

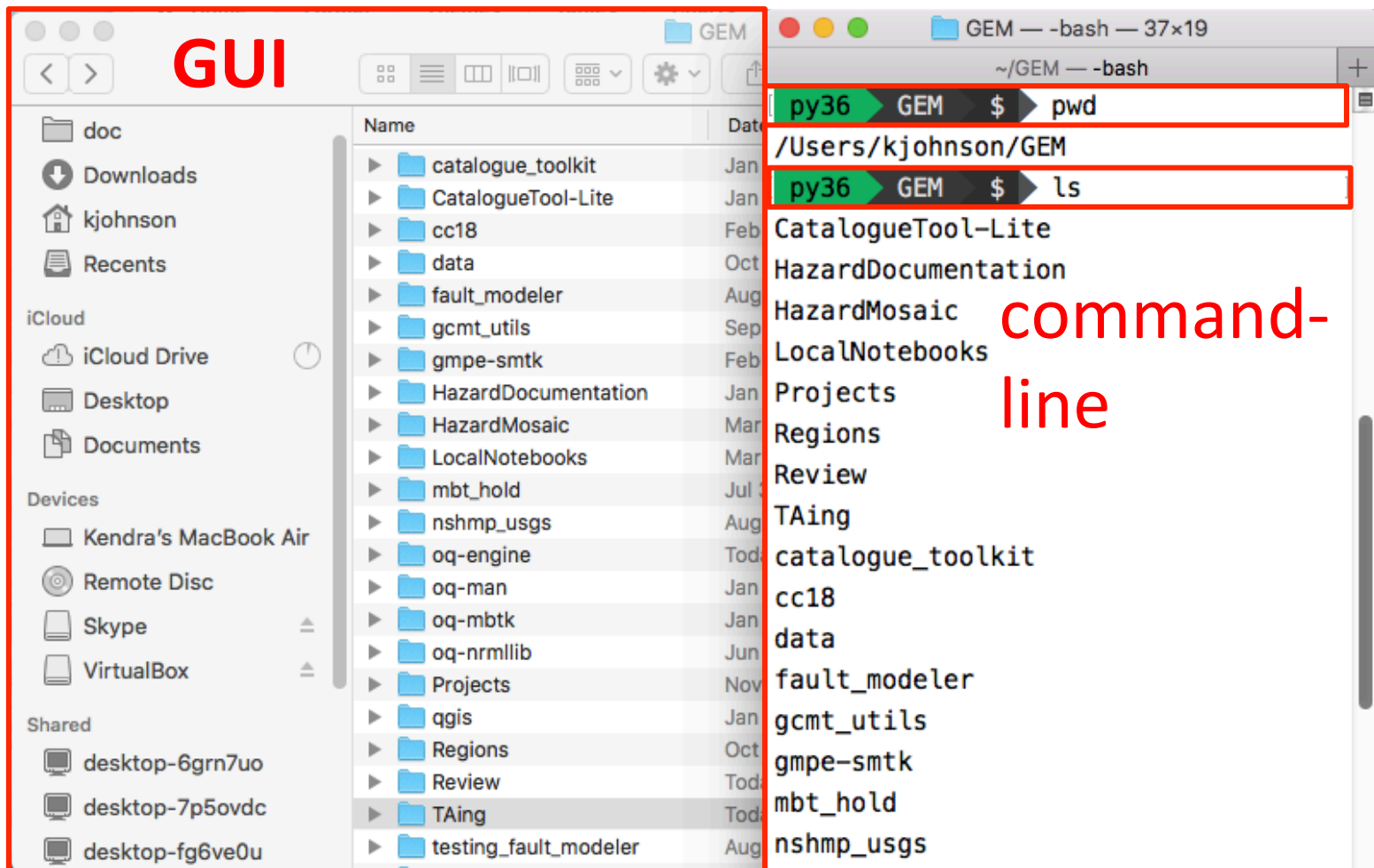
Tutorial 1: Introduction to Terminal

```
oq-engine — -bash — 90x23
~/GEM/oq-engine — -bash
Last login: Mon Apr 1 17:46:30 on ttys000
py36 kjohnson Kendras-MacBook-Air ~ $ pwd
/Users/kjohnson
py36 kjohnson Kendras-MacBook-Air ~ $ cd GEM/oq-engine
py36 kjohnson Kendras-MacBook-Air oq-engine $ pwd
/Users/kjohnson/GEM/oq-engine
py36 kjohnson Kendras-MacBook-Air oq-engine $ echo "I <3 OpenQuake"
I <3 OpenQuake
py36 kjohnson Kendras-MacBook-Air oq-engine $ ls
CONTRIBUTORS.txt          openquake
LICENSE                   openquake.engine.egg-info
MANIFEST.in               packager.sh
README.md                 pylintrc
__pycache__               pytest.ini
_config.yml               requirements-py36-linux64.txt
bin                       requirements-py36-macos.txt
debian                   rpm
demos                    setup.cfg
doc                      setup.py
docker                   tox.ini
helpers                  utils
py36 kjohnson Kendras-MacBook-Air oq-engine $
```

