

Capítulo 7

===Exercício 1===

Coloque em um vetor os seguintes animais:

Bicho papão
Burro quando foge
Galinho de Quintino
Leão de chácara

Agora liste os animais do vetor e seus índices.

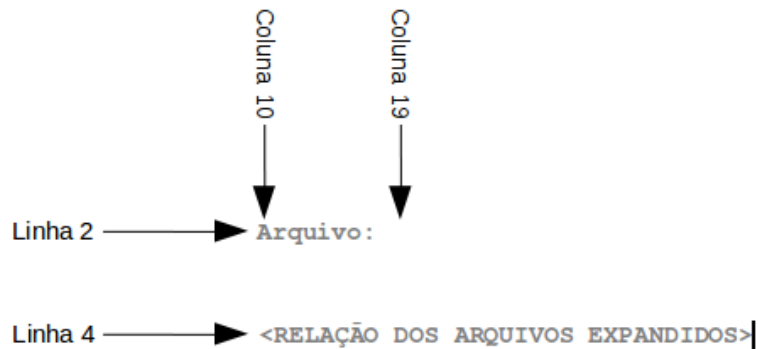
```
$ cat exerc7-1.sh
#!/bin/bash
# Exercício 1 do capítulo 7
# Gera um vetor e lista-o usando seus próprios índices

Bichos=('Bicho papão' 'Burro quando foge' 'Galinho de Quintino' 'Leão
de chácara')
Bichos+=([9]='Macaco de imitação')      # Só para o vetor ficar esperso
for Ind in ${!Bichos[@]}                  # Todos os índices do vetor
{
    echo -e "$Ind\t${Bichos[Ind]}"        # Listando um a um pelo índice
}
```

===Exercício 2===

Fazer um *script* que formate a tela da seguinte forma:

- A) Escreva na linha 2, coluna 10 a palavra **Arquivo;**
- B) Após isso, iniciaremos um *loop* de leitura de nomes de arquivos, onde serão aceitos metacaracteres de expansão de arquivos e será dado uma mensagem de erro para *StdErr* caso não exista nenhum arquivo;
- C) Se essa leitura de nomes de arquivos voltar um campo vazio (o operador deu um **<ENTER>**), sairemos desse *loop*;
- D) Após verificar que existe(m) arquivo(s) que atende(m) ao que foi digitado, será listado na linha 4, a partir da coluna 10 os nomes expandidos dos arquivos e estes serão incluídos no vetor **Arqs**, que será integralmente listado ao final.



```
$ cat exerc7-2.sh
```

```
#!/bin/bash
# Exercício 2 do capítulo 7
#+ Loop de leitura de arquivos em tela formatada
#+ aceitando metacaracteres e com exposição parcial e
#+ total dos arquivos informados após expansão dos coringas

clear
tput cup 2 10          # Posicionando na linha 2 coluna 10
echo "Arquivo:"
while true
do
    tput cup 2 19      # Posicionando na linha 2 coluna 10
    tput el           # Apaga nome do arquivo anterior
    read Arq
    [[ -z $Arq ]] && {
        tput cup 5 0
        [[ -n $Arqs ]] && {
            echo -e "Arquivos informados:\nÍndice\tArquivo"
            for Ind in ${!Arqs[@]}
            {
                echo -e "    $Ind\t${Arqs[Ind]}"
            }
        }
        exit
    }
    Trab=$(ls $Arq 2> /dev/null) || { # Variável de trabalho
        tput cup 5 0
        echo Arquivo $Arq não existe. Tecle algo... >&2
        read -n1
        tput cup 5 0; tput el      # Apaga a mensagem de erro
        continue
    }
    tput cup 4 10
    tput el                      # Apaga arquivos da relação anterior
    echo $Trab
    Arqs+=($Trab)
done
```

===Exercício 3===

Como posso listar os 5 últimos elementos desse vetor que acabamos de gerar (**Arqs**)?

```
$ echo ${Arqs[@]: -5:5}
```

OU:

```
$ echo ${Arqs[@]: (-5):5}
```

===Exercício 4===

Os comandos:

```
tr -d '[:punct:]' < ARQUIVO
```

OU:

```
tr -dc '[:alnum:]\n ' < ARQUIVO
```

Listam o arquivo **ARQUIVO**, sem sua pontuação. Use uma dessas linhas (aconselho que você entenda a segunda) em um *script* que, após excluir toda a pontuação, conte quantas vezes cada palavra ocorre em **ARQUIVO**, cujo nome foi passado por parâmetro e ao final diga a quantidade de vezes que cada palavra apareceu.

Se você quiser fazer legal mesmo, ponha todas as palavras em caixa alta ou caixa baixa.

```
$ cat exerc7-4.sh
```

```
#!/bin/bash
# Exercício 4 do capítulo 7
# Conta a quantidade de ocorrências de cada palavra
#+ num arquivo cujo nome será passado por parâmetro
declare -A Palavras # Array que será usado para contar palavras
Arq=$1
[[ -n $Arq ]] || { # Se não existir um arquivo com esse nome
    echo Arquivo não existe ou não é um arquivo regular >&2
    exit 1
}
# Passando para minúscula usando o sed, usando o tr para tirar
#+ a pontuação e guardando saída em um arquivo temporário, pois
#+ uma variável poderia estourar com grandes arquivos.
sed 's/\(.*\)/\L\1/' $Arq | tr -dc '[:alnum:]\n\t ' > /tmp/Arq$$
for Palavra in $(cat /tmp/Arq$$)
{
    ((${#Palavra} < 3)) && continue # Só conta palavras com 3 ou
mais caracteres
    let Palavras[$Palavra]++
}
> /tmp/Arq$$ # Esvaziando arquivo de trabalho
# Saída irá para /tmp/Arq$$ para ordenar e listar
#+ o arquivo terá <QTD OCORRENCIAS><TAB><PALAVRAS>
for Palavra in ${!Palavras[@]}
{
    echo -e "${Palavras[$Palavra]}\t$Palavra" >> /tmp/Arq$$
}
```

```
}  
sort -nr /tmp/Arq$$ | more
```