<epam>

While loop

- What is while
- Use cases and syntax

TRAINING

CENTER

<epam>

What is while loop

The **while** loop is used to performs a given set of commands an unknown number of times as long as the given condition evaluates to true.

The Bash **while** loop takes the following form:

```
while <condition>
do
  <commands>
done
```

The condition is evaluated before executing the commands. If the condition evaluates to true, commands are executed.

Otherwise, if the condition evaluates to false, the loop is terminated, and the program control will be passed to the command that follows.

While sample

In the example below, on each iteration, the current value of the variable i is printed and incremented by one.

```
i=0

while [ $i -le 2 ]
do
  echo Number: $i
  ((i++))
done
```

Due loop iterates as long as i is less or equal than two. It will produce the following output:

```
[root@localhost tmp]# bash while_sample.sh
Number: 0
Number: 1
Number: 2
[root@localhost tmp]#
```

While use cases

*Infinite **while** loop*

An infinite loop is a loop that repeats indefinitely and never terminates. If the condition always evaluates to true, you get an infinite loop.

```
while :
do
  echo "Press <CTRL+C> to exit."
  sleep 1
done
```

In the following example, we are using the built-in command ":" to create an infinite loop. ":" always returns true. You can also use the true built-in or any other statement that always returns true.

```
[root@localhost tmp]# bash infinite_while.sh
Press <CTRL+C> to exit.
Press <CTRL+C> to exit.
Press <CTRL+C> to exit.
Press <CTRL+C> to exit.
```

While use cases

Read a file line by line

One of the most common usages of the **while** loop is to read a file, data stream, or variable line by line.

Here is an example that reads the **~/.bashrc** file line by line and prints each line:

```
file=/etc/passwd

while IFS= read -r line; do
  echo $line
done < "$file"
```

Instead of controlling the **while** loop with a condition, we are using input redirection (**< "$file"**) to pass a file to the **read** command, which controls the loop. The while loop will run until the last line is read.

```
[root@localhost tmp]# bash read_while.sh
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
```

Example of useful while loop

The example below was written to copy pictures that are made with a webcam to a web directory. Every five minutes a picture is taken. Every hour, a new directory is created, holding the images for that hour. Every day, a new directory is created containing 24 subdirectories. The script runs in the background.

```
#!/bin/bash

PICSDIR=/home/carol/pics
WEBDIR=/var/www/carol/webcam

while true; do
    DATE=`date +%Y%m%d`
    HOUR=`date +%H`
    mkdir $WEBDIR/"$DATE"

    while [ $HOUR -ne "00" ]; do
        DESTDIR=$WEBDIR/"$DATE"/"$HOUR"
        mkdir "$DESTDIR"
        mv $PICDIR/*.jpg "$DESTDIR"/
        sleep 3600
        HOUR=`date +%H`
    done
done
```

Thanks for watching!