08/04/2019

What is the Linux “Terminal”

- A *terminal emulator* lets you interact with the *shell*
- The *shell* allows the user to communicate with the operating system by typing in a command-line interface



What can we do with the terminal?

Simple things:

- Navigate directories
- Move, duplicate, and rename files
- Search for files or file contents
- Edit text

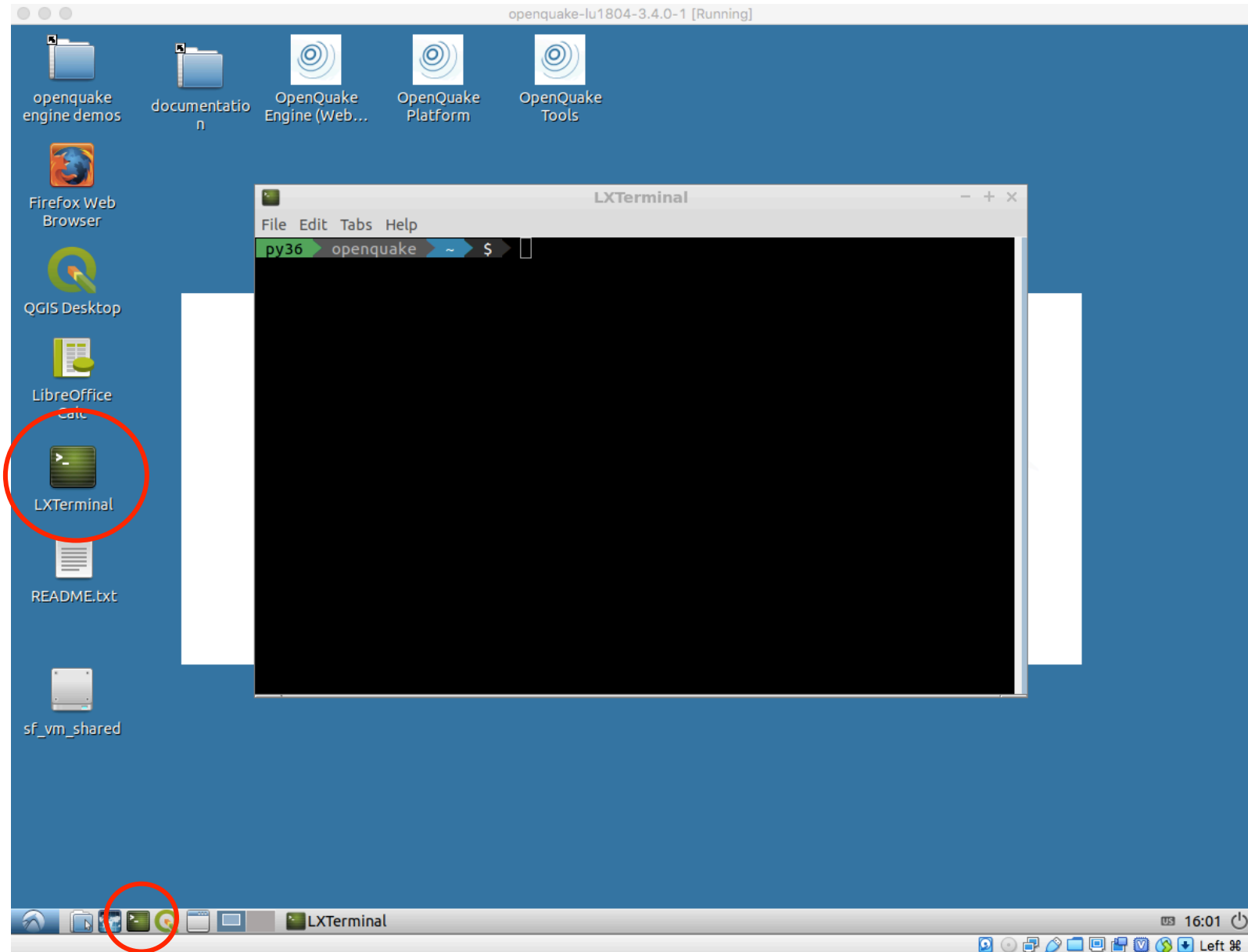
More complicated things:

