



MASSACHUSETTS  
GENERAL HOSPITAL

RADIATION ONCOLOGY

# Python Scripting in 3D Slicer

## Greg Sharp @ MGH

### July 19, 2019



# Disclosures

No conflict of interest

# Disclaimer

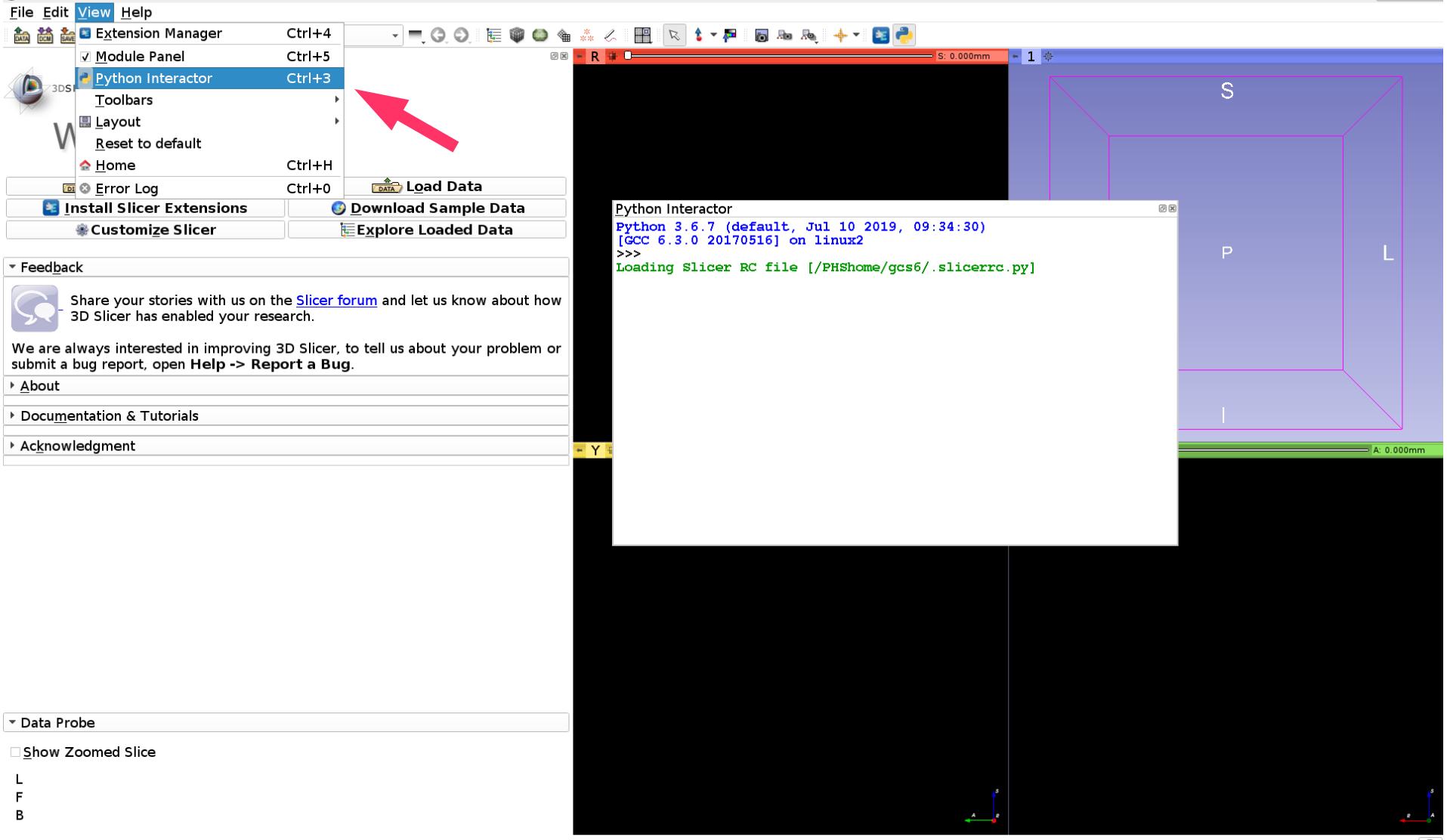
“The Software has been designed for research purposes only and has not been reviewed or approved by the Food and Drug Administration or by any other agency”

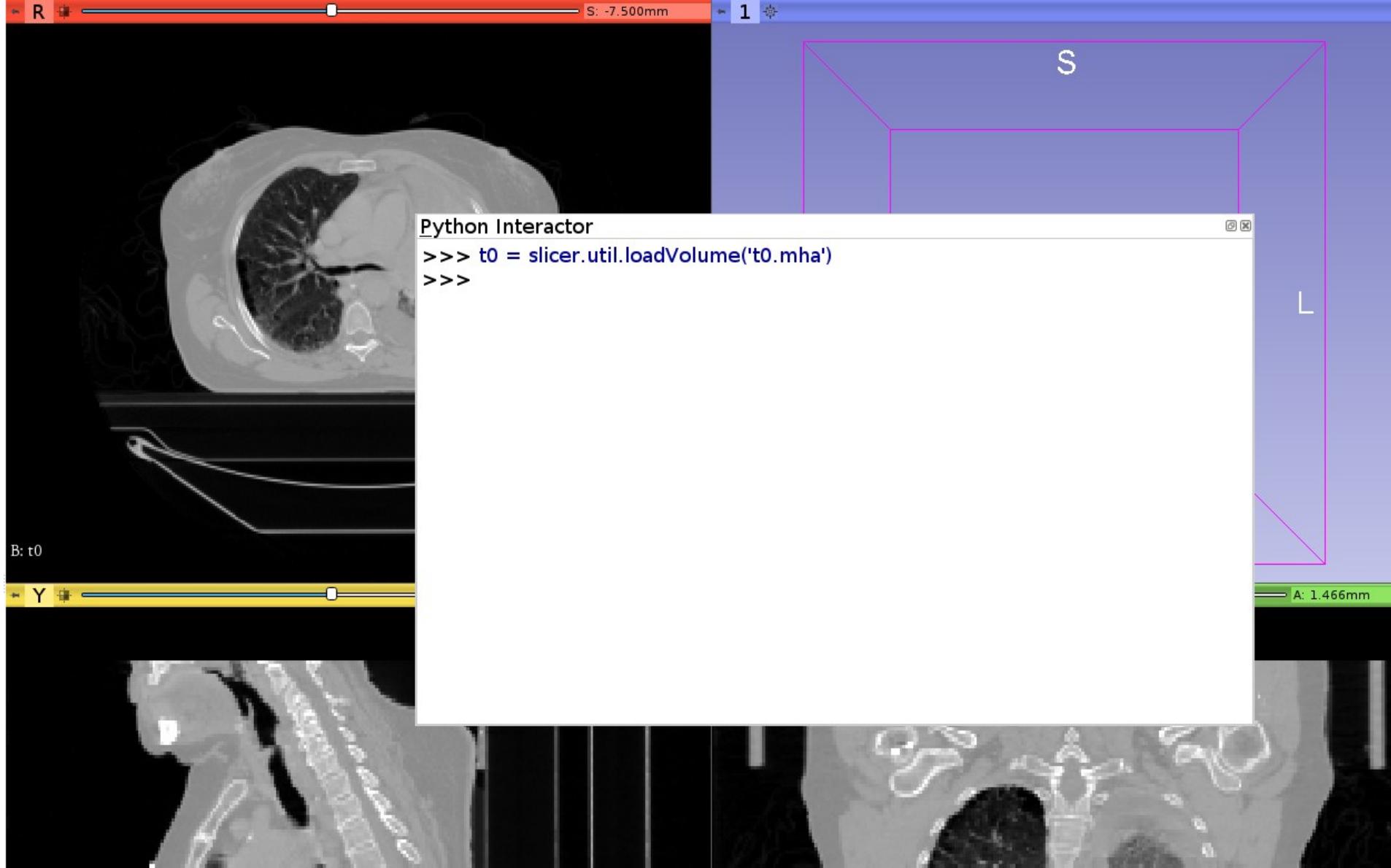
# Development in Slicer

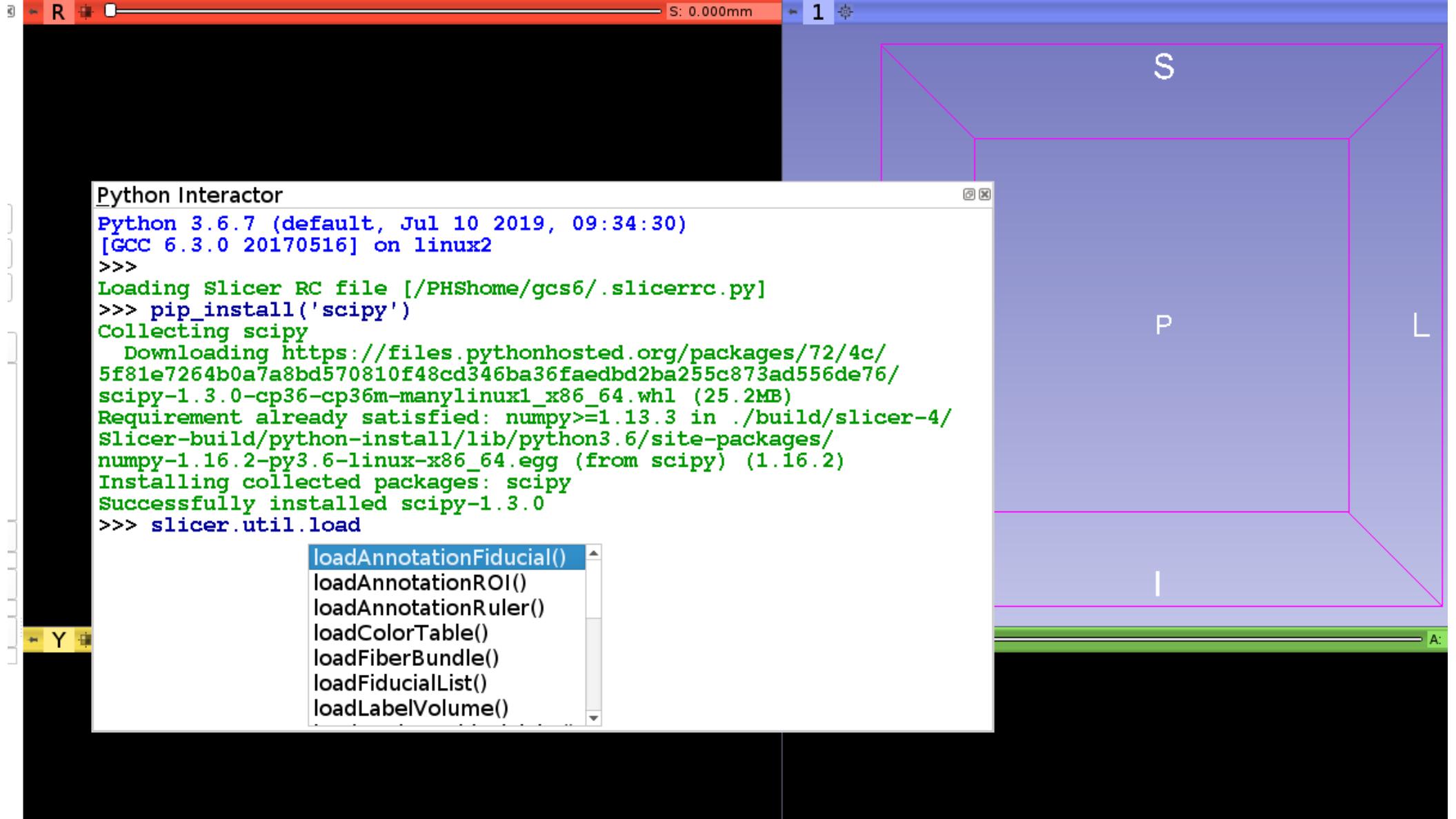
- Slicer modules
  - Command line module
  - Scripted module
  - Loadable module
- Matlab bridge

