### Software Carpentry I: The Shell & Version Control

Luís Pedro Coelho

Programming for Scientists

February 3, 2009



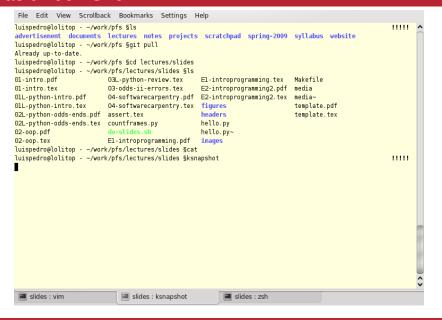


## Software Carpentry

Software engineering helps you build the equivalent of a bridge.

Software carpentry is what you need to build the equivalent of a garden shed.

#### Basic Tool: Shell



#### **Basic Concepts**

Current directory.

cd newdir

#### **Directories**

Your program also has a current directory.

```
import os
os.chdir('...')
```

#### **Paths**

- Absolute paths:
  - /home/luispedro/work/pfs/lectures/slides/04-softwa:
- Relative paths: 04-softwarecarpentry.pdf, images/04-konsole.png,../homeworks/02-oop.pdf

# Opening Files Inside Python

```
input = file('/home/luispedro/absolute.txt')
output = file('output.txt','w')
```

# **Executing Programs**

```
luispedro@lolitop - ~/work/pfs/lectures $
```

#### Input/Output

#### hello.py

print 'Hello World'

#### Shell

python hello.py

## Input/Output

Standard input/stdout.

#### grep

```
lp@lolitop ~/pfs/lects/slides Sgrep Exception *tex
01-intro.tex:\item Exception: implementing code from a pape
03-odds-ii-errors.tex:\frametitle{Exceptions}
03-odds-ii-errors.tex:\begin{block}{Exceptions}
03-odds-ii-errors.tex:\frametitle{Exceptions}
03-odds-ii-errors.tex:\frametitle{Exceptions}
03-odds-ii-errors.tex:\begin{block}{Exceptions}
03-odds-ii-errors.tex:\item Exceptions are \alert{objects}.
03-odds-ii-errors.tex:\item Exceptions have \alert{type} an
03-odds-ii-errors.tex:\frametitle{Exception Hierarchy}
03-odds-ii-errors.tex:
                                     Exception
03-odds-ii-errors.tex:\frametitle{Exception Handling}
03-odds-ii-errors.tex:\begin{block}{Exception Handling: Err
03-odds-ii-errors.tex: print 'Exception'
03-odds-ii-errors.tex:False\\Exception
03-odds-ii-errors.tex:True\\Exception
04-softwarecarpentry.tex:grep Exception *tex
```

WC

#### WC: Word Count

wc -1: count lines

lp@lolitop ~/pfs/lects/slides §grep Exception \*tex | wc -l
19

grep Exception \*tex

#### Levels

- konsole/xterm/gnome-terminal/iTerm.app/Terminal.app/...
- sh/bash/zsh/...
- your program

```
luispedro@lolitop - ~ Secho Hello World
Hello World
```

```
import sys
print len(sys.argv)
print sys.argv
```

```
$python printargs.py Hello World
3
['printargs.py', 'Hello', 'World']
```

```
Spython printargs.py "Hello World"
2
['printargs.py', 'Hello World']
```

# Command-line Argument Parsing

```
from optparse import OptionParser
parser = OptionParser()
parser.add option("-d", "--database", dest="database")
parser.add option("-i", "--ideal-scale", dest="ideal")
parser.add_option("-m", "--min-scale", dest="min")
parser.add_option("-M", "--max-scale", dest="max")
(options, args) = parser.parse args()
Elsevier DB=options.database
Ideal_Scale=float (options.ideal)
Min_Scale=float (options.min)
Max Scale=float(options.max)
```

#### Commands

- cd
- Is
- cat
- grep
- ...

#### SSH

#### Secure Shell

§ssh coupland.cbi.cmu.edu

## Password-less Login

Public key authentication.

# **Version Control** If your laptop exploded, how many hours of work would you lose?

## Advantages

- Maintain project history.
- Sync between computers.
- Sync between project members.
- ...

#### Subversion

#### Subversion: model

- Repository
- Checkout
- Commit

# Example

- Create a repository
- Create a checkout
- Edit
- Commit

## Version Control Etiquette

- Don't commit over my commit.
- Use the log.