

Conditional execution in BASH

Talk to a Teacher Project

<http://spoken-tutorial.org>

National Mission on Education through ICT

<http://www.sakshat.ac.in>

FOSSEE & Spoken Tutorial Team
IIT Bombay

August 26, 2013



Learning Objectives



Learning Objectives

► Use of test



Learning Objectives

- ▶ Use of `test`
- ▶ Conditional statements



Learning Objectives

- ▶ Use of `test`
- ▶ Conditional statements
- ▶ We will do this using a few examples



Prerequisite



Prerequisite

- ▶ You should be familiar with GNU/Linux Operating System



Prerequisite

- ▶ You should be familiar with GNU/Linux Operating System
- ▶ If not, for relevant tutorials please visit <http://spoken-tutorial.org>



System Requirements



System Requirements

- ▶ **Ubuntu Linux 12.04 OS**



System Requirements

- ▶ **Ubuntu Linux 12.04 OS**
- ▶ **GNU Bash version 4.1.10**



System Requirements

- ▶ Ubuntu Linux 12.04 OS
- ▶ GNU Bash version 4.1.10
- ▶ **GNU Bash version 4 or above is recommended for practice**



test



test

- ▶ `test` is a built-in command



test

- ▶ `test` is a built-in command
- ▶ 0 (True) and 1 (False)



test

- ▶ `test` is a built-in command
- ▶ **0 (True)** and **1 (False)**
- ▶ Return value depends on the evaluation of the expression



test



- Get the return status by typing `$?`



test

- ▶ Get the return status by typing `$?`
- ▶ An expression can be evaluated in two ways:



test

- ▶ Get the return status by typing `$?`
- ▶ An expression can be evaluated in two ways:
 - ▶ Using the keyword `test`



- ▶ Get the return status by typing `$?`
- ▶ An expression can be evaluated in two ways:
 - ▶ Using the keyword `test`
 - ▶ Using the expression enclosed within square brackets `[]`



Conditional Execution: if



Conditional Execution: if

Syntax:

```
► if [ expression ]; then  
  commands  
fi
```



Basic rules of condition



Basic rules of condition

- **Keep spaces between the brackets and the expression**



Basic rules of condition

- ▶ Keep spaces between the brackets and the expression
- ▶ Terminate the line using **‘;’** before keyword **then**



Basic rules of condition

- ▶ Keep spaces between the brackets and the expression
- ▶ Terminate the line using ';' before keyword `then`
- ▶ Semicolon is used to terminate the statement or an expression



Basic rules of condition



Basic rules of condition

- ▶ Quote string variables if you use them in conditions



Basic rules of condition

- ▶ Quote string variables if you use them in conditions
- ▶ Close conditional block with **fi**



Conditional Execution: if-else



Conditional Execution: if-else

Syntax:

```
► if [ expression ]; then  
    commands  
else  
    some other commands  
fi
```



Summary

- ▶ Usage of **test**
- ▶ Simple **if** statement
- ▶ **if-else** statement



Assignment

Write a script

- ▶ Take your name as an input
- ▶ It should check this name with your system's username
- ▶ If the username match, it should greet you by displaying **Hello**



Assignment

- ▶ Else, it should display **Try again**
- ▶ Hint: Your system's username is stored in a variable **\$USER**



About the Spoken Tutorial Project

- ▶ Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- ▶ It summarizes the Spoken Tutorial project



About the Spoken Tutorial Project

- ▶ Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- ▶ It summarizes the Spoken Tutorial project
- ▶ If you do not have good bandwidth, you can download and watch it



Spoken Tutorial Workshops

The Spoken Tutorial Project Team

- ▶ Conducts workshops using spoken tutorials
- ▶ Gives certificates to those who pass an online test
- ▶ For more details, please write to contact@spoken-tutorial.org



Acknowledgments

- ▶ Spoken Tutorial Project is a part of the Talk to a Teacher project
- ▶ It is supported by the National Mission on Education through ICT, MHRD, Government of India
- ▶ More information on this Mission is available at

<http://spoken-tutorial.org/NMEICT-Intro>