# Python scripting

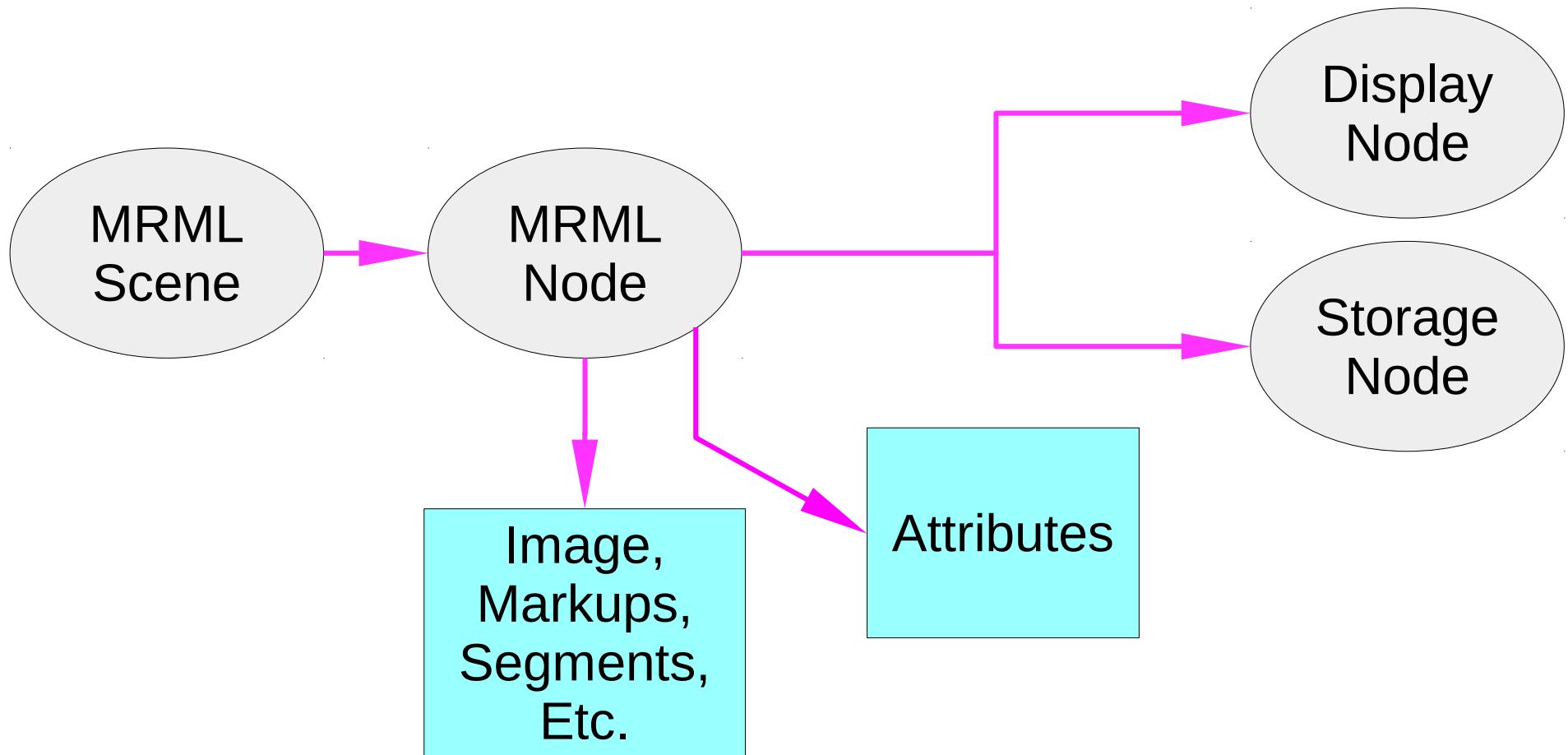
- Python only
- Very easy and powerful
  - You can make a GUI
  - Or just run commands
- Included: numpy, VTK, CTK, PyQt, SimpleITK







# MRML Nodes





File Edit View Help

DATA DCM SAVE Modules: Data



Help & Acknowledgement

Subject hierarchy Transform hierarchy All nodes

Nodes

View1

Red

Yellow

Green

Camera

SubjectHierarchy

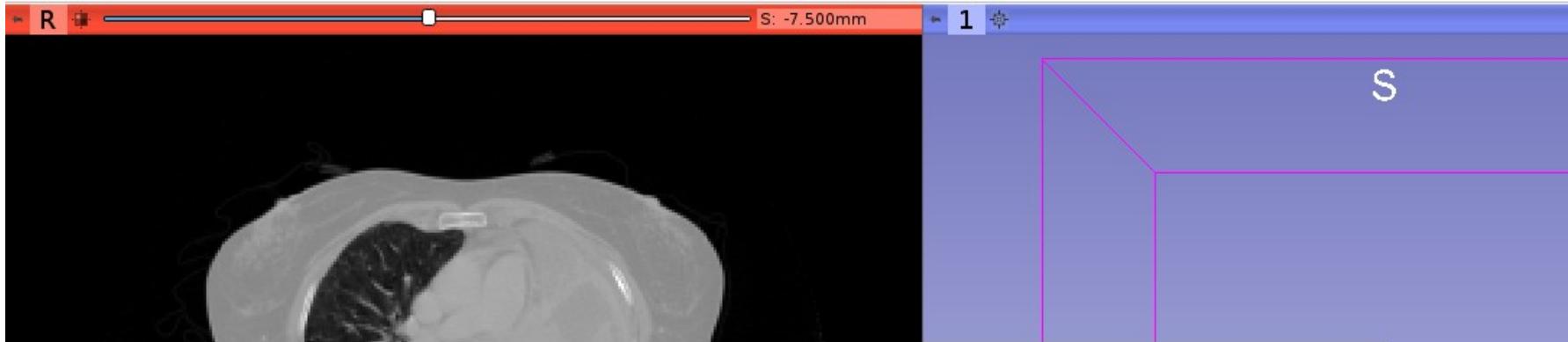
t0



Python Interactor

```
>>> t0 = slicer.util.loadVolume('t0.mha')
>>>
```





### Python Interactor

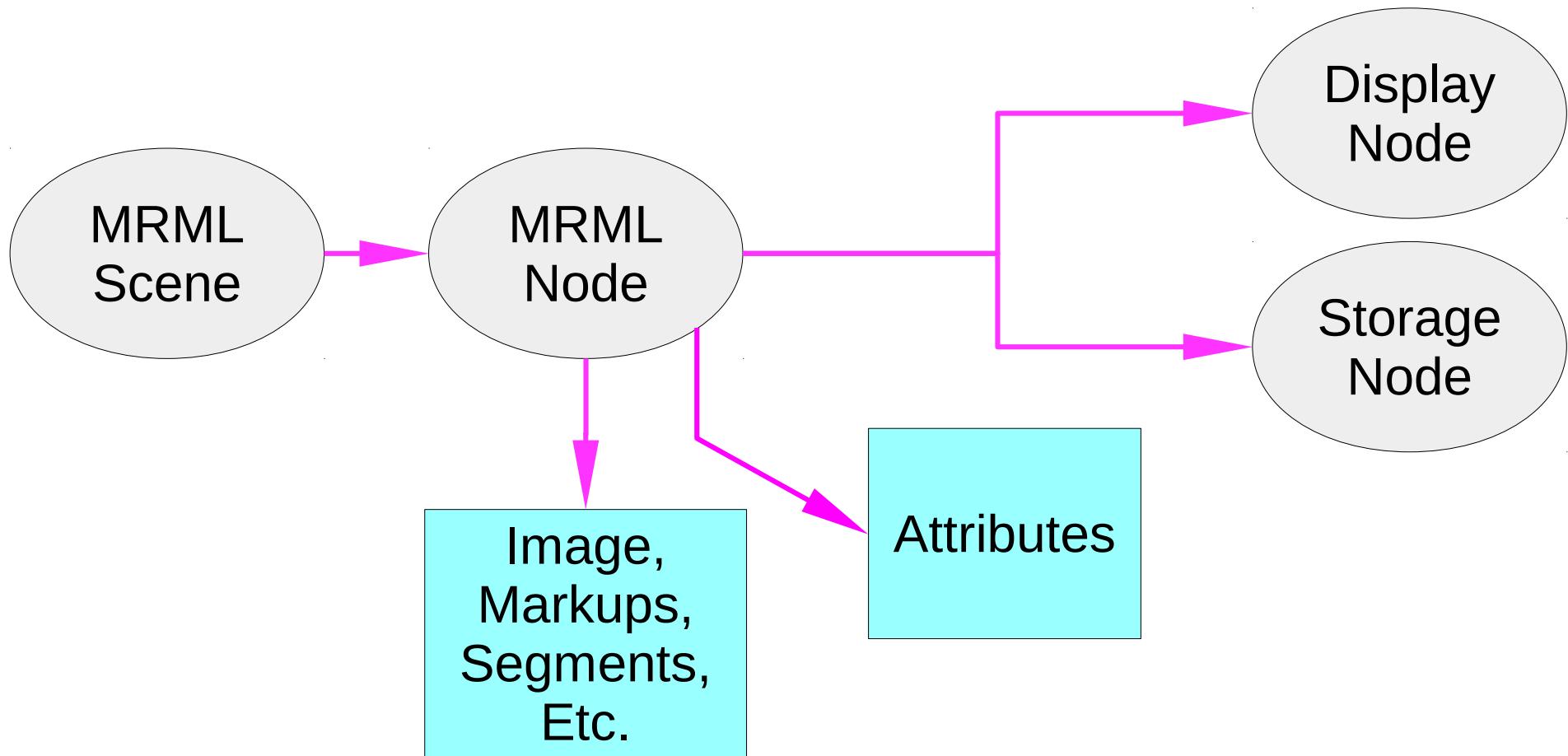
```
>>> t0 = slicer.util.loadVolume('t0.mha')
>>> t0.|
```

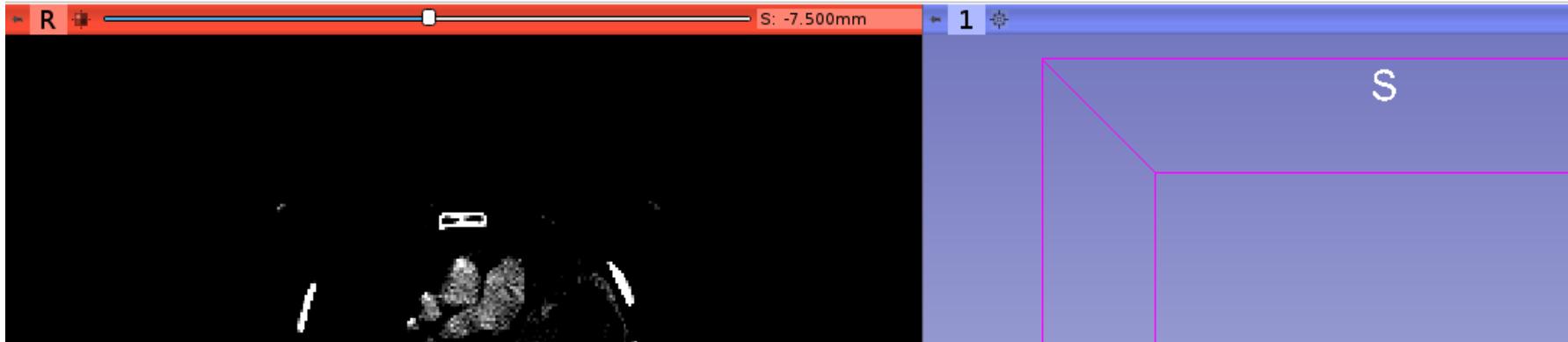
- AddAndObserveDisplayNodeID()
- AddAndObserveNodeReferenceID()
- AddAndObserveStorageNodeID()
- AddDefaultStorageNode()
- AddNodeReferenceID()
- AddNodeReferenceRole()
- AddObserver()

