# Publishing Notebooks

Martin Hunt
mmh@purdue.edu
Purdue University

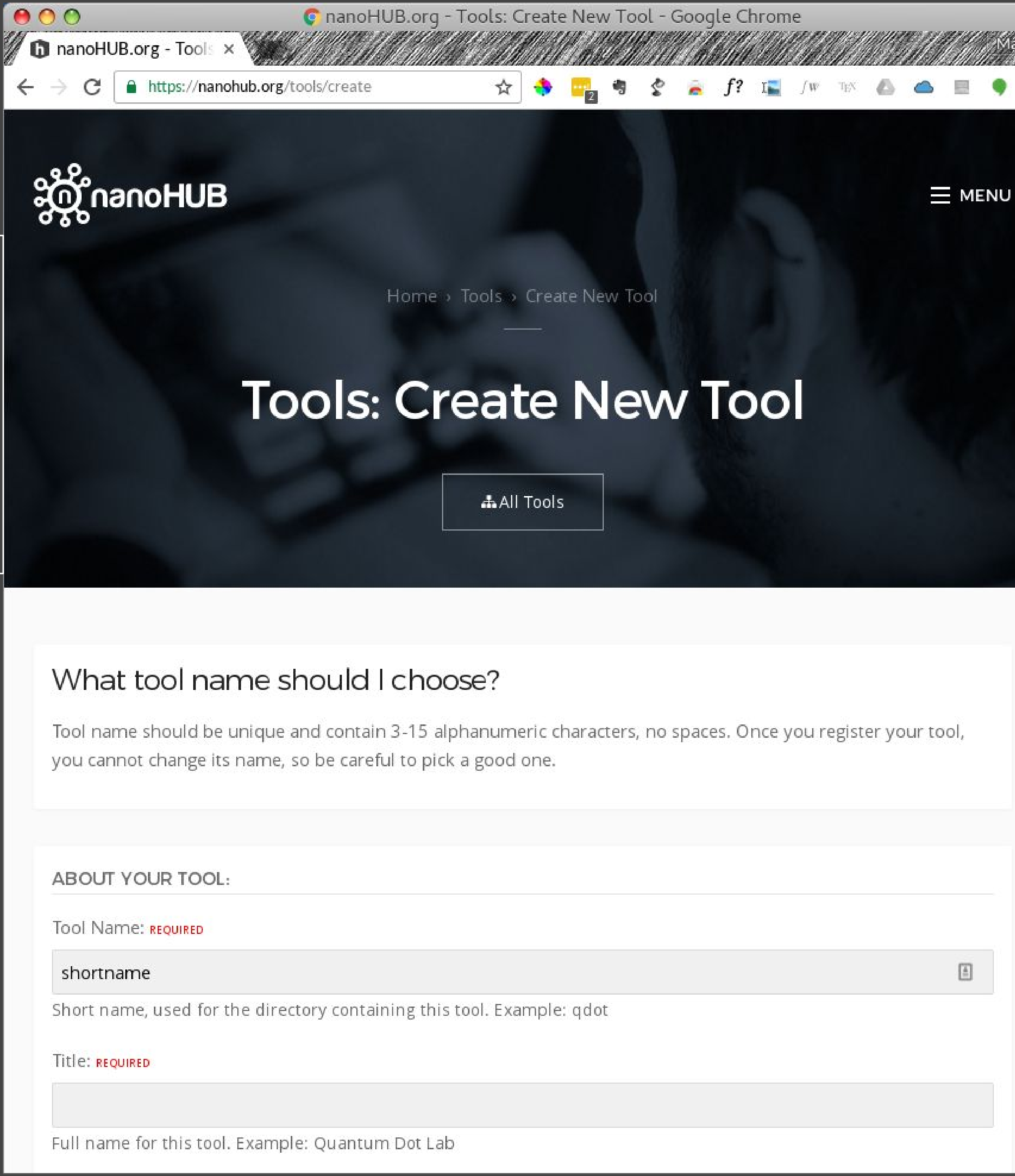
# How to Publish a Notebook on the Hubs

- Publishing a notebook shares it with the public or a group.

- The steps are very similar to publishing other materials.

- You should understand the publishing process before continuing. See https://nanohub.org/resources/3865/download/tools.pdf

To start, go to https://nanohub.org/tools/create.


nanoHUB.org - Tools: Create New Tool - Google Chrome
nanoHUB.org - Tools
https://nanohub.org/tools/create
nanoHUB
MENU
Home › Tools › Create New Tool
All Tools

# Tools: Create New Tool

## What tool name should I choose?

Tool name should be unique and contain 3-15 alphanumeric characters, no spaces. Once you register your tool, you cannot change its name, so be careful to pick a good one.

