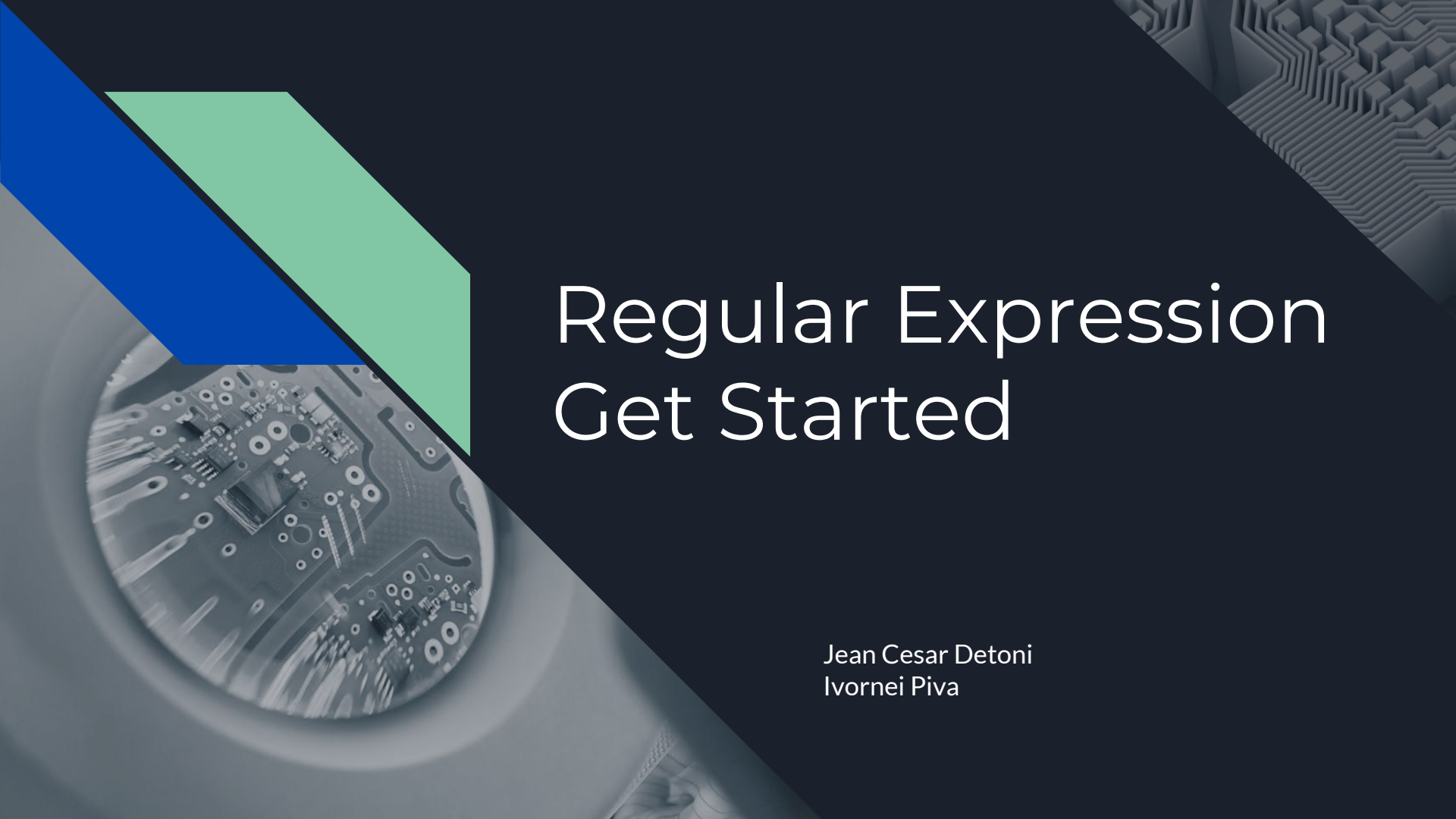


Regular Expression Get Started

Jean Cesar Detoni
Ivornei Piva

Cronograma

Introdução

Visão geral

História

Para que servem?

