



Bash essentials

S. Maccolini

<https://hsf-training.github.io/analysis-essentials/shell/README.html>

Starterkit, 22 November 2021

Bash essentials (1)

- > ssh -XY smaccoli@lxplus.cern.ch
- > whoami
smaccoli
- > hostname
lxplus727.cern.ch
- > pwd [print work directory]
/afs/cern.ch/user/s/smaccoli
- > mkdir starterkit21_bash [make directory]
- > cd starterkit21_bash [change directory]
- > mkdir subdirectory
- > cd ../
- > cd ~ (or cd)
- > cd -
- > touch test.txt
- > touch subdirectory/newtest.txt
- > ls
- > ls (ls -altrh . ; ls -altrhR . or ls -altrh *) [all, list, time, reverse, human, Recursive]
- > ls -d "\$PWD/"*.cpp

Bash essentials (2)

- `$ mkdir thesis`
- `$ emacs -nw main.tex`
- `$ cp main.tex main_backup.tex` [copy]
- `$ emacs -nw chapter1.tex`
- `$ mkdir chapter1`
- `$ mv chapter1.tex chapter1/.` [move]
- `$ mv chapter1 detector_chapter`
- `$ rm main.tex`
- `$ cp thesis thesis_nov21`
- `$ rm -r thesis (rm -r -i)`
- `$ history` [ctrl-R]

Bash tools

- ```
$ cp /afs/cern.ch/user/s/smaccoli/public/data-shell .
$ cd data-shell
$ ls */*.pdb
$ wc methane.pdb (wc -l ; wc -w; wc -c) [line, word, characters
counter]
$ wc -l *.pdb > lengths.txt
$ cat lengths.txt
$ less lengths.txt
$ sort -n lengths.txt
$ sort -n lengths.txt > sorted-lengths.txt
$ head -n 1 sorted-lengths.txt
$ wc -l *.pdb | sort -n | head -n 1
$ wc -l notes.txt
$ wc -l < notes.txt
$ cat notes.txt | wc -l
$ cat *.pdb > all.pdb
```

# Sort, uniq, cut, sed

---

- `$ sort data-shell/data/salmon.txt`  
`$ uniq data-shell/data/salmon.txt`  
`$ sort data-shell/data/salmon.txt | uniq`  
`$ sort -u data-shell/data/salmon.txt`  
`$ uniq -c salmon.txt`  
`$ cat animals.txt | head -n 5 | tail -n 3 | sort -r > final.txt`  
  
`$ cut -d, -f 2 animals.txt`  
`$ sed 's/coho/unicorn/g' salmon.txt`  
`$ sed 's+,+/,+g' animals.txt`

# Grep, find

---

- `$ grep "not" haiku.txt` (cat haiku.txt | grep "not")  
`$ grep "The" haiku.txt`  
`$ grep -w "The" haiku.txt`  
`$ grep -v "The" haiku.txt`  
`$ grep -n "it" haiku.txt`  
`$ grep -n -w "the" haiku.txt`  
`$ grep -n -w -i "the" haiku.txt`
  
- `$ find . -type d`  
`$ find . -type f`  
`$ find . -name "*.txt"`
  
- `$ wc -l $(find . -name '*.pdb')`  
`$ grep "FE" $(find . -name '*.pdb')`

# Awk, paste

---

- ```
$ cat cubane.pdb | awk '{print $4,$5}'  
$ cat cubane.pdb | awk '{a+=$6; print a}'  
$ cat cubane.pdb | awk '{a+=$6} END {print a}'  
$ awk -F',' '{ print $1 }' ../animals.txt  
$ paste yearpolarity.txt results.txt
```
- Floating point calculations with awk:
Calculate $ADK_{pi} = A(D2K_{pi}) - A(DKK)$ with its error.
 $A(D2K_{pi})$ and $A(DKK)$ from fit log files D2KK.txt and D2Kpi.txt
("A_sigD_blind")

```
$ paste <(grep "A_sigD" D2Kpi.txt | awk '{print $3,$5}') <(grep  
"A_sigD" D2KK.txt | awk '{print $3,$5}') | awk '{print $1-  
$3,sqrt($2*$2+$4*$4)}'
```

