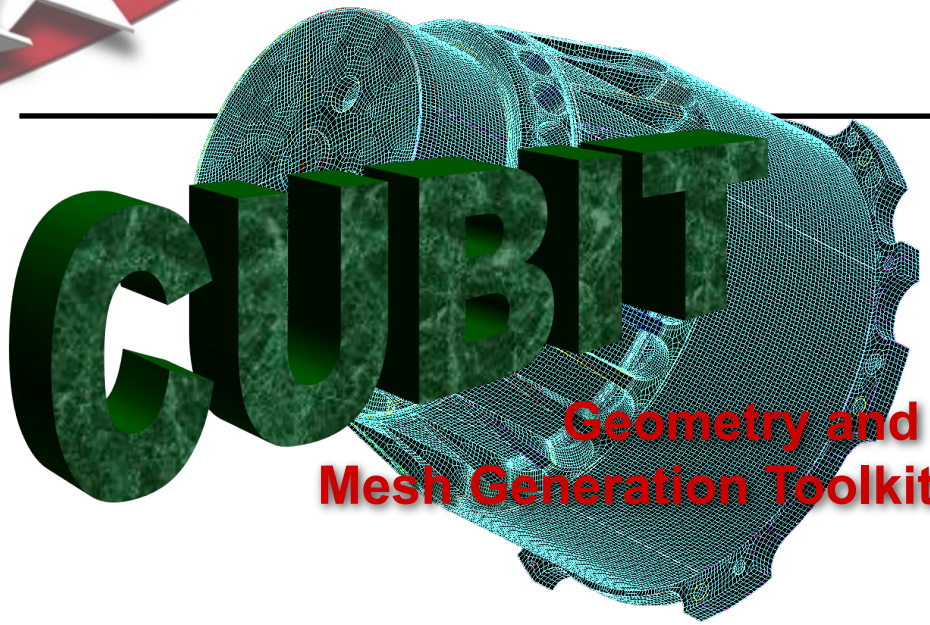


CUBIT Fast-Start Tutorial

16. Cubit Scripting with Python

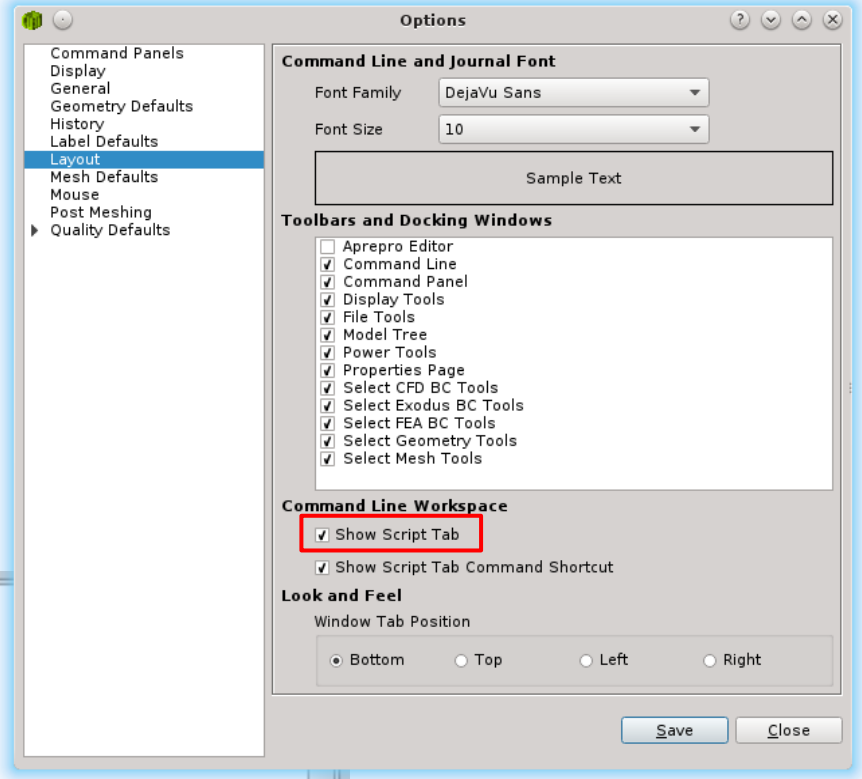
Python

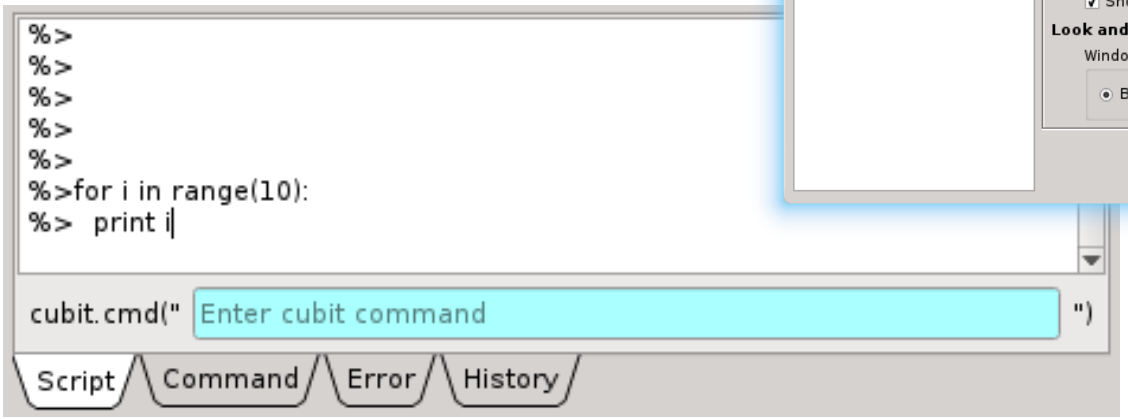
- **Python is a well established, widely accepted scripting language. Its use within the engineering community continues to grow.**
  - Abaqus
  - Paraview
  - PyTrilinos
- **Some useful links to learn Python**
  - Official website: www.python.org
  - Getting started: www.python.org/about/gettingstarted/
  - Tutorial (2.7): docs.python.org/2/tutorial/index.html
  - Reference (2.7): docs.python.org/2/reference/index.html

Enabling the Script Tab from Cubit

Simulation Modeling Sciences

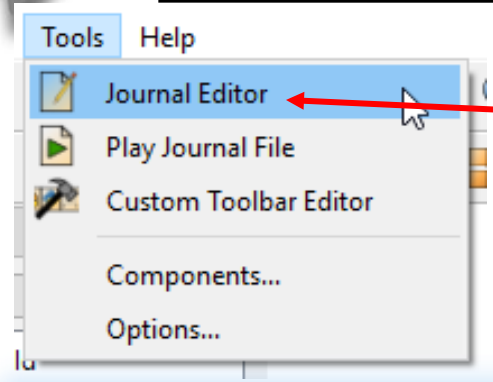
- Select Tools – Options – Layout
- Select “Show Script Tab”
- The script tab will allow direct entry of python commands.