- Connect with remote servers
- Install modules, software, and development tools
- Use version management for codes (e.g., Github)
- Streamline calculations
- Much more!

How will we use the terminal for class?

- Run the OpenQuake engine
- Run simple python scripts to visualize OQ inputs and outputs
- We will use only the Linux terminal
 - The basics in Mac OS are mostly the same
 - Windows is very different

Opening the terminal



Directories

pwd

Print working directory

ls [-options]

List directory contents

cd <directory>

Navigate to <directory>

cd ..

Go “up” one directory

mkdir <directory>

Create new dir. <directory>

<https://www.git-tower.com/blog/command-line-cheat-sheet/>



DIRECTORIES
\$ pwd
Display path of current working directory
\$ cd <directory>
Change directory to <directory>
\$ cd ..
Navigate to parent directory
\$ ls
List directory contents
\$ ls -la
List detailed directory contents, including hidden files
\$ mkdir <directory>
Create new directory named <directory>

OUTPUT
\$ cat <file>
Output the contents of <file>
\$ less <file>
Output the contents of <file> using the less command (which supports pagination etc.)
\$ head <file>
Output the first 10 lines of <file>
\$ <cmd> > <file>
Direct the output of <cmd> into <file>
\$ <cmd> >> <file>
Append the output of <cmd> to <file>
\$ <cmd1> <cmd2>
Direct the output of <cmd1> to <cmd2>
\$ clear
Clear the command line window

FILES
\$ rm <file>
Delete <file>
\$ rm -r <directory>
Delete <directory>
\$ rm -f <file>
Force-delete <file> (add -r to force-delete a directory)
\$ mv <file-old> <file-new>
Rename <file-old> to <file-new>
\$ mv <file> <directory>
Move <file> to <directory> (possibly overwriting an existing file)
\$ cp <file> <directory>
Copy <file> to <directory> (possibly overwriting an existing file)
\$ cp -r <directory1> <directory2>
Copy <directory1> and its contents to <directory2> (possibly overwriting files in an existing directory)
\$ touch <file>
Update file access & modification time (and create <file> if it doesn't exist)

PERMISSIONS
\$ chmod 755 <file>
Change permissions of <file> to 755
\$ chmod -R 600 <directory>
Change permissions of <directory> (and its contents) to 600
\$ chown <user>:<group> <file>
Change ownership of <file> to <user> and <group> (add -R to include a directory's contents)

SEARCH
\$ find <dir> -name "<file>"
Find all files named <file> inside <dir> (use wildcards [*] to search for parts of filenames, e.g. "file.*")
\$ grep "<text>" <file>
Output all occurrences of <text> inside <file> (add -i for case-insensitivity)
\$ grep -rL "<text>" <dir>
Search for all files containing <text> inside <dir>

NETWORK
\$ ping <host>
Ping <host> and display status
\$ whois <domain>
Output whois information for <domain>
\$ curl -O <url/to/file>
Download <file> (via HTTP[S] or FTP)
\$ ssh <username>@<host>
Establish an SSH connection to <host> with user <username>
\$ scp <file> <user>@<host>:/remote/path
Copy <file> to a remote <host>

PROCESSES
\$ ps ax
Output currently running processes
\$ top
Display live information about currently running processes
\$ kill <pid>
Quit process with ID <pid>

Files

rm <file>

Delete <file>

rm -r <directory>

Delete <directory>

mv <file-old> <file-new>

Rename <file-old> as <file-new>

cp <file> <directory>

Copy <file> to <directory>

Important! This command allows override of <file>!

How would we copy a directory?

