

Resumo Teórico e Prático de JavaScript: Regex

Conjuntos

Conjuntos em regex são definidos por colchetes [] e especificam um grupo de caracteres possíveis para uma correspondência.

Por exemplo, [abc] casa com 'a', 'b' ou 'c'.

Prático:

```
const regex = /[abc]/g;  
console.log("batata".match(regex)); // Saída: ['b', 'a', 'a', 'a']
```

Caracteres Especiais

Caracteres especiais como \d (dígitos), \w (caracteres alfanuméricos) e \s (espaços) facilitam correspondências específicas.

- \D: não dígitos, \W: não alfanuméricos, \S: não espaços.

Prático:

```
const regex = /\d+/g;  
console.log("123 abc 456".match(regex)); // Saída: ['123', '456']
```

Operador Not (^)

O operador not (^) usado dentro de conjuntos exclui os caracteres especificados.

Por exemplo, [^a-z] corresponde a qualquer coisa que não seja uma letra minúscula.

Prático:

```
const regex = /^[^aeiou]/g;  
console.log("batata".match(regex)); // Saída: ['b', 't', 't']
```

Operador Plus (+)

O operador + corresponde a uma ou mais ocorrências do padrão anterior.

Prático:

```
const regex = /a+/g;
```

```
console.log("baaaatata".match(regex)); // Saída: ['aaa', 'a', 'a']
```

Operador Question (?)

O operador ? torna o padrão anterior opcional, permitindo zero ou uma ocorrência.

Prático:

```
const regex = /colou?r/g;
```

```
console.log("color colour".match(regex)); // Saída: ['color', 'colour']
```

Operador de Precisão ({min,max})

Define o número exato ou intervalo de repetições do padrão.

Prático:

```
const regex = /a{2,3}/g;
```

```
console.log("a aa aaa aaaa".match(regex)); // Saída: ['aa', 'aaa']
```

Método exec

O método exec retorna detalhes da correspondência, incluindo o índice de início.

Prático:

```
const regex = /\d+/g;
```

```
const result = regex.exec("123 abc 456");
```

```
console.log(result); // Saída: ['123']
```

Método match

O método match retorna todas as correspondências de uma string em um array.

Prático:

```
const regex = /\w+/g;  
  
console.log("batata doce".match(regex)); // Saída: ['batata', 'doce']
```

Choice Pattern (|)

O operador | permite corresponder alternativas.

Prático:

```
const regex = /batata|doce/g;  
  
console.log("batata doce".match(regex)); // Saída: ['batata', 'doce']
```

Validar Domínios

Regex para validar domínios de site:

```
const regex = /^[a-zA-Z0-9-]+\.[a-zA-Z]{2,}$/;  
  
console.log(regex.test("example.com")); // Saída: true
```

Validar Emails

Regex para validar emails:

```
const regex = /^[^@\s]+@[^@\s]+\.[^@\s]+$/;  
  
console.log(regex.test("email@example.com")); // Saída: true
```

Validar Datas

Regex para validar datas no formato dd/mm/yyyy:

```
const regex = /^\d{2}\d{2}\d{4}$/;  
  
console.log(regex.test("10/12/2024")); // Saída: true
```