `1е место: Винни
2 2е место: Пятачок
3 3е место: Слонопотам`;
4
5 alert( str.match(/^d/g) ); // 1
```

Так происходит, потому что в обычном режиме каретка ^ – это только начало текста, а в многострочном – начало любой строки.

На заметку:

«Начало строки», формально, означает «сразу после перевода строки», то есть проверка ^ в многострочном режиме верна на всех позициях, которым предшествует символ перевода строки \n.

И в начале текста.

Поиск в конце строки \$

Символ доллара \$ ведёт себя аналогично.

Регулярное выражение \d\$ ищет последнюю цифру в каждой строке

```
1 let str = `Винни: 1
2 Пятачок: 2
3 Слонопотам: 3`;
4
5 alert( str.match(/\d$/gm) ); // 1,2,3
```

Без флага m якорь \$ обозначал бы конец всей строки, и была бы найдена только последняя цифра.

Ищем \n вместо ^ \$

Раздел

[Регулярные выражения](#)

Навигация по уроку

Поиск в начале строки ^

Поиск в конце строки \$

Ищем \n вместо ^ \$

Комментарии

Поделиться



[Редактировать на GitHub](#)

Для того, чтобы найти конец строки, можно использовать не только якоря ^ и \$, но и символ перевода строки \n.

В чём разница? Давайте посмотрим на примере.

Поищем \d\n вместо \d\$:



```
1 let str = `Винни: 1
2   Пятачок: 2
3   Слонопотам: 3`;
4
5 alert( str.match(/\d\n/gm) ); // 1\n,2\n
```



Как видим, совпадений теперь два, а не три.

Это потому, что после 3 нет перевода строки (а конец текста, подходящий под \$ – есть).

Ещё одно отличие: теперь в каждое найденное совпадение входит символ перевода строки \n. В отличие от якорей ^ \$, которые только проверяют условие (начало/конец строки), \n – символ и входит в результат.

Так что \n в шаблоне используется, когда нам нужен сам символ перевода строки в результате, а якоря – когда хотим найти что-то в начале/конце строки.

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

Комментарии

перед тем как писать...