Metacaracteres

Praticando

Conclusões

Referências e links úteis

Introdução

Pontos a destacar:

- Não se assuste;
- Interrompa e questione se precisar;
- Compartilhe o que você sabe sobre ER;
- Este treinamento não deve ser o suficiente para você;
- Faça uso do que aprender.

Visão geral

De forma detalhada, expressão regular é um conjunto de símbolos, caracteres com funções especiais, que, quando agrupados entre si e com caracteres literais, formam uma sequência, uma expressão. Essa expressão é interpretada como uma regra que indicará sucesso se uma entrada de dados qualquer que casar com essa regra, ou seja, obedecer exatamente a todas as suas condições.

De forma resumida, expressão regular é um método formal de se especificar um padrão de texto.

História

A ideia das expressões regulares surgiram em 1943 quando dois neurologistas decidiram publicar um estudo sobre o funcionamento dos neurônios, fundamentando o modelo através de sistemas de autômatos finitos e linguagens formais. Já em meados de 1950, um matemático chamado Stephen Kleene, descreveu este modelo proposto em formato de notação matemática.

O contato com o computador surgiu em 1968, em um algoritmo de busca utilizado no editor "qed", que depois virou "ed", editor padrão dos primeiros sistemas Unix.

O divisor de águas da utilização das expressões regulares na programação aconteceu em 1986, quando surgiu um pacote pioneiro em C chamado regex que tratava as expressões regulares qualquer desenvolvedor poderia utilizá-lo em seus programas.

Para que servem?

Servem para definir algo abrangente de uma forma específica. Elas definem padrões de texto que pode gerar uma lista de possibilidades de casamento de acordo com a cadeia de dados submetidos a ela.

Na prática, as expressões regulares são bastantes úteis quando se precisa buscar ou validar um padrão de texto que pode ser variável.

Dentre os exemplos de padrões que podem ser reconhecidos utilizando expressões regulares estão: data, horário, número IP, endereço de Internet, e-mail, tags HTML, RG, CPF, número de cartão de crédito, e mais uma infinidade de padrões que podem ser encontrados em cadeias de caracteres em um texto.

- Exemplo de uso.

Metacaracteres

- São os símbolos que formam as expressões regulares. Cada um deles possuem funções específicas que podem mudar dentro do contexto no qual estão inseridos.

. ? * + ^ $ | [ ] { } ( ) \

Metacaracteres

Metacaractere	Nome
.	Ponto
[ ]	Lista
[^]	Lista negada
?	Opcional
*	Asterisco
+	Mais
{ }	Chaves

Metacaractere	Nome
^	Circunflexo
$	Cifrão
\	Escape
( )	Grupo
\1 … \9	Retrovisor
|	Ou
\b	Borda

Classificação dos metacaracteres

- Representantes

  . [ ] [^]

- Quantificadores

  ? * + {n,m}

- Âncoras

  ^ $ \b

- Outros

  | (...) \1 ... \9

Ponto: .

O Ponto é o metacaractere que casa com todo mundo. Ele pode ser considerado um curinga em expressões regulares. O ponto deve ser usado com consciência.

Expressão	Casa com
n.o	não, nao, ...
.eclado	Teclado, teclado, ...
e.tendido	entendido, extendido, estendido,...
13.00	13.00, 13:00, 13 00, 13-00, ...

Lista: []

Bem mais exigente que o ponto, a lista não casa com qualquer caractere. Ela deixa explícito o que pode casar com ela e não casa com nada além daquilo.

Expressão	Casa com
n[a,ã]o	não e nao
[Tt]eclado	Teclado e teclado
e[s,x]tendido	estendido e extendido
13[:,-]00	13:00 e 13-00

Lista: []

Uma característica interessante das listas é a possibilidade utilizar intervalos em sua estrutura, ou seja, [0123456789] pode ser representado por [0-9]. O mesmo conceito pode ser aplicado para letras maiúsculas [A-Z], letras minúsculas [a-z] ou para ambas [A-Za-z]. Também funciona para símbolos [:-@]

Os intervalos respeitam a ordem numérica da tabela ASCII.