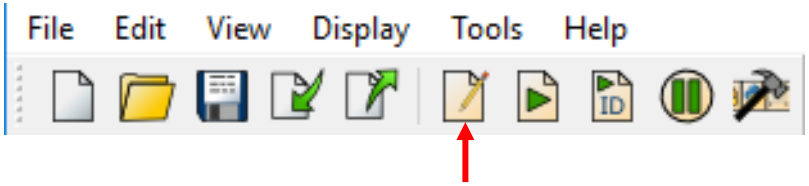


Sandia National Laboratories

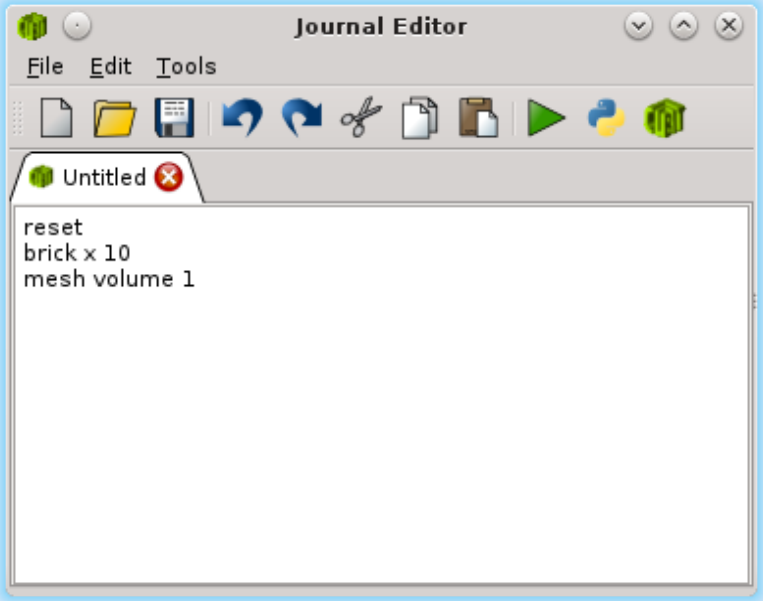
Python Journal Editor



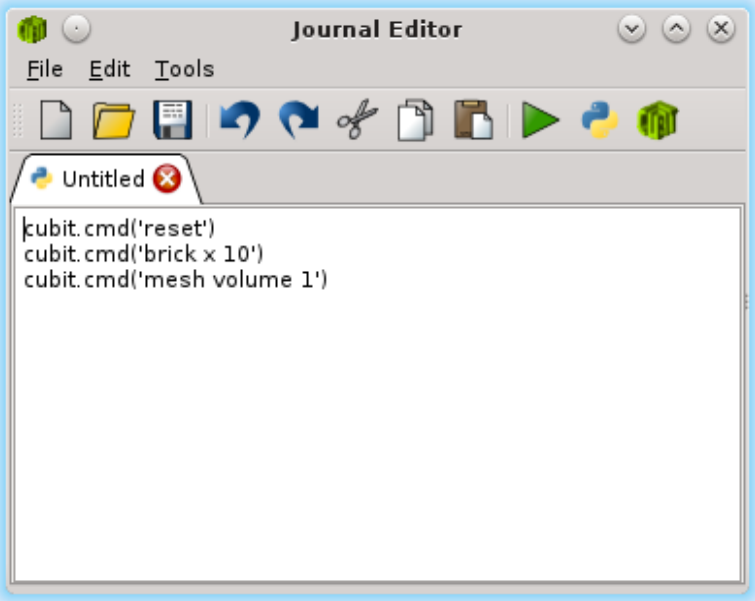
Access from menu



Access from toolbar

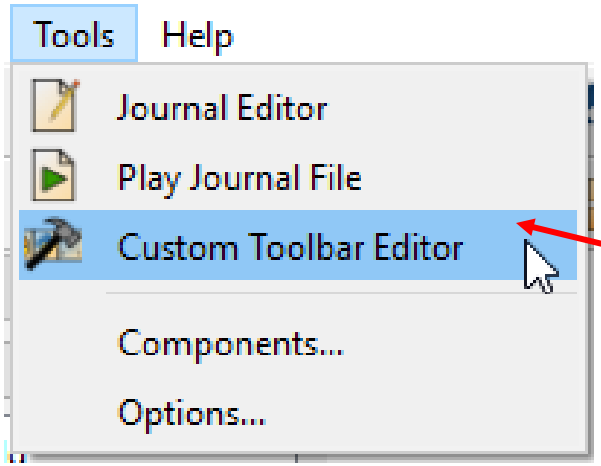


