

Shell Script do zero

Aula 4 – Condição IF e escrevendo o primeiro script

Condição IF

No inglês “if” significa “se”, (SE a condição for satisfeita eu executo o comando), condição que será testada usando o conceito da última aula. O teste de comparação acusou verdadeiro, então ele executa o comando que está dentro do if, o teste acusou falso então ele pula este if e segue com o script.

Pense nos operadores de comparação como se fossem chaves, e o if sendo a porta, se a chave for “verdadeira” ela abre a porta e executa o que tem lá dentro, se a chave for falsa o shell não consegue abrir a porta e conseqüentemente não executa o que esta lá dentro. Nesta aula aprenderemos como escrevê-lo na linguagem do shell e conseqüentemente o primeiro script.

Estrutura do if

```
if [ condição ];then
  comando
fi
```

- if → Abre o comando
- Condição → Condição comparativa para execução dos comandos dentro do if
- Comando → Qualquer comando do shell e quantos você quiser
- fi → Fecha o if (escrito ao contrário)

Exemplo na linguagem do shell:

█ → Espaço obrigatório

A palavra julia esta entre aspas porque é um texto (numéricos ficam sem aspas), a mesma coisa vale para a variável.

```
if [ "$USUARIO" = "julia" ];then
  mkdir $USUARIO
fi
```

```
#!/bin/bash
```

```
if [ "$USUARIO" = "julia" ];then
mkdir $USUARIO
fi
```

No Shell

Tradução ↑

Se o conteúdo da variável USUARIO é igual a palavra julia então execute o comando, que no caso cria um diretório com o nome de julia.

Comando auxiliar Else

Uma função complementar e muito útil no comando if é o else (senão), caso a condição do if não seja verdadeira, ele automaticamente executa o que está no else, seu uso é opcional.

Sentido do comando	Sintaxe no shell
se [condição é verdadeira]; então execute o comando mkdir \$USUARIO senão execute este outro comando mkdir \$USUARIO2	if [condição];then comando else comando fi

Abaixo um exemplo escrito no editor de texto:

Valor da variável a seguir

ESTADO=\$"operando"

Se a variável ESTADO é diferente de "desligado", então execute o comando killall... **senão** ...

```
#!/bin/bash

#IF simples

if [ "$ESTADO" != "desligado" ];then
    killall vlc
else
    sleep 10
    exit
fi
```

Este pequeno if compara a variável ESTADO com a palavra desligado, se a condição for verdadeira ele mata o player vlc, **senão** ele "dorme" 10 segundos e sai (exit).

Agora que sabemos escrever a condição if, vou mostrar como se escreve os operadores lógicos de comparação "-n" e "-z", prometidos na última aula, eles são escritos antes da variável a ser comparada como nula ou não nula.

```
*teste *
```

```
if [ -z "$PORTA" ];then
echo "Variavel sem valor"
fi
```

Se a variável PORTA estiver vazia ele mostra o texto na tela: Variável sem valor

```
if [ -n "$PORTA" ];then
echo "A variavel não esta vazia"
fi
```

Ela recebendo algum valor ele mostra: A variável não esta vazia

Poderíamos dar o "pulo do gato" e colocar este script com apenas um if → se a variável não tem valor, mostre na tela: "variável sem valor" **senão** mostre: "A variável não esta vazia"

```
#!/bin/bash

if [ -z "$PORTA" ];then
echo "Variavel sem valor"
# Senão quer dizer que ela tem algum valor
else
echo "A varivel não esta vazia"
fi
```

Então vamos supor que você necessite rodar um comando apenas se a variável conseguiu colher os dados do comando "ps" por exemplo (para verificar se determinado programa esta rodando), não hesite em usar o -z e o -n.

Antes de fazermos o nosso primeiro script vamos "juntar as peças": vimos os comandos básicos, o conceito de variáveis, os operadores de comparação e por último o comando if. Se juntarmos isto mais as pequenas dicas a seguir seremos capazes de fazer o primeiro script.

↓↓↓ **Nosso primeiro script**

Vamos fazer nosso primeiro script/exercício, no link do final da página temos o script pronto, mas é claro que você vai resolvê-lo antes de conferir, não precisa ficar igual, basta funcionar.

Passos a passo de como fazer o script, antes do exercício:

- Abra o editor de texto escolhido
- Escreva na primeira linha `#!/bin/bash`
- Agora vamos salvar para que as linhas fiquem coloridas
- Tem pessoas que colocam a extensão .sh no nome do arquivo (tanto faz)
- Para não termos problemas futuros, já vamos dar permissão de execução (`chmod +x script`)
- Agora já podemos escrever o script
- Quando o script estiver pronto entre no diretório em questão e dê o comando `./nome_do_script` para executá-lo

Lembretes:

- Preste muita atenção nos espaços, porque se der espaço a mais ou a menos não vai rodar.
- Lembre-se de sempre fechar o if → “fi”
- Os comandos sempre são executados de cima para baixo sequencialmente, ou seja, só executa a próxima linha quando terminar a atual, a não ser que usemos o “&”
- Quando houver erros, o terminal mostra em qual linha o mesmo está, nem sempre é aquela linha indicada, pode ser algumas depois
- Verticalmente pode saltar quantas linhas quiser

Exercício

Função do script → Temos 3 médicos, cada um atende num turno diferente, o usuário informa qual turno deseja se consultar e o programa mostra o nome do médico do turno escolhido, só isso!

Etapas do script:

- 1- Mostra um texto de boas vindas ao usuário
- 2- Pede que o usuário escolha qual turno em que deseja se consultar
- 3- Mostra o texto nestes moldes: Médico TAL é o único a atender no turno TAL
- 4- “Dorme” por 2 segundos
- 5- Informa que a consulta está marcada
- 6- O programa “dorme” por mais 4 segundos e sai

<http://www.mediafire.com/download/b52el0vr94nzp3y/primeiro>

Na próxima aula vamos aprender sobre os operadores lógicos de conexão, e assim poderemos usar mais de uma condição comparativa no if, fazendo scripts mais elaboradas.