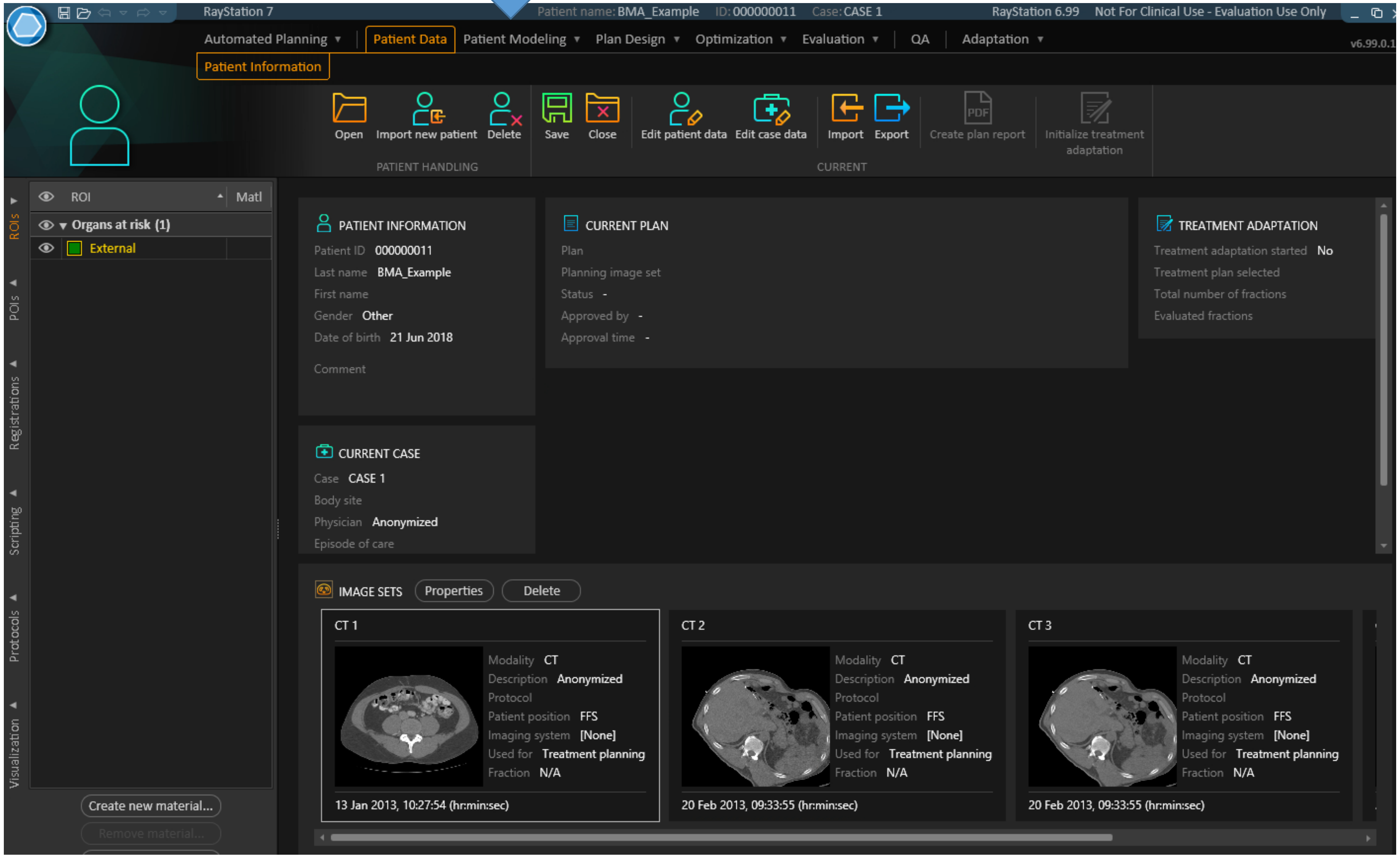
Raystation Scripting

Brian Mark Anderson

4/2/2020



Patient name: BMA_Example ID: 000000011 Case: CASE 1 RayStation 6.99 Not For Clinical Use - Evaluation Use Only

PATIENT INFORMATION

Patient ID 000000011
Last name BMA_Example
First name
Gender Other
Date of birth 21 Jun 2018
Comment

CURRENT PLAN

Plan
Planning image set
Status -
Approved by -
Approval time -

TREATMENT ADAPTATION

Treatment adaptation started No
Treatment plan selected
Total number of fractions
Evaluated fractions

CURRENT CASE

Case CASE 1
Body site
Physician Anonymized
Episode of care

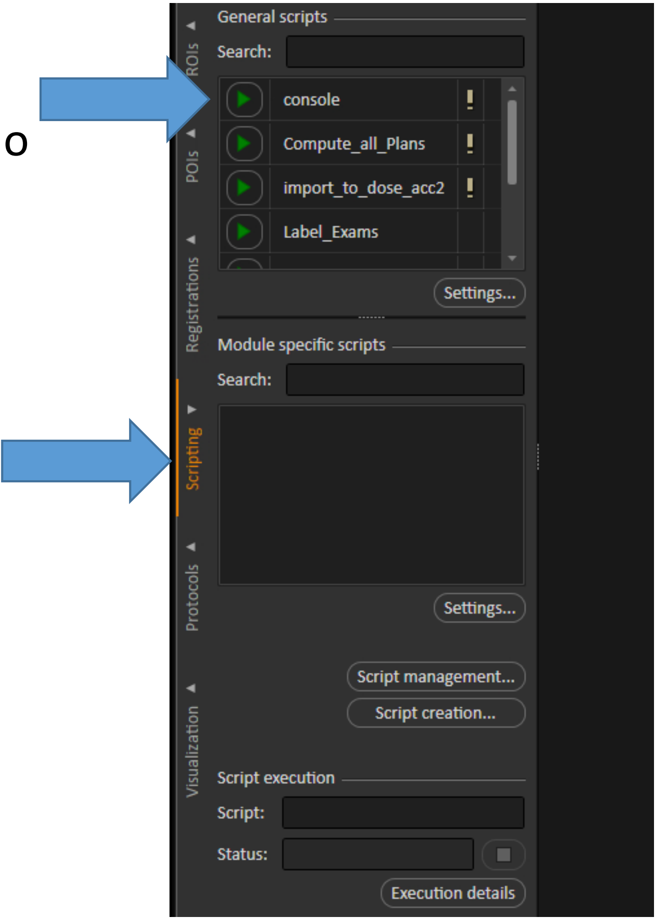
CT 1
Modality CT
Description Anonymized
Protocol
Patient position FFS
Imaging system [None]
Used for Treatment planning
Fraction N/A
13 Jan 2013, 10:27:54 (hr:min:sec)

CT 2
Modality CT
Description Anonymized
Protocol
Patient position FFS
Imaging system [None]
Used for Treatment planning
Fraction N/A
20 Feb 2013, 09:33:55 (hr:min:sec)