Journal Editor with Cubit commands



Journal Editor with Python commands

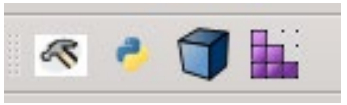
Custom Toolbars



Access editor from menu

Access editor from toolbar

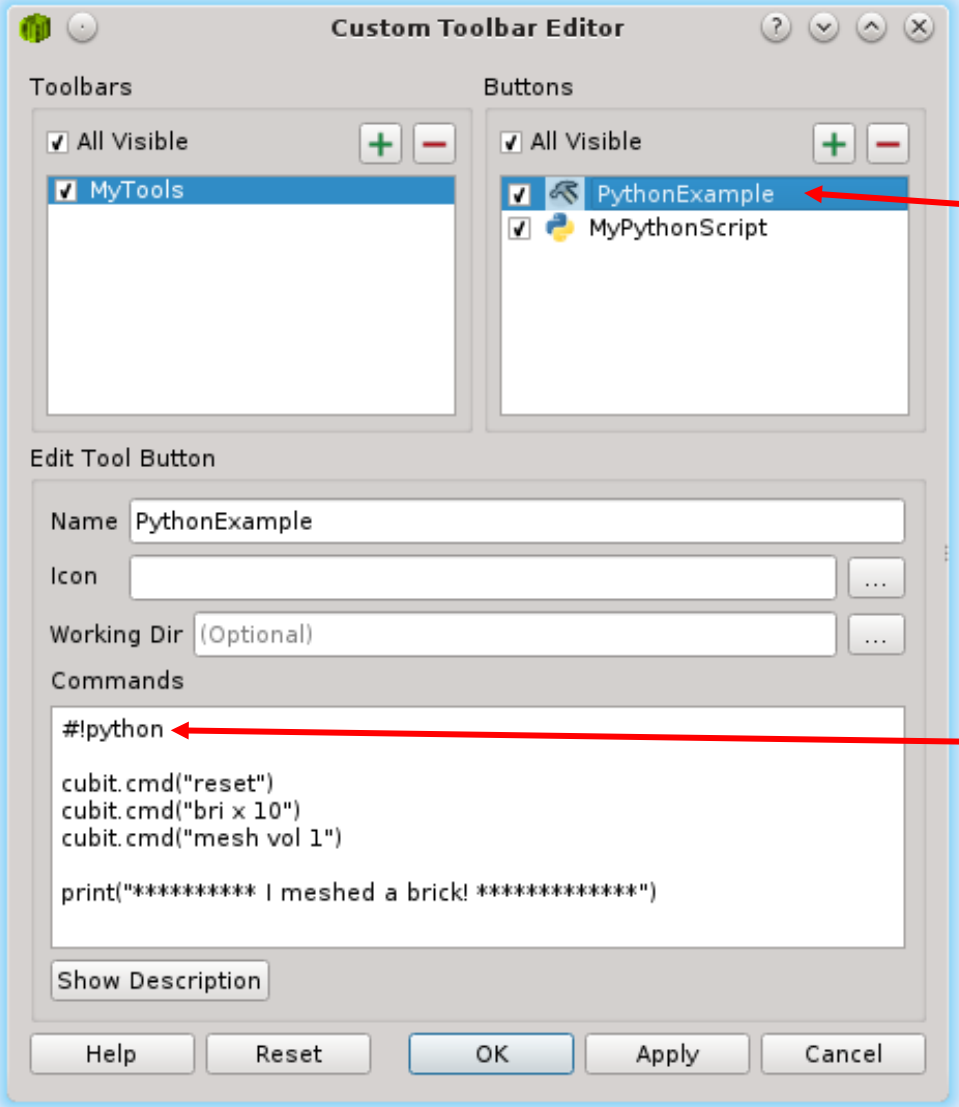
- **Execute a series of Python commands at the click of a button**



