# 学习使用Python操作ParaView

To get started with ParaView from python we recommend that you:

1 use Tools->Start Trace in the desktop application to record a python script.

2 try a few scripts in the Tools->Python Shell of the desktop application.

use the pvpython command line executable instead of the desktop application.

read the Paraview Guide, which has scripting instructions throughout.

see the online documentation (of just help() from within python) to see the detailed syntax.

如何使用Xbox360 Controller控制动画或in-situ visualization过程中，相机的视觉位置？通过VRPlugin的VRPN实现！

# 使用Python运行paraview

我们知道各种程序的脚本就是解决重复操作的，可以说是非常良心的功能啦。Paraview通过Python同样提供了丰富的脚本功能，按使用方式可以分成以下几种：

1、作为paraview客户端的一部分使用（Python shell），可以在菜单栏的view中打开；

2、支持MPI的批处理应用（pvbatch），可在安装目录中找到（必须是并行版）；

3、单独作为客户端使用（pvpython），可在Windows的开始目录中找到；

4、使用Python的任一种开发环境运行paraview。

前三种可以在paraview的user guide中找到使用方法，下面重点介绍第四种，在Python的开发环境pycharm中使用paraview。

首先需要编译paraview，然后按照下述步骤进行配置：

1、创建环境变量PYTHONPATH，用来链接paraview的动态库，下图是我的配置

2、将paraview编译好的可执行文件的路径加入环境变量PATH中，下图是我的配置。其中buildParaView552是我编译安装paraview的位置，bin/Debug是编译生成的可执行文件的位置。

3\配置好以上环境变量，下载安装pycharm（按照提示安装即可），下面通过一个小例子演示在pycharm中运行paraview。先看代码：

# 通过Python使用paraview需要加入这个模块

from paraview.simple import \*

# 创建一个球体

sphere = Sphere()

# 准备显示球体

Show()

# 渲染

Render()

以上代码会创建一个球体并显示，效果如下图

Ok，后边就可以在pycharm中愉快地使用paraview的各种功能啦。对于不了解的Python命令，可以使用paraview的Trace功能进行查看。

# ParaView\_5.8.1手册中Python脚本命令摘录

## 1.5 Getting started with pvpython

1.5.1 pvpython scripting interface

pvpython, pvbatch

1.5.2 Understanding the visualization process

1.5.3 Updating the pipeline

## 1.6 Scripting in paraview

1.6.1 The Python Shell

View -> python shell

1.6.2 Tracing actions for scripting

Tools -> Start Trace

Tools -> Stop Trace

## 2.2 Opening data files in pvpython

2.2.1 Handling temporal file series

2.2.2 Dealing with time

2.2.3 Common properties on readers

## 2.3 Reloading files

## 3.3 Getting data information in pvpython

## 4.1 Multiple views

4.1.2 Multiple views in pvpython

## 4.2 View properties

4.2.2 View properties in pvpython

## 4.3 Display properties

4.3.2 Display properties in pvpython

## 4.4 Render View

4.4.3 Render View in pvpython

## 4.5 Line Chart View

4.5.3 Line Chart View in pvpython

## 4.11 Slice View

4.11.2 Slice View in pvpython

## 4.12 Python View

Some Python libraries, such as matplotlib, are widely used for making publication-quality plots of data. The Python View provides a way to display plots made in a Python script right within paraview.

## 5.3 Creating filters in pvpython

5.3.1 Multiple input connections

5.3.2 Multiple input ports

5.3.3 Changing input connections

## 5.5 Changing filter properties in pvpython

## 5.6 Filters for sub-setting data

5.6.1Clip in pvpython

5.6.4 Threshold in pvpython

## 5.7 Filters for geometric manipulation

5.7.3 Transform in pvpython

## 5.8 Filters for sampling

5.8.3 Stream Tracer

## 8.1 Saving datasets

## 8.2 Saving rendered results

8.2.1 Saving screenshots

8.2.2 Exporting scenes

## 8.3 Saving animation

## 8.4 Saving state

# 维基百科

<https://kitware.github.io/paraview-docs/latest/python/>

ParaView offers rich scripting support through Python. This support is available as part of the ParaView client (paraview), an MPI-enabled batch application (pvbatch), the ParaView python client (pvpython), or any other Python-enabled application. Using Python, users and developers can gain access to the ParaView visualization engine.

