

Раздел

[Регулярные выражения](#)

Навигация по уроку

Опережающая проверка

Негативная опережающая проверка

Ретроспективная проверка

Скобочные группы

Итого

Задачи (2)

Комментарии

Поделиться

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

📅 2-го октября 2020

Опережающие и ретроспективные проверки

В некоторых случаях нам нужно найти соответствия шаблону, но только те, за которыми или перед которыми следует другой шаблон.

Для этого в регулярных выражениях есть специальный синтаксис: опережающая (lookahead) и ретроспективная (lookbehind) проверка.

В качестве первого примера найдём стоимость из строки 1 индейка стоит 30€. То есть, найдём число, после которого есть знак валюты €.

Опережающая проверка

Синтаксис опережающей проверки: X(?=Y).

Он означает: найди X при условии, что за ним следует Y. Вместо X и Y здесь может быть любой шаблон.

Для целого числа, за которым идёт знак €, шаблон регулярного выражения будет \d+(?=.*€):

```
1 let str = "1 индейка стоит 30€";
2
3 alert( str.match(/\d+(?=.*€)/) ); // 30, число 1 проигнор
```

Обратим внимание, что проверка – это именно проверка, содержимое скобок (?=.*€) не включается в результат 30.

При поиске X(?=Y) движок регулярных выражений, найдя X, проверяет есть ли после него Y. Если это не так, то игнорирует совпадение и продолжает поиск дальше.

Возможны и более сложные проверки, например X(?=Y)(?=Z) означает:

1. Найти X.
2. Проверить, идёт ли Y сразу после X (если нет – не подходит).
3. Проверить, идёт ли Z сразу после X (если нет – не подходит).
4. Если обе проверки прошли – совпадение найдено.

То есть, этот шаблон означает, что мы ищем X при условии, что за ним идёт и Y и Z.

Такое возможно только при условии, что шаблоны Y и Z не являются взаимно исключающими.

Например, \d+(?=.*\s)(?=.*30) ищет \d+ при условии, что за ним идёт пробел, и где-то впереди есть 30:

```
1 let str = "1 индейка стоит 30€";
2
3 alert( str.match(/\d+(?=.*\s)(?=.*30)/) ); // 1
```

В нашей строке это как раз число 1.

Негативная опережающая проверка

Допустим, нам нужно узнать из этой же строки количество индеек, то есть число \d+, за которым НЕ следует знак €.

Для этой задачи мы можем применить негативную опережающую проверку.

Синтаксис: X(?!Y)

Он означает: найди такой X, за которым НЕ следует Y.

Раздел

Регулярные выражения

Навигация по уроку

Опережающая проверка

Негативная опережающая проверка

Ретроспективная проверка

Скобочные группы

Итого

Задачи (2)

Комментарии

Поделиться



Редактировать на GitHub



```
1 let str = "2 индейки стоят 60€";
2
3 alert( str.match(/\d+(?!€)/) ); // 2 (в этот раз проигн
```

Ретроспективная проверка

Опережающие проверки позволяют задавать условия на то, что «идёт после».

Ретроспективная проверка выполняет такую же функцию, но с просмотром назад. Другими словами, она находит соответствие шаблону, только если перед ним есть что-то заранее определённое.

Синтаксис:

- Позитивная ретроспективная проверка: `(?<=Y)X`, ищет совпадение с `X` при условии, что перед ним ЕСТЬ `Y`.
- Негативная ретроспективная проверка: `(?<!Y)X`, ищет совпадение с `X` при условии, что перед ним НЕТ `Y`.

Чтобы протестировать ретроспективную проверку, давайте поменяем валюту на доллары США. Знак доллара обычно ставится перед суммой денег, поэтому для того чтобы найти `$30`, мы используем `(?<=\$)\d+` – число, перед которым идёт `$`:

```
1 let str = "1 индейка стоит $30";
2
3 // знак доллара экранируем \$, так как это специальный
4 alert( str.match(/(?<=\$)\d+/) ); // 30, одинокое число
```

Если нам необходимо найти количество индейек – число, перед которым не идёт `$`, мы можем использовать негативную ретроспективную проверку `(?<!\$)\d+`:

```
1 let str = "2 индейки стоят $60";
2
3 alert( str.match(/(?<!\$)\d+/) ); // 2 (проигнорировала
```

Скобочные группы

Как правило, то что находится внутри скобок, задающих опережающую и ретроспективную проверку, не включается в результат совпадения.