Expressão	Casa com
[012][0-9]:[0-5][0-9]	12:00, 13:30, 22:00, 29:00...

Você deve ter achado estranho casar com 29:00, porém, utilizando somente lista este é o máximo de precisão que se consegue. Isso pode ser corrigido utilizando listas em conjunto com outros metacaracteres que serão vistos mais à frente.

Lista: [] Pontos importantes

- Dentro da lista todo mundo é normal;
- Dentro da lista traço indica intervalo;
- Para representar um] literal, ele deve ser o primeiro elemento da lista;
- Para representar um - literal, ele deve ser o último elemento da lista;

Lista Negada: [^]

A lista negada possui a lógica inversa da lista normal. Ela casa com tudo, menos o que estiver listado dentro dela. Todas as regras de criação da lista normal se aplicam para a lista negada.

Obs. quando deseja-se um literal ^ dentro de uma lista negada, ele não deve ser o primeiro da lista.

Expressão	Casa com
e[^n]tendido	estendido, extendido, ...
n[^a]o	não, neo, ...
[^135790]	2, 4, 6, 8, A, B, C...

Opcional: ?

O opcional é um quantificador que casa a entidade anterior quando ela acontece 0 ou 1 vez.

Expressão	Casa com
ondas?	onda e ondas
</?div>	e
th?all?es	tales, talles, thales e thalles
</?[pib]>	, , , , e

Asterisco: *

O asterisco é um quantificador que casa a entidade anterior quando ela acontece, não acontece ou acontece infinitas vezes.

Expressão	Casa com
7*0	0, 70, 770, 77770, ...
car*o	cao, caro, carro, carrro, ...
bi*p	bp, bip, biip, biiip, ...
b[ip]*	b, bi, bp, biip, bpp, ...

O Curinga .*

A união dos dois metacaracteres abrangentes ponto e asterisco, formam um curinga muito útil nas expressões regulares. Ele representa qualquer caractere em qualquer quantidade.

Expressão	Casa com
.*	, Lorem Ipsum , ...
Bo[am].*!	Bom!, Bom dia! , Boa tarde!, Bom domingo!, ...
#.*#	##, #123#, #xyz#, #foo#, ...

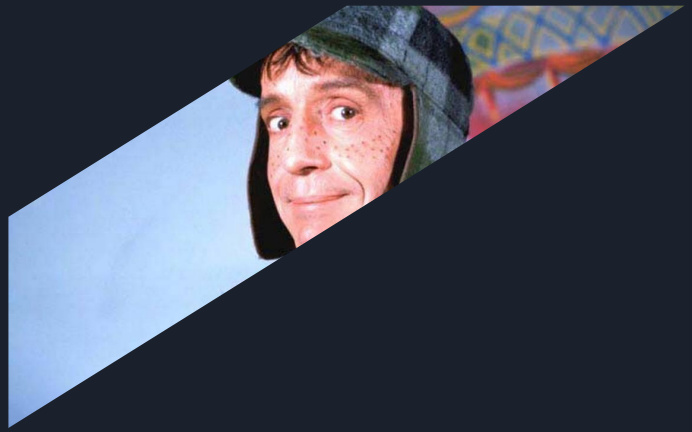
Mais: +

O mais é um quantificador idêntico ao asterisco, onde tudo o que vale para um se aplica para o outro, onde a única diferença é que a entidade anterior deve casar ao menos uma vez.

Expressão	Casa com
7+0	70, 770, 77770, ...
car+o	caro, carro, carrro, ...
bi+p	bip, biip, biiip, ...
b[ip]+	bi, bp, biip, bpp, ...

Chaves: {}

As chaves são uma solução para a quantificação mais controlada, onde se pode especificar exatamente quantas repetições se quer da entidade anterior.