ROOT Bash

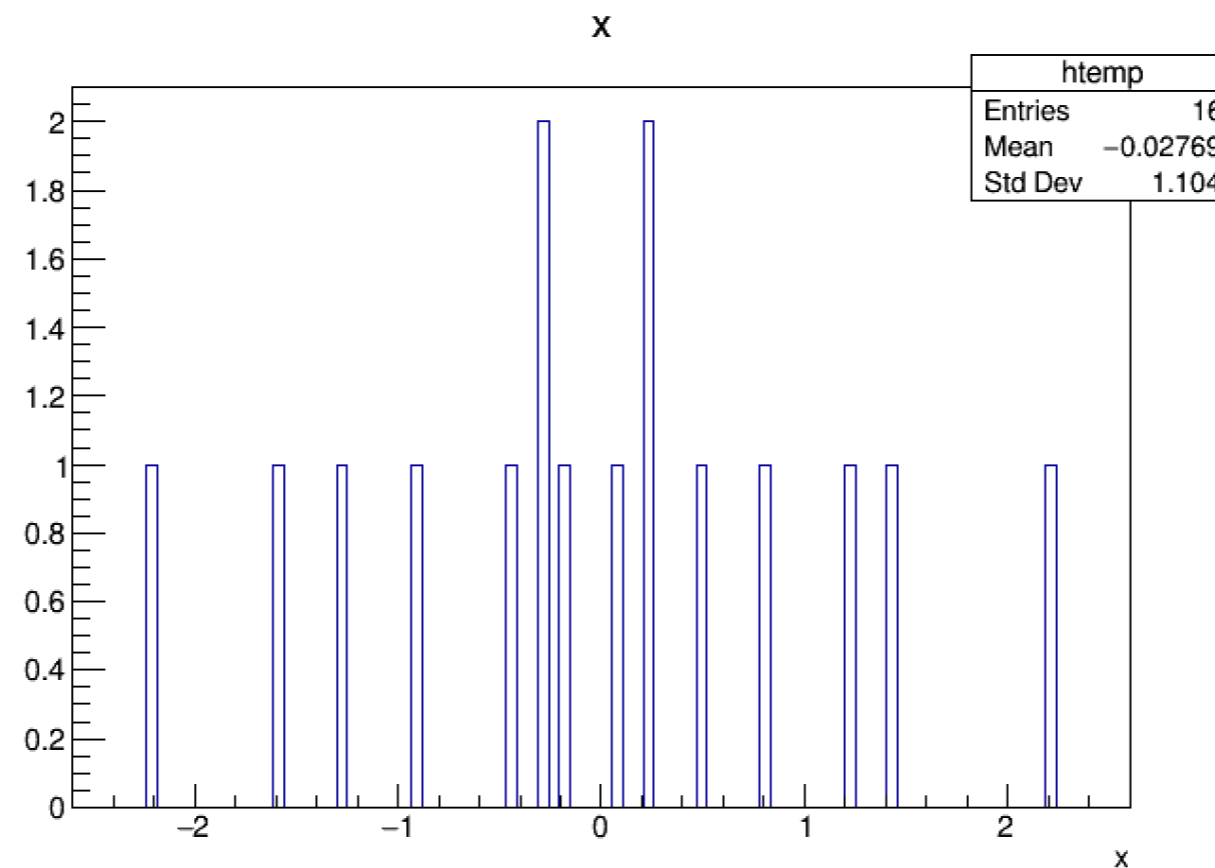
- ```
$ cat cubane.pdb | awk '{print $5,$6,$7}' | grep "\." | grep " " > inputfile.txt
```

```
$ root -l
```

```
$ TTree*a = new TTree("a","a")
```

```
$ a->ReadFile("inputfile.txt","x:y:z")
```

```
$ a->Draw("x")
```





# EMACS shortcuts

---

- `$ emacs -nw script.sh` [no window mode]

Lots of options that can be settled in your `~/.emacs` :

Eg. `/afs/cern.ch/user/s/smaccoli/public/.emacs`

Some useful commands:

- non standard cut and paste (ctrl-W, ctrl-Y)
- standard cut, copy and paste (ctrl-X ctrl-C ctrl-V) [optional]
- search (ctrl-S)
- substitute string (esc-%) then yes(y), no(n) or all(!)
- autocomplete (M-/) [optional]
- newline (ctrl-QJ)

