



Artigo

Invista em você! Saiba como a DevMedia pode ajudar sua carreira.



# Iniciando Expressões Regulares

Apresenta as sintaxes básicas das Expressões Regulares e também a classe Regex.



Marcar como concluído



Anotar

Artigos



.NET



Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

Aceitar



Não podemos esquecer que a sintaxe de Expressões Regulares são case-sensitive, ou seja, "abc" é diferente de "ABC".

Neste artigo apresentaremos os métodos mais utilizados da classe Regex, trabalharemos as sintaxes de referência de expressões regulares mais utilizadas, seguida de práticas e validações úteis do dia-a-dia.

## Regex

No [.NET](#) a classe [Regex](#) representa uma expressão regular imutável. Esta classe contém diversos métodos estáticos que nos **permite trabalhar com expressões regulares**. Para usarmos o objeto Regex devemos importar o namespace `System.Text.RegularExpressions`.

Segue relação de métodos mais utilizados da classe **Regex**, a saber:

Métodos Regex		
Framework	Descrição	Modelo
Escape	evita um conjunto de metacaracteres (\, *, +, ?,  , {, [, (, ^, \$, ., #, e espaço em branco)	Regex.Escape( input, pattern)

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

Aceitar



Match	<b>expressão regular</b> e retorna resultado preciso como um objeto Match único	pattern) regex.match(input, pattern, options)
Matches	pesquisa uma entrada de string para todas as ocorrências de uma expressão regular e retorna todos os resultados com sucesso, executa Match várias vezes	Regex.Matches( input, pattern) Regex.Matches( input, pattern, options)
Replace	dentro de uma entrada de string especificada, substitui string que associa com expressão regular por outra especificada para reposição	Regex.Replace( input, pattern, replacement) Regex.Replace( input, pattern, replacement, options)
Split	divide uma string em um array de substrings nas posições <b>definidas pela expressão regular</b>	Regex.Split( input, pattern) Regex.Split( input, pattern, options)
Unescape	substitui código de escape pelo caractere de escape	Regex.Unescape( input, pattern)

Opções de Execução		
Framework	Caractere Inline	Descrição
None	N/A	nenhuma opção especificada

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

Aceitar



		ou numerados do formato(1...)
SingleLine	s	modo linha simples; modifica o significado do caractere (.) assim ele associa todos caracteres (em lugar de cada caractere exceto \n)
IgnorePatternWhitespace	x	permite espaço em branco e comentários no pattern
Compiled	N/A	expressões regulares seriam compilados para um assembly
RightToLeft	N/A	procura mover da direita para esquerda em lugar da esquerda para direita
Global (não contém na framework)	g	pesquisa global para todas as ocorrências do padrão

Regex na Prática	
Framework	Exemplo
IsMatch	Dim new Boolean1 As Boolean = Regex.IsMatch("Editora Wrox", "wrox") Dim new Boolean2 As Boolean = Regex.IsMatch("Editora Wrox", "wrox", _ RegexOptions.IgnoreCase) Dim myRegex As New Regex("wrox", RegexOptions.IgnoreCase) Dim new Boolean3 As Boolean = myRegex.IsMatch("Editora Wrox") Obs.: new Boolean1 retorna 'False' por causa que não contém

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

Aceitar



	<code>match in newMatches response.write(myMatch.value) next</code>
Replace	<code>Dim newString1 As String = Regex.Replace("Editora Wrox", "wrox", "o'relly") Dim newString2 As String = Regex.Replace("Editora Wrox", "wrox", "o'relly", _ RegexOptions.IgnoreCase) Dim myRegex As New Regex("wrox", RegexOptions.IgnoreCase) Dim newString3 As String = myRegex.Replace("Editora Wrox", "o'relly")</code>
Split	<code>Dim newString1() As String = Regex.Split("Editora Wrox", "wrox", RegexOptions.IgnoreCase) For Each str As String In newString1 Response.Write(str &amp; " ") Next</code>

## Referências

Segue uma breve descrição dos caracteres especiais mais utilizados em expressão regular.

