**BASH Scripting**

**Exit code**  
In Linux, the exit codes range from **0 to 255**, with 0 representing success, and 1-255 indicating various failure conditions.

**If**

if [[ $? == 0 ]]; then

Echo “Code ran as expected”

Else

Echo ”code did not work!”

fi

**If Else if**

#!/bin/bash

echo "enter your favorite number!"

read myfnumber

if [[ $myfnumber -gt 10 ]]

then

echo "The variable is greater than 10."

elif [[ $VAR -eq 10 ]]

then

echo "The variable is equal to 10."

else

echo "The variable is less than 10."

fi

**Bash Arithmetic**

echo $((5\*3))

echo $((2+4))

x=5

y=3

echo $(($x\*$y))

echo $((x=100, y=50, x+y))

foo="$[ 3 + 2 ]" ; echo $foo

**Test**

test 10 -gt 5 ; echo $?

test 5 -gt 10 ; echo $?

**File test operators**

**Returns true if...**

-e

file exists

-a

file exists

This is identical in effect to -e. It has been "deprecated," and its use is discouraged.

-f

file is a *regular* file (not a directory or device file)

-s

file is not zero size

-d

file is a directory

-b

file is a block device

-c

file is a [character device](https://tldp.org/LDP/abs/html/devref1.html#CHARDEVREF)

**integer comparison**

-eq

is equal to

**if [ "$a" -eq "$b" ]**

-ne

is not equal to

**if [ "$a" -ne "$b" ]**

-gt

is greater than

**if [ "$a" -gt "$b" ]**

-ge

is greater than or equal to

**if [ "$a" -ge "$b" ]**

-lt

is less than

**if [ "$a" -lt "$b" ]**

-le

is less than or equal to

**if [ "$a" -le "$b" ]**

<

is less than (within [double parentheses](https://tldp.org/LDP/abs/html/dblparens.html))

**(("$a" < "$b"))**

<=

is less than or equal to (within double parentheses)

**(("$a" <= "$b"))**

>

is greater than (within double parentheses)

**(("$a" > "$b"))**

>=

is greater than or equal to (within double parentheses)

**(("$a" >= "$b"))**

**For loop**

#!/bin/bash

for file in /files/ldd /files/less /files/ln

do

crc32 $file

done

#!/bin/bash

for i in {1..10}

do

   echo "number is $i times"

done

**Arguments**

#!/bin/bash

echo $0

echo $1

echo $1

**Debugging with BASH!**

bash -x

Or  
set -x on your script

**Real life example:**

#!/bin/bash

if test -f /etc/apache2/apache2.conf; then

echo "Apache is installed no need to further action"

else

echo "apache not installed, and will install it soon"

apt install apache2 -y

fi