B: t0

+ Y

# MRML Nodes





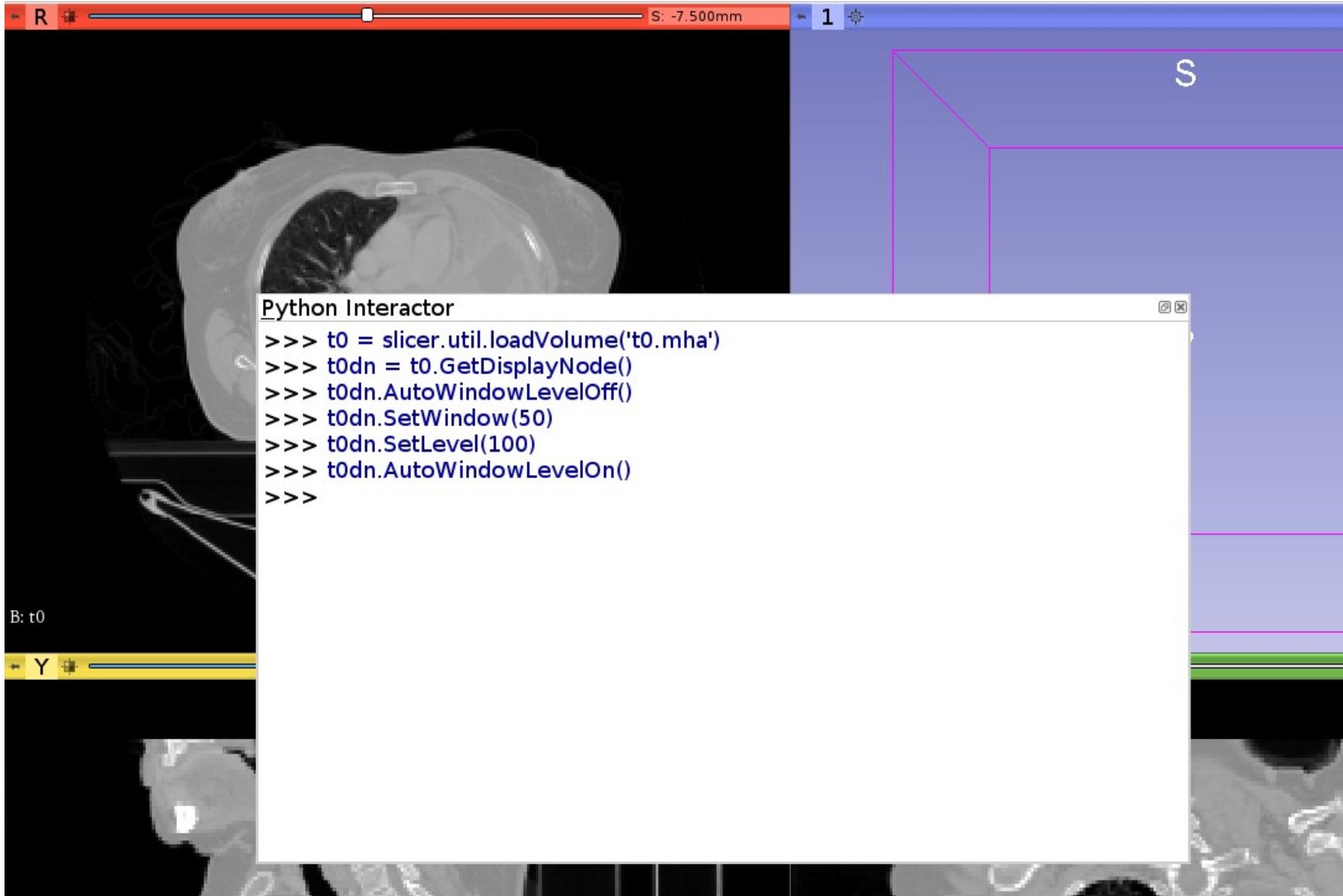
### Python Interactor

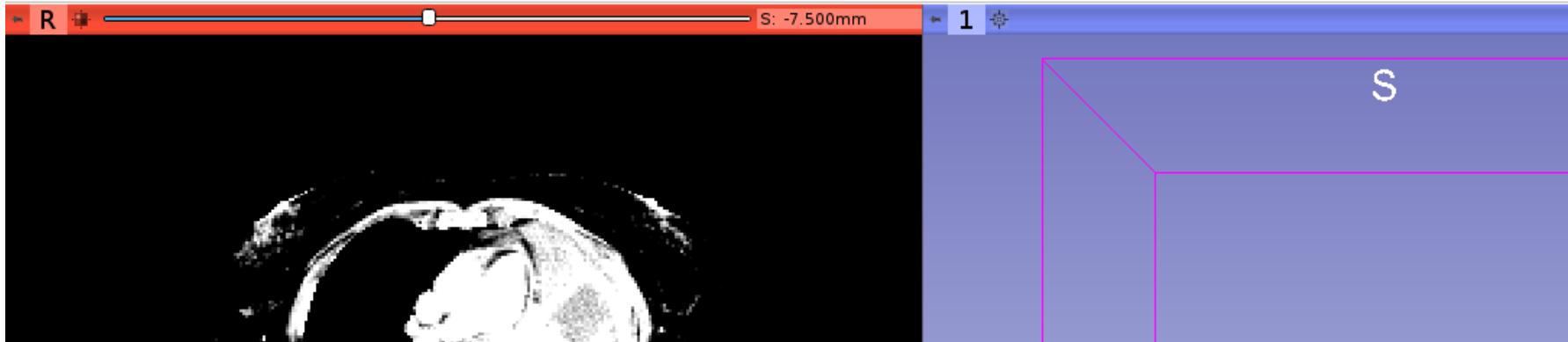
```
>>> t0 = slicer.util.loadVolume('t0.mha')
>>> t0dn = t0.GetDisplayNode()
>>> t0dn.AutoWindowLevelOff()
>>> t0dn.SetWindow(50)
>>> t0dn.SetLevel(100)
>>>
```

B: t0

+ Y







S

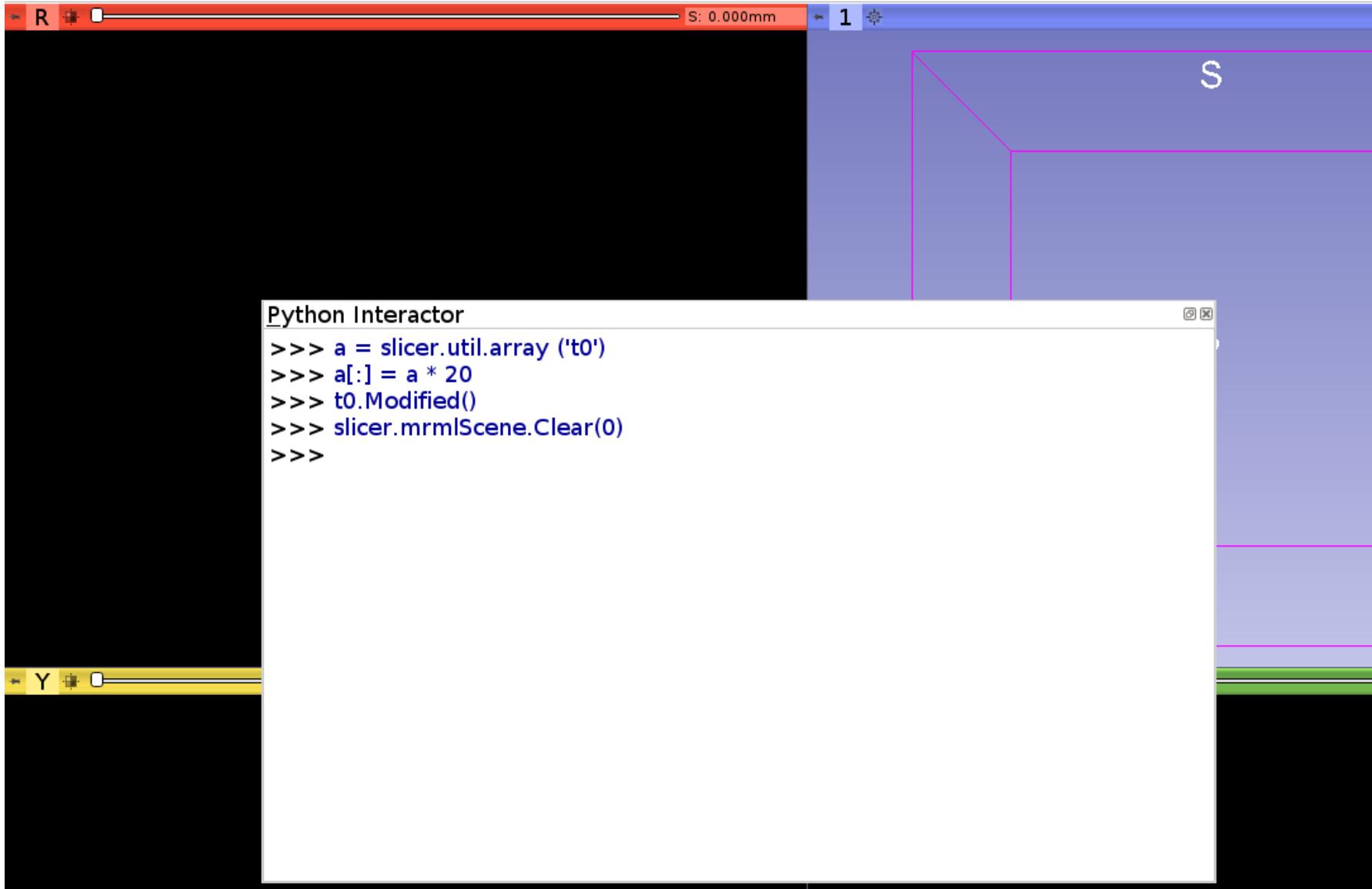
### Python Interactor

```
>>> a = slicer.util.array ('t0')
>>> a[:] = a * 20
>>> t0.Modified()
>>>
```

