//ABAIXO SEGUE UMA ESPÉCIE DE DICIONÁRIO DO REGEX:

//OBS: Ele começa com sinais de pontuação, números e depois letras

/\*

    .   =   O ponto "." é um meta-char universal que pode significar qualquer caractere.

    \   =   O escape "\" é usado para quando desejamos usar um meta-char especial ou um sinal comum sem ativar um ;

            meta-char;

    ?   =   A interrrogação "?" é um símbolo para apresentar um valor que pode ocorrer 0 ou 1 vez, ou seja, ele dá a

                opção, se tiver o caractere 1 vez retorne ele, mas se não tiver, não tem problema, pode retornar sem

                esse caractere também. Geralmente usamos ele após uma acentuação, onde a acentuação é opcional;

    {}  =   As chaves "{}" são usadas para quando desejamos usar uma determinada quantidade de meta-chars, inclusive as

            chaves são chamadas de "quantifier" no ReGex;

    []  =   Os colchetes "[]" delimitam uma classe de caracteres, classes de caracteres são um conjunto de caracteres

                que desejamos usar encontrar um determinado conjunto de caracteres. Podemos delimitá-los pela "," ou "-", usamos "," quando queremos um ou outro caractere, e usamos "-" quando desejamos ter valores de 1 determinado caractere até outro determinado caractere.

    \d  =   O espape d "\d" é um meta-char para encontrar dígitos;

\*/

//TARGET USADO EM TODOS OS EXEMPLO:

let targetDigitos = "João Almeida Augusto de Melo - Professor - Idade 36 - CPF 349.832.441-23 - RG 34.435.534-2 CNPJ 15.123.321/8883-22"

//USANDO PONTO:

let ExPonto = RegExp(/..\d/) //Perceba que o ponto vai pegar quaisquer 2 primeiros caracteres que estiverem antes do 1º

                                //dígito que for encontrado...

console.log("Exemplo de Ponto: " + ExPonto.exec(targetDigitos)) //Usamos aqui a função "exec" do RegExp essa função

                                                                    //retorna um array, onde o índice 0 é o resultado

                                                                    //que procuramos, o índice 1 traz o valor índice do

                                                                    //resultado que procuramos, o valor 2 traz o target

                                                                    //completo. Mas quando usamos o exec concatenado com

                                                                    //uma string ele traz automáticamente o índice 0 da

                                                                    //função "exec". Além da função exec temos muitas

                                                                    //outras, para mais informações consulte a

                                                                    //especificação oficial no site

                                                                    //https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Regular\_Expressions

//USANDO DÍGITOS: (Aqui a missão é encontrar o CPF)

let ExDigitos = RegExp(/\d\d\d\.\d\d\d\.\d\d\d-\d\d/) //Perceba que podemos colocar um "\d" um atrás do outro...

let ExDigitos2 = RegExp(/\d{3}\.\d{3}\.\d{3}-\d{2}/) //Mas uma solução mais elegante seria usar o quantifier para

                                                        //delimitar a quantidade de dígitos. Perceba também que estamos

                                                        //usando o "\." para referenciar as pontuações do CPF

console.log("Exemplos de Dígitos: " + ExDigitos.exec(targetDigitos)) //Perceba que o resultado será o mesmo...

console.log("Exemplos de Dígitos: " + ExDigitos2.exec(targetDigitos))

//USANDO ESCAPE PARA PEGAR CARACTERES ESPECIAIS:

let ExBarra = RegExp(/\d{2}\.\d{3}.\d{3}\/\d{4}-\d{2}/) //Quando queremos a barra, temos que usar o escape junto com a

                                                            //barra, dessa forma "\/" para que consigamos referenciar a

                                                            //barra...

console.log("Exemplos de Barra: " + ExBarra.exec(targetDigitos)) //Aqui queremos como resultado o CNPJ da pessoa...

//USANDO CLASSES DE CARACTERE:

//EXEMPLO COM VÍRGULA:

let ExClasseVirgula = RegExp(/[0,1,2,3][0,1,2,3][0,1,2,3]/)//Perceba que dentro das classes estamos usando números

                                                                //separados por vírgula de 0 á 3, cada classe vai representar apenas 1 dígito, ou seja, estamos pesquisando pelo conjunto de 3 dígitos onde cada dígito deverá estar entre 0 e 3....

console.log("Exemplo de Classe com separação por vírgula para para os valores 0 ou 3: " + ExClasseVirgula.exec(targetDigitos)) //veja que pegamos o segundo conjunto de dígitos do CNPJ...

//EXEMPLO COM TRAÇO:

let ExClasseTraco = RegExp(/[0-9][0-3][0-2]/)//Perceba que dentro das classes estamos usando números separados por

                                                //traço tentando pegar um conjunto de 3 dígitos onde o 1º dígito poderá conter valores que vão de 0 á 9, o 2º poderá conter valores de 0 á 3 e o 3º poderá ir de 0 á 2...

console.log("Exemplo de Classe com separação por vírgula para para os valores 0 ou 3: " + ExClasseTraco.exec(targetDigitos)) //veja que pegamos o segundo conjunto de dígitos do CPF...

//USANDO ? PARA PEGAR 0 OU ALGUM VALOR:

let ExSemInterrogacao = RegExp(/[a-z][a-z]/) //Nosso objetivo aqui é pegar o 1º conjunto de 2 letras minúsculas sem

                                                //acentuação.

let ExInterrogacao = RegExp(/[a-z][a-z]?/) //Mas aqui nós já damos a opção de que pode ser 1 letra minúscula sem

                                                //acentuação, mas se tiver 2 pode trazer também.

console.log("Exemplo com Interrogação: " + ExSemInterrogacao.exec(targetDigitos)) //Veja que quando queremos 2 letras ele

                                                                                        //força o encontro do primeiro conjunto e letras minúsculas sem acentuação, indo até o "lm" do Almeida..

console.log("Exemplo com Interrogação: " + ExInterrogacao.exec(targetDigitos)) //Mas quando tornamos o encontro da 2

                                                                                //letra opcional, ele pega para nós o 1º caractere "o" do nome João...

RESULTADO NO CONSOLE...

[Running] node "c:\Users\Almoxarifado\Documents\Gah\javascript\arquivos\_das\_aulas\199-REGEX\_\_Dicionario\_do\_ReGeX.js"

Exemplo de Ponto: e 3

Exemplos de Dígitos: 349.832.441-23

Exemplos de Dígitos: 349.832.441-23

Exemplos de Barra: 15.123.321/8883-22

Exemplo de Classe com separação por vírgula para para os valores 0 ou 3: 123

Exemplo de Classe com separação por vírgula para para os valores 0 ou 3: 832

Exemplo com Interrogação: lm

Exemplo com Interrogação: o

[Done] exited with code=0 in 0.116 seconds