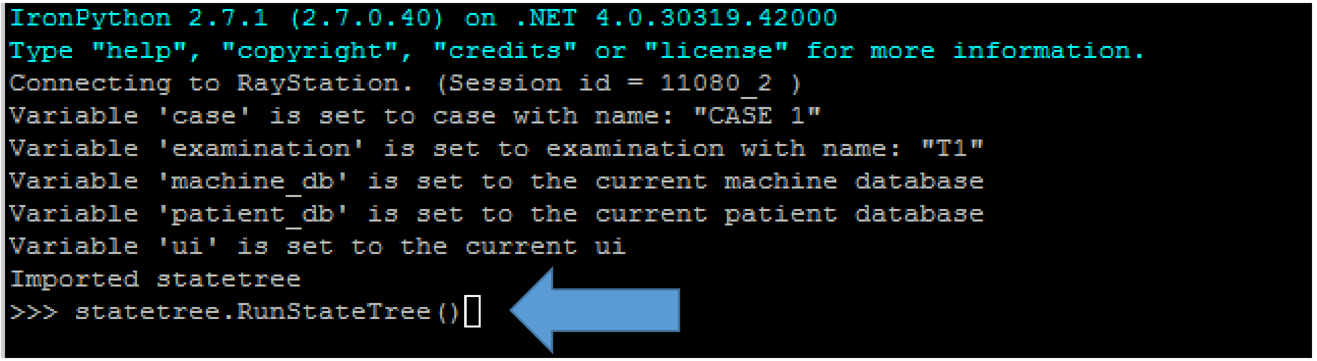
CT 3
Modality CT
Description Anonymized
Protocol
Patient position FFS
Imaging system [None]
Used for Treatment planning
Fraction N/A
20 Feb 2013, 09:33:55 (hr:min:sec)

First things first

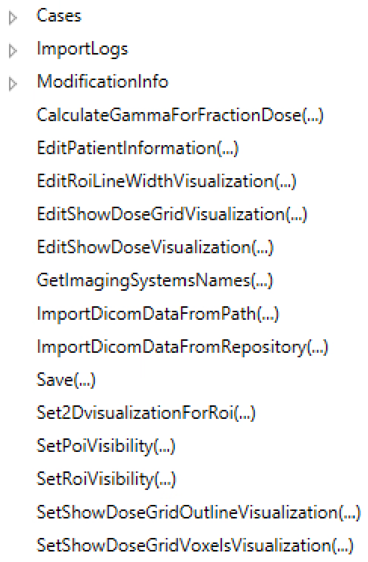
Console can do everything



StateTree is your best friend.. Everything that can be done On Raystation is in the state tree