<https://www.git-tower.com/blog/command-line-cheat-sheet/>



DIRECTORIES

```
$ pwd
Display path of current working directory

$ cd <directory>
Change directory to <directory>

$ cd ..
Navigate to parent directory

$ ls
List directory contents

$ ls -la
List detailed directory contents, including hidden files

$ mkdir <directory>
Create new directory named <directory>
```

OUTPUT

```
$ cat <file>
Output the contents of <file>

$ less <file>
Output the contents of <file> using the less command (which supports pagination etc.)

$ head <file>
Output the first 10 lines of <file>

$ <cmd> > <file>
Direct the output of <cmd> into <file>

$ <cmd> >> <file>
Append the output of <cmd> to <file>

$ <cmd1> | <cmd2>
Direct the output of <cmd1> to <cmd2>

$ clear
Clear the command line window
```

FILES

```
$ rm <file>
Delete <file>

$ rm -r <directory>
Delete <directory>

$ rm -f <file>
Force-delete <file> (add -r to force-delete a directory)

$ mv <file-old> <file-new>
Rename <file-old> to <file-new>

$ mv <file> <directory>
Move <file> to <directory> (possibly overwriting an existing file)

$ cp <file> <directory>
Copy <file> to <directory> (possibly overwriting an existing file)

$ cp -r <directory1> <directory2>
Copy <directory1> and its contents to <directory2> (possibly overwriting files in an existing directory)

$ touch <file>
Update file access & modification time and create <file> if it doesn't exist
```

PERMISSIONS

```
$ chmod 755 <file>
Change permissions of <file> to 755

$ chmod -R 600 <directory>
Change permissions of <directory> (and its contents) to 600

$ chown <user>:<group> <file>
Change ownership of <file> to <user> and <group> (add -R to include a directory's contents)
```

SEARCH

```
$ find <dir> -name "<file>"
Find all files named <file> inside <dir> (use wildcards [*] to search for parts of filenames, e.g. "file.*")

$ grep "<text>" <file>
Output all occurrences of <text> inside <file> (add -i for case-insensitivity)

$ grep -rl "<text>" <dir>
Search for all files containing <text> inside <dir>
```

NETWORK

```
$ ping <host>
Ping <host> and display status

$ whois <domain>
Output whois information for <domain>

$ curl -O <url/to/file>
Download <file> (via HTTP[S] or FTP)

$ ssh <username>@<host>
Establish an SSH connection to <host> with user <username>

$ scp <file> <user>@<host>:/remote/path
Copy <file> to a remote <host>
```

PROCESSES

```
$ ps ax
Output currently running processes

$ top
Display live information about currently running processes

$ kill <pid>
Quit process with ID <pid>
```