Escape	
Caractere	Descrição
\	caractere de escape (usado para anular um efeito de um metacaractere; TOME CUIDADO AO UTILIZAR ESTE CARACTERE POIS "\d" REPRESENTA UM METACARACTERE DÍGITO E PARA ANULAR DEVE-SE ACRESCENTAR MAIS UM CARACTERE DE

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

Aceitar



\e	escape \u001B
\040	caractere ASCII como octal (3 dígitos). Por exemplo, um caractere \040 representa um espaço
\x20	caractere ASCII usando representação hexadecimal (3 dígitos)
\cC	caractere ASCII de controle. Por exemplo, \cC é equivalente a Ctrl-C
\u0020	caractere ASCII usando representação hexadecimal (4 dígitos)

Agrupamento	
Caractere	Descrição
[]	usado para indicar o conjunto de caracteres que podem conter no texto pesquisado. (veja quantificadores)
[^]	usado para indicar o conjunto de caracteres que NÃO podem conter na pesquisa.
(...)	indica o início e o fim de um grupo; ex.: /(abc)+(def)/ encontra um ou mais ocorrências de "abc" seguido por uma ocorrência de "def". obs.: para usar o parêntese, literalmente, utilize o caractere de escape "\" (" ou "\)", ou ainda, "[ ( )" ou "[ ] ]"

Âncora				
Caractere	Descrição	Modelo	Representação	Explicação
				associa texto que

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

[Aceitar](#)



\b	fronteira do caractere	"car\b"	"car"; "tocar"	associa a fronteira da palavra "car", embora não "carburador".
\B	não fronteira do caractere	"car\B"	"carburetor"	não associa a fronteira da palavra "car", mas pode assegurar que a sequência ocorra no meio da palavra, como "carburador", embora não "tocar".
\A	início da string	"\Apattern"	"patterns"	--
\Z	fim da string	"pattern\Z"	"topattern"	--

Quantificadores				
Caractere	Descrição	Modelo	Representação	Explicação
*	zero ou mais vezes	"zo*"	"z" ; "zoo"	o caractere que precede o asterisco pode ser repetido 0 ou mais vezes (equivalente a "{0,}")

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

Aceitar



:	vez	do(es):	do , does	interrogação pode ocorrer 0 ou 1 vez (equivalente a "{0,1}")
{n}	n vezes	"o{2}"	"food"	o caractere que precede o sinal pode ocorrerá n vezes
{n,}	pelo menos n vezes	"o{2,}"	"foooooo"	a sequência "{0,}" é equivalente a "o*"; a sequência "{1,}" é equivalente a "o+"
{n,m}	no mínimo n e no máximo m	"o{1,3}"	"foooooo"	associa aos três primeiros "ooo"; é equivalente a "o?"
.	um caractere simples, exceto "\n"	"o."	"oa"	representa um carecteres simples, exceto "\n" ( para associá-lo utilize o padrão "[\s\S] " )
[xyz]	um dos caracteres que está dentro dos colchetes	"[abc]"	"plain"	associa "a" contido na palavra "plain"
	um dos			associa um dos

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

Aceitar





Caractere	Descrição	Modelo	Representação	Explicação
" " (pipe)	uma ou outra ocorrência	"(z f)ood"	"zood" ou "food"	o caractere   (pipe) representa uma ocorrência ou outra

Classes	
Classe	Descrição
[digit:]	dígito; equivalente a [0-9]
[alnum]	letras e números ; equivalente a [A-Za-z0-9]
[space:]	caracteres brancos ; equivalente a [ \t\n\r\f\v]
[alpha:]	letras; equivalente a [A-Za-z]
[lower:]	minúsculas; equivalente a [a-z]
[upper:]	maiúsculas; equivalente a [A-Z]
[punct:]	caracteres de pontuação; equivalente a [.,!?:...]
[xdigit:]	números hexadecimais; equivalente a [0-9A-Fa-f]
[word:]	qualquer caractere alfanumérico, mais underscore (_); equivalente a [[:alnum:]_]
[blank:]	espaço em branco e TAB; equivalente a [\t]
[punct:]	pontuação; equivalente a [!"#\$%&'()*+,-./:;?@[\\]_`{ }~]

