

HPC workshop online

- session 3: Bash scripting (recap) and Torque job management -



<http://hpc.dccn.nl>



helpdesk@fcdonders.ru.nl



DCCN#HPC users

- presentation and demo
 - recap on Bash scripting: Hong Lee
 - Torque job management: Edward Gerrits/Hong Lee
- Q&A
 - post your questions in the Zoom chat

What is a bash script?

- A text file with the "executable" permission
- It may contain
 - Linux commands
 - variables
 - logical blocks (e.g. if-statement, for-loop, case-switch)

First dummy bash script

```
$ touch dummy.sh  
$ chmod +x dummy.sh  
$ ./dummy.sh
```

- empty text file
- make it "executable"
- run it
 - note the `./`
 - the `PATH` environment variable

Add some commands

```
#!/bin/bash

# this is a comment

ls

ls | wc -l

ls | wc -l > newfile.txt
```

- The shebang (`#!`)
- The comment (`#`)
- The linux commands
 - simple command
 - pipeline, I/O redirect

Variables

```
#!/bin/bash

# declare variables
dir1=/tmp
dir2=${HOME}
dir3=$(pwd)
dir4=${1}

# access variables
ls ${dir4}
```

- declare variables with values using `var=val`
 - space around `=` is NOT allowed
 - `val` may be:
 - literal
 - another (environment) variable
 - output of a command
 - argument(s) to the script
- access variables with `${var}`

if-statement

```
#!/bin/bash

# get number of files in a given
# directory
nof=$(ls ${1} | wc -l)

if [ ${nof} -lt 10 ]; then
    echo "${1} has less than 10 files"
else
    echo "${1} has more than 10 files"
fi
```

- test condition enclosed by `[]`
 - sometimes `[[]]`
- test operators
 - number: `-lt` `-le` `-gt` `-ge`
 - string: `==` `>=` `<=` `!=`
 - other

for-loop

```
#!/bin/bash

# print 1~5
for i in 1 2 3 4 5; do
    echo ${i}
done

# print script arguments each per line
for arg in ${@}; do
    echo ${arg}
done

# list files in a given directory,
# each per line
for f in $(ls ${1}); do
    echo ${f}
done
```

- repeat code-block over a list of items:
 - space/line-separated items