Metacaractere	Repetições
{1,3}	De 1 a 3
{3,}	Pelo menos 3 (3 ou mais)
{0,3}	Até 3
{3}	Exatamente 3
{1}	Exatamente 1
{0,1}	Zero ou 1 (igual ao opcional ?)
{0,1}	Zero ou mais (igual ao asterisco *)
{1,}	Um ou mais (igual ao mais)

Circunflexo: ^

O Circunflexo é um metacaractere que indica o início de uma linha. Ele também está presente nas lista negadas, porém, fora da lista seu efeito é diferente.

Expressão	Casa com
^[0-9].*	1, 123, 1abc, 0a2b3c, 100pressão, ...
^[^0-9].*	a123, @#345, d&t0n1, ...
^[a-z_]+@email.com	foo@email.com, foo_bar@email.com, ...
^<.*	<?php echo "x", , ...

Cifrão: $

O cifrão é similar e complementar ao circunflexo. Ele marca o final de uma linha e só é válido no final de uma ER.

Expressão	Casa com
[0-9]$	teste1, .38, abcd123, ...
^[0-9]+$	1, 234, 98878, 032, ...
^[A-Z][A-Za-z]+[.]$	Teste., Validando texto., ...
^$	Linha vazia

Obs: isso é cifrão, não dolar. Lugar de dólar é no bolso.

Escape: \

O metacaractere de escape serve para atender a necessidade de tornar outro metacaractere ou ele mesmo um caractere literal.

Escapar um caractere também funciona colocando-o em uma lista, como visto anteriormente.

Expressão	Casa com
\(teste\)	(teste)
*+	*, ***, ****, ...
\++	+, ++, +++, ...

Ou: |

O metacaractere pipe, permite que seja possível criar alternativas para uma ER.

Expressão	Casa com
bom-dia|boa-tarde|boa-noite	bom-dia, boa-tarde e boa-noite
http://|https://|ftp://	http://, https:// e ftp://

Grupo: ()

Os grupos têm a função de juntar caracteres em um único local. Como em expressões matemáticas, os parênteses definem um grupo e seu conteúdo pode ser visto como um bloco na expressão. Eles são importantes para aumentar a abrangência dos metacaracteres apresentados anteriormente.

Os grupos podem ter sub grupos aninhados.

Expressão	Casa com
boa-(tarde|noite)	boa-tarde e boa-noite
pederneiras (ai)+	pederneiras ai, pederneiras aiai, pederneiras aiaiai, ...
((su|hi)per)?mercado	mercado, supermercado e hipermercado

Retrovisor: \1 ... \9

O retrovisor é um recurso usado nas expressões regulares para "olhar pra trás" e pegar um grupo que foi casado dentro de uma expressão.

Expressão	Casa com
(corre)-\1	corre-corre
(mata)-\1	mata-mata
(blá)(-\1)+	blá, blá-blá, blá-blá-blá
(lenta)(mente) é \2 \1	lentamente é mente lenta

Praticando

Análise de script de um case de uso real na IXC, por Ivornei Piva

Conclusão

As expressões regulares podem ser utilizadas em inúmeras situações, linguagens e ferramentas. O estudo dessa abordagem é uma ótima carta na manga para quem trabalha com tecnologia, principalmente para desenvolvedores e administradores de sistema.

Aproveite os recursos que elas podem oferecer :)

Referências e links úteis

- JARGAS, Aurélio Marinho. Expressões regulares: uma abordagem divertida. 4. ed. rev. e ampl. São Paulo: Novatec, 2012. 207 p. ISBN 9788575223376 (broch.).

- Testador de expressões regulares https://regex101.com/
- Blog do Aurélio : http://aurelio.net/
- Tabela ASCII : https://www.invertexto.com/tabela-ascii
- Expressões regulares Apêndice: http://piazinho.com.br/download/expressoes-regulares-3-tabelas.pdf
- Meu github : https://github.com/jcdetoni/get-started-er/