ПМ3 Разработка модулей ПО.

РО 3.1 Понимать и применять принципы объектно-ориентированного и асинхронного программирования.

Тема 4. Регулярные выражения.

Лекция 20. Метасимволы. Символьные классы. Квантификаторы.

Цель занятия:

Познакомиться регулярными выражениями в JavaScript как инструментом для поиска и обработки текста.

Учебные вопросы:

1. Метасимволы.
2. Символьные классы.
3. Квантификаторы.

1. Метасимволы.

Метасимволы — это специальные символы, которые обозначают не конкретные буквы, а группы символов или позиции в строке.

Они нужны, чтобы гибко описывать шаблон текста: цифры, буквы, пробелы и т.д.

◇ Основные метасимволы

1. Точка .

Обозначает любой один символ, кроме перевода строки \n.

```
console.log("cat".match(/c.t/));   // ['cat'] (подходит "c"+"a"+"t")
console.log("c_t".match(/c.t/));   // ['c_t'] (подходит "c"+"_"+"t")
console.log("scotch".match(/c.t/));   // ['scotch'] (подходит "c"+"o"+"t"..)
```

2. \d

Означает цифру (от 0 до 9).

```
console.log("Room 101".match(/\d/));   // ['1'] (первая цифра)
console.log("Room 101".match(/\d/g));  // ['1','0','1'] (все цифры)
```

3. \w

\w — это "символ слова"

Под это понятие в JavaScript попадают:

- латинские буквы a–z, A–Z
- цифры 0–9
- символ подчёркивания _

То есть \w = [A-Za-z0-9_]

⚠ Важно: кириллица (А, Б, Я, ё) не входит в **\w**.

Примеры:

```
console.log("JS_2025".match(/\w/g));
// ['J', 'S', '_', '2', '0', '2', '5']
// Нашли все буквы, цифры и подчёркивание

console.log("Привет2025".match(/\w/g));
// ['2', '0', '2', '5']
// Кириллические буквы не подходят!

console.log("Hi, JS!".match(/\w/g));
// ['H', 'i', 'J', 'S']

const text = "Price_2025: $500!";
const count = (text.match(/\w/g) || []).length;
console.log(count);  // 13
```

4. \s

Означает любой пробельный символ: пробел, табуляция, перевод строки.

```
console.log("a b\tc\nd".match(/\s/g));
// [' ', '\t', '\n'] (пробел, таб, перевод строки)
```

5. Отрицания (заглавные аналоги)

\D — не цифра

\W — не символ слова

\S — не пробел

Примеры:

```
console.log("123abc".match(/\D/g));  // ['a','b','c']
console.log("123 abc".match(/\S/g)); // ['1','2','3','a','b','c']

//Найти первый не-символ слова
console.log("Hello, JS!".match(/\W/));
// [","] → первая запятая, так как она не буква/цифра/подчёркивание

//Найти все не-символы слова (с флагом g)
console.log("Hello, JS!".match(/\W/g));
// [",", " ", "!"]
```

📌 Важно

Если нужен буквальный символ (например, точка "."), его нужно экранировать: **\.**

```
console.log("mail.com".match(/mail\.com/)); // ['mail.com']
```

Метасимволы в регулярных выражениях

Метасимвол	Значение	Пример	Результат
.	Любой символ (кроме \n)	"cat".match(/c.t/)	["cat"]
\d	Цифра (0–9)	"a1b".match(/\d/)	["1"]
\D	Не цифра	"a1b".match(/\D/g)	["a","b"]
\w	Символ слова (латиница, цифра, _)	"JS_2025".match(/\w/g)	["J","S","_","2","0","2","5"]
\W	Не символ слова	"Hi!".match(/\W/)	["!"]
\s	Пробельный символ (пробел, таб, \n)	"a b".match(/\s/)	[" "]
\S	Не пробел	"a b".match(/\S/g)	["a","b"]

🎯 Итог

- Метасимволы помогают находить не только фиксированные слова, но и шаблоны текста.
- Мы можем искать цифры, буквы, пробелы или любые символы.
- Для противоположных значений используются заглавные варианты (\D, \W, \S).

2. Символьные классы

Символьный класс — это конструкция в регулярках, которая позволяет указать набор допустимых символов в одной позиции.

Записывается в квадратных скобках: [ ].

