**RegEx em JavaScript**

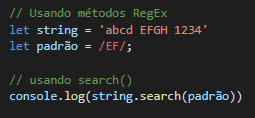
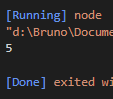
**Criando uma expressão em JS**

**Sintaxe básica:**  /expressão/

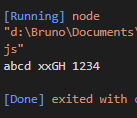
**Nota:** não precisa de aspas

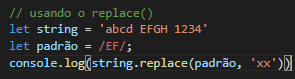
**Métodos RegExs**

**string.search(padrão ou substring):** retorna a posição de uma substring ou padrão em uma string.

**Ex.:**

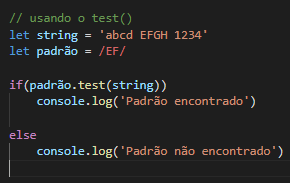
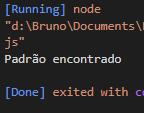
**string.replace(padrão/substring, substituto):** substitui um padrão/substring na string pela string substituta. Retorna a nova string.

**Ex.:**



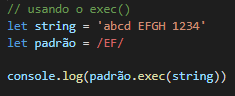
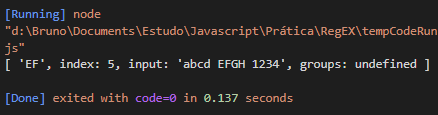
**padrão.test(string):** verifica se um padrão está presente em uma string.

**Ex.:**



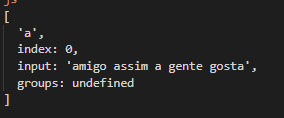
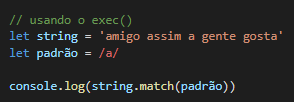
**padrão.exec(string):** retorna os padrões encontrados e suas posições, além da string completa, como um objeto. Caso não encontre, retorna Null

**Ex.:**



**string.match(padrão):** retorna um array com as substrings que seguem o padrão. Se o modificador ‘g’ não for usado, apenas uma sequência será retornada.

**Ex.:**



**Modificadores**

* **i** => realiza uma pesquisa case-insensitive;
* **g** => faz uma pesquisa global;
* **m** => faz uma pesquisa multilinha(cada linha é considerada como uma string nova)

**Quantificadores**

* **a+ =>** pesquisa por 1 ou vários ‘a ‘ consecutivos;
* **a\* =>** pesquisa zero ou vários ‘a’ consecutivos;
* **a?** (opcional) **=>** pesquisa por zero ou 1 ‘a’;
* **a?\*(lazy) =>** pega a menor sequência possível de ‘a’;
* **a{x} =>** pesquisa por ‘a’ x vezes consecutivas;
* **a{x, y} =>** pesquisa por ‘a’ x vezes consecutivas no mínimo e y vezes consecutivas no máximo;
* **a{x,} =>** pesquisa por ‘a’ x vezes consecutivas no mínimo;

**Greedy :** retorna a maior sequência possível

**Lazy:** retorna a menor sequência possível

**Âncoras:**

* **^ =>** começo da string;
* **$ =>** final da string;
* **\b =>** limites da palavra. Dependendo da posição, pode ser o começo \bword ou o final word\b;
* **\B =>** tudo que não for os limites da palavra, ou seja, as letras.

**Classes de Caracteres**

* **.(dot ou peridod) =>** substitui qualquer caractere;

**Agrupamento**

* **()**