Variáveis em SHELL

$variavel

Para criar uma variável é muito simples, vamos editar o meuscript.sh e alterar/ adicionar uma linha, com a seguinte sintaxe

```
#!/bin/bash
# Meu Comentário
OLA="Olá, mundo!"
echo $OLA
```

Variáveis em SHELL

✔ OLA="conteudo" ⟵ CERTO

✗ OLA = "conteudo" ⟵ ERRADO

Variáveis de Ambiente

printenv

env

sh-utils, Shellutils, Fileutils, coreutils, …

https://www.gnu.org/software/shellutils/shellutils.html

Variáveis em SHELL

- O comando **set** é usado pra *mostrar/modificar* variáveis de ambiente.
- O comando **export** *exporta* uma variável para as variáveis de ambiente.
- Dica: **set -o** (mostra **off** e **on**) e **noclubber** (**>** *para não sobrepôr*).

Variáveis Especiais

São variáveis composta de números e caracteres que passam parâmetros para funções/scripts e exibem status do script.

Variáveis

Variável	Parâmetros Posicionais
$0	Parâmetro número 0 (nome do comando ou função)
$1	Parâmetro número 1 (da linha de comando ou função)
...	Parâmetro número N ...
$9	Parâmetro número 9 (da linha de comando ou função)
${10}	Parâmetro número 10 (da linha de comando ou função)
...	Parâmetro número NN ...
$#	Número total de parâmetros da linha de comando ou função
$*	Todos os parâmetros, como uma string única
$@	Todos os parâmetros, como várias strings protegidas
Variável	**Miscelânia**
$$	Número PID do processo atual (do próprio script)
$!	Número PID do último job em segundo plano
$_	Último argumento do último comando executado
$?	Código de retorno do último comando executado

Expansão de Variáveis em SHELL

Caracteriza-se pela combinação de comando único , e usa-se um cifrão:

$(comando), ${comando} seguido de parenteses/colchetes , que o levam a uma subshell, ou seja, um comando executado em segundo plano "invisível", alguns usam assim `comando` , eu não indico pois além de ser uma sintaxe feia, você pode confundir com a aspas simples e alguns casos você precisa sempre escapar essas crases.

Expansão de Variáveis em

Sintaxe	Expansão Condicional
${var:-texto}	Se var não está definida, retorna 'texto'
${var:=texto}	Se var não está definida, defina-a com 'texto'
${var:?texto}	Se var não está definida, retorna o erro 'texto'
${var:+texto}	Se var está definida, retorna 'texto', senão retorna o vazio
Sintaxe	**Expansão de Strings**
${var}	É o mesmo que $var, porém não ambíguo
${#var}	Retorna o tamanho da string
${!var}	Executa o conteúdo de $var (igual 'eval \$$var')
${!texto*}	Retorna os nomes de variáveis começadas por 'texto'
${var:N}	Retorna o texto a partir da posição 'N'
${var:N:tam}	Retorna 'tam' caracteres a partir da posição 'N'
${var#texto}	Corta 'texto' do início da string
${var##texto}	Corta 'texto' do início da string (* guloso)
${var%texto}	Corta 'texto' do final da string
${var%%texto}	Corta 'texto' do final da string (* guloso)
${var/texto/novo}	Substitui 'texto' por 'novo', uma vez
${var//texto/novo}	Substitui 'texto' por 'novo', sempre
${var/#texto/novo}	Se a string começar com 'texto', substitui 'texto' por 'novo'
${var/%texto/novo}	Se a string terminar com 'texto', substitui 'texto' por 'novo'

Curiosidade

Tente rodar o comando abaixo e veja o resultado!!!

```
echo t{r,igr,rist}es
```

Arrays em Shell

(Vetores de Variável)

Array é uma estrutura de dados que consiste em itens relacionados entre si. É um grupo de valores alocados em um único elemento declarado. A referência é feita a uma posição em particular. Além disso, no caso do *shell BASH* ele possibilita o uso por blocos de elementos.

Um array pode ser declarado como lista de elementos de uma só vez:

```
mundo=("Shell Script" "Bash" "GNU" "Linux" "Debian")
```

Arrays em Shell

Ou uma declaração elemento a elemento:

```
mundo[0]="Shell Script"
mundo[1]="Bash"
mundo[2]="GNU"
mundo[3]="Linux"
mundo[4]="Debian"
```

Não esquecendo que em *arrays* o primeiro elemento é sempre o zero 0, para referenciar o 4° elemento q é o número "3", chamamos assim:

```
${mundo[3]} # Linux
```

Você também pode declarar o array com o comando bult-in do Bash, declare , o -a aponta para o nome do array. (Comandos bult-in, são comandos embutidos no próprio Bash, cada *Shell* tem seus comandos embutidos assim que você instala eles, os comandos são daquele Shell e não do sistema.) $ declare -a mundo

Alguns exemplos para chamar elemento(s) de um array

- Contar o número de elementos de um array: **${#mundo[@]}** # 5
- Exibir todos os elemento: ${mundo[*]}
- Deletar um determinado elemento do array: **unset mundo[2]**
- Deletar todo array: **unset mundo**
- Mostrar do 3° (que tem número 2) até o final: **${mundo[@]:2}** # GNU Linux Debian
- Mostrar o 2° elemento (que tem o número 1) até o 4° elemento: **${mundo[@]:1:3}** # Bash GNU Linux

Veja nesse link 45 formas de exibir elementos de um array:

http://terminalroot.com.br/2015/08/45-exemplos-de-variaveis-e-arrays-em_19.html

Vamos adicionar esse conhecimento ao script *meuscript.sh*