Para saber mais : Tudo que não é!

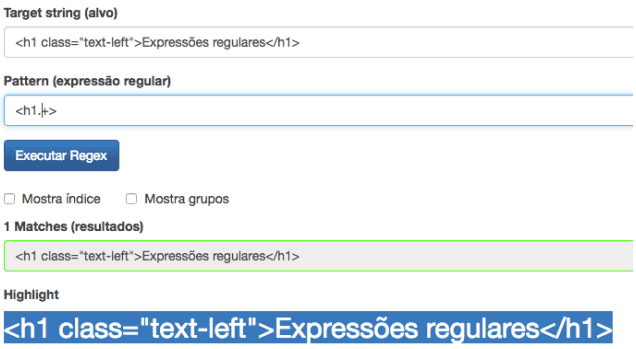
PRÓXIMA ATIVIDADE

Na aula vimos como selecionar uma tag específica dentro de um alvo e chegamos juntos a regex em baixo:

```
<h1.+?>([\w\sõãí.]+)</h1>
```

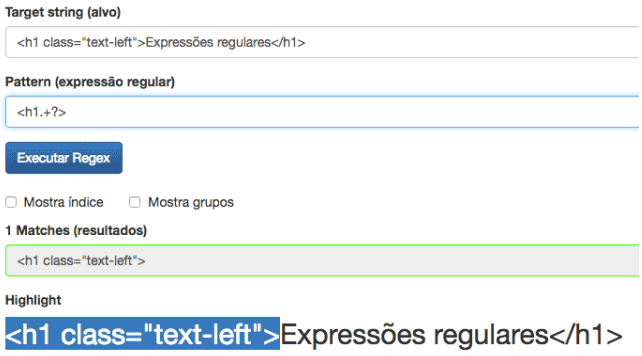
Com ela pegamos a tag `<h1>`, e o grupo devolve o conteúdo da tag. Repare também que deixamos o *quantifier preguiçoso* para não pegar a tag inteira.

Por exemplo a regex `<h1.+>` seleciona o tag inteira:



Mas se usarmos o quantifier preguiçosamente selecionamos apenas a abertura da tag:

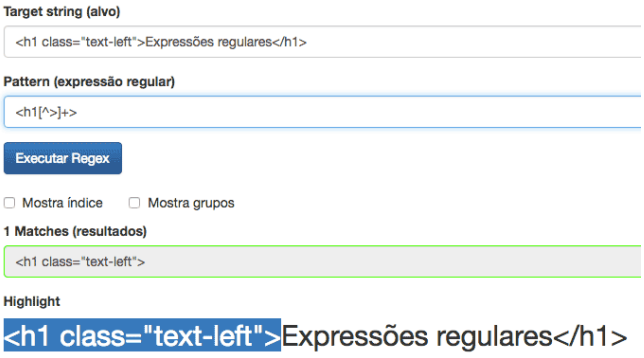
```
<h1.+?>
```



Há uma alternativa para resolver esse problema de abertura da tag. Podemos definir uma classe de caracteres que seleciona *tudo que não é um* `>`. Essa negação é feita através da meta-char `^`.

Teste o exemplo abaixo sem usar um quantifier preguiçoso:

```
<h1[^>]+>
```



Repare que usamos a classe em conjunto com o quantifier *ganancioso*: `[^>]+`

Mesmo assim foi encontrado apenas a abertura da tag pois a nossa classe de caractere **não inclui** a caractere `>`

Opinião do Instrutor

Essa negação é algo muito comum nas regexes. Há circunstâncias em que é mais fácil definir o que **não queremos** em vez de dizer o que queremos. A negação `^` ajuda nisso. Isso também é a razão da existência de classes como `\W` (com W maiúsculo) e `\D` (com D maiúsculo).

O `\W` é a *non-word char*, ou seja tudo que não é um *word char*. `\W` é um atalho para `[^\w]` .

A classe `\D` , por sua vez, é um *non-digit*, ou seja, `\D` é um atalho para `[^\d]`

Repare também que **não usamos a meta-char `^` como âncora** pois aparece dentro de uma classe `[^>]` .