B: t0

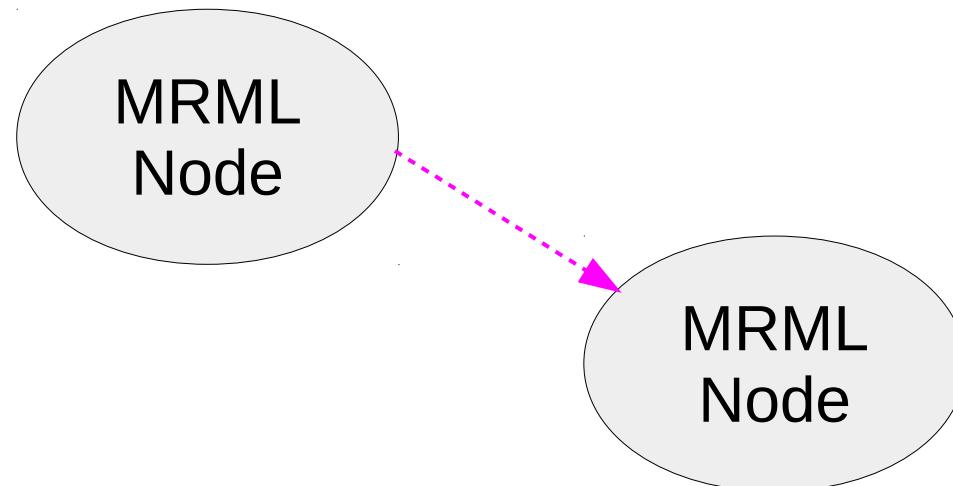
+ Y





# Node references

- Nodes can refer to other nodes using the concept of a node reference



# Node reference

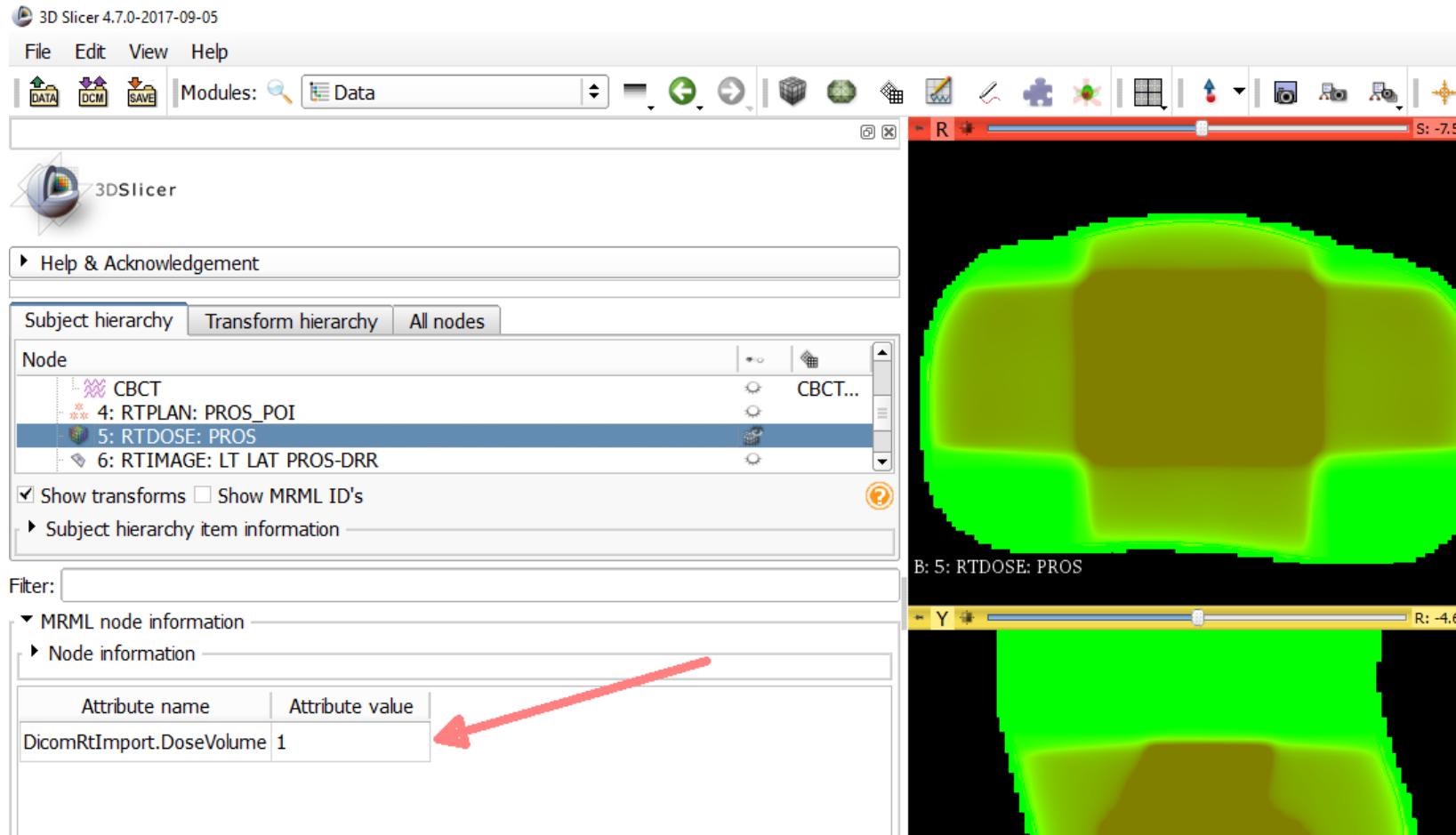


```
# Create model node
model = slicer.vtkMRMLModelNode()

# Create model node
transform = slicer.vtkMRMLLinearTransformNode()

# Transform model by reference to transform node
model.SetAndObserveTransformNodeID(
    transform.GetID())
```

# Node attributes

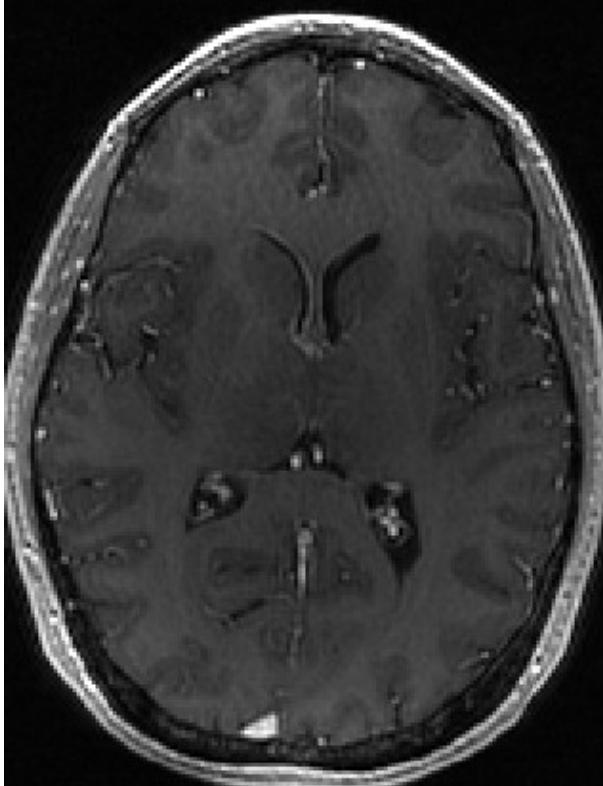


# SimpleITK

- An easy way to use ITK functions in python
- But, requires extra step to move images between SimpleITK and Slicer

S: -0.700mm

1

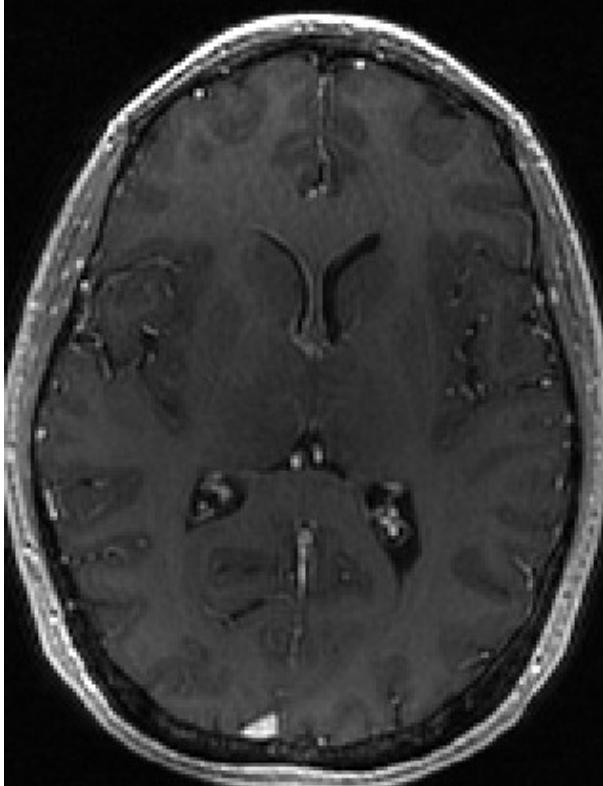


### Python Interactor

```
>>> import SimpleITK  
>>> import sitkUtils  
>>> input = sitkUtils.PullVolumeFromSlicer('MRBrainTumor1')  
>>> output = SimpleITK.DiscreteGaussian (input, 1.0, 5)  
>>>
```

S: -0.700mm

1

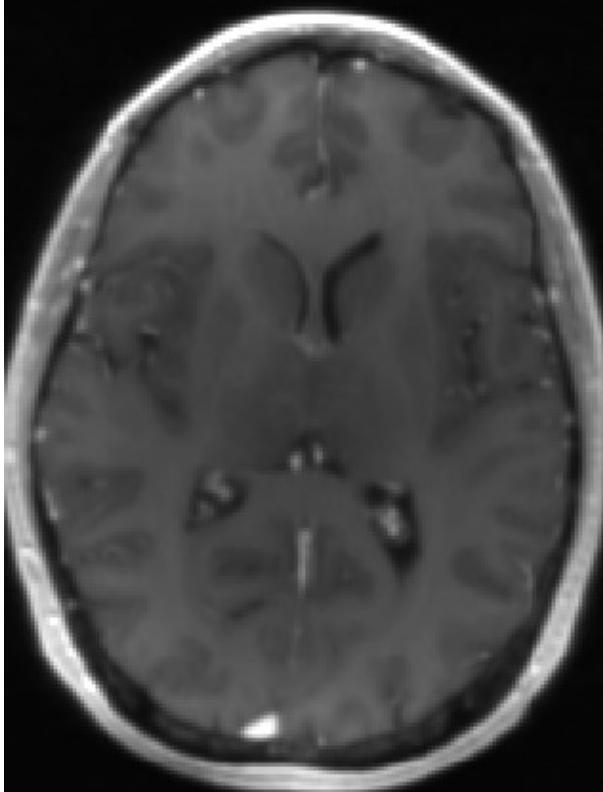


### Python Interactor

```
>>> node = sitkUtils.PushVolumeToSlicer(output)
>>>
```

S: -0.700mm

1



### Python Interactor

```
>>> node = sitkUtils.PushVolumeToSlicer(output)
>>> slicer.util.setSliceViewerLayers(node)
>>>
```

