

Регулярні вирази у JavaScript

*Морозов Андрій Васильович,
к.т.н, доц.,
декан факультету інформаційно-комп'ютерних
технологій ЖДТУ*

Створення регулярного виразу у JavaScript:

1) `var regex = new RegExp("рег. вираз", "прапорці");`

2) `var regex = /рег. вираз/прапорці;`

Тут прапорці:

- **i** – ігнорувати регістр символів у рядку;
- **g** – шукати усі співпадіння (без прапорця **g** – буде знайдено тільки перше співпадіння);
- **m** – багаторядковий режим

Приклад.

```
var r1 = new RegExp("\\d{2}-\\d{2}-\\d{2}", "gi");
```

```
var r2 = /\d{2}-\d{2}-\d{2}/gi;
```

Операції з використанням регулярних виразів:

1. Пошук у рядку:

```
var str = "рядок";
```

```
var reg = /рег.вир/прапорці;
```

```
var res = str.search(reg);
```

Результат: позиція першого співпадіння або -1.

Метод `search` шукає завжди перше співпадіння незалежно від наявності прапорця `g`.

Приклад.

```
var str = "Я вивчаю регулярні вирази у " +  
          "мові JavaScript";  
  
var res1 = str.search( /Ви/i );    // 2  
var res2 = str.search( /Ви/im );   // 2  
var res3 = str.search( /Ви/img );  // 2
```

2. Пошук входжень регулярного виразу у рядок:

```
var str = "рядок";  
var reg = /рег.вир/прапорці;  
var res = str.match(reg);
```

Результат:

1) якщо не використано прапорець **g**, то знаходиться перше входження регулярного виразу,

res – це масив,

res[0] – знайдений фрагмент тексту;

res[i] – знайдений текст, що відповідає *i*-м круглим дужкам регулярного виразу;

res.index – позиція входження рег.виразу;

res.input – весь рядок, в якому виконувався

пошук

Приклад.

```
var str = "15,1,5,11,2,12,15";  
var res = str.match(/(\d{2}),(\d{1})/);  
console.log(res.length); // 3  
console.log(res[0]); // 15,1  
console.log(res[1]); // 15  
console.log(res[2]); // 1  
console.log(res.index); // 0  
console.log(res.input);  
// 15,1,5,11,2,12,15
```

```
var str = "рядок";  
var reg = /рег.вир/прапорці;  
var res = str.match(reg);
```

Результат:

2) якщо використано прапорець **g**, то знаходяться усі входження регулярного виразу,
res – це масив,

res – масив, що містить усі співпадіння;

res.length – кількість співпадінь;

- текст, що відповідає круглим дужкам у результат не потрапляє, у результат заносяться підрядки, які відповідають усьому регулярному виразу
- якщо співпадіння не знайдено, повертається

Приклад.

```
var str = "15,1,5,11,2,12,15";  
var res = str.match(/(\d{2}),(\d{1})/g);  
console.log(res.length); // 3  
console.log(res[0]); // 15,1  
console.log(res[1]); // 11,2  
console.log(res[2]); // 12,1  
console.log(res.index); // undefined  
console.log(res.input); // undefined
```


3. Пошук у рядку усіх співпадінь і дужкових груп

```
var str = "рядок";  
var reg = /рег.вир/прапорці ;  
// якщо прапорця g не вказано, то  
// це теж саме, що і str.match(reg) ;  
// якщо прапорець g вказано, то використовують  
цикл:  
while (res = reg.exec(str)) {  
    // res[0] – все співпадіння;  
    // res[i] – i-ті круглі дужки;  
    // result.index – позиція входження  
}
```

4. Перевірка входження у рядок регулярного виразу

```
var reg = /рег.вир/прапорці ;  
var res = reg.test("рядок") ;
```

Результат: `res` – `true/false`;

5. Багаторядковий режим

У регулярних виразах можна використовувати символи:

^ - початок рядка

\$ - кінець рядка

Рядок, у якому є `\n` вважається одним

```
var str = '1е місце: Іваненко\n' +  
          '2е місце: Петренко\n' +  
          '5е місце: Сидоренко';  
console.log(str.match(/^d+/g)); // 1
```

Для того, щоб оброблювати кожний підрядок між `\n` розглядався як окремий рядок, потрібно використати прапорець `m`:

```
var str = '1е місце: Іваненко\n' +  
          '2е місце: Петренко\n' +  
          '5е місце: Сидоренко';  
console.log(str.match(/^d+/gm)); // 1,2,5
```

6. Заміна

```
var reg = /рег.вир/прапорці ;  
var str = "рядок тексту" ;  
var res = str.replace(reg, "на що замінюємо") ;
```

У рядку заміни, можна використовувати спеціальні

СИМВОЛИ:

\$\$ СИМВОЛ "\$"

\$& усе знайдене співпадіння

\$` частина рядка до співпадіння

\$' частина рядка після співпадіння

\$n **n** - число, що позначає вміст **n**-их круглих дужок

Приклад.

```
var str = "Мене звати Петренко Сергій.";
var res = str.replace(
    /([А-ЯІЄЇ][а-яііє]+) ([А-ЯІЄЇ][а-яііє]+)/,
    '$2 $1');
console.log(res);
// "Мене звати Сергій Петренко."
```

\$&	Петренко Сергій
-----	-----------------

\$`	Мене звати
-----	------------

\$'	.
-----	---

\$1	Петренко
-----	----------

\$2	Сергій
-----	--------