<epam>

If Statement

- What is If statement
- Which conditions can If have

TRAINING

CENTER

<epam>

What is If statement

At times you need to specify different courses of action to be taken in a shell script, depending on the success or failure of a command, or any other conditions. The **if** construction allows you to specify such conditions.

A basic **if** statement effectively says, if a particular test is true, then perform a given set of actions. If it is not true then don't perform those actions. If follows the format below:

```
if [ <some test> ]
then
   <commands>
fi
```

Anything between **then** and **fi** (if backwards) will be executed only if the test (between the square brackets) is true.

If with [] operator

The [] operator is an alias for a **test** command in bash and returns an exit code of **0** when the condition is **true** and **1** when the condition is **false.**

Let's look at a simple example:

```
#!/bin/bash

if [ $1 -gt 100 ]
then
  echo "Test condition is true"
fi
```

A simple script that uses the test ([]) statement as a condition

```
[root@EVUAKYIA001B tmp]# bash if_test.sh 150
Test condition is true
[root@EVUAKYIA001B tmp]# bash if_test.sh 50
[root@EVUAKYIA001B tmp]#
```

Execution example

If with [[]] operator

The double square brackets [[]] is a shell keyword. It is similar in behavior to the single square bracket and is used to evaluate conditional expressions and is a Bash, Zsh, and Korn shell specific.

```
[root@EVUAKYIA001B tmp]# cat if_test.sh
#!/bin/bash

echo $HOSTNAME

if [[ $HOSTNAME =~ "epam.com" ]]
then
  echo "Test condition is true"
fi

[root@EVUAKYIA001B tmp]# bash if_test.sh
EVUAKYIA001B.kyiv.epam.com
Test condition is true
```

Simple script with [[]] operator

This construct can handle more complex conditions and is less error-prone and is the recommended way to use conditions in an **if** statement

If with (()) operator

The double parentheses (()) is used to test an arithmetic expression. It does support the && and || binary operators.

```
[root@EVUAKYIA001B tmp]# cat if_test.sh
#!/bin/bash

if (( $1 % 2 == 0 ))
then
  echo "The number is even"
fi
[root@EVUAKYIA001B tmp]# bash if_test.sh 24
The number is even
[root@EVUAKYIA001B tmp]# bash if_test.sh 23
```

Simple script with (()) operator

Using else statement

Sometimes we want to perform a certain set of actions if a statement is **true**, and another set of actions if it is **false**. We can accommodate this with the **else** mechanism.

```
if [ <some test> ]
then
   <commands>
else
  <other command>
fi
```

If statement schema using the "**else**" operator

Simple script with "**else**" operator

```
[root@EVUAKYIA001B tmp]# cat if_test.sh
#!/bin/bash

echo $HOSTNAME

if [[ $HOSTNAME =~ "minsk" ]]
then
  echo "Test condition is true"
else
  echo "Test condition is false"
fi

[root@EVUAKYIA001B tmp]# bash if_test.sh
EVUAKYIA001B.kyiv.epam.com
Test condition is false
```

Using elif statement

Sometimes we may have a series of conditions that may lead to different paths. In that case we can use **elif** operator, that create another branch in **if** statement. You can have as many **elif** branches as you like

```
if [ <some test> ]
then
   <commands>
elif [ <another test> ]
then
   <other command>
fi
```

If statement schema using the "**elif**" operator

```
[root@EVUAKYIA001B tmp]# cat if_test.sh
#!/bin/bash

echo $HOSTNAME

if [[ $HOSTNAME =~ "minsk" ]]
then
  echo "True in first condition"
elif [[ $HOSTNAME =~ "kyiv" ]]
then
  echo "True in second condition"
fi
[root@EVUAKYIA001B tmp]# bash if_test.sh
EVUAKYIA001B.kyiv.epam.com
True in second condition
```

Simple script with "**else**" operator

Using if without test

You can also use **if** without condition because for **if** it is important to get the exit code of the command/condition and it does not matter what these commands will be

```
[root@EVUAKYIA001B tmp]# cat if_test.sh
#!/bin/bash

FOO=$1
BAR=$2

if echo $FOO | grep $BAR; then echo "BAR is substring of FOO string"
else echo "BAR isn't substing of FOO string"
fi
[root@EVUAKYIA001B tmp]# bash if_test.sh $HOSTNAME kyiv
EVUAKYIA001B.kyiv.epam.com
BAR is substring of FOO string
[root@EVUAKYIA001B tmp]# bash if_test.sh $HOSTNAME minsk
BAR isn't substing of FOO string
```

Thanks for watching!