# Python modules

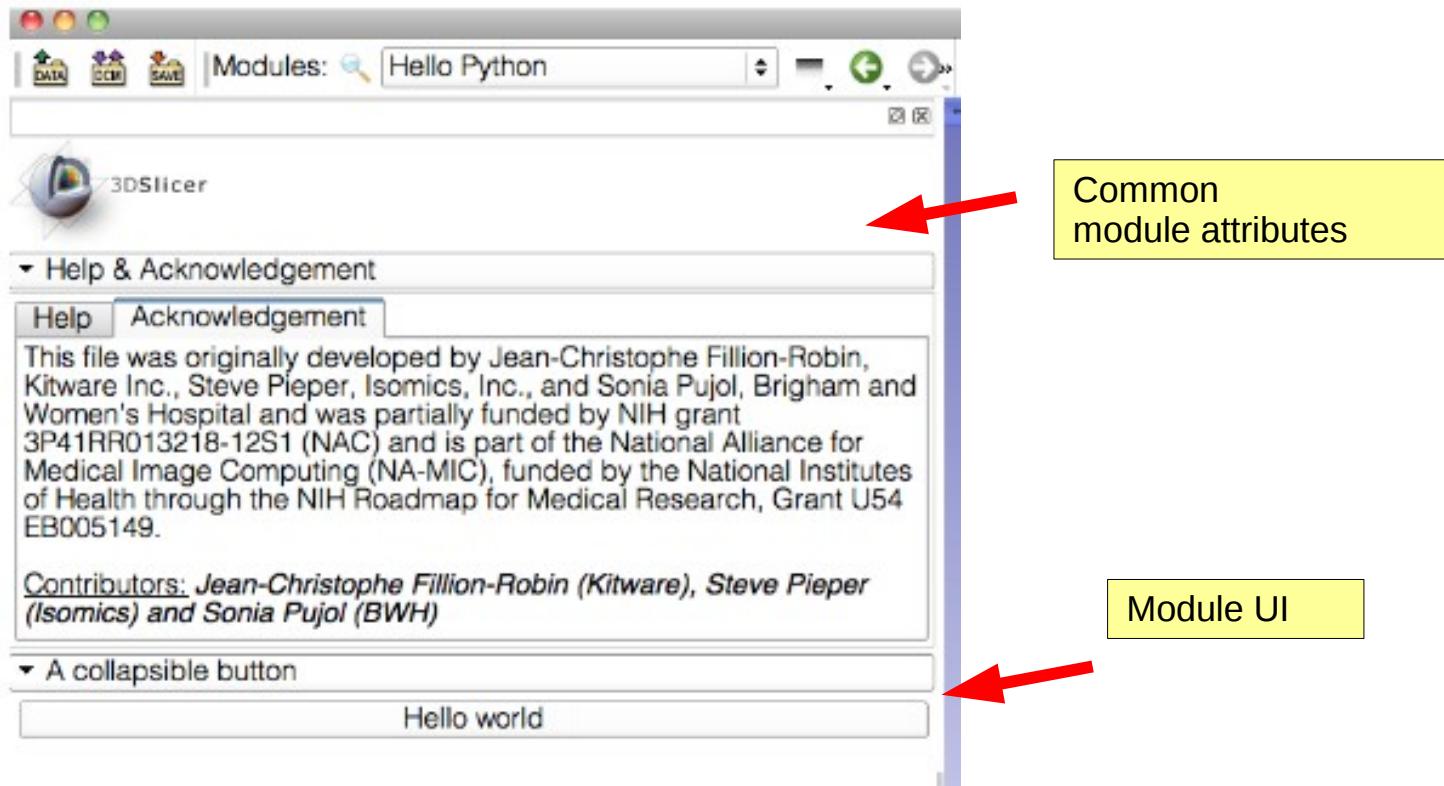
```
class ExampleModule(ScriptedLoadableModule):  
    def __init__(self, parent):  
        ScriptedLoadableModule.__init__(self, parent)  
        self.parent.title = "Hello Python"  
        self.parent.categories = ["Examples"]  
        self.parent.dependencies = []  
        self.parent.contributors = ["Jean-Christophe ..."]  
        self.parent.helpText = ...
```

Common  
module attributes

```
class ExampleModuleWidget(ScriptedLoadableModuleWidget):  
    def __init__(self, parent = None):  
        ScriptedLoadableModuleWidget.__init__(self, parent)  
        self.logic = ExampleModuleLogic()  
  
    def setup(self):  
        ScriptedLoadableModuleWidget.setup(self)  
        inputCollapsibleButton = ctk.ctkCollapsibleButton()  
        inputCollapsibleButton.text = "A Collapsible Button"  
        self.layout.addWidget(inputCollapsibleButton)
```