## Quick-Start

Getting Started

设置环境变量PYTHONPATH，定位ParaView的二进制路径和modules的路径。paraview/simple.py, paraview/vtk.py etc

例如:

export PYTHONPATH=/Users/berk/work/paraview3-build/lib:/Users/berk/work/paraview3-build/lib/site-packages

还要设置搜索动态链接库的路径的环境变量LD\_LIBRARY\_PATH

export LD\_LIBRARY\_PATH=/Users/berk/work/paraview3-build/lib

如果用pvpython或pvbatch运行脚本，就不用设置PYTHONPATH。

加载servermanager module：

>>> from paraview.simple import \*

Tab键补全代码功能

创建变量PYTHONSTARTUP

export PYTHONSTARTUP = /home/<username>/.pythonrc

where .pythonrc is:

# ~/.pythonrc

# enable syntax completion

try:

import readline

except ImportError:

print "Module readline not available."

else:

import rlcompleter

readline.parse\_and\_bind("tab: complete")

Creating a Pipeline

Start by creating a Cone object:

>>> cone = Cone()

You can get some documentation about the cone object using help().

>>> help(cone)

Next, apply a shrink filter to the cone:

>>> shrinkFilter = Shrink(cone)

Rendering

Now that you’ve created a small pipeline, render the result. You will need two objects to render the output of an algorithm in a scene: a representation and a view. A representation is responsible for taking a data object and rendering it in a view. A view is responsible for managing a render context and a collection of representations. Simple creates a view by default. The representation object is created automatically with Show().

>>> Show(shrinkFilter)

>>> Render()

In this example the value returned by Cone() and Shrink() was assigned to Python variables and used to build the pipeline. ParaView keeps track of the last pipeline object created by the user. This allows you to accomplish everything you did above using the following code:

## paraview Package

paraview Package

The paraview package provides modules used to script ParaView. Generally, users should import the modules of interest directly e.g.:

from paraview.simple import \*

Modules

* [\_backwardscompatibilityhelper Module](https://kitware.github.io/paraview-docs/latest/python/paraview._backwardscompatibilityhelper.html)
* [\_colorMaps Module](https://kitware.github.io/paraview-docs/latest/python/paraview._colorMaps.html)
* [collaboration Module](https://kitware.github.io/paraview-docs/latest/python/paraview.collaboration.html)
* [coprocessing Module](https://kitware.github.io/paraview-docs/latest/python/paraview.coprocessing.html)
* [cpstate Module](https://kitware.github.io/paraview-docs/latest/python/paraview.cpstate.html)
* [live Module](https://kitware.github.io/paraview-docs/latest/python/paraview.live.html)
* [lookuptable Module](https://kitware.github.io/paraview-docs/latest/python/paraview.lookuptable.html)
* [numeric Module](https://kitware.github.io/paraview-docs/latest/python/paraview.numeric.html)
* [numpy\_support Module](https://kitware.github.io/paraview-docs/latest/python/paraview.numpy_support.html)
* [pv-vtk-all Module](https://kitware.github.io/paraview-docs/latest/python/paraview.pv-vtk-all.html)
* [python\_view Module](https://kitware.github.io/paraview-docs/latest/python/paraview.python_view.html)
* [selection Module](https://kitware.github.io/paraview-docs/latest/python/paraview.selection.html)
* [servermanager Module](https://kitware.github.io/paraview-docs/latest/python/paraview.servermanager.html)
* [simple Module](https://kitware.github.io/paraview-docs/latest/python/paraview.simple.html)
* [Available readers, sources, writers, filters and animation cues](https://kitware.github.io/paraview-docs/latest/python/paraview.servermanager_proxies.html)
* [smstate Module](https://kitware.github.io/paraview-docs/latest/python/paraview.smstate.html)
* [smtesting Module](https://kitware.github.io/paraview-docs/latest/python/paraview.smtesting.html)
* [smtrace Module](https://kitware.github.io/paraview-docs/latest/python/paraview.smtrace.html)
  + [Developer Documentation](https://kitware.github.io/paraview-docs/latest/python/paraview.smtrace.html#developer-documentation)
  + [Notes about references](https://kitware.github.io/paraview-docs/latest/python/paraview.smtrace.html#notes-about-references)
* [spatiotemporalparallelism Module](https://kitware.github.io/paraview-docs/latest/python/paraview.spatiotemporalparallelism.html)
* [variant Module](https://kitware.github.io/paraview-docs/latest/python/paraview.variant.html)
* [vtk Module](https://kitware.github.io/paraview-docs/latest/python/paraview.vtk.html)

