Loops em Shell

A maioria das línguas tem o conceito de loops. Se quisermos repetir uma tarefa vinte vezes, não queremos digitar o código “vinte vezes”, com talvez uma ligeira mudança a cada vez.

TIPOS DE LOOP FOR

```
for ((i=0;i<10;i++));
do
    echo $i
done
```

```
for i in {2..8}
 do
    echo $i
done
```

```
for i in $(seq 2 8)
 do
    echo $i
done
```

Exemplo:

- Wildcards/Coringas terminal

```
for i in Começando 1 * 2 Finalizando
    do
        echo "Criando um laç
```

WHILE

```
contador=0;
while [ $contador -lt 10 ]; do
    echo O valor de contador = $contador
    (( contador = contador + 1 ))
done
```

UNTIL

```
COUNTER=20
until [  $COUNTER -lt 10 ]; do
    echo COUNTER $COUNTER
    let COUNTER-=1
done
```

CONTINUE, BREAK e EXIT

- A instrução continue é usada para retomar a iteração seguinte do loop *FOR, WHILE ou UNTIL*.
- Use a instrução break para sair de dentro de um loop *FOR, WHILE ou UNTIL*, isto é, pare a execução do loop.
- O exit é usado para definir o status da execução do programa, se o valor de $? for 0 então tudo ocorreu naturalmente, se não houve
- erro. Você pode pôr o exit 0 no final do seu script para sair sem problemas, e um echo $? para saber qual foi o código de retorno na excução do programa.
- No exemplo ao lado só irá imprimir 8 e 9 , perceba o uso do continue , break e exit.


```
for i in $(seq 1 10);
do
        if [[ "$i" < "8" ]]; then
                continue
        fi

        if [[ "$i" > "9" ]]; then
                break;
        fi

        echo $i;

done

exit 0;
```