Module UI

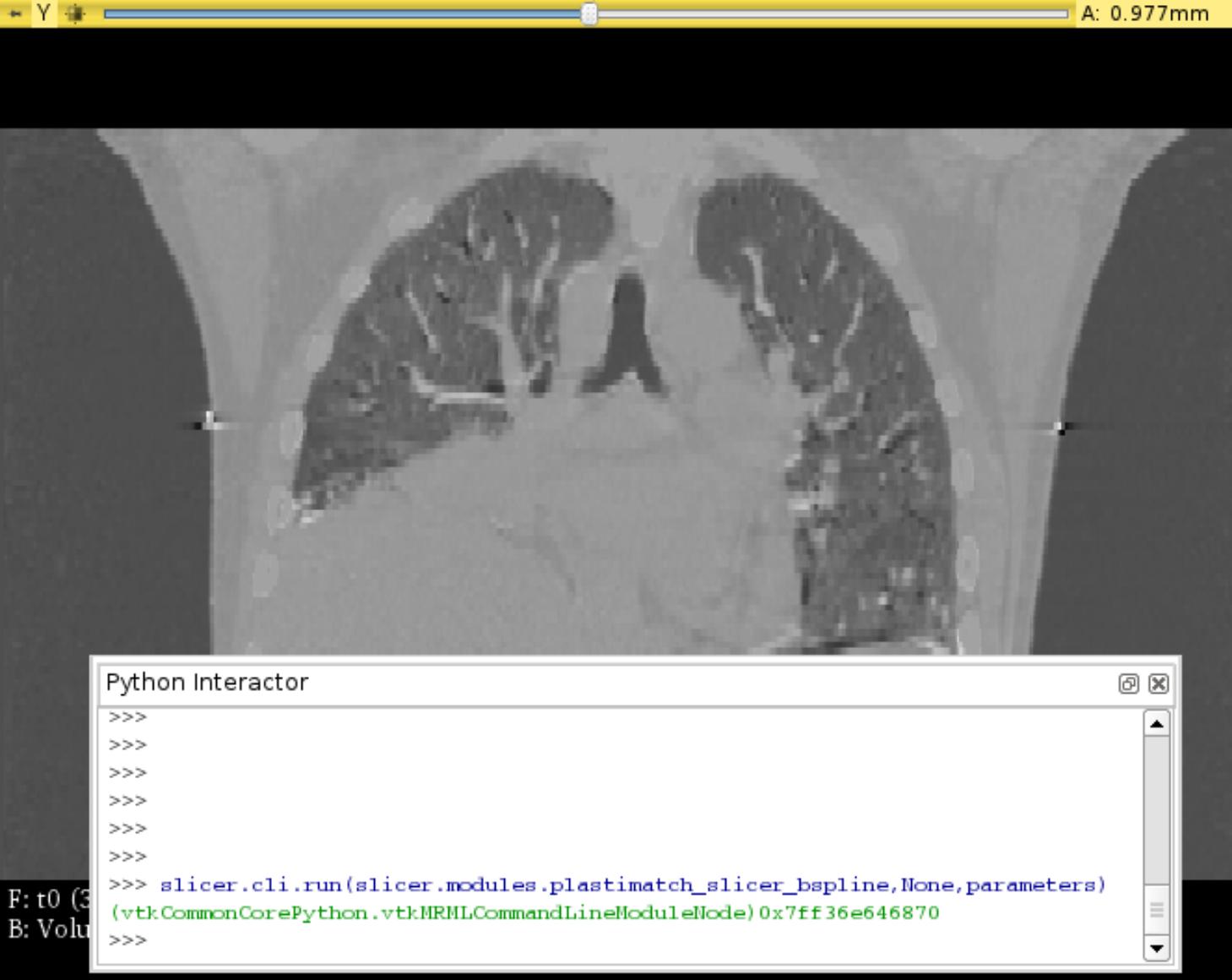
# Python modules

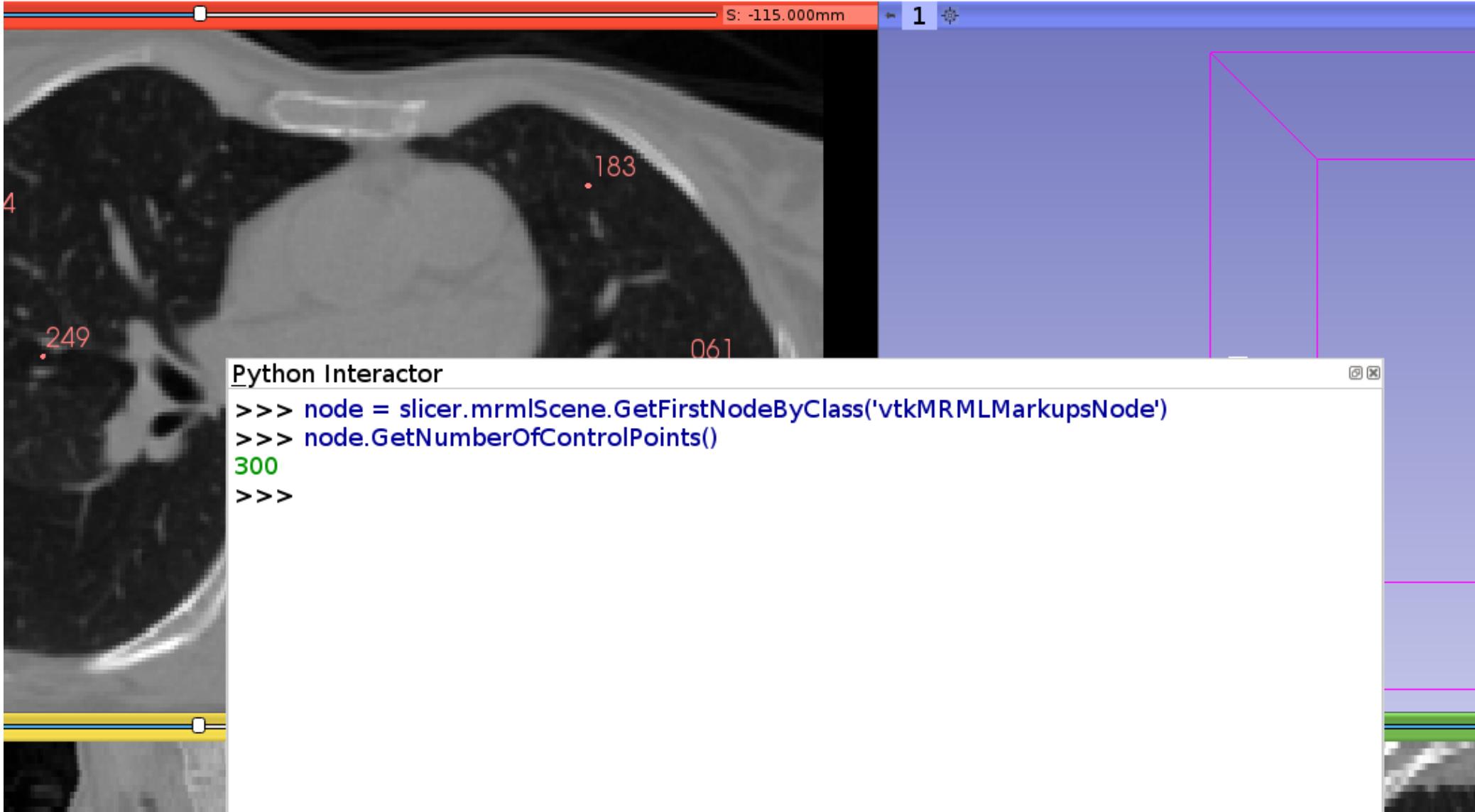


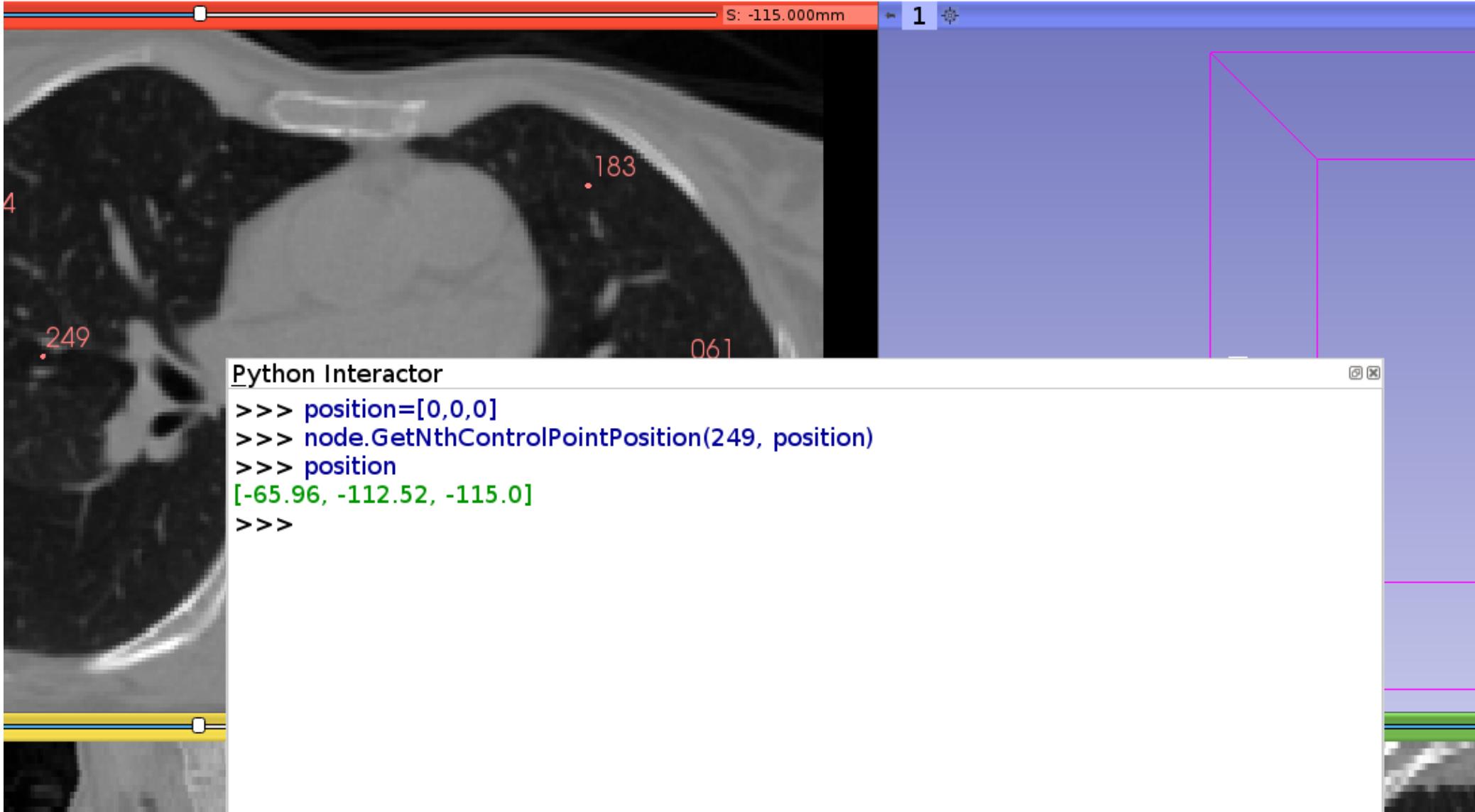
# Calling Slicer modules

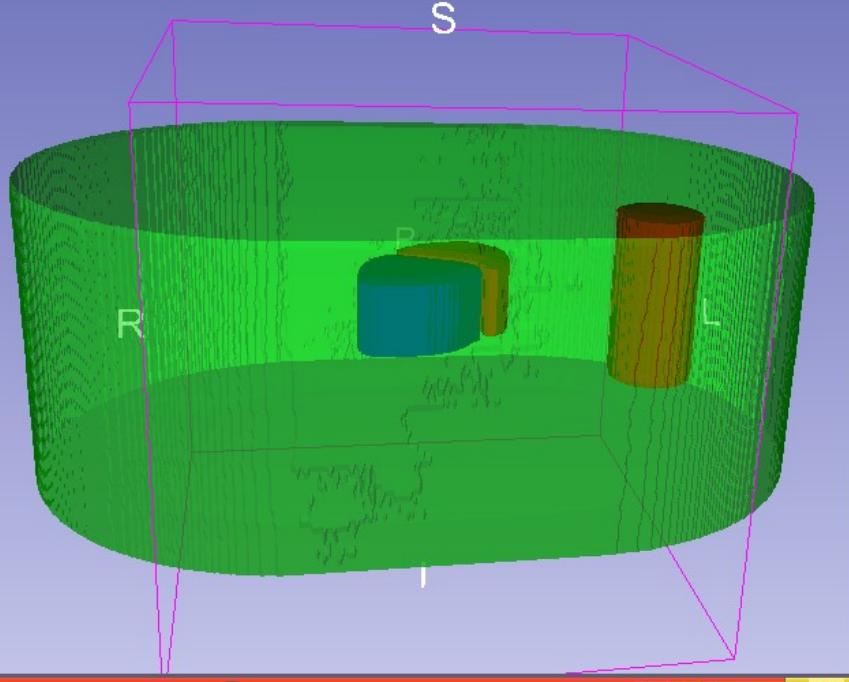
- Slicer modules can be called from python
- CLI modules follow this pattern:
  - Create any needed output nodes
  - Set parameters
  - Call CLI function







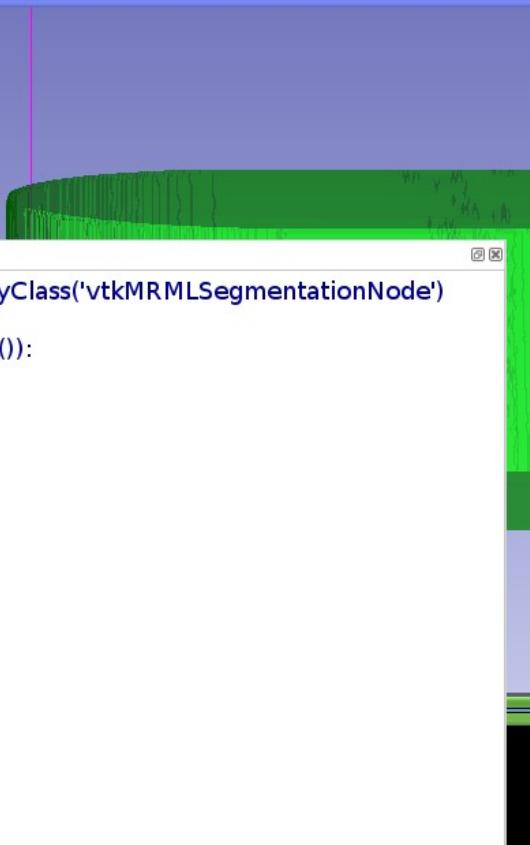
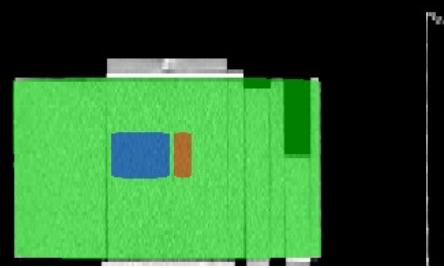
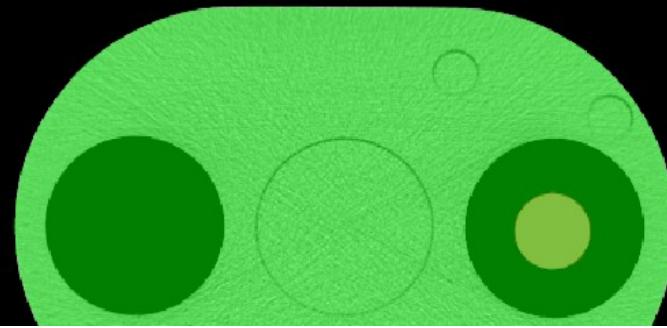


