

ПМЗ Разработка модулей ПО.

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

Тема 4. Регулярные выражения.

Лекция 21. Якоря Группы и альтернации .

Цель занятия:

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

Учебные вопросы:

- 1. Якоря в регулярных выражениях.**
- 2. Группы и альтернативы в регулярных выражениях.**
- 3. Замены с \$1, \$2 в регулярных выражениях.**

1. Якоря в регулярных выражениях.

Якоря — это специальные символы в регулярных выражениях, которые указывают не на сами символы текста, а на позицию внутри строки.

Они помогают задать, где именно должно находиться совпадение: в начале, в конце, на границе слова и т. д.

◆ Основные якоря

1. ^ — начало строки

Соответствует позиции перед первым символом строки (или начала строки в многострочном режиме).

```
console.log(/^Hello/.test("Hello world"));  
// true → строка начинается с "Hello"
```

```
console.log(/^world/.test("Hello world"));  
// false → строка НЕ начинается с "world"
```

2. \$ — конец строки

Соответствует позиции после последнего символа строки.

```
console.log(/world$/.test("Hello world"));
```

```
// true → строка заканчивается на "world"
```

```
console.log(/Hello$/.test("Hello world"));
```

```
// false → строка не заканчивается на "Hello"
```

3. **\b** — граница слова

Соответствует позиции между:

символом слова (\w) и не-символом слова (\W),
или началом/концом строки.

```
console.log(/\bcat\b/.test("black cat white"));
```

```
// true → есть отдельное слово "cat"
```

```
console.log(/\bcat\b/.test("concatenation"));
```

```
// false → "cat" внутри другого слова
```


4. \B — НЕ-граница слова

Противоположность \b.

Совпадает с позицией, где нет границы слова.

```
console.log(/\Bcat\B/.test("concatenation"));  
// true → "cat" окружено буквами  
  
console.log(/\Bcat\B/.test("black cat white"));  
// false → "cat" стоит отдельно
```

◆ Применение якорей

Проверка формата строки

Проверить, что строка состоит только из цифр:

```
console.log(/^\\d+$/ .test("2025")); // true  
console.log(/^\\d+$/ .test("2025abc")); // false
```

Поиск слов

Найти отдельное слово "JS":

```
console.log("I like JS!".match(/\bJS\b/));  
// ["JS"]
```

Валидация ввода (имя, email, телефон)

Проверить, что имя содержит только буквы:

```
console.log(/^([A-Za-zА-Яа-я])+$/i.test("Ivan")); // true  
console.log(/^([A-Za-zА-Яа-я])+$/i.test("Ivan123")); // false
```

Итоги:

^ — начало строки

\$ — конец строки

\b — граница слова

\B — не граница слова

2. Группы и альтернации в регулярных выражениях.

◆ Группы ()

1. Обычные группы

- Круглые скобки позволяют:
- объединять части шаблона;
- применять квантификаторы к целой группе;
- сохранять совпадения для последующего использования.

```
console.log("abscabc".match(/(abc)+/));  
// ["abscabc"] → повторяющаяся группа
```

2. Сохраняющие группы

Каждая пара скобок запоминает найденную часть строки.

Нумерация идёт слева направо: \$1, \$2, ...

```
const text = "2025-09-17";  
const re = /(\d{4})-(\d{2})-(\d{2})/;  
const result = text.match(re);  
console.log(result);  
// ["2025-09-17", "2025", "09", "17"]
```

Здесь:

- \$1 → "2025"
- \$2 → "09"
- \$3 → "17"

3. Несохранивающие группы (?: ...)

Если нужно сгруппировать выражение, но не сохранять результат:

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

```
console.log("color colour".match(/colo(?:r|ur)/g));  
// ["color", "colour"] → но без запоминания
```

◆ Альтернатива |

Символ | означает "или".

