

Expressões Regulares



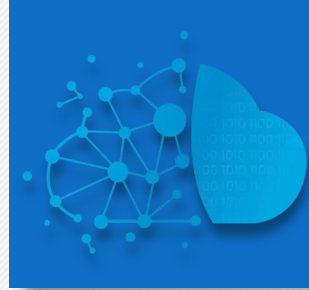
```
/^((?>[a-zA-Z\d!#$%&'"+\-/=?^_`{|}~]+\x20*|"(?=[\x01-\x7f])[^"\\]|\\[\x01-\x7f])*"\x20*)*(?<angle><))?(?!\.)(?>\.?(?>[a-zA-Z\d!#$%&'"+\-/=?^_`{|}~]+\x20*|"(?=[\x01-\x7f])[^"\\]|\\[\x01-\x7f])*")@((?!-)[a-zA-Z\d\-\-]+(?<!--)\.)+[a-zA-Z]{2,}|\[(((?<!--\[)\.)(25[0-5]|2[0-4]\d|[01]?\d?\d)){4}|[a-zA-Z\d\-\-]*[a-zA-Z\d]:((?=[\x01-\x7f])[^\\[\]]|\\[\x01-\x7f])+\])?(?<angle>)>)$/
```

Expressões regulares

Representam um modo declarativo de expressar os strings que queremos aceitar

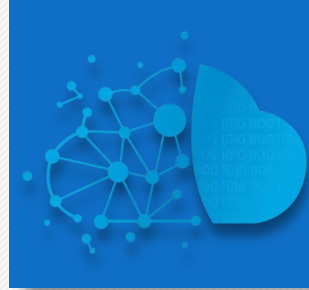


Expressões regulares



- Formalismo gerador, permite gerar palavras de uma linguagem regular
- Pode ser convertida em AFND/AFD
- Muito usado para construção de analisadores léxicos de compiladores, mecanismos de busca (crawlers e spiders)
- Muitas linguagens de programação e frameworks possuem funções para identificação de ER

Terminologia



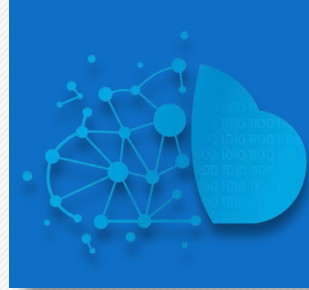
RegEx (*Regular Expression*) - termo muito utilizado na internet.

Match - casar, conferir, combinar, igualar, encontrar, encaixar, equiparar (Ex: função *preg_match()* do PHP).

Pattern - o padrão que se deseja encontrar com a expressão regular.

Metacaracteres - símbolos utilizados na construção das expressões regulares.

Tipos de Metacaracteres



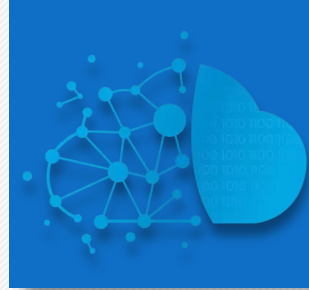
Representante - tem a função de representar um ou mais caracteres.

Qual(is) caractere(s) pode(m) aparecer?

Quantificador - serve para indicar o número de repetições permitidas para a entidade imediatamente anterior.

O(s) caractere(s) é(são) opcional(is)? Pode(m) aparecer quantas vezes?

Tipos de Metacaracteres

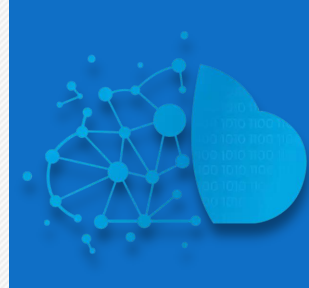


Âncora - serve para marcar uma posição específica na linha.

A entidade deve aparecer no início? No final?

Outros metacaracteres - têm funções específicas e não relacionadas entre si.

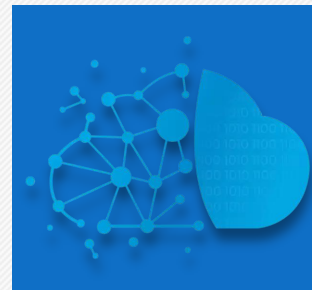
Metacaracteres tipo Representante



- . = qualquer caractere (ASCII) (necessitado)
- [] = lista de caracteres (exigente)
- [^] = lista negada (experiente)

Exemplo	Reconhece
...	Qualquer palavra com 3 caracteres
[a-z]	Conjunto das letras minúsculas
[^"]	Qualquer coisa que não seja aspas

Metacaracteres tipo Representante



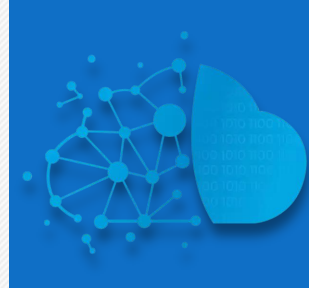
Atenção ao utilizar a lista!

