ПМ3 Разработка модулей ПО.

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

Тема 2. Регулярные выражения.

Лекция 2. Основы регулярных выражений

Цель занятия:

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

Учебные вопросы:

1. Флаги регулярных выражений.

1. Флаги регулярных выражений.

Флаги меняют поведение поиска. Указываются после /.../

◇ Основные

g — global (глобальный поиск)

По умолчанию регулярное выражение ищет только первое совпадение.

С **g** будут найдены все совпадения.

```
// ищем одну цифру → находит только первую: ["1"]
"a1 b2 c3".match(/\d/);

// то же, но с флагом g → находит все цифры: ["1", "2", "3"]
"a1 b2 c3".match(/\d/g);
```

i — ignore case (без учёта регистра)

Поиск не различает большие и маленькие буквы.

```
/hello/.test("Hello");
// false, потому что ищем строго "hello" в нижнем регистре

/hello/i.test("Hello");
// true, потому что i игнорирует регистр
```

m — multiline (многострочный режим)

Когда в тексте есть перенос строки (**\n**), строка становится многострочной.

```
const text = "one\ntwo";
// это не одна строка "one\ntwo"
// а две строки:
//    1) "one"
//    2) "two"
```

Некоторые проверки работают только для всей строки целиком.

Из-за этого "two" может не находиться, если смотреть на текст как на одну длинную строку.

Флаг m включает режим, в котором каждая строка внутри текста проверяется отдельно.

Теперь "two" можно найти и во второй строке.

```
const text = "one\ntwo";

console.log(text.match(/^two$/));    // null
console.log(text.match(/^two$/m));   // ["two"]
```

Дополнительные (знать полезно, но не обязательно для старта):

- s — dotAll, позволяет . захватывать \n
- u — unicode, корректная работа с юникодом (например, эмодзи)
- y — sticky, ищет только с позиции lastIndex

Контрольные вопросы:

- Что делают флаги g, i, m?
- Чем отличается test от match?

Домашнее задание:

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