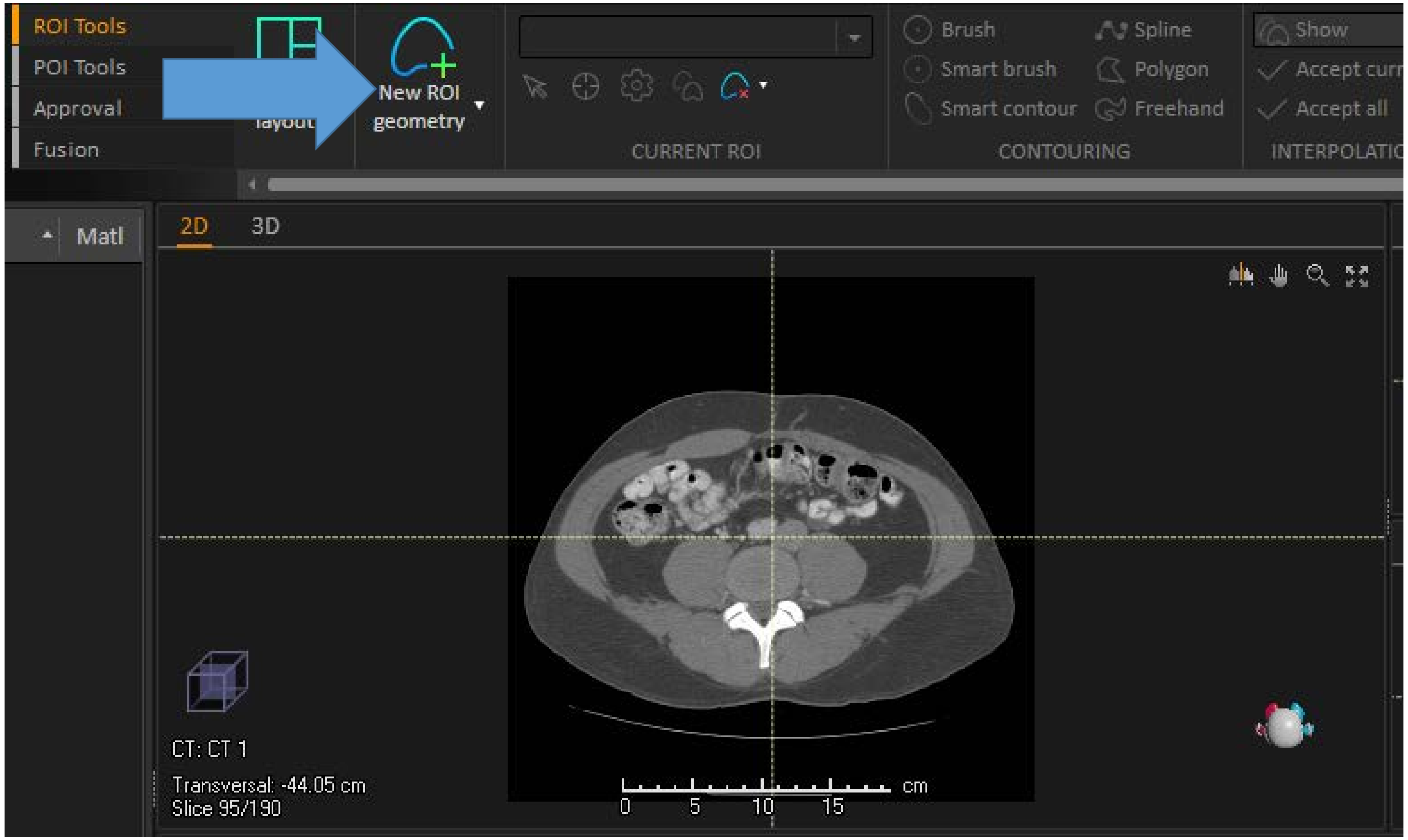


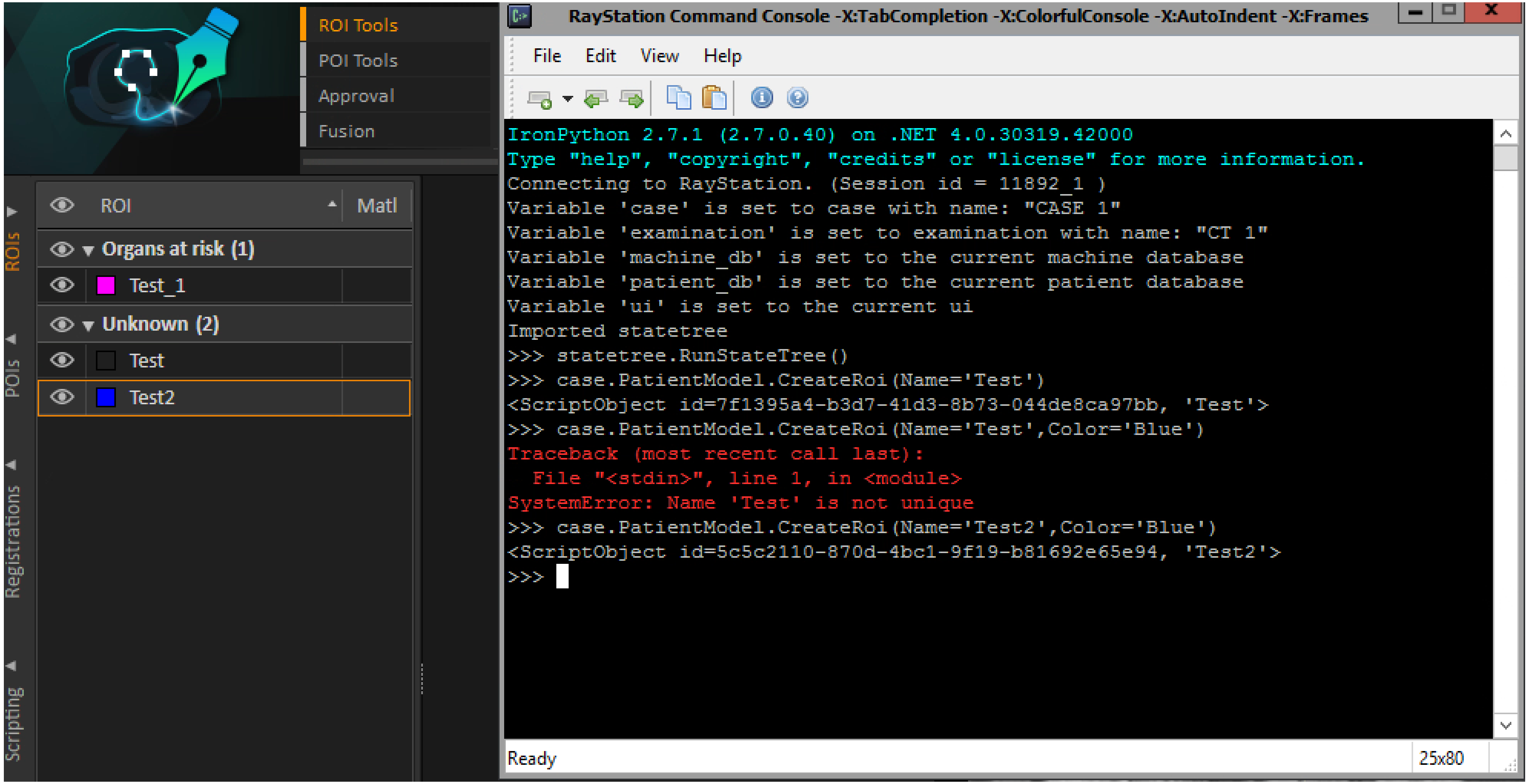
```
IronPython 2.7.1 (2.7.0.40) on .NET 4.0.30319.42000
Type "help", "copyright", "credits" or "license" for more information.
Connecting to RayStation. (Session id = 11080_2 )
Variable 'case' is set to case with name: "CASE 1"
Variable 'examination' is set to examination with name: "T1"
Variable 'machine_db' is set to the current machine database
Variable 'patient_db' is set to the current patient database
Variable 'ui' is set to the current ui
Imported statetree
>>> statetree.RunStateTree()
```



```
▷ Cases
▷ ImportLogs
▷ ModificationInfo
  CalculateGammaForFractionDose(...)
  EditPatientInformation(...)
  EditRoiLineWidthVisualization(...)
  EditShowDoseGridVisualization(...)
  EditShowDoseVisualization(...)
  GetImagingSystemsNames(...)
  ImportDicomDataFromPath(...)
  ImportDicomDataFromRepository(...)
  Save(...)
  Set2DvisualizationForRoi(...)
  SetPoiVisibility(...)
  SetRoiVisibility(...)
  SetShowDoseGridOutlineVisualization(...)
  SetShowDoseGridVoxelsVisualization(...)
```

Lets make an ROI

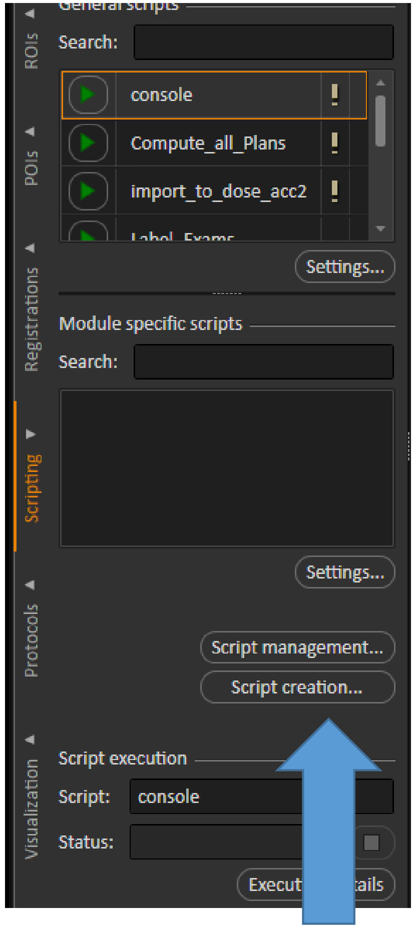




```
IronPython 2.7.1 (2.7.0.40) on .NET 4.0.30319.42000
Type "help", "copyright", "credits" or "license" for more information.
Connecting to RayStation. (Session id = 11892_1 )
Variable 'case' is set to case with name: "CASE 1"
Variable 'examination' is set to examination with name: "CT 1"
Variable 'machine_db' is set to the current machine database
Variable 'patient_db' is set to the current patient database
Variable 'ui' is set to the current ui
Imported statetree
>>> statetree.RunStateTree()
>>> case.PatientModel.CreateRoi(Name='Test')
<ScriptObject id=7f1395a4-b3d7-41d3-8b73-044de8ca97bb, 'Test'>
>>> case.PatientModel.CreateRoi(Name='Test',Color='Blue')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
SystemError: Name 'Test' is not unique
>>> case.PatientModel.CreateRoi(Name='Test2',Color='Blue')
<ScriptObject id=5c5c2110-870d-4bc1-9f19-b81692e65e94, 'Test2'>
>>>
```

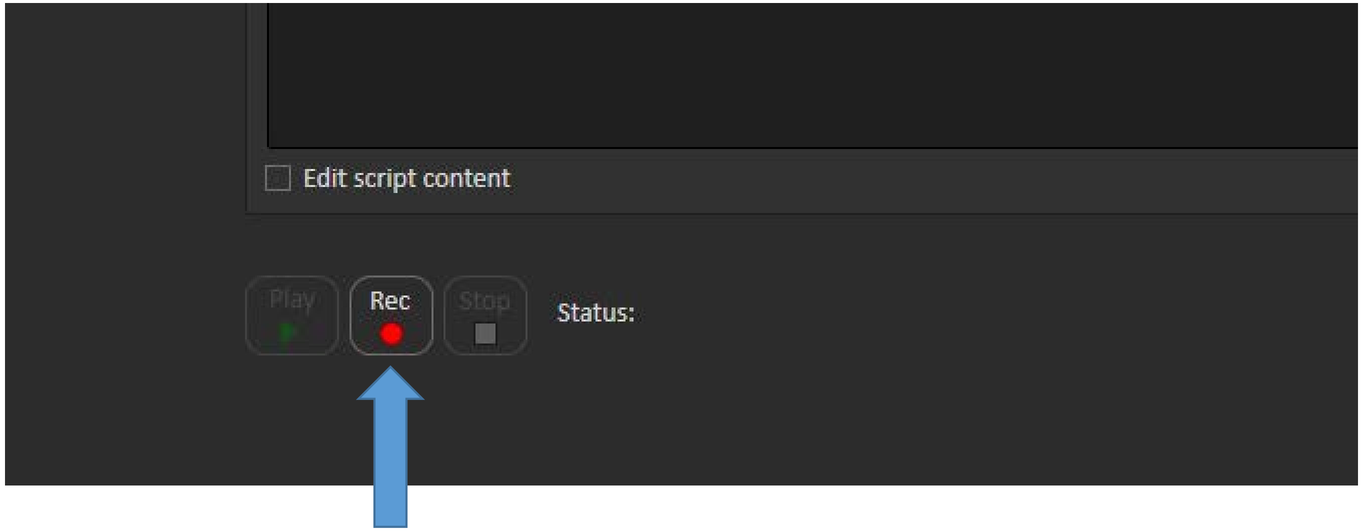
How to know what to type?