### Caracteres de Classes Predefinidos

Classe	Caractere
--------	-----------

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

[Aceitar](#)



Parâmetros BackReference	
Caracteres	Descrição
\number	backreference que associa o conteúdo do grupo de mesmo número. Grupos são numerados iniciando de 1. Por exemplo, (.+)\n1 associa 'abc abc' ou '777 777', embora não 'abc def'. O número máximo de backreferences é limitado para 9 (\1..\9).
\k	backreference nomeada. ( <a href="http://msdn2.microsoft.com/en-us/library/ks2azbh.aspx">http://msdn2.microsoft.com/en-us/library/ks2azbh.aspx</a> )

Comentário	
Caractere	Descrição
#	insere comentário dentro da expressão

## Praticando

É interessante criarmos uma página .aspx (ou qualquer outra hosting application que desejar) para testarmos as expressões regulares. O código abaixo facilita bastante a execução das sintaxes, inclusive dos métodos da classe Regex (veja Métodos Regex).

Utilizamos cookies para fornecer uma melhor experiência para nossos  
usuários. Para saber mais sobre o uso de cookies, consulte nossa  
política de privacidade. Ao continuar navegando em nosso site, você  
concorda com a nossa política.

Aceitar



End Sub

Perceba que o método `IsMatch` retorna um valor Booleano e é necessário informar, neste caso, apenas o texto e a sintaxe da expressão regular (veja Métodos Regex). Para qualquer teste que queira fazer basta colocar o pattern - sintaxe - da expressão regular e testar. Por exemplo, `^\d+$` valida somente dígitos, isso significa que se for digitado alguma coisa que não for número então retornará False. Abaixo segue uma lista de algumas das validações simples que utilizamos no dia a dia.

Validações Úteis com Expressões Regulares	
Tipo de Validação	Expressão
Dígito	<code>^\d+\$</code>
Decimal	<code>^[+-]?((\d+ \d{1,3}(\.\d{3})+)(\,\d*)?) \,\d+\$ ^[+-]?([0-9]*\,[0-9]+  [0-9]+)\$</code>
Letra	<code>^[:,alpha:]]+\$</code>
URL	<code>^((http) (https) (ftp)):\V/([ \w]+\.)+\w{2,3}(\V [% \w]+ (\.\w{2,})?)*\$</code>
E-mail	<code>^([ \w\-\-]+\.)*[ \w\-\- ]+@([ \w\-\- ]+\.)+([ \w\-\-]{2,3})\$</code>
Endereço IP	<code>\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b</code>

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

Aceitar



## Referências

- Microsoft Visual Basic .NET Programmer's Cookbook / by Matthew MacDonald;
- Sams Teach Yourself Regular Expressions in 10 Minutes / By Ben Forta;
- Mastering Regular Expressions, 3rd Edition / By Jeffrey E. F. Friedl;
- Beginning Regular Expressions / Andrew Watt;
- Microsoft .NET Framework 2.0 Application Development Foundation / by Tony Northrup, Shawn Wildermuth and Bill Ryan.

### Saiu na DevMedia!

- React com Redux:

Redux é uma biblioteca JavaScript criada pelo Facebook para resolver um problema inerente de aplicações front-end conforme elas crescem em tamanho e complexidade.

### Saiba mais sobre .NET ;)

- Linguagem C#:

Neste guia de consulta você encontrará diversos conteúdos que podem ser

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários. Para saber mais sobre o uso de cookies, consulte nossa [política de privacidade](#). Ao continuar navegando em nosso site, você concorda com a nossa política.

Aceitar