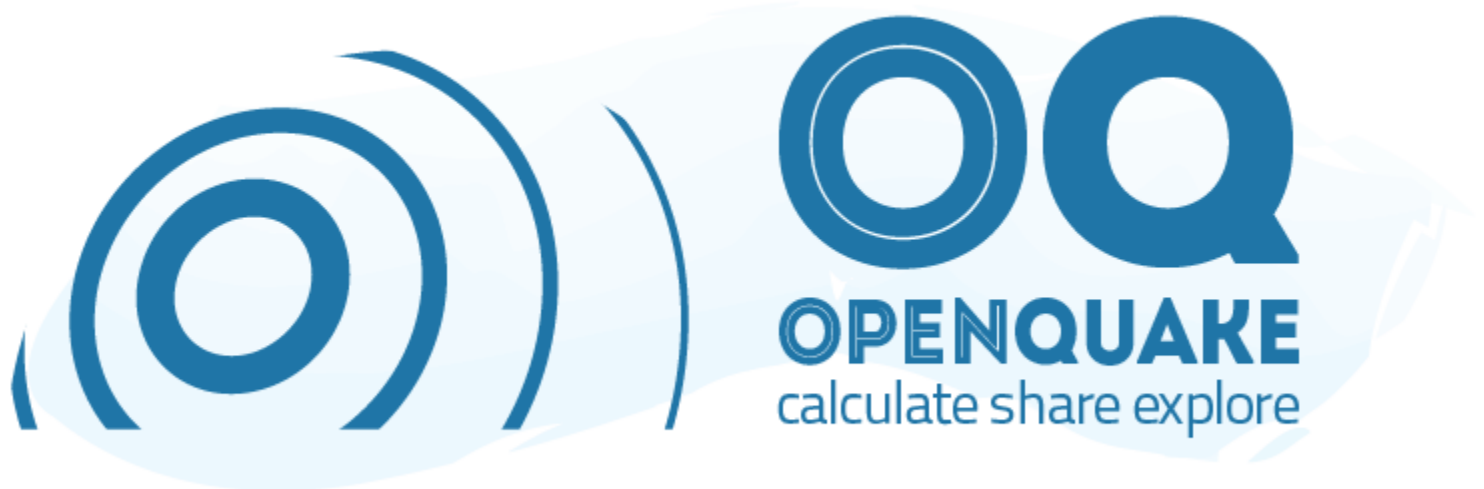

Other useful things

- Press “tab” to see options or complete file names
- * includes all possibilities
 - **ls *** shows the contents of all subdirectories
 - **grep ‘fault’ *.xml** finds instances of the word “fault” in all the xml files
- Use . to refer to “here” (the current working directory)
 - **mv ~/Documents/file.txt .** moves file.txt to the current directory (~ indicates the home directory)
- Use / to refer to the root directory
 - **cd /** takes you to the root directory

Other useful things

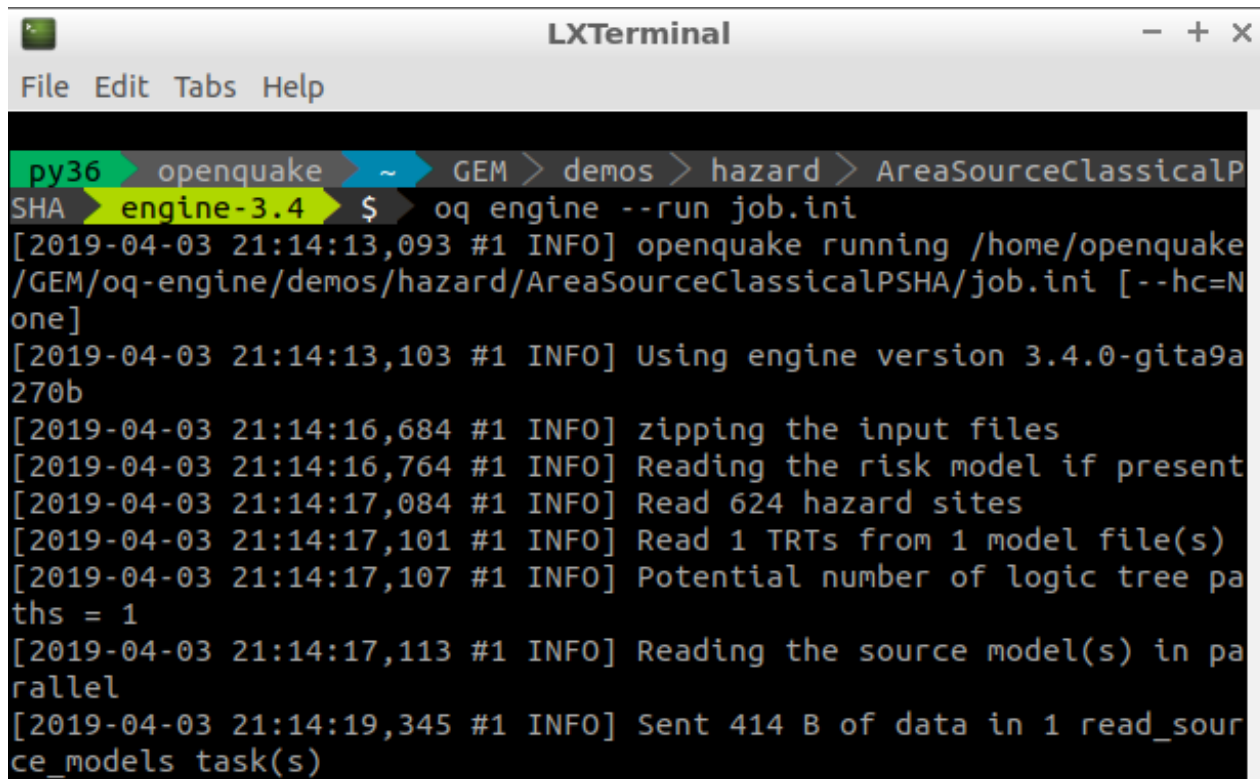
- File or directory names with spaces have a special syntax
 - e.g., `~/Desktop/openquake\ engine\ demos`
- **xdg-open <filename>** or **xdg-open .** opens the file or the directory in File Manager
- Arrow up/down to review command history

