Shell Script

If - Case – For - While

Cores e Formatação

Tabela de códigos:

CORES DE TEXTO		CORES DE FU	NDO	MODOS ANSI	
PRETO	30	PRETO	40	NORMAL	0
VERMELHO	31	VERMELHO	41	NEGRITO	1
VERDE	32	VERDE	42	BAIXA INTENSID	2
AMARELO	33	AMARELO	43	ITALICO	3
AZUL	34	AZUL	44	SUBLINHADO	4
MAGENTA	35	MAGENTA	45	PISCANDO	5
CIANO	36	CIANO	46	PISCA RAPIDO	6
BRANCO	37	BRANCO	47	INVERSO	7
				INVISIVEL	8

Forma de uso

echo "\033[05;32;45mTeste1\033[00;36mTeste2 \n"

\033[→ inicio da formatação

 1^{o_S} dois $n^{o_S} \rightarrow estilo$

 2°_S} dois $n^{\circ_S} \rightarrow cor texto$

 3^{o_S} dois $n^{o_S} \rightarrow cor fundo$

m → fim da formatação e após a palavra ou variável desejada

OBS.: não é necessário usar todos os formatos.

• crie um shell script e veja executando o código usando #!/bin/sh no inicio do arquivo se usar ./ para executar o arquivo, ou auxiliar -e após o "echo" se for usar interpretador "bash".

Testes

	Testes em variáveis			Testes em arquivos
-lt	Núm. é menor que (LessThan)		-d	É um diretório
-gt	Núm. é maior que (GreaterThan)		-f	É um arquivo normal
-le	Núm. é menor igual (LessEqual)		-r	O arquivo tem permissão de leitura
-ge	Núm. é maior igual (GreaterEqual)		-s	O tamanho do arquivo é maior que zero
-eq	Núm. é igual (EQual)		-w	O arquivo tem permissão de escrita
-ne	Núm. é diferente (NotEqual)	_	-nt	O arquivo é mais recente (NewerThan)
=	String é igual	-	-ot	O arquivo é mais antigo (OlderThan)
!=	String é diferente	[[-	ef	O arquivo é o mesmo (EqualFile)
-n	String é não nula		-a	E lógico (AND)
-z	String é nula		-0	OU lógico (OR)

Exemplo

```
#!/bin/bash
echo "Digite seu nome:
read nome
if [ -z $nome ]
then
    echo "Você não digitou seu nome!"
else
    echo "Olá, $nome"
fi
```

Exemplo

```
#!/bin/bash
echo "Digite um número"
read numero
if [ $numero -gt 0 ];
then
    echo "Número positivo"
elif [ $numero -lt 0 ]
then
    echo "Número negativo"
elif [ $numero -eq 0 ]
then
    echo "Número é zero"
else
    echo "O valor fornecido não é um número!"
fi
```

Estrutura Case

```
case $variável in
    comandos
    comandos
esac
```

Sobre Case

- •Note que o ? deve ser substituído pelo valor a esperado seja numero ou caracteres, sendo necessário o uso de aspas no segundo caso.
- •O fim de cada CASE é dado pelo duplo ponto e virgula.
- •O fim global do método CASE é seu próprio nome invertido ESAC

Exercício de fixação

Construa um shell script que leia um nome e após questiona se este usuário é gremista ou colorado por entradas de caracteres ou números.

Com o uso do método CASE mostrar fulano é gremista nas cores do time no texto e fundo, ou fulano é colorado nas cores do time no texto e fundo.

For

for var in sequencia

do

comandos

done

OBS.: var assume os valores persentes na sequencia na mesma ordem

Teste de funcionalidade

Execute o código do exemplo para ver seu resultado.

for i in 4 2 6 0

do

echo \$i

done

For com seq

No lugar de digitar uma sequencia de valores podemos usar a funcionalidade \$seq() com 1, 2 ou 3 campos.

- \cdot \$(seq 5) \rightarrow vai de 1 a 5
- \cdot \$(seq 3 6) \rightarrow vai de 3 a 6
- $-\$(\text{seq 4 2 12}) \rightarrow \text{vai de 4 a 12 de 2 em 2}$

Exercício de fixação

Crie um script que leia um valor numérico e por meio de um for mostre a tabuada de 1 a 10 com este valor lido do usuário.

Ex.: Multiplicar: 3

$$3 \times 1 = 3$$

$$3 X 2 = 6$$

$$3 X 3 = 9$$

• • •

Funcionalidade arq

Se refere a arquivos, no caso de querermos listar os arquivos de nosso diretório contabilizando ou apenas numerando podemos usar o seguinte código. Teste.

done

Exercício de fixação

Crie um script que mostre o numérico total de arquivos, quantas pastas e arquivos comuns existem na pasta do usuário padrão linux por meio do comando "for":

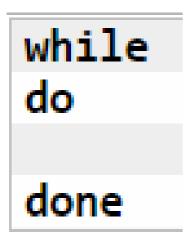
Ex.:

Total = 7

Pastas = 5

Arq. Com. = 2

While



```
i=0
while [ $i -lt 10 ]
do
    echo $i
    i=$((i+1))
done
```

While usando break

```
i=0
while true
do
    if [ $i -ge 10 ]
    then
        break
    fi
    i=$((i+1))
done
```

```
while read i
do

if [ $i -eq 0 ]
then

break
fi
done
```

Escrevendo em arquivo

```
i="-"
while [ "#" != $i ]
do
    read i
    echo 'Voce digitou ' $i
    echo $i >> texto.txt
done
```

Lendo arquivos

```
cat 'texto.txt' | while read i
do
    if [ "#" != $i ]
        then
            echo $i
        else
            break
    fi
done
```