Example custom toolbar

- **Execute a Python script**

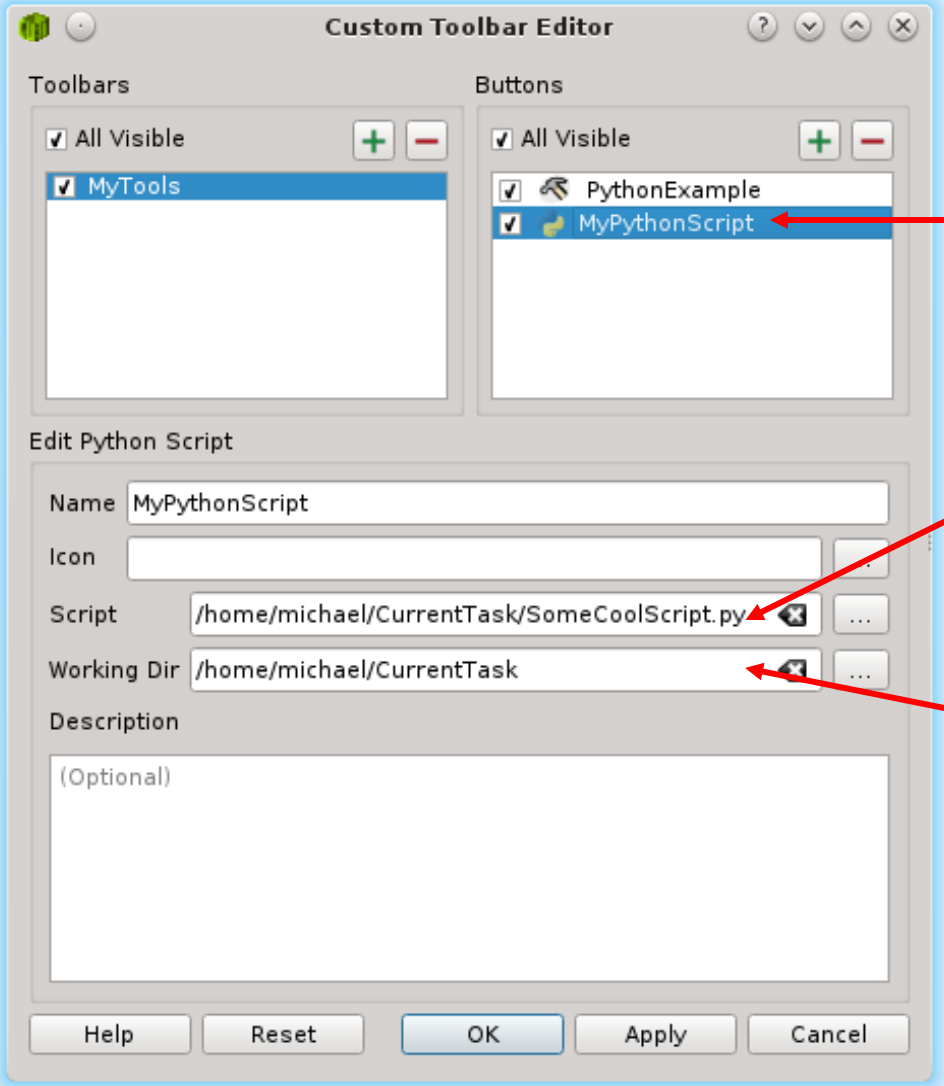
Custom Toolbars



Create a custom tool button and write Python commands in place, just like Cubit commands.

