

//REGEX:
//O ReGex é uma expressão regular (regular expression ou Regex) é uma sequência de caracteres que forma um padrão de pesquisa. Ao procurar dados em um texto, você pode usar esse padrão de pesquisa para descrever o que está procurando. Uma expressão regular pode ser um único caractere ou um padrão mais complicado. Expressões regulares podem ser usadas para executar todos os tipos de pesquisa de texto e operações de substituição de texto.

//Onde o JavaScript usa o Regex?
//Em JavaScript, expressões regulares são frequentemente usadas com os dois métodos string: search() e replace(). O método search() usa uma expressão para procurar uma correspondência e retorna a posição da correspondência enquanto o método replace() retorna uma string modificada onde o padrão é substituído.
//Mas não fica apenas nisso, pois as expressões possuem métodos próprios que também podem ser utilizadas para diversos tipos de verificações.

// Sintaxe do Regex
//A sintaxe padrão do Regex funciona da seguinte maneira:
// /padrão/modificadores;

// Exemplo:
// let regex = /mundo/i;

//Explicação:

// /mundo/i é um REGEX ou uma expressão regular.

//“mundo” é a palavra de busca (para ser usado em uma pesquisa).
//“i” é o modificador (modifica a pesquisa para não diferenciar maiúsculas de minúsculas).

//Regex com Funções JavaScript

// Utilizando REGEX com Search()

// É possível utilizar o Regex em conjunto com o método Search para efetuar buscas mais inteligentes. No caso do exemplo abaixo, você consegue saber onde foi a primeira ocorrência da palavra “mundo” sem precisar se importar se estão sendo usadas letras maiúsculas ou minúsculas.

```
let str1 = "Olá Mundo!";
let n = str1.search(/mundo/i);
console.log(n); //retornará 4, que é a posição da palavra na string
```

// Utilizando REGEX com Replace

// Em conjunto com o replace, é possível fazer algo que muitas vezes causa problemas para aqueles que precisam lidar um strings enormes. No caso abaixo, o regex ajuda a fazer um replace em todas as vezes que a palavra algo for encontrada.

```
let str2 = "Olá mundo! Programando JavaScript pelo mundo"
let txt = str2.replace(/mundo/g,"universo");
console.log(txt); //altera o texto onde a palavra “mundo” aparece, independentemente de ser maiúscula ou minúscula.
```

// Modificadores de Expressões Regulares

// Os modificadores podem ser usados para realizar pesquisas mais globais, sem distinção entre maiúsculas e minúsculas:

// Tipos de modificadores

// letra i: Executa a busca ignorando letras maiúsculas e minúsculas.

// letra g: Executa uma pesquisa global (encontre todas os resultados em vez de parar após a primeira ocorrência).

// letra m: Executar o ajuste de considerando as quebras de linha.

// Padrões de Expressão Regular

// Os colchetes são usados para encontrar um intervalo de caracteres:

// Descrição da Expressão

// [abc] Encontre qualquer um dos caracteres entre os colchetes

// [0-9] Encontre qualquer um dos dígitos entre os parênteses

// (x | y) Encontre uma das alternativas separadas por |

```
// Metacaracteres são caracteres com um significado especial:

// Metacaractere Descrição
// \ d Encontrar um dígito
// \ s Encontrar um caractere de espaço em branco
// \ b Encontrar uma correspondência no começo ou no final de uma palavra
// \ uxxxx Encontre o caractere Unicode especificado pelo número hexadecimal xxxx

// Descrição do Quantificador
// n + Corresponde a qualquer string que contenha pelo menos um n
// n * Corresponde a qualquer cadeia que contenha zero ou mais ocorrências de n

// Exemplos:
// script usa o método exec para encontrar uma instancia dentro de uma string.
let myRe = /d(b+)d/g;
let myArray = myRe.exec('cdbbdsbz');
console.log(myArray);

// Retorna todas as ocorrencias das letras "a" a "h", independente de serem maiúsculas ou minúsculas
let str3 = "Cara, cadê o meu carro?";
let patt = /[a-h]/gi;
console.log(str3.replace(patt, 'i'));

// Procura por ocorrências da letra "o" na string
let str4 = "Hellooo World! Hello Mundoooooooo!";
let patt1 = /o+/g;
let result = str4.match(patt1);
console.log(result);
```

RESULTADO NO CONSOLE...

```
[Running] node "c:\Users\Almoxarifado\Documents\javascript\arquivos_das_aulas\128-ReGeX.js"
```

```
Olá universo! Programando JavaScript pelo universo  
[ 'dbbd', 'bb', index: 1, input: 'cdbbdsbz', groups: undefined ]  
iiri, iiiê o miu iirro?  
[ 'ooo', 'o', 'o', 'ooooo' ]  
  
[Done] exited with code=0 in 0.112 seconds
```