

# ***Programação Shell Script***

*REDES DE COMPUTADORES*

Prof. Dr. João Paulo Aramuni

# RegexOne

- \* As expressões regulares são extremamente úteis na extração de informações de texto, como o código, arquivos, planilhas ou até mesmo documentos de registo.
- \* As seguintes lições e exemplos irão explorar os usos mais práticos de expressões regulares para que você possa usá-los de forma simples e útil.

# RegexOne

- \* A primeira coisa a reconhecer quando usar expressões regulares é que tudo é essencialmente um caractere, e estamos escrevendo padrões para coincidir com uma sequência específica de caracteres (também conhecido como uma string).
- \* A maioria dos padrões usar ASCII normal, que inclui letras, números, pontuação e outros símbolos do seu teclado como # \$% @ !, mas caracteres Unicode também podem ser usados para combinar com qualquer tipo de texto internacional.

# RegexOne

- \* Vamos praticar...
- \* Tentar escrever um padrão que corresponda a todos as três linhas apresentadas em cada exercício:
- \* <http://regexone.com/>

# Exercício

## Lesson 3: Matching specific characters

The dot metacharacter from the last lesson is pretty powerful, but sometimes **too** powerful. If we are matching phone numbers for example, we don't want to validate the letters "(abc) def-ghij" as being a valid number!

There is a method for **matching specific characters** using regular expressions, by defining them inside **square brackets**. For example, the pattern `[abc]` will only match a **single** a, b, or c letter and nothing else.

Below are a couple lines, where we only want to match the first three strings, but not the last three strings. Notice how we can't avoid matching the last three strings if we use the dot, but have to specifically define what letters to match using the notation above.

### Exercise 3: Matching Characters

Task	Text
Match	can
Match	man
Match	fan
Skip	dan
Skip	ran
Skip	pan



`[c,m,f]an`

Continue >

Solve the above task to continue on to the next problem, or read the [Solution](#).

### Lesson Notes

abc...	Letters
123...	Digits
\d	Any Digit
\D	Any Non-digit character
.	Any Character
\.	Period
[abc]	Only a, b, or c
[^abc]	Not a, b, nor c
[a-z]	Characters a to z
[0-9]	Numbers 0 to 9
\w	Any Alphanumeric character
\W	Any Non-alphanumeric character
{m}	m Repetitions
{m,n}	m to n Repetitions
*	Zero or more repetitions
+	One or more repetitions
?	Optional character
\s	Any Whitespace
\S	Any Non-whitespace character
^...\$	Starts and ends
(...)	Capture Group
(a(bc))	Capture Sub-group
(.*)	Capture all
(abc def)	Matches abc or def

# Desafio

- \* Utilize o comando `grep` para encontrar a palavra “test” dentro de um arquivo de `.txt`.
- \* O comando também deverá trazer o número da linha onde está cada incidência da palavra “test”.
- \* Escreva no arquivo `.txt` o texto:

# Desafio

- \* César é o tester da equipe de teste do sistema Sol. Seu trabalho é executar testes e encontrar bug's em tudo que é produzido pelo João. Estes testes costumam ser maliciosos e chatos. Fazer o que. Este é o trabalho de um tester. Tester tem que testar!

# Solução

- \* # -n para mostrar o número da linha
- \* `grep -n "test" arquivoTeste.txt`
- \* Ou
- \* # -n para mostrar o número da linha
- \* `grep -n "test.." arquivoTexte.txt`



# Exercícios

- \* exer12\_userinterface.sh
- \* exer13\_userinterface.sh
- \* exer14\_testeFuncao.sh
- \* exer15\_palindromo.sh
- \* exer16\_fatorial.sh
- \* exer17\_parImpar.sh
- \* exer18\_parImparTOP.sh
- \* exer19\_maiorNumero.sh
- \* exer20\_ips.sh
- \* exer21\_zip.sh
- \* **exer22\_zipTOP.sh**



# Desafio

- \* Agora que vimos como compactar arquivos para o formato .zip, pesquise e escreva um script .sh para compactar arquivos para o formato .tar.gz.
- \* Vamos primeiro entender brevemente o que é o processo de compactação de arquivos. Lembrando que o tar não compacta nada, ele apenas junta. Quem realiza a compactação é o gz.
- \* Após compactar, tente descompactar o arquivo .tar.gz através de uma função do seu script.

Obrigado.

*Contato:* [joaopauloaramuni@gmail.com](mailto:joaopauloaramuni@gmail.com)