◇ Основные возможности

1. Перечисление символов

/[abc]/ // означает "a" или "b" или "c"

```
console.log("cat".match(/[abc]/));  // ["c"]
console.log("dog".match(/[abc]/));  // null
```

2. Диапазоны

Можно задавать диапазоны символов:

- [a-z] — любая строчная латинская буква
- [A-Z] — любая заглавная латинская буква
- [0-9] — любая цифра

```
console.log("Room 101".match(/[0-9]/)); // ["1"]
```

3. Объединение диапазонов

/[A-Za-z]/ // любая латинская буква

/[A-Za-z0-9]/ // любая латинская буква или цифра

4. Отрицание

Если первым символом внутри скобок стоит ^, то класс обозначает всё, кроме указанных символов.

/[^0-9]/ // любой символ, кроме цифры

```
console.log("2025!".match(/[^0-9]/)); // ["!"]
```

5. Спецсимволы внутри классов

Точка **.** внутри **[]** теряет своё особое значение → это обычная точка.

То же самое с ****, если явно не экранировать.

Но сокращения (**\d**, **\w**, **\s**) можно использовать и внутри **[]**.

```
/[.\d]/   // точка ИЛИ цифра
console.log("v2.0".match(/[.\d]/g));
// ["2", ".", "0"]
```

Сравнение: классы и метасимволы

\d ⇔ [0-9]

\w ⇔ [A-Za-z0-9_]

\s ⇔ [\t\n\r]

Символьные классы в RegExp

Запись	Значение	Пример	Результат
[abc]	любой символ из списка	"cat".match(/[abc]/)	["c"]
[0-9]	любая цифра (как \d)	"2025".match(/[0-9]/)	["2"]
[A-Z]	заглавная латинская буква	"Hi".match(/[A-Z]/)	["H"]
[a-z]	строчная латинская буква	"Hi".match(/[a-z]/)	["i"]
[A-Za-z0-9_]	буква, цифра или _ (как \w)	"JS_2025".match(/[A-Za-z0-9_]/g)	["J","S","_","2","0","2","5"]
[^0-9]	любой символ, кроме цифры	"A1!".match(/[^0-9]/g)	["A","!"]
[.\d]	точка или цифра	"v2.0".match(/[.\d]/g)	["2",".","0"]
[ ] (пробел внутри)	сам пробел	"a b".match(/[ ]/)	[" "]

📌 Итог

[abc] — любой из перечисленных символов.

[a-z] — диапазон символов.

[^...] — отрицание (всё, кроме).

Символьные классы дают гибкость: можно комбинировать цифры, буквы, спецсимволы.

3. Квантификаторы

Квантификаторы показывают, сколько раз должен повторяться символ или группа символов в шаблоне.

- Основные квантификаторы

1. **+**

Один или больше раз.

```
console.log("123".match(/\d+/)); // ["123"]
console.log("abc".match(/\d+/)); // null (цифр нет)
```

2. *

Ноль или больше раз.

```
console.log("123".match(/\d*/)); // ["123"]

// [""] (совпадение пустой строки перед "a")
console.log("abc".match(/\d*/));
```

3. ?

Ноль или один раз.

```
console.log("color".match(/colou?r/)); // ["color"]
console.log("colour".match(/colou?r/)); // ["colour"]
```

4. **{n}**

Ровно n раз.

```
console.log("aaa".match(/a{3}/)); // ["aaa"]
console.log("aa".match(/a{3}/));  // null
```

5. **{n,}**

Не меньше n раз.

```
console.log("aaaaa".match(/a{3,}/)); // ["aaaaa"]
console.log("aa".match(/a{3,}/));    // null
```

6. **{n,m}**

От n до m раз.

```
console.log("aaa".match(/a{2,4}/)); // ["aaa"]
console.log("aaaaa".match(/a{2,4}/)); // ["aaaa"]
```

📌 **Итог:**

+ → 1 и больше

***** → 0 и больше

? → 0 или 1

{n} → ровно n

{n,} → n и больше

{n,m} → от n до m

Важные моменты:

- Квантификаторы относятся к предыдущему символу или группе.

```
/ab+/  // "a" + одна или больше "b"
```

- По умолчанию квантификаторы **жадные** (захватывают максимум).

