

Linux Basics IV: Basic shell scripting

ADRIANO ANGELONE, GRAZIANO GIULIANI

Course Outline

- UNIX/Linux Basics
- Intermediate shell commands
- Editing and compiling source code
- Text file manipulation
- **Basic shell scripting**

Download slides and exercise files with the command

```
git clone https://github.com/AA24KK/LinuxBasics.git
```

or download a ZIP archive at

```
https://github.com/AA24KK/LinuxBasics/archive/master.zip
```

Adriano: **aangelon@ictp.it**, Room 263, ICTP

Graziano: **ggiulian@ictp.it**

Why shell scripting

Reuse multiple times the same command:

less work, less bugs

What are the basics:

- Variables
- Conditionals
- Loops
- Arithmetic operations

Variables

Variables contain data
they have a label (**name**) and a **content**



In bash, variables **have no type**:
everything is a string, with arithmetic sometimes possible

How to use variables

Variables are initialized when you use them the first time:

you can call non-existing variables, usually trouble
(unless you put **set -u** in your script)

Assignment: give a variable a value

<variable name>=<value>

Expansion: access the value

\${<variable name>}

```
~ » cat example_script
#!/bin/bash

a="txt"
echo $a

-----
~ » ./example_script
txt
```

Usually it's good to use **"\${<variable name>}"**
to avoid problems if variable contains spaces

Easy to join
variable values and strings

Use **\${<variable name>}**

to avoid expanding another variable

```
~ » cat example_script
#!/bin/bash

a="txt"
echo ${a}_stuff

-----
~ » ./example_script
txt_stuff
```