- Look in statetree, or cross fingers and hope you can record it

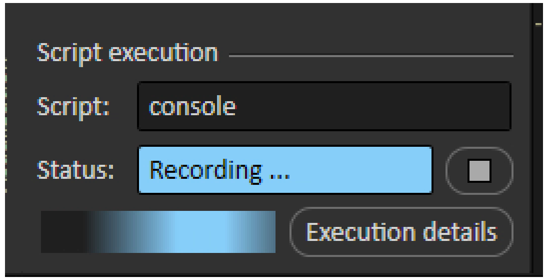


Script: console

Status:



Status:

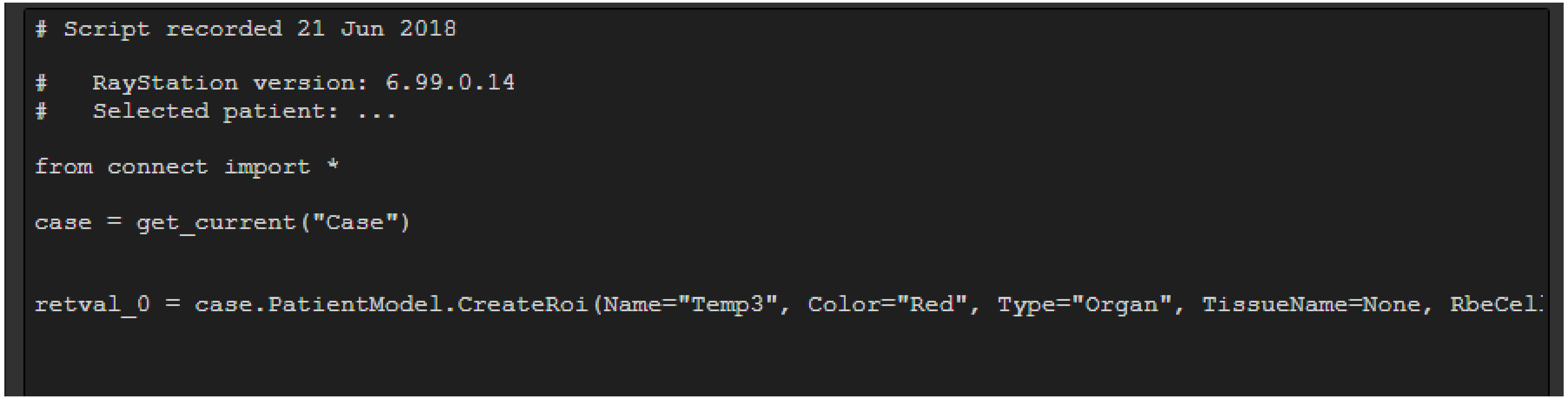


Script: console

Status: Recording ...

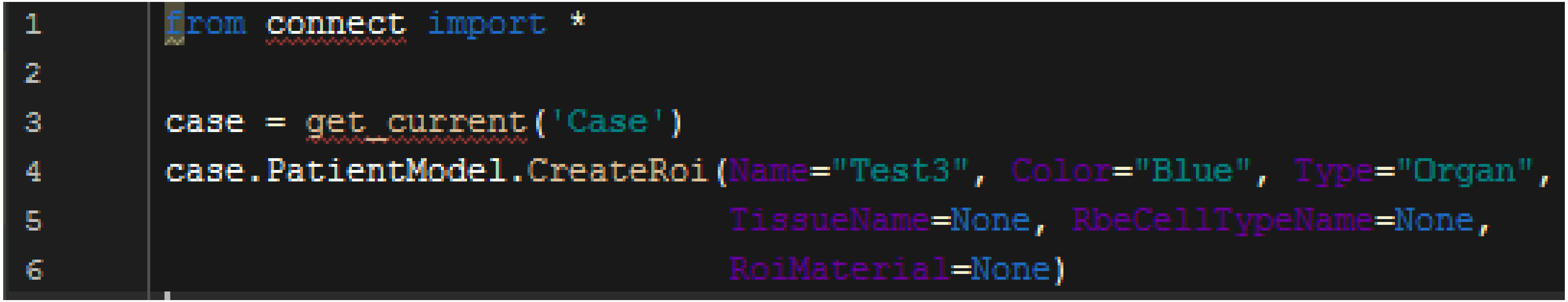
Results of Record

This can at least give you a hint of where to look in the statetree. Note that this is case.PatientModel



```
# Script recorded 21 Jun 2018

#   RayStation version: 6.99.0.14
#   Selected patient: ...

from connect import *

case = get_current("Case")


retval_0 = case.PatientModel.CreateRoi(Name="Temp3", Color="Red", Type="Organ", TissueName=None, RbeCel
```

Making A Script



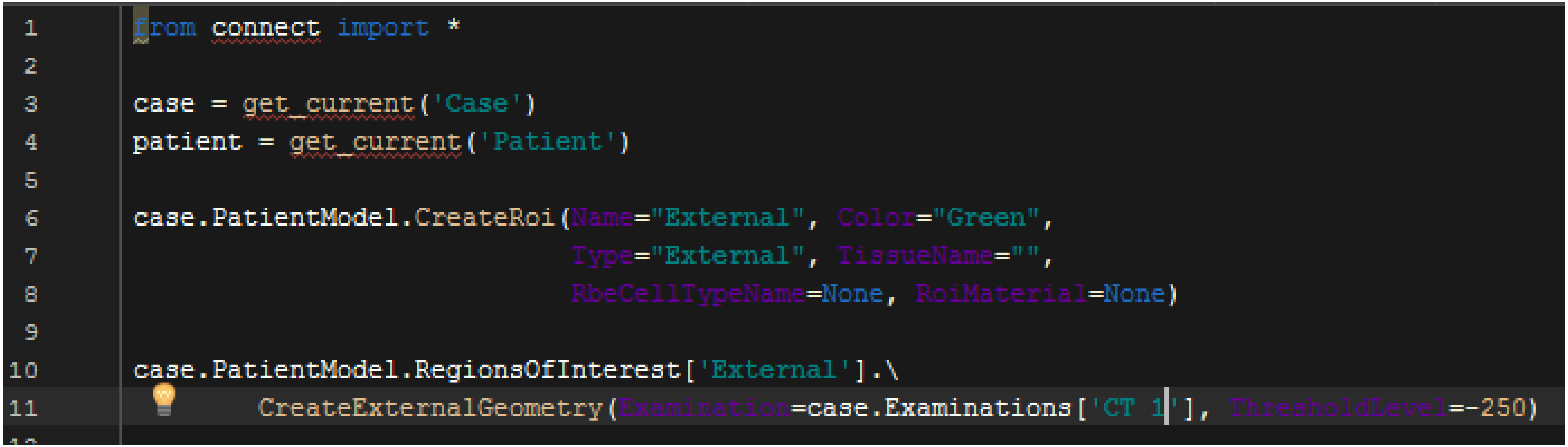
```
from connect import *

case = get_current('Case')
case.PatientModel.CreateRoi(Name="Test3", Color="Blue", Type="Organ",
                            TissueName=None, RbeCellTypeName=None,
                            RoiMaterial=None)
```