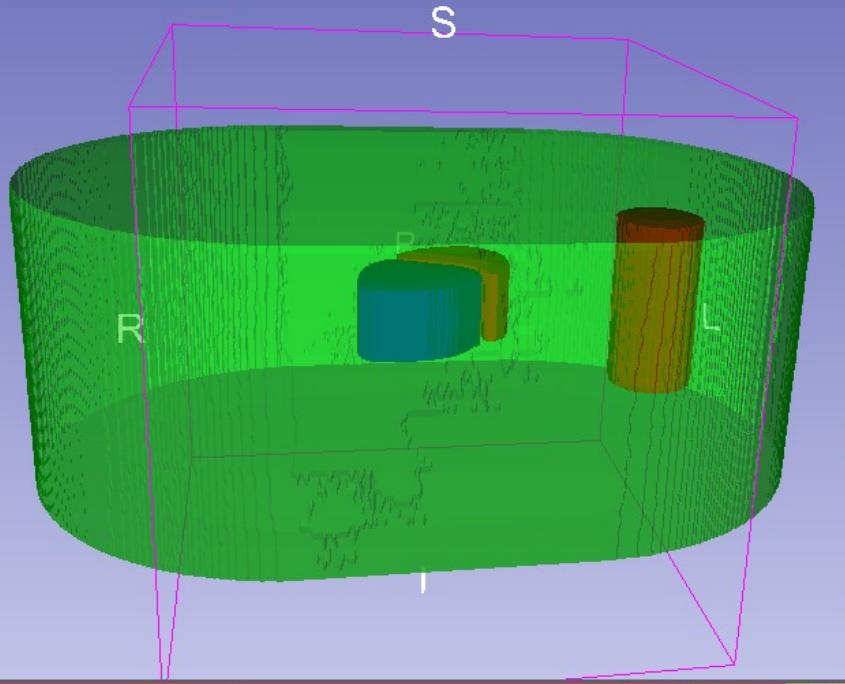


2

Python Interactor

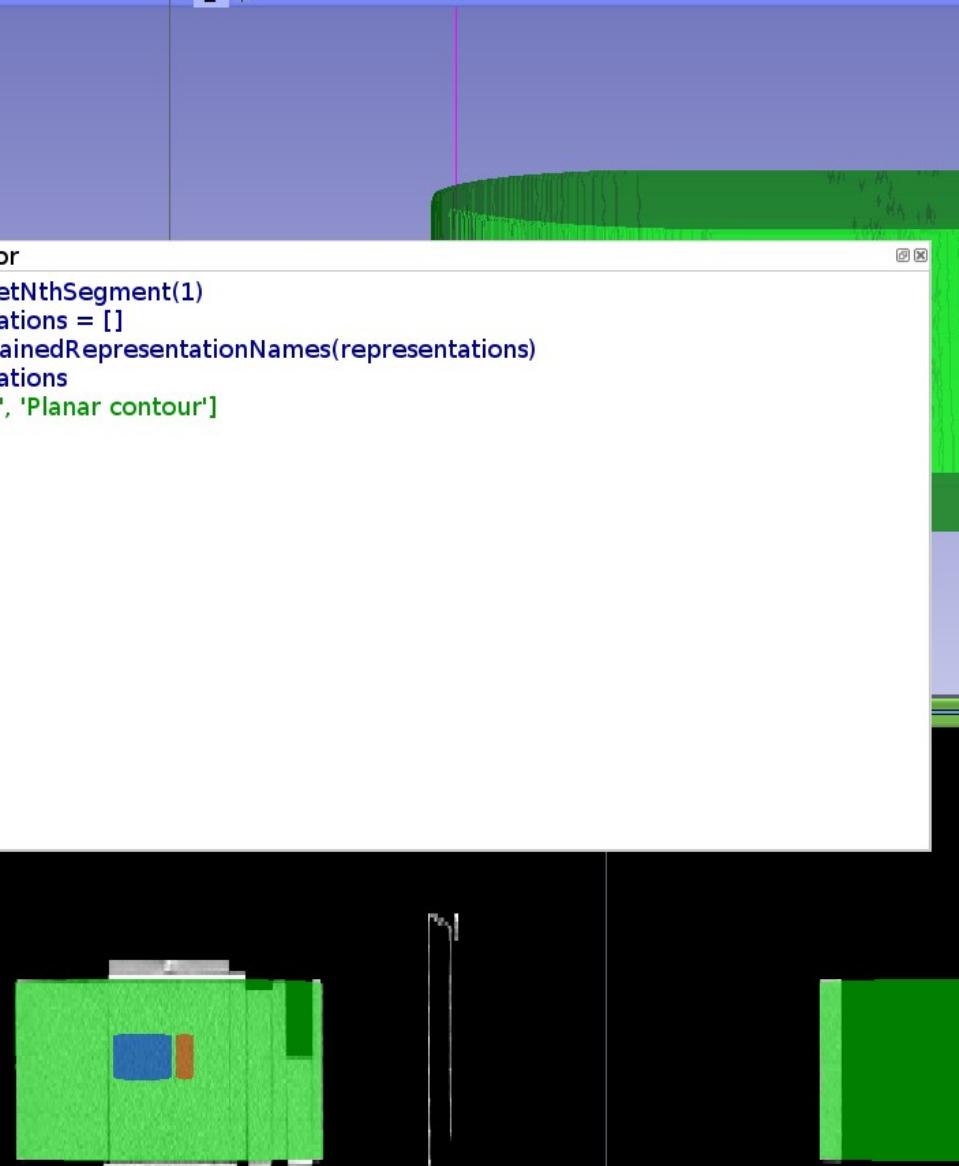
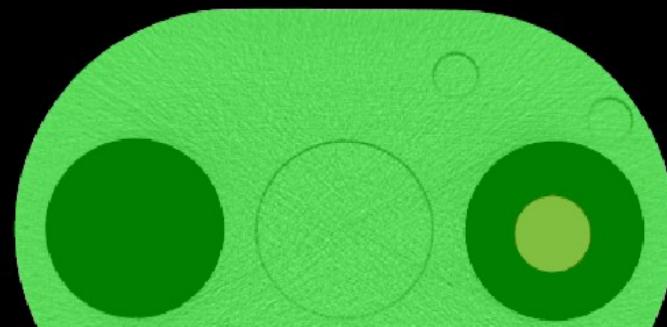
```
>>> seg_node = slicer.mrmlScene.GetFirstNodeByClass('vtkMRMLSegmentationNode')
>>> seg = seg_node.GetSegmentation()
>>> for i in range (0, seg.GetNumberOfSegments()):
...     seg.GetNthSegment(i).GetName()
...
'Insert'
'External'
'GTV'
'OAR'
>>>
```





### Python Interactor

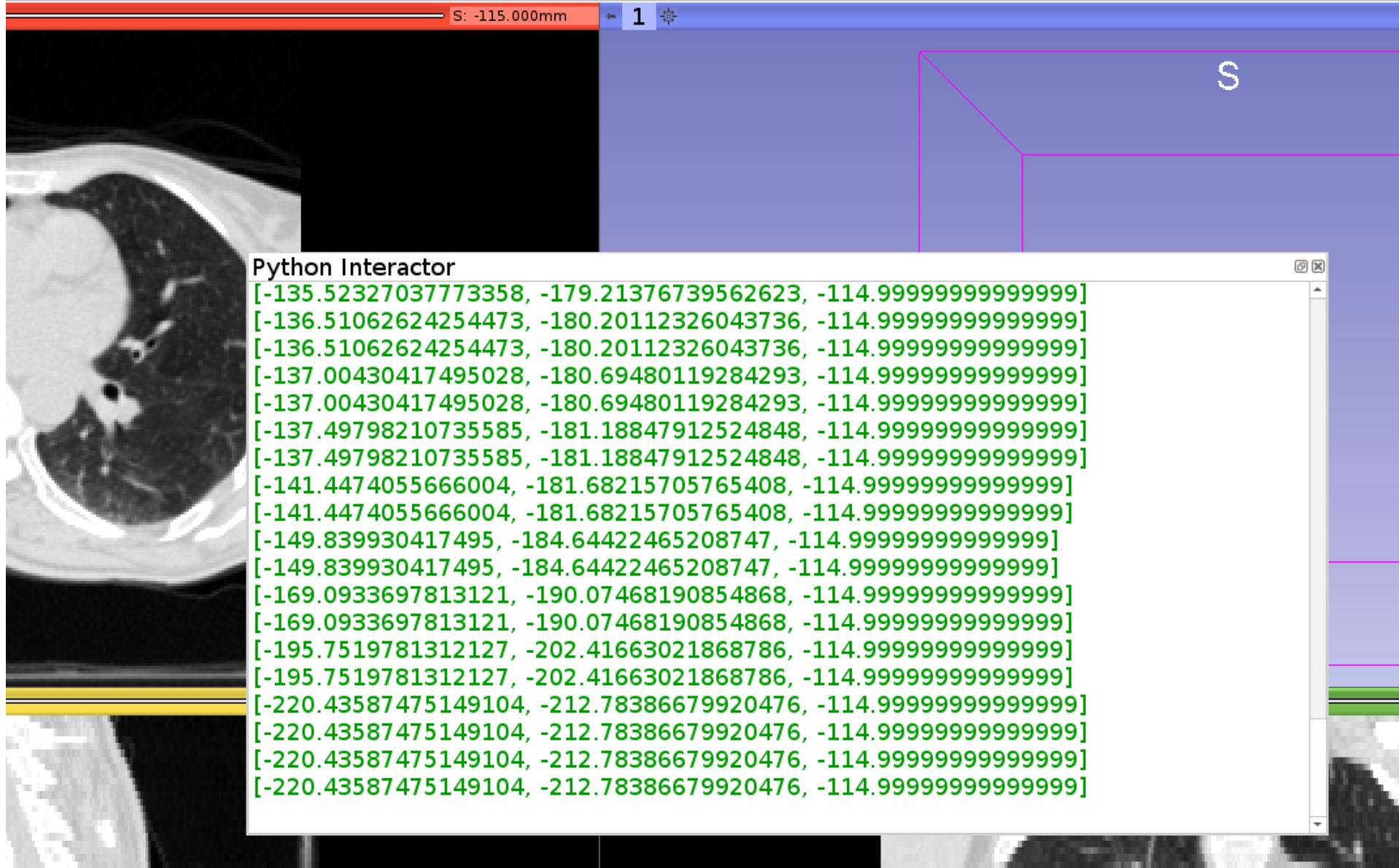
```
>>> s = seg.GetNthSegment(1)
>>> representations = []
>>> s.GetContainedRepresentationNames(representations)
>>> representations
['Closed surface', 'Planar contour']
>>>
```





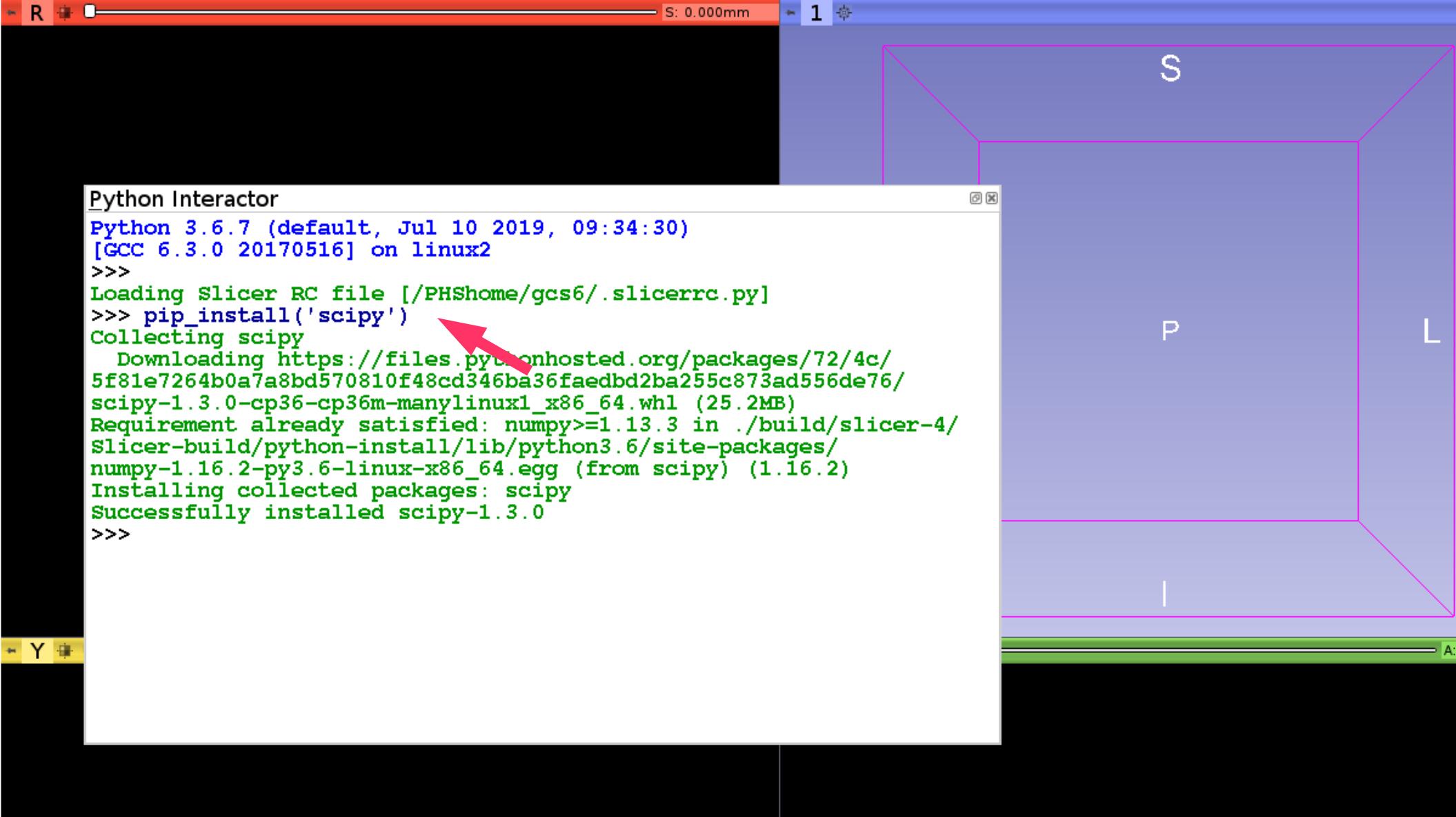
### Python Interactor

```
>>> def onMouseMoved(observer,eventid):
...     ras=[0,0,0]
...     crosshairNode.GetCursorPositionRAS(ras)
...     print(ras)
...
>>> crosshairNode=slicer.util.getNode('Crosshair')
>>>
crosshairNode.AddObserver(slicer.vtkMRMLCrosshairNode.CursorPositionModifiedEvent,
onMouseMoved)
15
>>>
```



# Python scripting

- As of June 2019, python binary packages can be used



## Python Interactor

```
Python 3.6.7 (default, Jul 10 2019, 09:34:30)
[GCC 6.3.0 20170516] on linux2
>>>
Loading Slicer RC file [/PHShome/gcs6/.slicerrc.py]
>>> pip_install('scipy')
Collecting scipy
  Downloading https://files.pythonhosted.org/packages/72/4c/
  5f81e7264b0a7a8bd570810f48cd346ba36faedb2ba255c873ad556de76/
  scipy-1.3.0-cp36-cp36m-manylinux1_x86_64.whl (25.2MB)
Requirement already satisfied: numpy>=1.13.3 in ./build/slicer-4/
  Slicer-build/python-install/lib/python3.6/site-packages/
  numpy-1.16.2-py3.6-linux-x86_64.egg (from scipy) (1.16.2)
Installing collected packages: scipy
Successfully installed scipy-1.3.0
>>>
```



# Documentation/Nightly/Developers

[Home](#) < [Documentation](#) < [Nightly](#) < [Developers](#)

New Developers: Welcome ! Check these [instructions](#) !