Можно искать одно из нескольких выражений.

```
console.log("I like cats and dogs".match(/cat|dog/g));  
// ["cat", "dog"]
```

Комбинация групп и альтернации

Можно объединять **()** и **|**:

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

```
console.log("cat, dog, bird".match(/(cat|dog)/g));  
// ["cat", "dog"]
```

◆ Применение

Форматы дат

```
const date = "17/09/2025";  
console.log(date.match(/(\d{2})[-\/](\d{2})[-\/](\d{4})/));  
// ["17/09/2025", "17", "09", "2025"]
```

Проверка расширений файлов

```
console.log("file.doc".match(/\.(jpg|png|doc)$/));  
// [".doc", "doc"]
```

Поиск имён

```
console.log("Hello Ivan and Anna".match(/(Ivan|Anna)/g));  
// ["Ivan", "Anna"]
```

 Итоги:

(...) — группа символов

(?: ...) — группа без сохранения

| — альтернатива ("или")

Группы можно использовать для квантификаторов, замен с \$1, \$2, валидации форматов

3. Замены с \$1, \$2 в регулярных выражениях.

◆ Что это такое?

Когда мы используем группы (...) в регулярных выражениях, они сохраняют части совпадений.

При замене (.replace) к этим группам можно обратиться через \$1, \$2, \$3 и т. д.

\$1 → первая группа

\$2 → вторая группа

\$3 → третья группа ...

◆ Примеры

1. Перестановка слов

```
const text = "Ivan Petrov";  
console.log(text.replace(/(\w+)\s+(\w+)/, "$2 $1"));  
// "Petrov Ivan"
```

(\w+) → имя "Ivan" (группа 1)

(\w+) → фамилия "Petrov" (группа 2)

В замене " \$2 \$1 " → меняем местами

2. Преобразование даты

```
const date = "2025-09-17";  
console.log(date.replace(/(\d{4})-(\d{2})-(\d{2})/, "$3.$2.$1"));  
// "17.09.2025"
```

\$1 → 2025

\$2 → 09

\$3 → 17

3. Email (извлечение логина и домена)

```
const email = "student@mail.com";  
console.log(email.replace(/(\w+)@(\w+\.\w+)/, "Логин: $1, Домен: $2"));  
// "Логин: student, Домен: mail.com"
```

4. Телефон (форматирование)

```
const phone = "87011234567";  
console.log(phone.replace(/(\d{1})(\d{3})(\d{3})(\d{2})(\d{2})/, "+$1 ($2) $3-$4-$5"));  
// "+8 (701) 123-45-67"
```

5. Несколько замен сразу

```
const text = "Price: 5000 KZT";  
console.log(text.replace(/(\d+)\s*KZT/, "$1 ₸"));  
// "Price: 5000 ₸"
```

◆ Использование с функцией

В `replace` можно использовать не только `$1`, `$2`, но и функцию, которая получает группы:

```
const text = "2025-09-17";  
const newDate = text.replace(/(\d{4})-(\d{2})-(\d{2})/,  
  (match, year, month, day) => `${day}/${month}/${year}`  
);  
console.log(newDate);  
// "17/09/2025"
```

Итоги:

- \$1, \$2 ... → содержимое скобочных групп
- Удобно для перестановки элементов, форматирования дат, телефонов, цен
- Работает в методе `.replace`
- Можно использовать функцию для более гибкой замены

Контрольные вопросы:

- Что такое якоря в регулярных выражениях?
- Чем отличаются символы `^` и `$`?
- Для чего используется `\b`? Приведите пример.
- В чём отличие `\b` и `\B`?
- Можно ли якоря комбинировать с квантификаторами? Пример.
- Для чего нужны круглые скобки `()` в регулярных выражениях?
- Чем отличаются сохраняющая группа `(...)` и не сохраняющая `(?: ...)`?
- Как нумеруются группы внутри регулярного выражения?
- Что делает символ `|` в регулярных выражениях?
- Как найти в тексте слова `cat` или `dog` с помощью альтернативы?
- Как можно использовать группы `(...)` при замене в методе `.replace`?
- Что означает `$1`, `$2` и т. д. в строке замены?
- Как поменять местами имя и фамилию в строке `"Ivan Petrov"` с помощью регулярки?
- Как преобразовать дату из `"2025-09-17"` в `"17.09.2025"` с использованием `$1`, `$2`, `$3`?
- Чем замена с `$1`, `$2` отличается от замены с функцией в `replace`?

Домашнее задание:

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

7	Жадность	Выясняем, как перестать жадничать
8	Группировка и обратная связь	Учимся переиспользовать группы в регулярных выражениях
9	Модификаторы	Знакомимся с управлением через флаги внутри регулярного выражения
10	Просмотр вперед и назад	

2. Повторить материалы лекции.

Материалы лекций:

<https://github.com/ShViktor72/Education2025>

Обратная связь:

colledge20education23@gmail.com