Making External ROI

- Can be helpful for rigid registrations

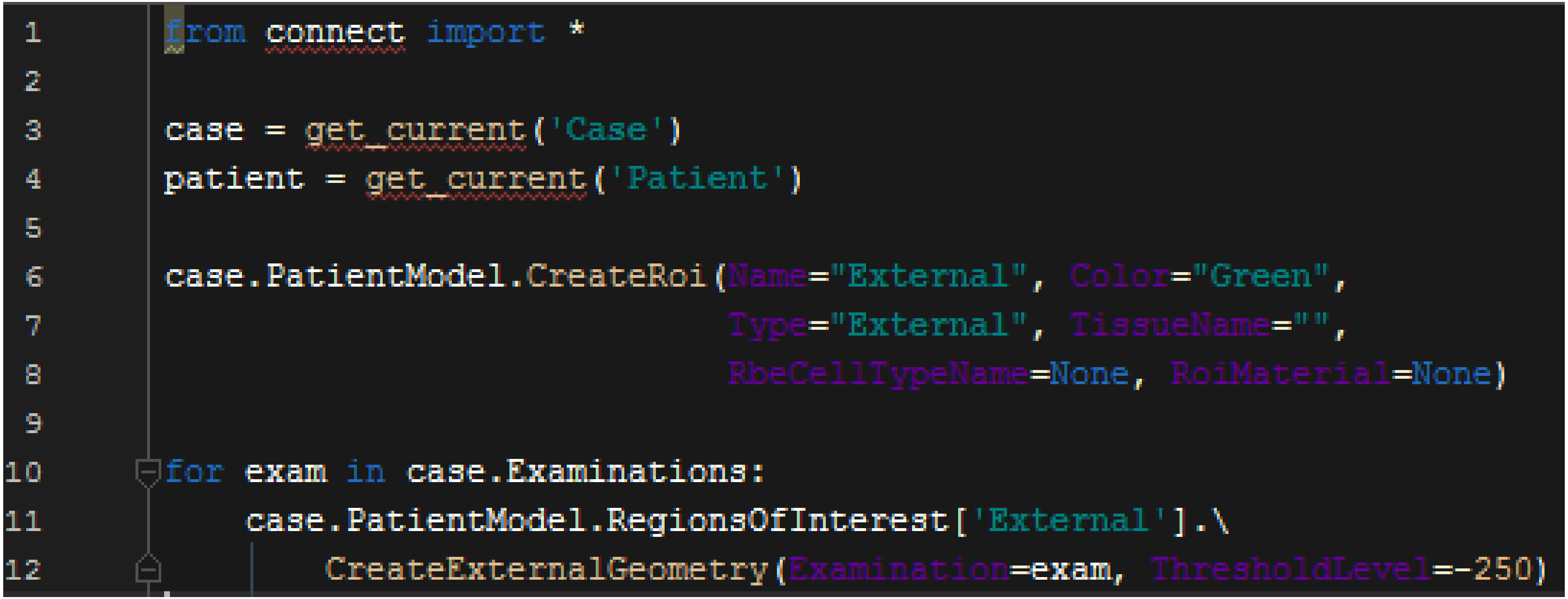
- Record action* But we want to repeat it across all exams

Make External ROI



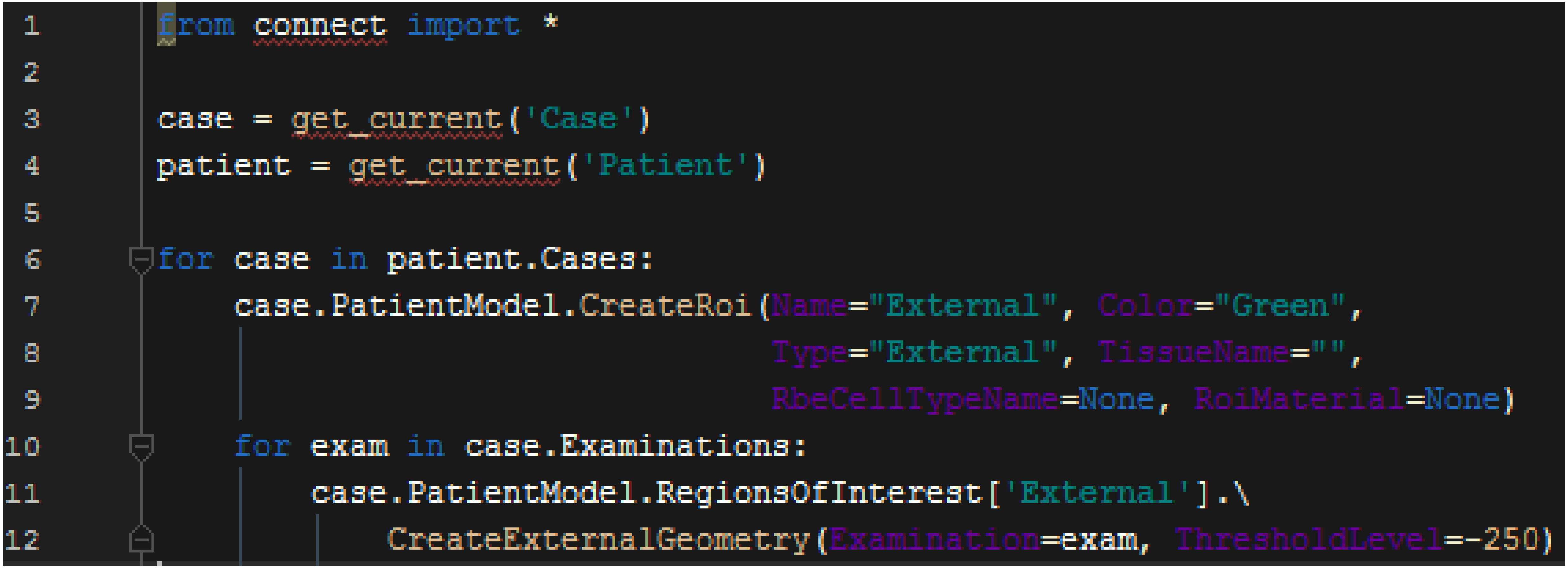
```
from connect import *

case = get_current('Case')
patient = get_current('Patient')

case.PatientModel.CreateRoi(Name="External", Color="Green",
                            Type="External", TissueName="",
                            RbeCellTypeName=None, RoiMaterial=None)

case.PatientModel.RegionsOfInterest['External'].\
        CreateExternalGeometry(Examination=case.Examinations['CT 1'], ThresholdLevel=-250)
```

Make External ROI across exams



```
from connect import *

case = get_current('Case')
patient = get_current('Patient')

case.PatientModel.CreateRoi(Name="External", Color="Green",
                            Type="External", TissueName="",
                            RbeCellTypeName=None, RoiMaterial=None)

for exam in case.Examinations:
    case.PatientModel.RegionsOfInterest['External'].\
        CreateExternalGeometry(Examination=exam, ThresholdLevel=-250)
```

