

08

[Ruby] Regex com Ruby

Transcrição

No Ruby também existe uma sintaxe literal para definir a expressão regular, por exemplo:

```
> regex = /(\d\d)(\w)/
```

A partir daí, podemos usar o método `match` para aplicar a regex no alvo:

```
> alvo = "12a34b56c"
> resultado = regex.match("12a34b56c")
```

O método `match` devolve um objeto que representa o primeiro *match* e através dele podemos pegar o *match* inteiro, grupos e os índices:

```
> resultado[0]
12a
> resultado[1]
12
> resultado.begin 0 #inicio do match inteiro
0
> resultado.begin 1 #inicio do grupo
0
> resultado.end 0 #fim do match inteiro
3
> resultado.end 1 #fim do grupo
2
```

Para pegar todos os grupos e fazer um *match* global, devemos utilizar o método `scan`, da classe `String`:

```
> regex = /(\d\d)(\w)/ #lembrando dois grupos
> alvo = "12a34b56c"
> resultados = alvo.scan regex
=> [["12", "a"], ["34", "b"], ["56", "c"]]
```

Repare que cada posição possui agora dois elementos, que podemos testar:

```
> resultados[2][1]
=> "c"
```

Nos exercícios veremos como usar os dois métodos fundamentais: `sub` e `gsub`, para substituir uma parte da string através de uma regex.

