



Introduction to Bash Video lecture

01 Some key concepts



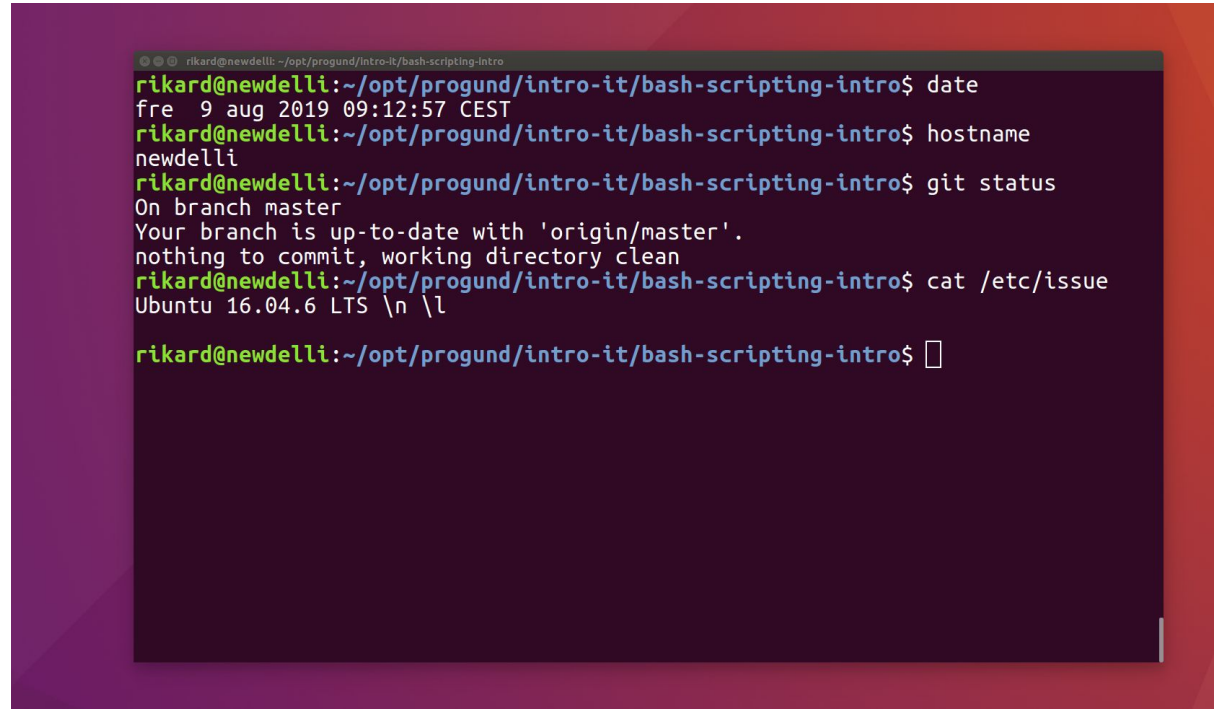
Terminal, command line, shell, Bash

- You open a *terminal* in order to work in the *command line*
- The terminal runs a *shell*
- The shell interprets command lines
- A command line is a series of commands ending with a newline
- The shell is *interactive* which means you take turns when communicating
- Since the shell is *line-based* it does nothing until you hit Enter
- If you hit Enter before you have written a complete command line, the shell kindly asks for more

Terminal

Gnome-terminal

running bash

A screenshot of a terminal window with a dark purple background and a red title bar. The terminal shows a user named 'rikard' at a machine named 'newdelli' in the directory '~/opt/progund/intro-it/bash-scripting-intro'. The user runs several commands: 'date' (output: 'fre 9 aug 2019 09:12:57 CEST'), 'hostname' (output: 'newdelli'), 'git status' (output: 'On branch master', 'Your branch is up-to-date with 'origin/master'.', 'nothing to commit, working directory clean'), and 'cat /etc/issue' (output: 'Ubuntu 16.04.6 LTS \n \l'). The prompt is always shown in green text.

```
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$ date
fre 9 aug 2019 09:12:57 CEST
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$ hostname
newdelli
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$ cat /etc/issue
Ubuntu 16.04.6 LTS \n \l

rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$ █
```

Shell

- In this course material, we are using the Bash shell
- A UNIX shell and command language
- Written by Brian Fox
- A free software replacement for the Bourne shell
- First version came out 1989
- Default shell on Solaris, most GNU/Linux distros, macOS
- Available also on Windows

<https://www.windowscentral.com/how-install-bash-shell-command-line-windows-10>

Command line

- Bash interprets and executes command lines
- It is called a command line, because Bash operates on lines of text
- You type in your command(s) and then hit Enter (generating a line)
- Bash interprets your command(s) and if they make sense, it executes the command(s)

Interactive

- An interactive program patiently waits for your input, and replies when you give input
- You “take turns” communicating
- You enter some text, the interactive program responds with a result
- Bash can be run interactively (which is the default behavior for the Bash running in your terminal)

Commands need to be complete

- If you enter a command line that is not complete, you will get a secondary *prompt* reminding you that the shell needs more text before it regards the command line as complete
- Not closing quotes is an example of an uncomplete command line:

```
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$ echo "
```

```
> hej  
> hurray  
> let's  
> sway"
```

```
hej  
hurray  
let's  
sway
```

Commands need to be complete

```
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$ echo "
```

```
> hej  
> hurray  
> let's  
> sway"
```

```
hej  
hurray  
let's  
sway
```

prompt

secondary prompt

Prompt

```
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$
```

user computer

current directory

\$ ends the prompt