Include “#!python” to tell Cubit to interpret the commands as Python (necessary for this tool only)

Custom Toolbars



Create a Python script button and choose a Python script to run.

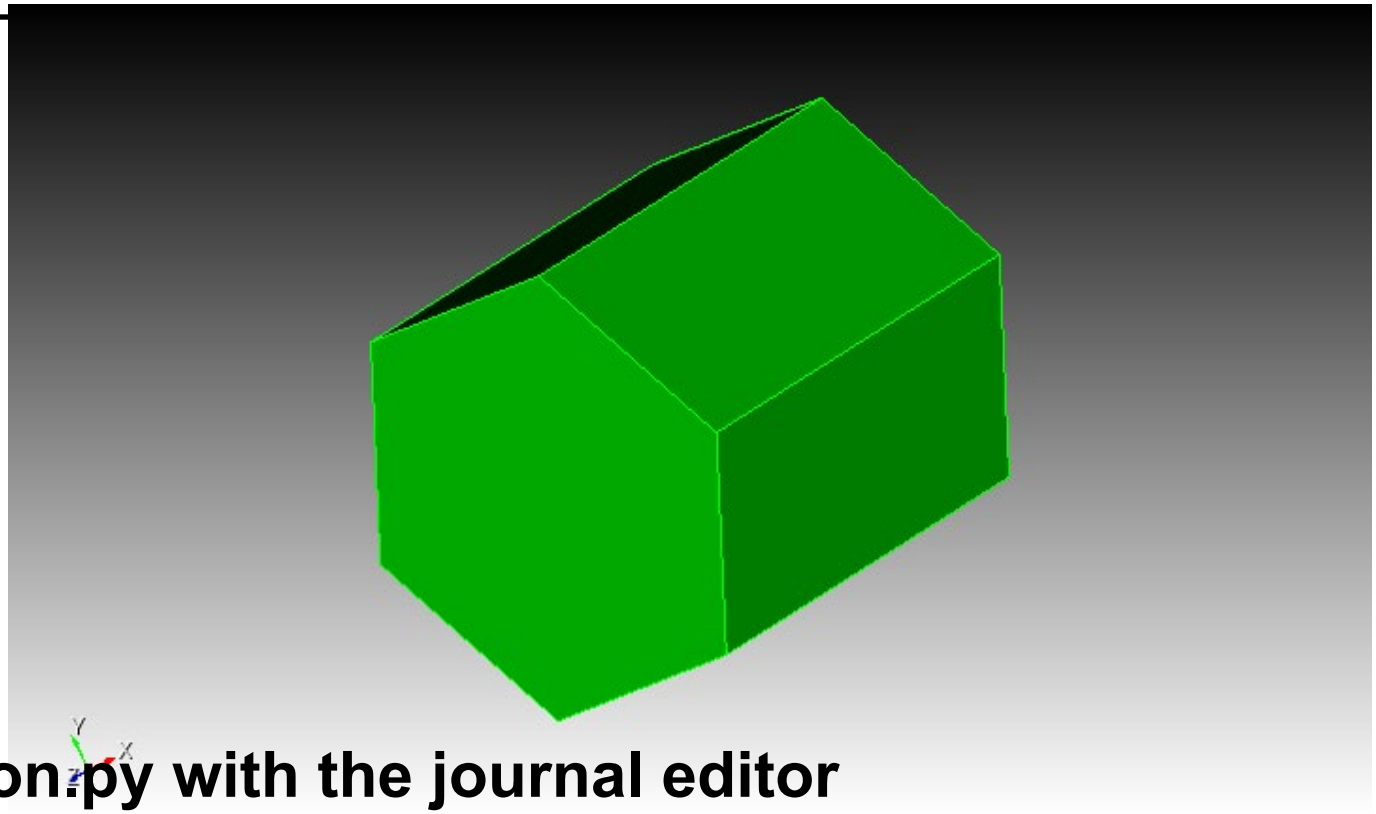
Select the Python script to run.