Например, в шаблоне `\d+(?=€)` знак `€` не будет включён в результат. Это логично, ведь мы ищем число `\d+`, а `(?=€)` – это всего лишь проверка, что за ним идёт знак `€`.

Но в некоторых ситуациях нам может быть интересно захватить и то, что в проверке. Для этого нужно обернуть это в дополнительные скобки.

В следующем примере знак валюты `(€|kr)` будет включён в результат вместе с суммой:

```
1 let str = "1 индейка стоит 30€";
2 let regexp = /\d+(?=(€|kr))/; // добавлены дополнительные
3
4 alert( str.match(regexp) ); // 30, €
```

То же самое можно применить к ретроспективной проверке:

```
1 let str = "1 индейка стоит $30";
2 let regexp = /(?<=(\$|£))\d+/;
3
4 alert( str.match(regexp) ); // 30, $
```

Раздел

Регулярные выражения

Навигация по уроку

Опережающая проверка

Негативная опережающая проверка

Ретроспективная проверка

Скобочные группы

Итого

Задачи (2)

Комментарии

Поделиться



Редактировать на GitHub



Итого

Опережающая и ретроспективная проверки удобны, когда мы хотим искать шаблон по дополнительному условию на контекст, в котором он находится.

Для простых регулярных выражений мы можем сделать похожую вещь «вручную». То есть, найти все совпадения, независимо от контекста, а затем в цикле отфильтровать подходящие.

Как мы помним, `regex.match` (без флага `g`) и `str.matchAll` (всегда) возвращают совпадения со свойством `index`, которое содержит позицию совпадения в строке, так что мы можем посмотреть на контекст.

Но обычно регулярные выражения удобнее.

Виды проверок:

Шаблон	Тип	Совпадение
<code>X(?:Y)</code>	Позитивная опережающая	<u>X</u> , если за ним следует <u>Y</u>
<code>X(?:!Y)</code>	Негативная опережающая	<u>X</u> , если за ним НЕ следует <u>Y</u>
<code>(?<=Y)X</code>	Позитивная ретроспективная	<u>X</u> , если следует за <u>Y</u>
<code>(?!Y)X</code>	Негативная ретроспективная	<u>X</u> , если НЕ следует за <u>Y</u>

✓ Задачи

Найдите неотрицательные целые

Есть строка с целыми числами.

Создайте регулярное выражение, которое ищет только неотрицательные числа. Ноль разрешён.

Пример использования:

```
1 let regexp = /ваше регулярное выражение/g;
2
3 let str = "0 12 -5 123 -18";
4
5 alert( str.match(regexp) ); // 0, 12, 123
```

решение

Вставьте после фрагмента

Есть строка с HTML-документом.

Вставьте после тега `<body>` (у него могут быть атрибуты) строку `<h1>Hello</h1>`.

Например:

```
1 let regexp = /ваше регулярное выражение/;
2
3 let str = `
4 <html>
5   <body style="height: 200px">
6     ...
7   </body>
8 </html>
9 `;
10
11 str = str.replace(regexp, `<h1>Hello</h1>`);
```

После этого значение `str`:

```
1 <html>
```

Раздел

Регулярные выражения

Навигация по уроку

Опережающая проверка

Негативная опережающая
проверка

Ретроспективная проверка

Скобочные группы

Итого

Задачи (2)

Комментарии

Поделиться



Редактировать на GitHub



```
2 <body style="height: 200px"><h1>Hello</h1>
3 ...
4 </body>
5 </html>
```

решение

Проводим курсы по JavaScript и фреймворкам.



Комментарии

перед тем как писать...

© 2007–2020 Илья Кантор | о проекте | связаться с нами | пользовательское соглашение | политика конфи