Subpackages

* [algorithms Package](https://kitware.github.io/paraview-docs/latest/python/paraview.algorithms.html)
  + [algorithms Package](https://kitware.github.io/paraview-docs/latest/python/paraview.algorithms.html#id1)
  + [Modules](https://kitware.github.io/paraview-docs/latest/python/paraview.algorithms.html#modules)
    - [algorithms.openpmd Module](https://kitware.github.io/paraview-docs/latest/python/paraview.algorithms.openpmd.html)
* [apps Package](https://kitware.github.io/paraview-docs/latest/python/paraview.apps.html)
  + [apps Package](https://kitware.github.io/paraview-docs/latest/python/paraview.apps.html#id1)
  + [Modules](https://kitware.github.io/paraview-docs/latest/python/paraview.apps.html#modules)
    - [apps.\_\_main\_\_ Module](https://kitware.github.io/paraview-docs/latest/python/paraview.apps.__main__.html)
    - [apps.\_internals Module](https://kitware.github.io/paraview-docs/latest/python/paraview.apps._internals.html)
    - [apps.divvy Module](https://kitware.github.io/paraview-docs/latest/python/paraview.apps.divvy.html)
    - [apps.flow Module](https://kitware.github.io/paraview-docs/latest/python/paraview.apps.flow.html)
    - [apps.glance Module](https://kitware.github.io/paraview-docs/latest/python/paraview.apps.glance.html)
    - [apps.lite Module](https://kitware.github.io/paraview-docs/latest/python/paraview.apps.lite.html)
    - [apps.visualizer Module](https://kitware.github.io/paraview-docs/latest/python/paraview.apps.visualizer.html)
* [benchmark Package](https://kitware.github.io/paraview-docs/latest/python/paraview.benchmark.html)
  + [benchmark Package](https://kitware.github.io/paraview-docs/latest/python/paraview.benchmark.html#id1)
  + [Modules](https://kitware.github.io/paraview-docs/latest/python/paraview.benchmark.html#modules)
    - [benchmark.basic Module](https://kitware.github.io/paraview-docs/latest/python/paraview.benchmark.basic.html)
    - [benchmark.logbase Module](https://kitware.github.io/paraview-docs/latest/python/paraview.benchmark.logbase.html)
    - [benchmark.logparser Module](https://kitware.github.io/paraview-docs/latest/python/paraview.benchmark.logparser.html)
    - [benchmark.manyspheres Module](https://kitware.github.io/paraview-docs/latest/python/paraview.benchmark.manyspheres.html)
    - [benchmark.waveletcontour Module](https://kitware.github.io/paraview-docs/latest/python/paraview.benchmark.waveletcontour.html)
    - [benchmark.waveletvolume Module](https://kitware.github.io/paraview-docs/latest/python/paraview.benchmark.waveletvolume.html)
* [catalyst Package](https://kitware.github.io/paraview-docs/latest/python/paraview.catalyst.html)
  + [catalyst Package](https://kitware.github.io/paraview-docs/latest/python/paraview.catalyst.html#id1)
  + [Modules](https://kitware.github.io/paraview-docs/latest/python/paraview.catalyst.html#modules)
    - [catalyst.bridge Module](https://kitware.github.io/paraview-docs/latest/python/paraview.catalyst.bridge.html)
    - [catalyst.detail Module](https://kitware.github.io/paraview-docs/latest/python/paraview.catalyst.detail.html)
    - [catalyst.importers Module](https://kitware.github.io/paraview-docs/latest/python/paraview.catalyst.importers.html)
    - [catalyst.v2\_internals Module](https://kitware.github.io/paraview-docs/latest/python/paraview.catalyst.v2_internals.html)
* [demos Package](https://kitware.github.io/paraview-docs/latest/python/paraview.demos.html)
  + [Modules](https://kitware.github.io/paraview-docs/latest/python/paraview.demos.html#modules)
    - [demos.export\_catalyst\_state Module](https://kitware.github.io/paraview-docs/latest/python/paraview.demos.export_catalyst_state.html)
    - [demos.filedriver\_miniapp Module](https://kitware.github.io/paraview-docs/latest/python/paraview.demos.filedriver_miniapp.html)
    - [demos.show\_grid\_as\_background Module](https://kitware.github.io/paraview-docs/latest/python/paraview.demos.show_grid_as_background.html)
    - [demos.wavelet\_miniapp Module](https://kitware.github.io/paraview-docs/latest/python/paraview.demos.wavelet_miniapp.html)
    - [demos.wavelet\_miniapp\_plugin Module](https://kitware.github.io/paraview-docs/latest/python/paraview.demos.wavelet_miniapp_plugin.html)
* [detail Package](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.html)
  + [detail Package](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.html#id1)
  + [Modules](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.html#modules)
    - [detail.annotation Module](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.annotation.html)
    - [detail.calculator Module](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.calculator.html)
    - [detail.catalyst\_export Module](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.catalyst_export.html)
    - [detail.cdbwriter Module](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.cdbwriter.html)
    - [detail.exportnow Module](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.exportnow.html)
    - [detail.extract\_selection Module](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.extract_selection.html)
    - [detail.loghandler Module](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.loghandler.html)
    - [detail.python\_selector Module](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.python_selector.html)
    - [detail.pythonalgorithm Module](https://kitware.github.io/paraview-docs/latest/python/paraview.detail.pythonalgorithm.html)
* [tests Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tests.html)
  + [Modules](https://kitware.github.io/paraview-docs/latest/python/paraview.tests.html#modules)
    - [tests.validate\_extracts Module](https://kitware.github.io/paraview-docs/latest/python/paraview.tests.validate_extracts.html)
    - [tests.verify\_eyedomelighting Module](https://kitware.github.io/paraview-docs/latest/python/paraview.tests.verify_eyedomelighting.html)
* [tpl Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.html)
  + [Subpackages](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.html#subpackages)
    - [cinema\_python Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinema_python.html)
      * [Subpackages](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinema_python.html#subpackages)
        + [adaptors Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinema_python.adaptors.html)