Tutorial 2: Introduction to Running OpenQuake (OQ)



Running OQ

This tutorial demonstrates how to run a basic OpenQuake calculation using the command line

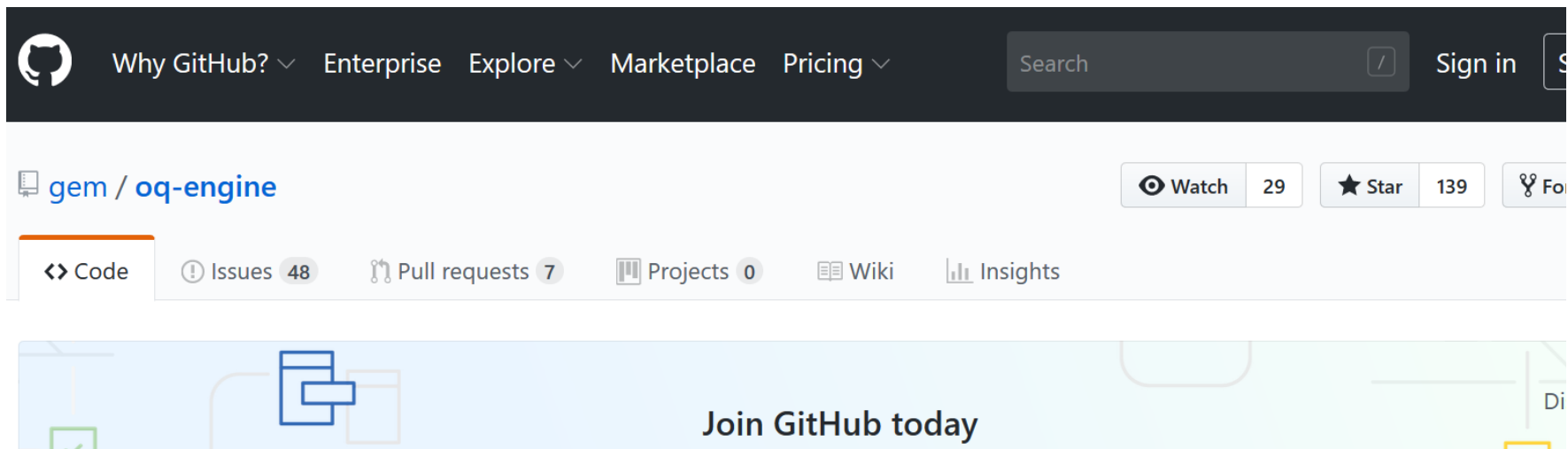


```
LXTerminal
File Edit Tabs Help

py36 openquake ~ GEM > demos > hazard > AreaSourceClassicalP
SHA engine-3.4 $ oq engine --run job.ini
[2019-04-03 21:14:13,093 #1 INFO] openquake running /home/openquake
/GEM/oq-engine/demos/hazard/AreaSourceClassicalPSHA/job.ini [--hc=N
one]
[2019-04-03 21:14:13,103 #1 INFO] Using engine version 3.4.0-gita9a
270b
[2019-04-03 21:14:16,684 #1 INFO] zipping the input files
[2019-04-03 21:14:16,764 #1 INFO] Reading the risk model if present
[2019-04-03 21:14:17,084 #1 INFO] Read 624 hazard sites
[2019-04-03 21:14:17,101 #1 INFO] Read 1 TRTs from 1 model file(s)
[2019-04-03 21:14:17,107 #1 INFO] Potential number of logic tree pa
ths = 1
[2019-04-03 21:14:17,113 #1 INFO] Reading the source model(s) in pa
rallel
[2019-04-03 21:14:19,345 #1 INFO] Sent 414 B of data in 1 read_sour
ce_models task(s)
```