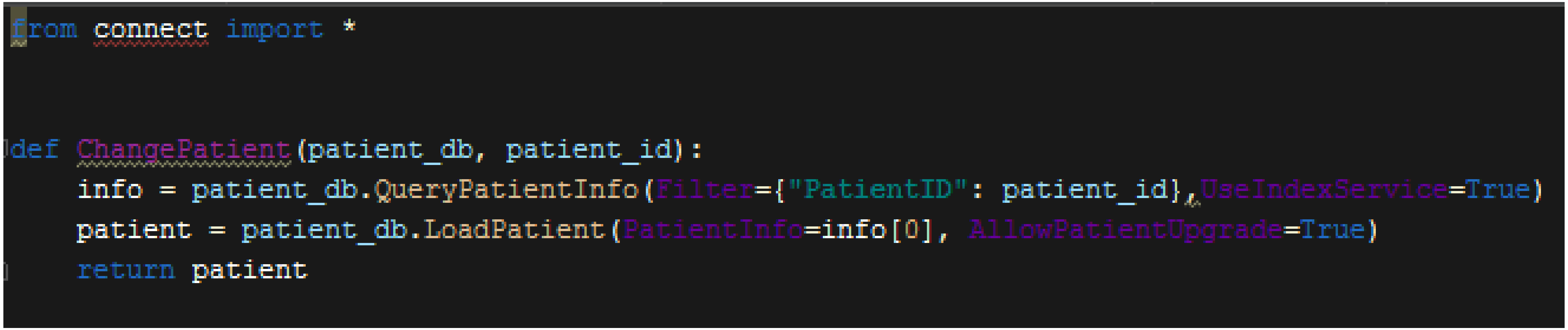
Make External ROI across Cases



```python
from connect import *

case = get_current('Case')
patient = get_current('Patient')

for case in patient.Cases:
    case.PatientModel.CreateRoi(Name="External", Color="Green",
                                Type="External", TissueName="",
                                RbeCellTypeName=None, RoiMaterial=None)
    for exam in case.Examinations:
        case.PatientModel.RegionsOfInterest['External'].\
            CreateExternalGeometry(Examination=exam, ThresholdLevel=-250)
```

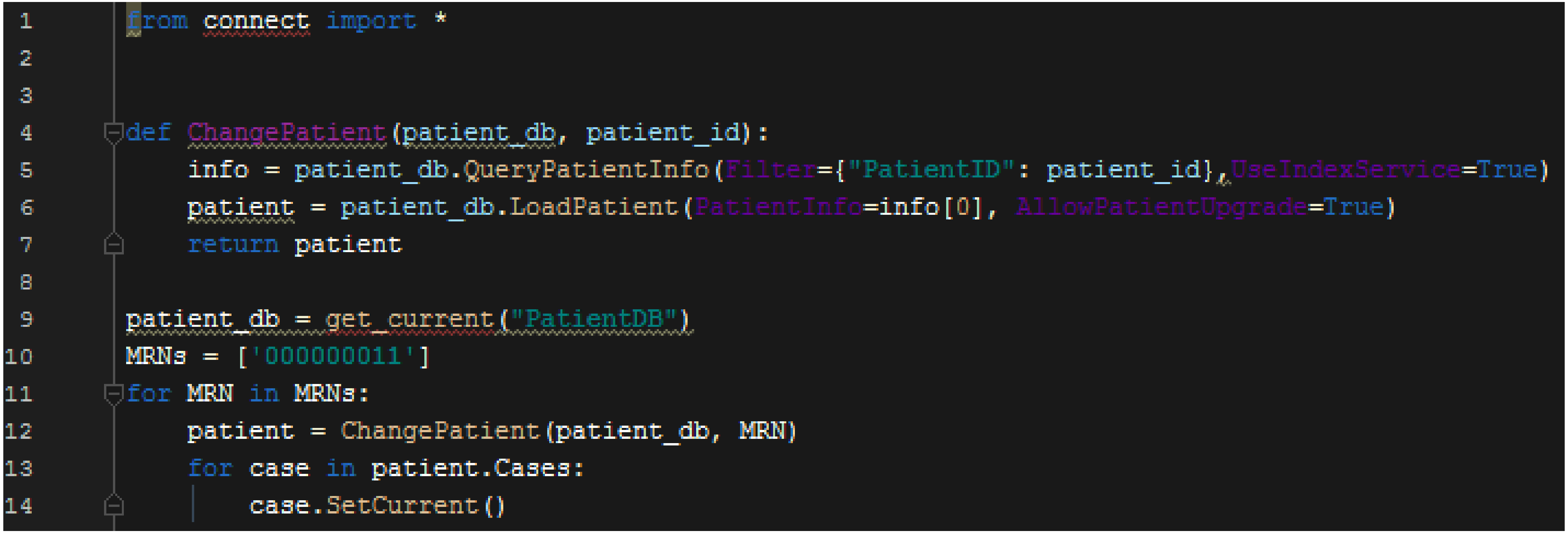
Make External across Patients and cases

- First need to load up a new patient..
- Define a way to query patients:

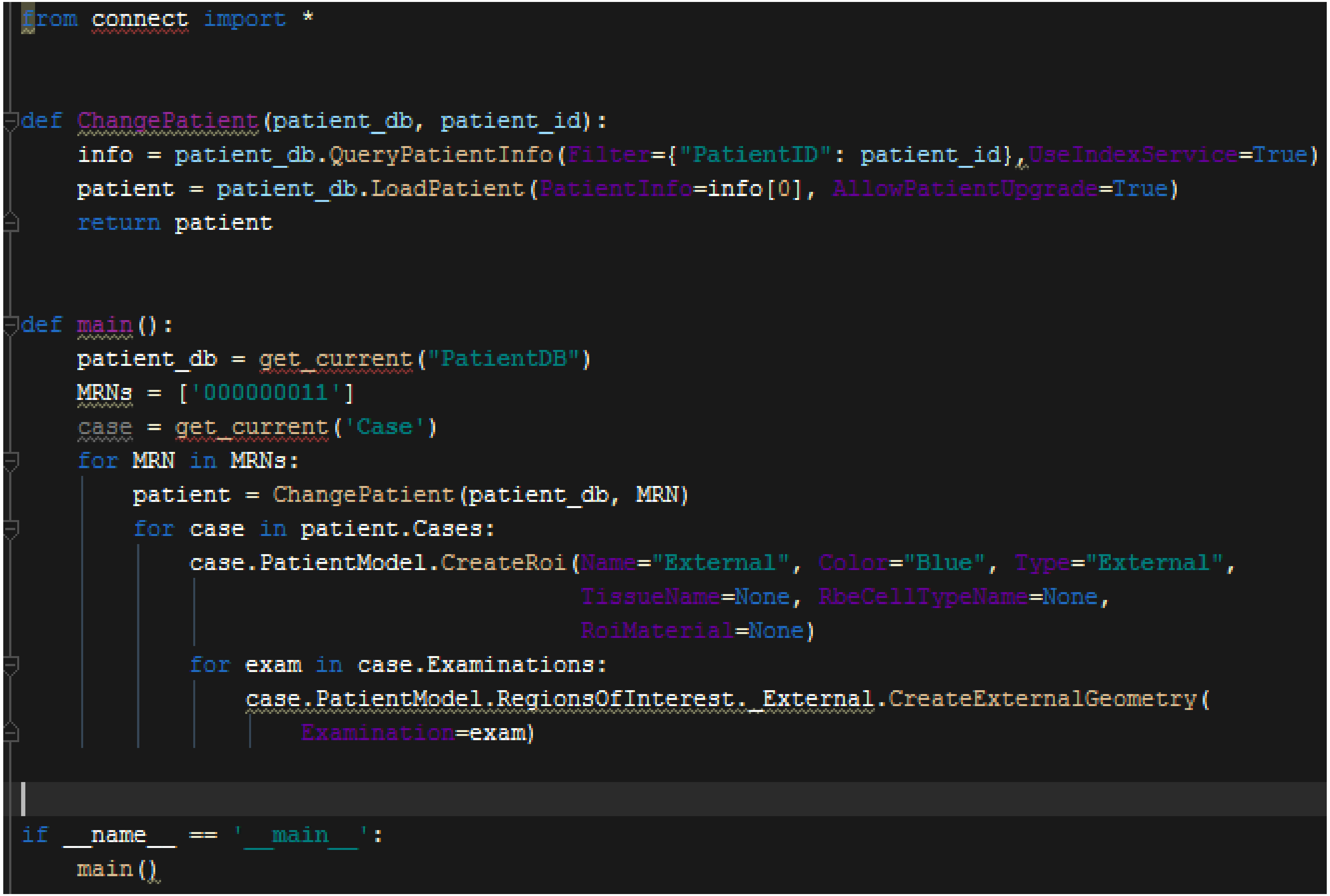


```python
from connect import *


def ChangePatient(patient_db, patient_id):
    info = patient_db.QueryPatientInfo(Filter={"PatientID": patient_id},UseIndexService=True)
    patient = patient_db.LoadPatient(PatientInfo=info[0], AllowPatientUpgrade=True)
    return patient
```