Примеры:

Найти трёхзначное число:

```
console.log("Код 727".match(/\d{3}/)); // ["727"]
```

Найти номер квартиры (1–3 цифры):

```
console.log("Кв. 105".match(/\d{1,3}/)); // ["105"]
```

Найти вариант написания слова:

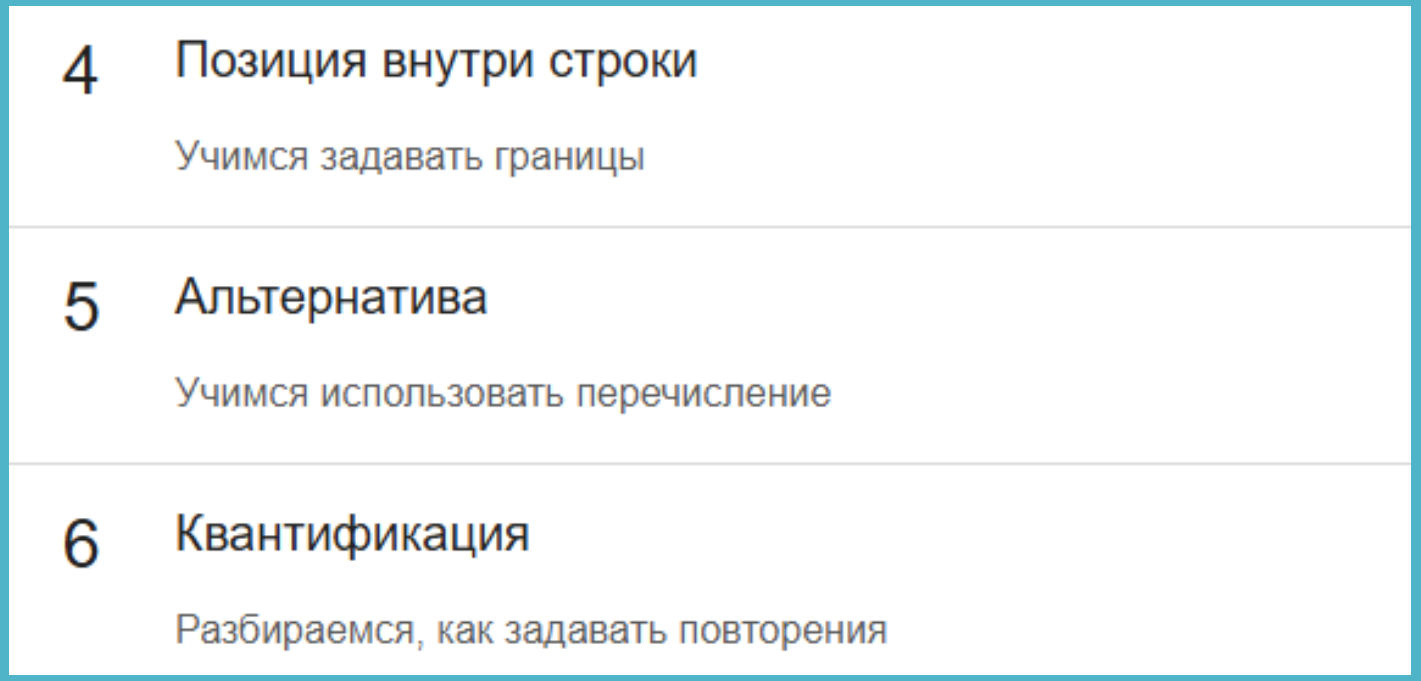
```
console.log("color / colour".match(/colou?r/g));
// ["color", "colour"]
```

Контрольные вопросы:

- Что обозначает метасимвол .?
- Чем отличаются \d и \D?
- Какие символы включает в себя \w?
- Чем отличаются \s и \S?
- Как найти в тексте именно точку . (а не любой символ)?
- Что означает запись [abc]?
- Что означает диапазон [0-9]?
- Чем отличаются [0-9] и \d?
- Как работает отрицание [^...] в символьных классах?
- Что вернёт выражение "v2.0".match(/[.\d]/g)?
- Что означает квантификатор +?
- Чем отличаются * и +?
- Что означает ? после символа или группы?
- Как работает {n}? Приведите пример.
- Что означает запись {2,4}?

Домашнее задание:

1. https://ru.hexlet.io/courses/regexp



2. Повторить материалы лекции.

Материалы лекций:

https://github.com/ShViktor72/Education2025

Обратная связь:

colledge20education23@gmail.com