# Scripts

---

- \$ bash script.sh

```
$ for filename in basilisk.dat unicorn.dat
do
 head -n 3 $filename
done
```

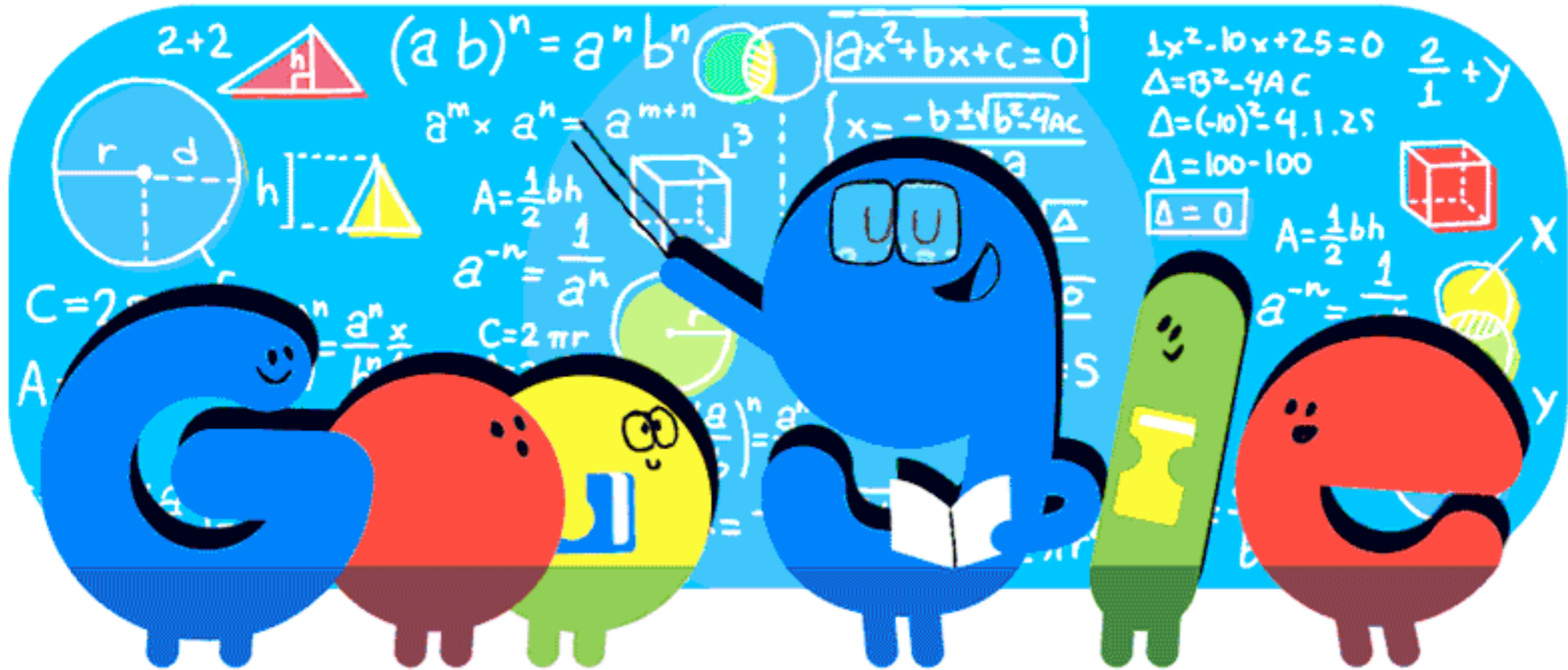
```
$ for file in $(ls -d *.txt); do echo $file; done
```

```
$ for year in 15 16 17 18; do echo $year; done
```

```
$ for year in $(seq 15 18); do echo $year; done
```

```
$ for year in $(seq 15 18); do
 for pol in $(echo "Up" "Dw"); do
 echo $year $pol;
 root -l -b -q myscript.cpp '\("${year}\",\("${pol}")' > log_${year}_${pol}.txt

 condor_submit exec/$year.$pol.sub -batch-name $year$pol
 done
done
```



***Google is your best friend!***