Load a patient



```python
from connect import *


def ChangePatient(patient_db, patient_id):
    info = patient_db.QueryPatientInfo(Filter={"PatientID": patient_id},UseIndexService=True)
    patient = patient_db.LoadPatient(PatientInfo=info[0], AllowPatientUpgrade=True)
    return patient

patient_db = get_current("PatientDB")
MRNs = ['000000011']
for MRN in MRNs:
    patient = ChangePatient(patient_db, MRN)
    for case in patient.Cases:
        case.SetCurrent()
```



```python
from connect import *


def ChangePatient(patient_db, patient_id):
    info = patient_db.QueryPatientInfo(Filter={"PatientID": patient_id},UseIndexService=True)
    patient = patient_db.LoadPatient(PatientInfo=info[0], AllowPatientUpgrade=True)
    return patient


def main():
    patient_db = get_current("PatientDB")
    MRNs = ['000000011']
    case = get_current('Case')
    for MRN in MRNs:
        patient = ChangePatient(patient_db, MRN)
        for case in patient.Cases:
            case.PatientModel.CreateRoi(Name="External", Color="Blue", Type="External",
                                        TissueName=None, RbeCellTypeName=None,
                                        RoiMaterial=None)
            for exam in case.Examinations:
                case.PatientModel.RegionsOfInterest._External.CreateExternalGeometry(
                    Examination=exam)


if __name__ == '__main__':
    main()
```

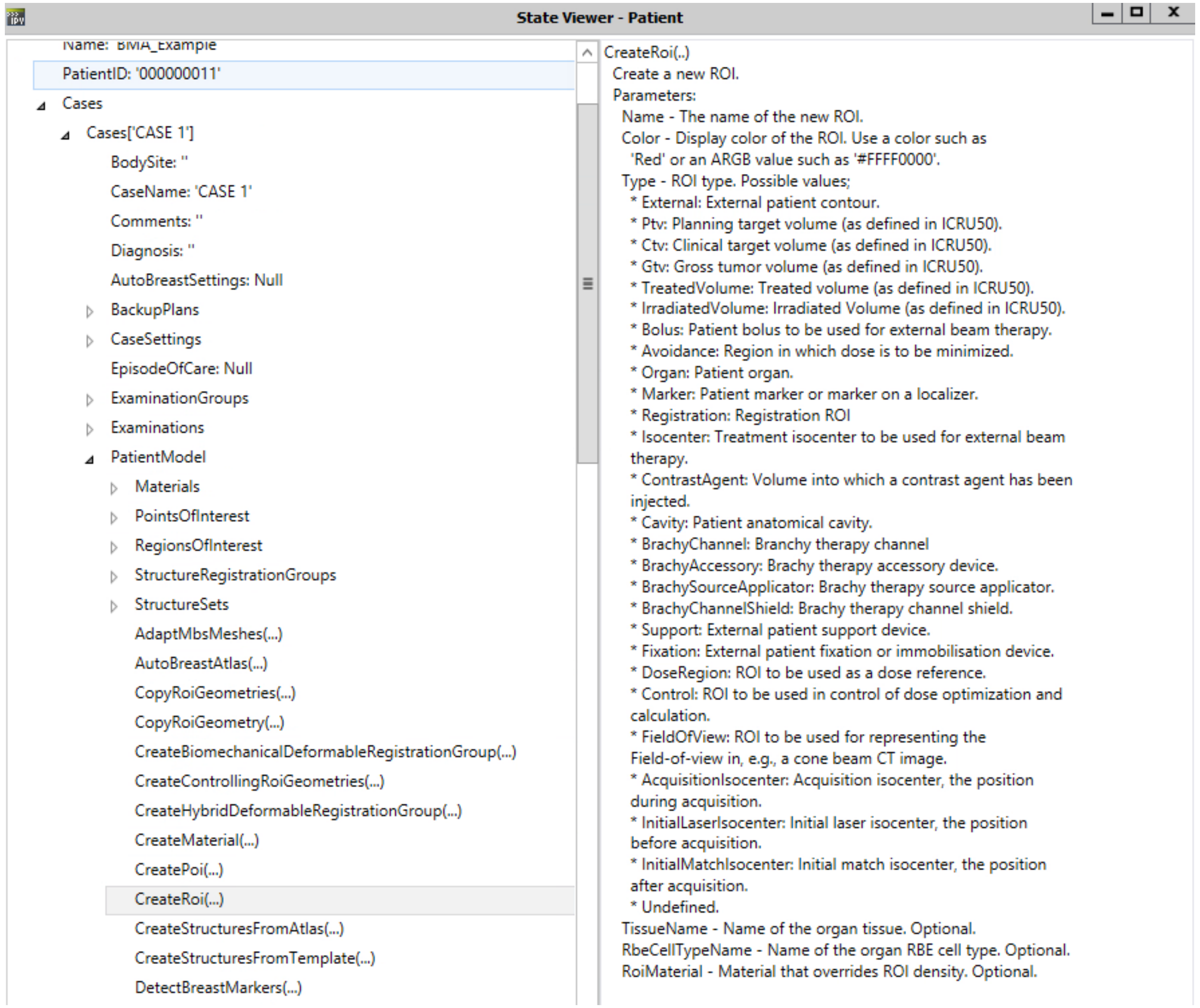
Script create a new ROI

- Open the console

- Run statetree.RunStateTree()