## Getting involved

### Start here

New community member checklist

### Developer meetings

It is open to everyone, feel free to join.

### Discussion Forum

The most effective way to get help from the community

### FAQ

Set of common development questions/answers

## Resources

### Roadmap

What's next ? What's the plan ?

### Labs

Keep track of on-going experiments.

### Doxxygen

Source code API documentation.

### Source code repository

Github [or](#) SVN [or](#)

### CDash Quality Dashboard: SlicerStable [,](#) SlicerPreview [,](#)

Nightly, continuous and experimental dashboards. - [Setup a dashboard](#)

### Bug tracker [/ Register](#) [/ Report a problem](#) / Contribute a patch

Web-based bug tracking system - [Configuration](#)

### Slicer Style Guidelines

Consistency and Readability for a manageable code base

### Script repository

Collection of python scripts manipulating various Slicer components.

### Change logs and release details

Informations about Slicer releases

## Resources

List Slicer resources and who to contact in case of problem.

## Build instructions

### Create Slicer extensions

UPDATED

Build, test, package and distribute extensions

### Build Slicer application

Compiling and installing Slicer from source.

### Build Module

Compiling slicer modules outside of the slicer source tree.

## How-tos

### Migration Guide

How to update Slicer extension code following breaking changes in Slicer or dependent toolkits (VTK, ITK, Qt, ...)

### Source code How-tos

How to checkout the source and contribute patches

### Module How-tos

How to create or build modules, how to add or update remote modules

### Testing How-tos

How to write tests and manage testing data

### Documentation How-tos

How to document module or extension

### Extension How-tos

How to contribute, build or package extensions

### Misc How-tos

How to setup QtCreator, setup dashboard, work with QtDesigner, port modules

### Debugging How-tos

How to debug Slicer, troubleshoot crash or memory leaks

## Quick links

[Stable Dashboard](#) [,](#) [Preview Dashboard](#) [,](#)

[Source](#) [,](#)

[C++ API](#) [,](#)

[Contribute](#)

[Style Guide](#)

[Issue Tracker](#) [,](#)

[WikiCheatSheet](#), [Wiki Editing Guide](#)

## Design & Implementation

### Modules

Comparison between the different supported module types

### Data Model / MRML

Objects and their organization. MRML Library provides API for managing medical image data type

### Logics

Details the role of MRML/Slicer/Application/Modules logics and Displayable managers

### Layouts

UPDATED

How to control the layout of the views

### Slice Orientation Presets

How to manage slice orientation presets

### Sliclets

Create simple standalone applications (sliclets)

### IO Mechanism

How to read or write nodes from file

### Python scripting

Presents the underlying infrastructure.

### Charts

Description of the Charting (jqPlot) architecture.

### Plots

NEW

Description of the Plotting (VTK) architecture.

### Directory Structure

Files location in the build and install tree.

### Qt Plugins

How to build and load Qt plugins.

### Build system / Release process / Factory description

Details how Slicer is built and packaged.

### QtTesting

Testing framework to test Slicer application. It complements unit tests.

# Python script repository



3DSlicer

**About slicer**  
+ Introduction  
+ Acknowledgments  
+ News  
+ Contact Us  
+ Licensing  
+ Commercial Use

**Publication**  
+ Publication DB  
+ Image Gallery  
+ SlicerCommunity  
+ Citing Slicer

**Documentation**  
+ Slicer Training  
+ User manual  
+ Developer manual

**Help**  
+ Help  
+ User Q&A  
+ Developer FAQ  
+ Discussion Forum

**Links**  
+ Downloads  
+ Wiki  
+ Homepage  
+ Recent Changes

**Search**  
 Search  
Go  
Search

**Tools**  
+ What links here  
+ Related changes  
+ Special pages  
+ Printable version  
+ Permanent link  
+ Page information

[page](#) [discussion](#) [view source](#) [history](#)

## Documentation/Nightly/ScriptRepository

Home < Documentation < Nightly < ScriptRepository

For the stable Slic

Contents (hide)

### 1 Community-contributed modules

1.1 Filters

1.2 DICOM

1.3 Informatics

### 2 Community-contributed examples

2.1 Capture

2.2 Launching Slicer

2.3 Load volume from file

2.4 Show volume rendering automatically when a volume is loaded

2.5 Automatically load volumes that are copied into a folder

2.6 DICOM

2.6.1 How to access top level tags of DICOM images imported into Slicer? For example, to print the first patient's first study's first series' "0020,0032" field:

2.6.2 How to access DICOM tags nested in a sequence

2.6.3 How to access tag of a volume loaded from DICOM? For example, get the patient position stored in a volume:

2.6.4 How to access tag of an item in the Subject Hierarchy tree? For example, get the content time tag of a structure set:

2.6.5 How to get path and filename of a loaded DICOM volume?

2.6.6 How can I convert DICOM to NRRD on the command line?

2.6.7 Export a volume to DICOM file format

2.6.8 Customize table columns in DICOM browser

2.7 Toolbar functions

2.8 Manipulating objects in the slice viewer

2.9 Set slice position and orientation from 3 markup fiducials

2.10 Switching to 3D view in fiducial placement mode

2.11 Set up markup fiducial display

2.12 Get a notification if a markup point position is modified

2.13 Get a notification if a transform is modified

2.14 Show a context menu when a markup point is clicked in a slice or 3D view

2.15 Write markup positions to JSON file

2.16 Write annotation ROI to JSON file

2.17 Show a simple surface mesh as a model node

2.18 Add a texture mapped plane to the scene as a model

2.19 Get scalar values at surface of a model

2.20 Export entire scene as VRML

2.21 Export model to Blender, including color by scalar

2.22 Export a tract (FiberBundle) to Blender, including color

2.23 Import a tract (FiberBundle) streamline points

2.24 Clone a node

2.25 Clone a volume

2.26 Create a new volume

2.27 Modify voxels in a volume

2.28 Get volume voxel coordinates from markup fiducial RAS coordinates

2.29 Get markup fiducial RAS coordinates from volume voxel coordinates

2.30 Get the values of all voxels for a label value

2.31 Access values in a DTI tensor volume

2.32 Change window/level (brightness/contrast) or colormap of a volume

2.33 Make mouse left-click and drag on the image adjust window/level

2.34 Create custom color table

2.35 Manage a slice view

2.35.1 Change slice offset

2.35.2 Change slice orientation

2.35.3 Show slice views in 3D window

2.35.4 Reset field of view to show background volume maximized

2.35.5 Rotate slice views to volume plane

2.35.6 Iterate over current visible slice views, and set foreground and background images

2.36 Synchronize zoom factor between slice views

2.37 Show a volume in slice views

2.38 Change opacity of foreground volume in slice views

2.39 Fit slice plane to markup fiducials

2.40 Save a series of images from a Slice View

2.41 Save the scene into a new directory

- 2.42 Save the scene into a new directory
- 2.42 Save the scene into a single MRB file
- 2.43 Save a node to file
- 2.44 Center the 3D View on the Scene
- 2.45 Rotate the 3D View
- 2.46 Display text in a 3D view or slice view
- 2.47 Hide slice view annotations (DataProbe)
- 2.48 Turning off interpolation
- 2.49 Customize viewer layout
- 2.50 Customize keyboard shortcuts
- 2.51 Disable certain user interactions in slice views
- 2.52 Change default slice view orientation
- 2.53 Set all slice views linked by default
- 2.54 Set up custom units in slice view ruler
- 2.55 Show a slice view outside the view layout
- 2.56 Show a 3D view outside the view layout
- 2.57 Running an ITK filter in Python using SimpleITK
- 2.58 Get current mouse coordinates in a slice view
- 2.59 Get DataProbe text
- 2.60 Get reformatted image from a slice viewer as numpy array
- 2.61 Thick slab reconstruction and maximum/minimum intensity volume projections
- 2.62 Change default file type for nodes (that have never been saved yet)
- 2.63 Change file type for saving for all volumes (with already existing storage nodes)
- 2.64 Sequences
  - 2.64.1 Concatenate all sequences in the scene into a new sequence
- 2.65 Segmentations
  - 2.65.1 Create a segmentation from a labelmap volume and display in 3D
  - 2.65.2 Export labelmap node from segmentation node
  - 2.65.3 Export model nodes from segmentation node
  - 2.65.4 Show a segmentation in 3D
  - 2.65.5 Get a representation of a segment
  - 2.65.6 Convert all segments using default path and conversion parameters
  - 2.65.7 Convert all segments using custom path or conversion parameters
  - 2.65.8 Re-convert using a modified conversion parameter
  - 2.65.9 Get centroid of a segment in world (RAS) coordinates
  - 2.65.10 How to run segment editor effects from a script
- 2.66 Accessing views, renderers, and cameras
- 2.67 Hide view controller bars
- 2.68 Change 3D view background color
- 2.69 Subject hierarchy
  - 2.69.1 Get the pseudo-singleton subject hierarchy node
  - 2.69.2 Create subject hierarchy item
  - 2.69.3 Get subject hierarchy item
  - 2.69.4 Traverse children of a subject hierarchy item
  - 2.69.5 Manipulate subject hierarchy item
  - 2.69.6 Filter items in TreeView or ComboBox
  - 2.69.7 Listen to subject hierarchy item events
- 2.70 Plotting
  - 2.70.1 Create histogram plot of a volume
- 2.71 Execute external applications
  - 2.71.1 Run process in default environment

More than 100 simple examples

Welcome to the Slicer forum!

3D Slicer ('Slicer') is an open source, extensible software platform for image visualization and analysis. Slicer has a large community of users in medical imaging and surgical navigation, and is also used in fields such as astronomy, paleontology, and 3D printing.

- [Click here for answers to common questions](#)
- [Click here for tutorials and documentation](#)
- [Click here to post a new question.](#)

To receive updates about new software releases and major events, sign-in with Google, GitHub, Facebook, or email.

[all categories ▶](#)[Categories](#)[Latest](#)[Top](#)

#### Category

#### Topics

#### Latest

### Announcements

1 / week

Low-traffic category for 3D Slicer, extension, and community news and announcements.

■ Release Notes ■ Jobs

### Support

36 / week

The Support category is for all usage questions and general discussion of Slicer and extensions.

■ Feature requests

### Development

9 / week

The Development category is for discussion of Slicer application and extension programming, software testing, and related topics - similarly to the former slicer-devel mailing list.

### Community

2 / week

Community information and project/topic sub-forums.

■ SlicerCMF ■ SlicerSALT ■ SlicerDMRI ■ ProjectWeek  
■ SlicerCIP ■ Hangout

### Forum and website feedback

57

Discussion about this site, its organization, how it works, and how we can improve it.



Welcome to the 3D Slicer Forum!

2

Aug '18



Simplify Slicer or Remove some modules from Slicer

2

35m



Update 3D view if node changes

4

1h



rebuild the surface of skull bone

5

1h



Emsegmentation cannot found

8

1h



Slicer Tractography ROI Selection Not Working

10

1h



Python codes for diffusion module

4

3h

■ Development

# Links

- <https://www.slicer.org/>
- <https://www.slicer.org/wiki/Documentation/Nightly/Training>
- <https://www.slicer.org/wiki/Documentation/Nightly/Developers>
- <https://discourse.slicer.org/>