[4-G]

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Metacaracteres tipo Quantificador



- ? = nenhuma ou uma vez (opcional)
- * = nenhuma ou várias vezes (tanto faz)
- + = uma ou várias vezes (tem que ter)
- {1,3} = no mínimo 1 e no máximo 3 vezes (controle)
- {4} = exatamente 4 vezes (controle)
- {4,} = no mínimo 4 vezes (controle)

Exemplo

.+

[aeiou]*

[a-z]?

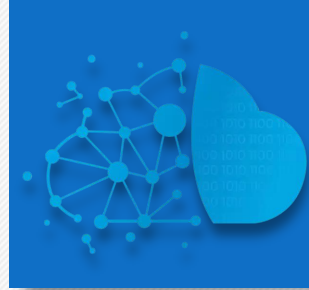
Reconhece

Qualquer palavra

Símbolos da lista, nenhuma ou várias vezes

Uma letra opcional

Metacaracteres tipo Âncora

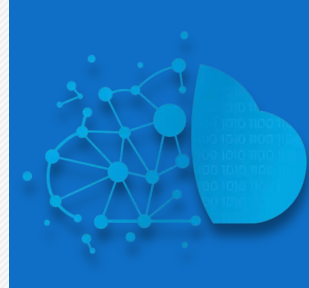


^ = início da linha
\$ = fim da linha
\b = borda

Exemplo	Reconhece
---------	-----------

^Este	Linhas iniciando com Este
;\$	Linhas terminando com ;
\bdia	Palavras iniciadas com “dia”

Outros Metacaracteres



**** = escape
| = ou
() = delimitador de grupos

Exemplo

\.

a|b

(a|b)+

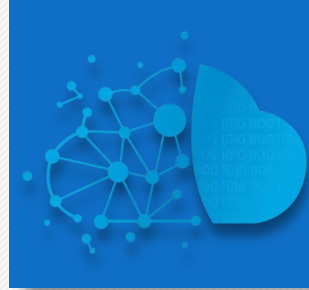
Reconhece

Somente “.”

a ou b

Sequências de as e bs

Exemplos



`[0-9]+`

= números inteiros

`[0-9]+ \. [0-9]+`

= números reais

`[a-zA-Z][^0-9]`

= letra não seguida de número

`(0[1-9] | [12][0-9] | 3[01])`

= um dia entre 01 e 31

`d..?cente`

= docente, discente, decente,...

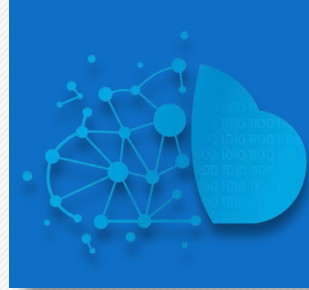
`[Ii][Ff]`

= if, iF, If, IF

`[0-9]{4}`

= anos DC com 4 dígitos.

Exercícios



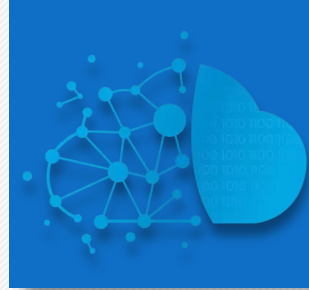
Crie as expressões regulares que reconheçam:

- Binário pares
- Palavras com 00 no final
- Strings entre aspas
- E-mail .br ou .com.br
- Telefones em SC
- Placas de veículos no Brasil
- Comentários de linha //
- Comentários de múltiplas linhas /* ... */

Referência: guia-er.sourceforge.net

Testes: regex101.com

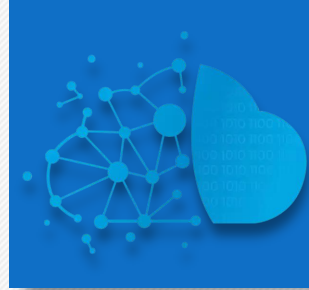
Homework



Crie as expressões regulares que reconheçam:

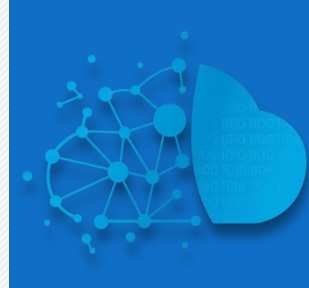
- tags contendo links no HTML (href...)
- tags contendo imagens no HTML (img...)
- trechos javascript dentro de um HTML

Particularidades do GALS



- Ausência do quantificador {min, max}
- Definições Regulares
- Casos especiais
- Pontuadores
- Ordem de avaliação
- Tratamento dos espaços

Referências



- AHO, Alfred V.; VIEIRA, Daniel. Compiladores: princípios, técnicas e ferramentas. 2. ed. São Paulo, SP: Pearson/ Addison Wesley, c2007, c2008. x, 634 p. ISBN 9788588639249.
- DELAMARO, Márcio Eduardo. Como construir um compilador utilizando ferramentas Java. São Paulo: Novatec, c2004. 308 p. ISBN 8575220551