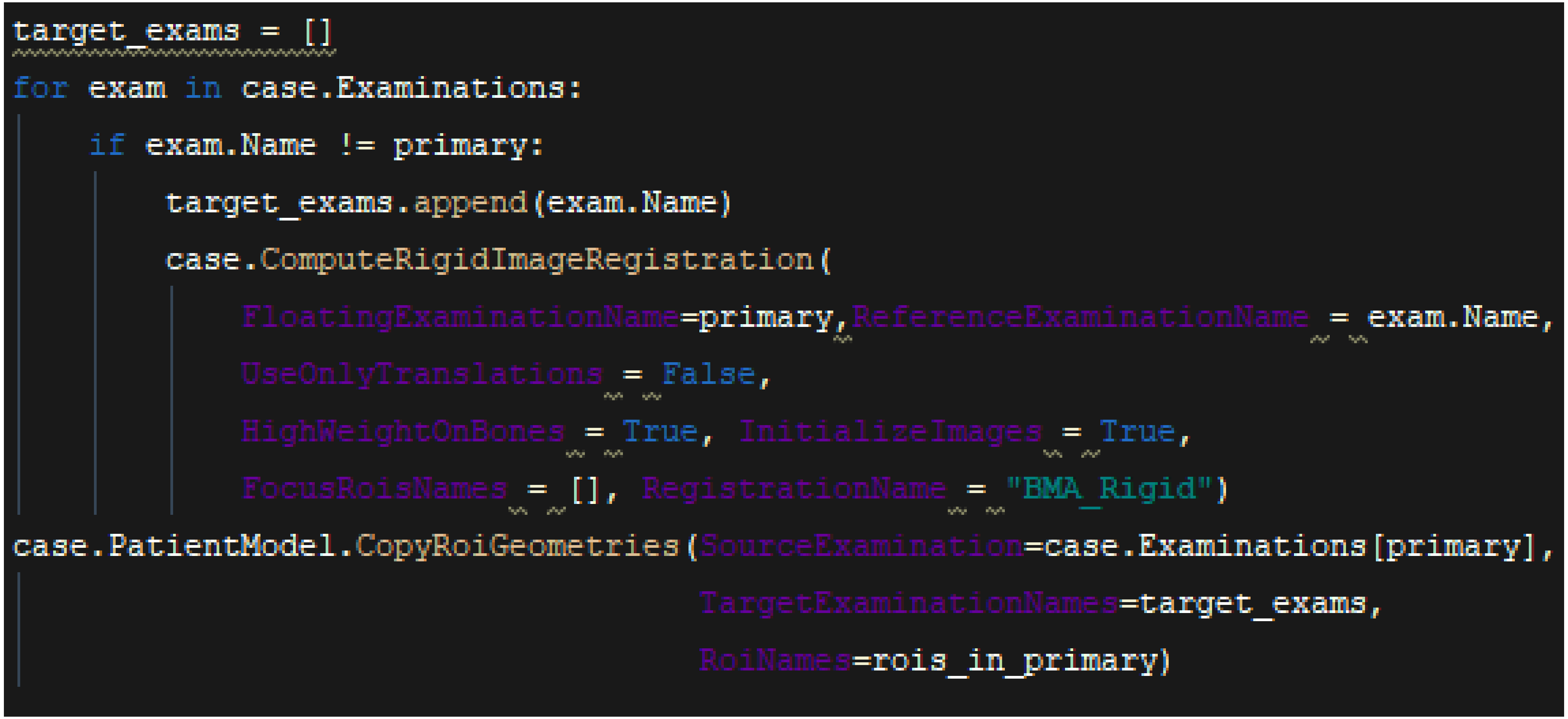
- See all available options

- Remember, we were under the PatientModel in raystation



```
Name: 'BMA_Example'
PatientID: '000000011'
◢ Cases
  ◢ Cases['CASE 1']
      BodySite: ''
      CaseName: 'CASE 1'
      Comments: ''
      Diagnosis: ''
      AutoBreastSettings: Null
    ▷ BackupPlans
    ▷ CaseSettings
      EpisodeOfCare: Null
    ▷ ExaminationGroups
    ▷ Examinations
    ◢ PatientModel
      ▷ Materials
      ▷ PointsOfInterest
      ▷ RegionsOfInterest
      ▷ StructureRegistrationGroups
      ▷ StructureSets
        AdaptMbsMeshes(...)
        AutoBreastAtlas(...)
        CopyRoiGeometries(...)
        CopyRoiGeometry(...)
        CreateBiomechanicalDeformableRegistrationGroup(...)
        CreateControllingRoiGeometries(...)
        CreateHybridDeformableRegistrationGroup(...)
        CreateMaterial(...)
        CreatePoi(...)
        CreateRoi(...)
        CreateStructuresFromAtlas(...)
        CreateStructuresFromTemplate(...)
        DetectBreastMarkers(...)
```

```
CreateRoi(..)
 Create a new ROI.
 Parameters:
  Name - The name of the new ROI.
  Color - Display color of the ROI. Use a color such as
   'Red' or an ARGB value such as '#FFFF0000'.
  Type - ROI type. Possible values;
   * External: External patient contour.
   * Ptv: Planning target volume (as defined in ICRU50).
   * Ctv: Clinical target volume (as defined in ICRU50).
   * Gtv: Gross tumor volume (as defined in ICRU50).
   * TreatedVolume: Treated volume (as defined in ICRU50).
   * IrradiatedVolume: Irradiated Volume (as defined in ICRU50).
   * Bolus: Patient bolus to be used for external beam therapy.
   * Avoidance: Region in which dose is to be minimized.
   * Organ: Patient organ.
   * Marker: Patient marker or marker on a localizer.
   * Registration: Registration ROI
   * Isocenter: Treatment isocenter to be used for external beam
   therapy.
   * ContrastAgent: Volume into which a contrast agent has been
   injected.
   * Cavity: Patient anatomical cavity.
   * BrachyChannel: Branchy therapy channel
   * BrachyAccessory: Brachy therapy accessory device.
   * BrachySourceApplicator: Brachy therapy source applicator.
   * BrachyChannelShield: Brachy therapy channel shield.
   * Support: External patient support device.
   * Fixation: External patient fixation or immobilisation device.
   * DoseRegion: ROI to be used as a dose reference.
   * Control: ROI to be used in control of dose optimization and
   calculation.
   * FieldOfView: ROI to be used for representing the
   Field-of-view in, e.g., a cone beam CT image.
   * AcquisitionIsocenter: Acquisition isocenter, the position
   during acquisition.
   * InitialLaserIsocenter: Initial laser isocenter, the position
   before acquisition.
   * InitialMatchIsocenter: Initial match isocenter, the position
   after acquisition.
   * Undefined.
  TissueName - Name of the organ tissue. Optional.
  RbeCellTypeName - Name of the organ RBE cell type. Optional.
  RoiMaterial - Material that overrides ROI density. Optional.
```



```python
target_exams = []
for exam in case.Examinations:
    if exam.Name != primary:
        target_exams.append(exam.Name)
        case.ComputeRigidImageRegistration(
            FloatingExaminationName=primary,ReferenceExaminationName = exam.Name,
            UseOnlyTranslations = False,
            HighWeightOnBones = True, InitializeImages = True,
            FocusRoisNames = [], RegistrationName = "BMA_Rigid")
case.PatientModel.CopyRoiGeometries(SourceExamination=case.Examinations[primary],
                                    TargetExaminationNames=target_exams,
                                    RoiNames=rois_in_primary)
```

Scriptability of actions

-
  ```
  case.ComputeRigidImageRegistration(FloatingExaminationName=Reference_Image,
                                     ReferenceExaminationName=exam.Name,
                                     UseOnlyTranslations=False,
                                     HighWeightOnBones=True,
  InitializeImages=True,
                                     FocusRoisNames=[],
                                     RegistrationName="BMA_Rigid")
  ```

Thank you!