* Expressões regulares
  + O que são

“Expressões Regulares são padrões usados para pegar combinações de caracteres em cadeias de caracteres (Strings), em JS expressões regulares também são objetos”. (LINHA DE CÓDIGO, 2022).

“Expressões Regulares são padrões de caracteres que associam sequências de caracteres no texto. Podemos usar expressões regulares para extrair ou substituir porções de texto, bem como, endereço ou link de imagens em uma página HTML, modificar formato de texto ou remover caracteres inválidos”. (DEVMEDIA, 2022).

* + Como construir

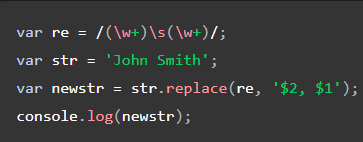
“Um padrão de expressão é composto por um conjunto de caracteres simples, como /abc/, ou uma combinação de caracteres simples e especiais, como /ab\*c/ ou /Capitulo (\d+)\.\d\*/”. (MDN, 2022).

“As RegEx são baseadas em dois elementos: um padrão e modificadores. Esses elementos, em conjunto, formam a chamada expressão regular, cuja sintaxe básica é bastante simples: /padrão/modificadores”. (DEVMEDIA, 2022).

* + Como aplicar no JavaScript

“O seguinte script usa o método replace() da instância de String para casar o nome no format nome sobrenome e produzir no formato sobrenome, nome. No texto substituto, o script usa $1 e $2 para indicar os respectivos parênteses de casamento no padrão da expressão regular”. (MDN, 2022).

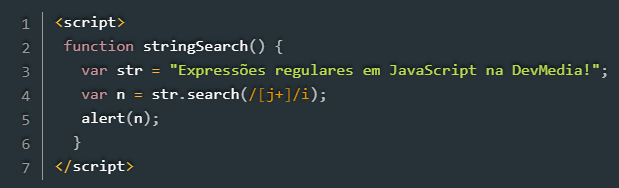
**Figura 1 - Usando uma expressão regular para mudar o formato dos dados**



Fonte: MDN (2022).

“As buscas com o auxílio de expressões regulares no JavaScript podem ser realizadas de duas formas: a partir do método search(), do tipo string, ou do método exec(), do tipo RegExp. O valor de retorno de cada um dos métodos, no entanto, é diferente: enquanto search() retorna o índice da ocorrência encontrada, exec() retorna o primeiro caractere encontrado que satisfaça a expressão regular. Vamos começar, então, pelo search(), que por ser um método do tipo string, será executado em cima da própria variável, ou seja, do texto no qual desejamos buscar determinada expressão. O código da Listagem 1 mostra em detalhes o uso desse método. Ali, note que estamos declarando uma string com um texto de teste, e realizando a operação “search()” sobre ele. No caso, estamos fazendo a busca por ao menos uma ocorrência da letra “j”, como mostra a expressão regular especificada, com o quantificador “+”. Assim, o valor retornado na variável n é 24, indicando o índice da letra “J” na string”. (DEVMEDIA, 2022).

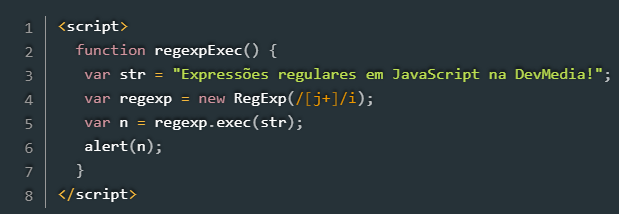
**Figura 2 - Exemplo de uso do método search()**

****

Fonte: DEVMEDIA (2022).

“Algo similar poderia ser obtido com o método exec() de RegExp, como mostra o código da Listagem 2. Nesse caso, estamos criando uma expressão regular (um objeto RegExp) e comparando-a e com a string criada. Repare que essa operação é justamente o contrário do que fizemos com o método search(), quando comparamos uma string com a expressão regular. O valor retornado em n, dessa vez, será a string encontrada através da expressão regular (no caso, a letra J), e não a posição em que ela se encontra no texto”. (DEVMEDIA, 2022).

**Figura 3 - Exemplo de uso do método exec() de RegExp**



Fonte: DEVMEDIA (2022).

Referências

DEVMEDIA. **Expressões regulares em JavaScript**. Disponível em: https://www.devmedia.com.br/expressoes-regulares-em-javascript/37015. Acesso em: 24 out. 2022.

DEVMEDIA. **Iniciando Expressões Regulares**. Disponível em: https://www.devmedia.com.br/iniciando-expressoes-regulares/6557. Acesso em: 06 out. 2022.

LINHA DE CÓDIGO. **JavaScript - Expressões Regulares**. Disponível em: http://www.linhadecodigo.com.br/artigo/328/javascript-expressoes-regulares.aspx. Acesso em: 06 out. 2022.

MDN. **Expressões Regulares**. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Regular\_Expressions. Acesso em: 24 out. 2022.

MDN. **RegExp**. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\_Objects/RegExp. Acesso em: 24 out. 2022.