(Optional) choose a directory from which to run the script.

Cubit Interface

- **Primarily, a query interface into Cubit**
  - `double mesh_size = cubit.get_mesh_size("volume", 22);`
- **Accessible via C++ or python**
- **Change state by using cubit.cmd(" ... ")**
  - `import cubit`
  - `cubit.cmd("create brick x 10 y 10 z 10")`
  - `cubit.cmd("mesh volume 1,3,5")`

Example 1



- Open ngon.py with the journal editor
- Play the script
- Change the parameters
- Play again

Example 2

In the Cubit Journal File (Python) Editor

- **Create a python script to compute and print the minimum shape metric for all volumes. Consider using the following CubitInterface functions**

```
get_entities()
get_volume_hexes()
get_quality_value()
```

Example 2


```
all_vols = cubit.get_entities("volume")
min_quality = 1.0
for vol in all_vols:
    vhexes = cubit.get_volume_hexes(vol)
    for hex in vhexes:
        q = cubit.get_quality_value("hex", hex, "shape")
        if q < min_quality:
            min_quality = q

print 'min quality = ', min_quality
```

Cubit Extended Interface

- **Create "pythonic" objects in Cubit**
- **Reduce (but not eliminate) id issues**

```
bri = cubit.brick(10,5,3)
cyl = cubit.cylinder(12,2,2,2)
vols = cubit.subtract([cyl], [bri])
v = vols[0].volumes()
v[0].mesh()
print dir(v[0])
print v[0].id()
v[0].mesh()
```

Python Help

- **Documentation**
  - Help Manual online or built-in Appendix/Python
- **Python prompt**
  - print dir(object)

Black Box Cubit

- **Cubit can also be run from inside python**
  - Set your environment variable PATH to include the installed Cubit libraries
  - You may also need to set PYTHONPATH to the same place

```
Run Python
import cubit
cubit.init([“”])
cubit.cmd(“brick x 10”)
```

- **This allows you to run Cubit programatically and interact with other tools.**

Example 3

In the native operating system using python 2.7

- **Copy your script from Example 2 to a text editor**
- **Add the ability to import a mesh**
- **Make the script you created above run on the hexes in the mesh and print the result**

Example 3


```
#!python
import sys
# add Cubit libraries to your path
sys.path.append('/Applications/Cubit-15.4/Cubit.app/Contents/MacOS')

import cubit
cubit.init(['cubit','-nojournal'])

cubit.cmd('import mesh geom "mesh.g"')

all_vols = cubit.get_entities("volume")
min_quality = 1.0
for vol in all_vols:
    vhexes = cubit.get_volume_hexes(vol)
    for hex in vhexes:
        q = cubit.get_quality_value("hex", hex, "shape")
        if q < min_quality:
            min_quality = q

print 'min quality = ', min_quality
```

Customization

- **Cubit can support some additions to the GUI**
  - Add new menu items
  - Add new dialogs
  - Cannot currently add new control panels
- **Use PyQt5 - a python interface to Qt**

  from PyQt5 import QtGui

  QtGui.QMessageBox.question(None, “Title”, “Hello”)