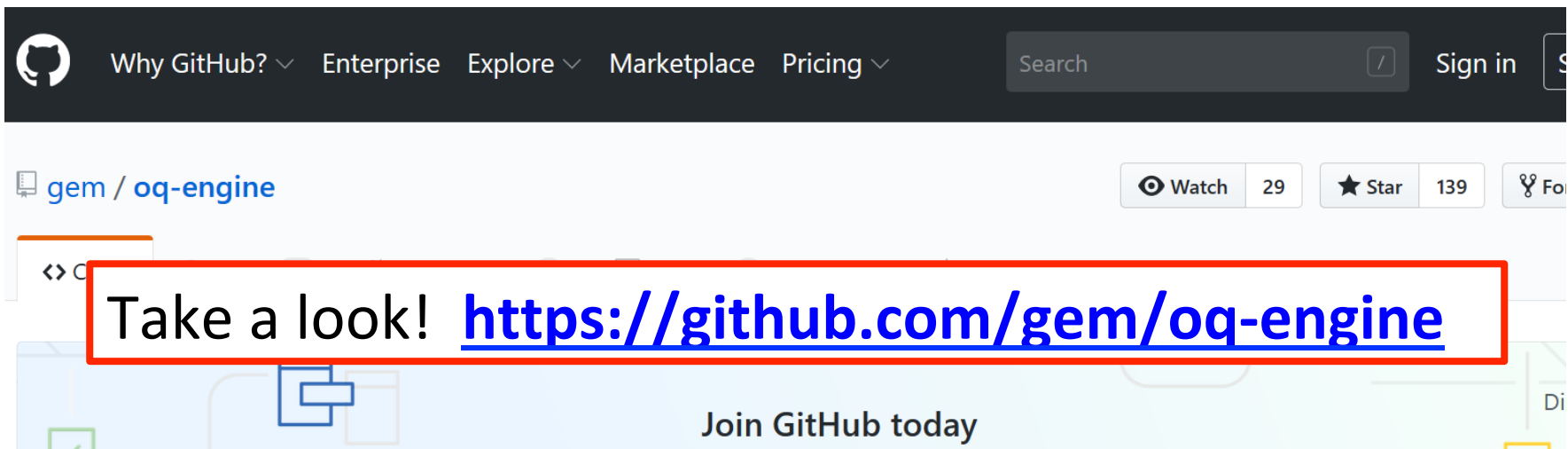
OpenQuake

- Seismic hazard and risk calculation software developed by the Global Earthquake Model (GEM) Foundation.
- Written in Python (currently ~80k lines)
- Code is open-source, available for everyone to see.



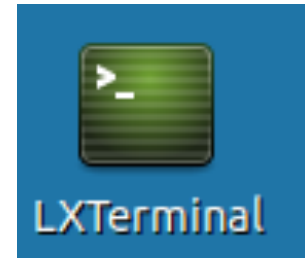
OpenQuake

- Seismic hazard and risk calculation software developed by the Global Earthquake Model (GEM) Foundation.
- Written in Python (currently ~80k lines)
- Code is open-source, available for everyone to see.



Find the the OQ input files

We will start by running a demo calculation. Hazard and risk demos are located in the 'demos' folder in your Virtualbox.



In the terminal, navigate to the demos using 'cd':

```
cd /home/openquake/GEM/demos/hazard
```

Find the the OQ input files

We will run the “AreaSourceClassicalPSHA” demo:

```
cd AreaSourceClassicalPSHA
```

Trick: use ‘tab’ button to automatically finish the folder name after typing the first few letters

List the input files

List the input files using 'ls':

ls

List the input files

List the input files using 'ls':

ls

You will see 4 files (in addition to the README). These are the 4 files required to run a hazard calculation:

```
gmpe_logic_tree.xml  
job.ini  
README.txt  
source_model_logic_tree.xml  
source_model.xml
```

ground motion
configuration file
seismic source

See how to run each command

We will use the “oq engine” command to run hazard calculations. Get information about this command using:

```
oq help <command>
```

```
oq help engine
```

Run a hazard calculation

Run a hazard calculation using the 'oq engine' command:

```
oq engine --run job.ini --exports=csv
```

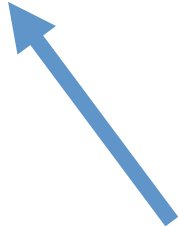
Command



Configuration
file



Immediately save
results as csv
(text) format



Let's take a look at the results..

Go to the folder with the out results:

```
cd /tmp
```

What's inside?

```
ls
```

Let's take a look at the results..

Go to the folder with the out results:

`cd /tmp`

What's inside?

