

## Trabalhando com Grupos

### Transcrição

Vamos voltar novamente no nosso exemplo CSV e relembrar a regex que extrai a data:

```
[0123]?\\d\\s+de\\s+[A-Z][a-zç]{1,8}\\s+de\\s+[12]\\d{3}
```

Ao executar, a regex devolve a data completa, como planejamos. Agora imagine que queremos sim fazer um *match* da data, mas selecionar apenas o ano. Para tal existem os grupos. Um grupo é definido através de parênteses `()`. Ou seja, basta colocar a parte da regex que define o ano em parênteses:

```
[0123]?\\d\\s+de\\s+[A-Z][a-zç]{1,8}\\s+de\\s+([12]\\d{3})
```

Para ver os grupos dentro do nosso formulário, devemos habilitar o checkbox "Mostra grupos":

### Expressões regulares

Target string (alvo)

João Fulano,123.456.789-00,21 de Maio de 1993,(21) 3079-9987,Rua do Ouvidor,50,20040-030,Rio de Janeiro

Pattern (expressão regular)

[0123]?\\d\\s+de\\s+[A-Z][a-zç]{1,8}\\s+de\\s+([12]\\d{3})

Executar Regex

☐ Mostra índice ☒ Mostra grupos

1 Matches (resultados)

21 de Maio de 1993 | 1993

Highlight

João Fulano,123.456.789-00,21 de Maio de 1993,(21) 3079-9987,Rua do Ouvidor,50,20040-030,Rio de Janeiro

Dentro do código JavaScript, o método `exec` devolve um array, com o *match* inteiro e os grupos da regex. Esses resultados é que estamos mostrando no formulário. Sabendo disso, podemos definir mais grupos para também selecionar o dia e o mês:

```
([0123]?\\d)\\s+de\\s+([A-Z][a-zç]{1,8})\\s+de\\s+([12]\\d{3})
```

A regex com o alvo `21 de Maio de 1993` devolve, além do *match* inteiro, os grupos `21`, `Maio` e `1993`.

### Grupos opcionais

Através de um grupo, podemos também definir um conjunto de caracteres como opcional. Já conhecemos o *quantifier* que significa opcional, o `?` (zero ou uma vez). Agora só falta combinar o `?` com um grupo. Por exemplo, podemos deixar a preposição `de` como opcional: `(de\\s+)?`, e aplicando isso na regex nas duas preposições:

```
([0123]?\\d)\\s+(de\\s+)?([A-Z][a-zç]{1,8})\\s+(de\\s+)?([12]\\d{3})
```

Essa regex pega datas como: 21 Maio 1993 , 21 Maio de 1993 e 21 de Maio de 1993

## Non-capturing groups

Agora temos um novo problema: usamos os grupos justamente para selecionar o dia, mês e ano. Com os dois novos grupos, também recebemos as preposições como resposta. Por exemplo, usando o alvo 21 de Maio de 1993 , o nosso formulário devolve os grupos:

```
21 ||| de ||| Maio ||| de ||| 1993
```

Os caracteres ||| são apenas o separador dos grupos, utilizado no nosso código JavaScript e não importam na avaliação. Agora, a pergunta é: como podemos usar um grupo que não é devolvido pela regex? Para tal existem os **non-capturing groups**!

Novamente usaremos o símbolo ? , mas agora no início do grupo junto com os dois pontos, por exemplo para a preposição de : (? :de\s+)?

Aplicando isso na regex inteira, no alvo 21 de Maio de 1993 :

```
([0123]? \d) \s+ (? :de\s+)? ([A-Z][a-z]{1,8}) \s+ (? :de\s+)? ([12] \d{3})
```

Isso devolve os grupos:

```
21 de Maio de 1993 ||| 21 ||| Maio ||| 1993
```

Perfeito, a preposição de , que faz parte da regex, não aparece como grupo!