[Subpackages](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinema_python.adaptors.html#subpackages)

[paraview Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinema_python.adaptors.paraview.html)

* + - * + [database Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinema_python.database.html)
        + [images Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinema_python.images.html)
    - [cinemasci Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinemasci.html)
      * [Subpackages](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinemasci.html#subpackages)
        + [cis Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinemasci.cis.html)

[Subpackages](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinemasci.cis.html#subpackages)

[read Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinemasci.cis.read.html)

[write Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinemasci.cis.write.html)

* + - * + [server Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinemasci.server.html)
        + [viewers Package](https://kitware.github.io/paraview-docs/latest/python/paraview.tpl.cinemasci.viewers.html)
* [util Package](https://kitware.github.io/paraview-docs/latest/python/paraview.util.html)
  + [util Package](https://kitware.github.io/paraview-docs/latest/python/paraview.util.html#id1)
  + [Modules](https://kitware.github.io/paraview-docs/latest/python/paraview.util.html#modules)
    - [util.vtkAlgorithm Module](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html)
      * [Introduction](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#introduction)
      * [Decorator Basics](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#decorator-basics)
      * [*smproxy* Decorators](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproxy-decorators)
        + [Common decorator parameters](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#common-decorator-parameters)
        + [*smproxy.source*](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproxy-source)
        + [*smproxy.filter*](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproxy-filter)
        + [*smproxy.reader*](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproxy-reader)
        + [*smproxy.writer*](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproxy-writer)
      * [*smproperty* Decorators](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproperty-decorators)
        + [Common decorator parameters](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#id1)
        + [*smproperty.xml*](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproperty-xml)
        + [*smproperty.intvector*, *smproperty.doublevector*, *smproperty.idtypevector*, *smproperty.stringvector*](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproperty-intvector-smproperty-doublevector-smproperty-idtypevector-smproperty-stringvector)
        + [*smproperty.proxy*](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproperty-proxy)
        + [*smproperty.input*](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproperty-input)
        + [*smproperty.dataarrayselection*](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smproperty-dataarrayselection)
      * [*smdomain* Decorators](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smdomain-decorators)
      * [*smhint* Decorators](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#smhint-decorators)
      * [Examples](https://kitware.github.io/paraview-docs/latest/python/paraview.util.vtkAlgorithm.html#examples)

## simple Module

simple is a module for using paraview server manager in Python. It provides a simple convenience layer to functionality provided by the C++ classes wrapped to Python as well as the servermanager module.

A simple example:

from paraview.simple import \*

# Create a new sphere proxy on the active connection and register it

# in the sources group.

sphere = Sphere(ThetaResolution=16, PhiResolution=32)

# Apply a shrink filter

shrink = Shrink(sphere)

# Turn the visibility of the shrink object on.

Show(shrink)

# Render the scene

Render()

## servermanager Module

servermanager is a module for using paraview server manager in Python. One can always use the server manager API directly. However, this module provides an interface easier to use from Python by wrapping several VTK classes around Python classes.

Note that, upon load, this module will create several sub-modules: sources, filters and rendering. These modules can be used to instantiate specific proxy types. For a list, try “dir(servermanager.sources)”

Usually users should use the paraview.simple module instead as it provide a more user friendly API.

A simple example:

**from** paraview.servermanager **import** **\***

# Creates a new built-in session and makes it the active session.

Connect()

# Creates a new render view on the active session.

renModule = CreateRenderView()

# Create a new sphere proxy on the active session and register it

# in the sources group.

sphere = sources.SphereSource(registrationGroup="sources", ThetaResolution=16, PhiResolution=32)

# Create a representation for the sphere proxy and adds it to the render

# module.

display = CreateRepresentation(sphere, renModule)

renModule.StillRender()

## coprocessing Module

This module is designed for use in co-processing Python scripts. It provides a class, Pipeline, which is designed to be used as the base-class for Python pipeline. Additionally, this module has several other utility functions that are appropriate for co-processing.

## Available readers, sources, writers, filters and animation cues