`ls`

```
l py36 openquake tmp $ ls
config-err-1NqPUF
hazard_curve-mean-PGA_4.csv
hazard_curve-mean-PGV_4.csv
'hazard_curve-mean-SA(0.025)_4.csv'
'hazard_curve-mean-SA(0.05)_4.csv'
'hazard_curve-mean-SA(0.1)_4.csv'
'hazard_curve-mean-SA(0.2)_4.csv'
'hazard_curve-mean-SA(0.5)_4.csv'
'hazard_curve-mean-SA(1.0)_4.csv'
'hazard_curve-mean-SA(2.0)_4.csv'
hazard_map-mean_4.csv
hazard_uhs-mean_4.csv
```

Let's take a look at the results..

more hazard_curve-mean-PGA_<id>.csv

```
# mean, investigation_time=50.0, imt="PGA", checksum=1671584369
lon,lat,depth,poe-0.005,poe-0.007,poe-0.0098,poe-0.0137,poe-0.0192,poe-0.0269,poe-
0.0376,poe-0.0527,poe-0.0738,poe-0.103,poe-0.145,poe-0.203,poe-0.284,poe-0.397,poe-
-0.556,poe-0.778,poe-1.09,poe-1.52,poe-2.13
-1.00005,-0.01933,0.00000,9.999832E-01,9.997750E-01,9.967527E-01,9.686500E-01,8.39
2978E-01,5.686073E-01,2.847404E-01,1.056937E-01,3.005412E-02,6.456052E-03,8.671796
E-04,4.718129E-05,0.000000E+00,0.000000E+00,0.000000E+00,0.000000E+00,0.000000E+00
,0.000000E+00,0.000000E+00
-1.00005,0.07060,0.00000,9.999751E-01,9.996504E-01,9.951093E-01,9.569483E-01,8.027
575E-01,5.177675E-01,2.468916E-01,8.780215E-02,2.394965E-02,4.858518E-03,5.896870E
-04,2.256332E-05,0.000000E+00,0.000000E+00,0.000000E+00,0.000000E+00,0.000000E+00,
0.000000E+00,0.000000E+00
-1.00005,-0.10926,0.00000,9.999875E-01,9.998394E-01,9.976490E-01,9.758000E-01,8.64
8527E-01,6.085742E-01,3.170958E-01,1.219225E-01,3.581845E-02,8.021189E-03,1.158783
E-03,7.793093E-05,0.000000E+00,0.000000E+00,0.000000E+00,0.000000E+00,0.000000E+00
,0.000000E+00,0.000000E+00
-1.00005,0.16053,0.00000,9.999592E-01,9.994033E-01,9.921098E-01,9.385205E-01,7.538
059E-01,4.581362E-01,2.064435E-01,6.990276E-02,1.808397E-02,3.418243E-03,3.572737E
-04,7.669055E-06,0.000000E+00,0.000000E+00,0.000000E+00,0.000000E+00,0.000000E+00,
0.000000E+00,0.000000E+00
-1.00005,-0.19920,0.00000,9.999897E-01,9.998721E-01,9.981193E-01,9.798787E-01,8.81
0679E-01,6.365591E-01,3.414713E-01,1.348359E-01,4.055145E-02,9.345791E-03,1.417882
--More--(0%)
```

Your OQ reference for this course

- The OQ manual explains all the input parameters required to run hazard (and risk) calculations, including examples.
- The manual can be found here: <https://docs.openquake.org/manuals/OpenQuake%20Manual%20%28latest%29.pdf>



Your OQ reference for this course

- Mistakes? Anything unclear?
- Add your comments to here, and we'll include your suggestion in future versions of the manual:

<https://docs.google.com/document/d/1OFVbSjIR6YYs1IUTt7FsZXuGjgm3QuxLIHtDLq6na4U/edit>



Cheat Sheet

- A sheet containing the most useful commands will be provided

OpenQuake Cheat Sheet

<> indicates a variable

General:

Check which version of OpenQuake you are running:

```
oq -version
```

Show what commands are available:

```
oq help
```

Get help using a specific command:

```
oq help <command>
```

|

Running OQ:

Run a hazard calculation (without automatic export of results; sufficient for plotting in QGIS)

```
oq engine --run job.ini
```

Show calculations recently computed and their calculation ids

```
oq engine --lhc
```

Export the results of a specific calculation (provide calculation id)

```
oq engine --eos <calc_id>
```

Run a hazard calculation and automatically export to csv

```
oq engine --run job.ini --exports=csv
```

Coming soon

- More details about OQ in next tutorials
 - How the software works
 - The format of the input files
 - How you can create the input files