ABOUT YOUR TOOL:

Tool Name: REQUIRED

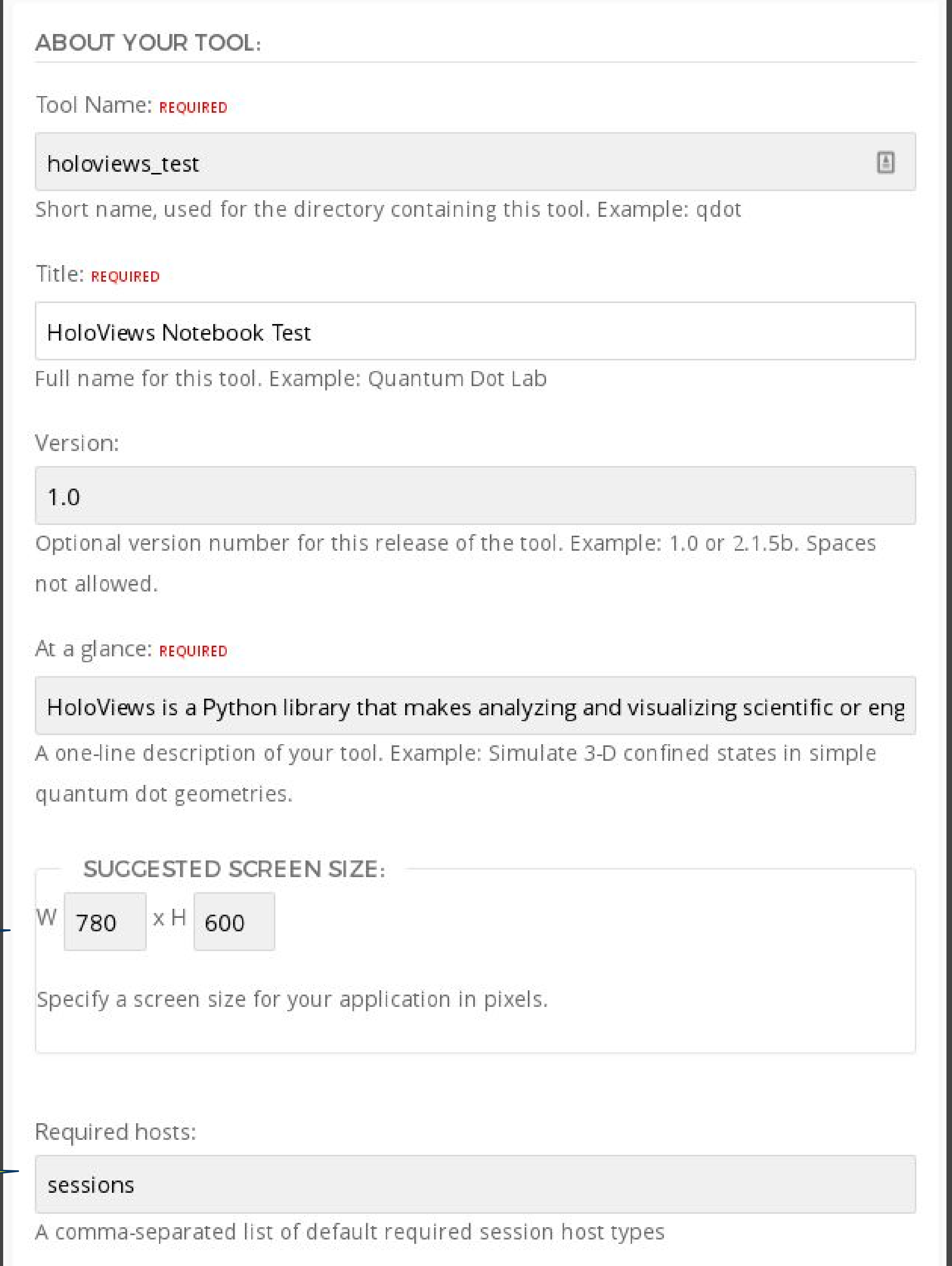
shortname

Short name, used for the directory containing this tool. Example: qdot

Title: REQUIRED

Full name for this tool. Example: Quantum Dot Lab

Fill in the fields.



ABOUT YOUR TOOL:

Tool Name: REQUIRED

holoviews_test

Short name, used for the directory containing this tool. Example: qdot

Title: REQUIRED

HoloViews Notebook Test

Full name for this tool. Example: Quantum Dot Lab

Version:

1.0

Optional version number for this release of the tool. Example: 1.0 or 2.1.5b. Spaces not allowed.

At a glance: REQUIRED

HoloViews is a Python library that makes analyzing and visualizing scientific or eng

A one-line description of your tool. Example: Simulate 3-D confined states in simple quantum dot geometries.

SUGGESTED SCREEN SIZE:

W 780 x H 600

Specify a screen size for your application in pixels.

Ignore

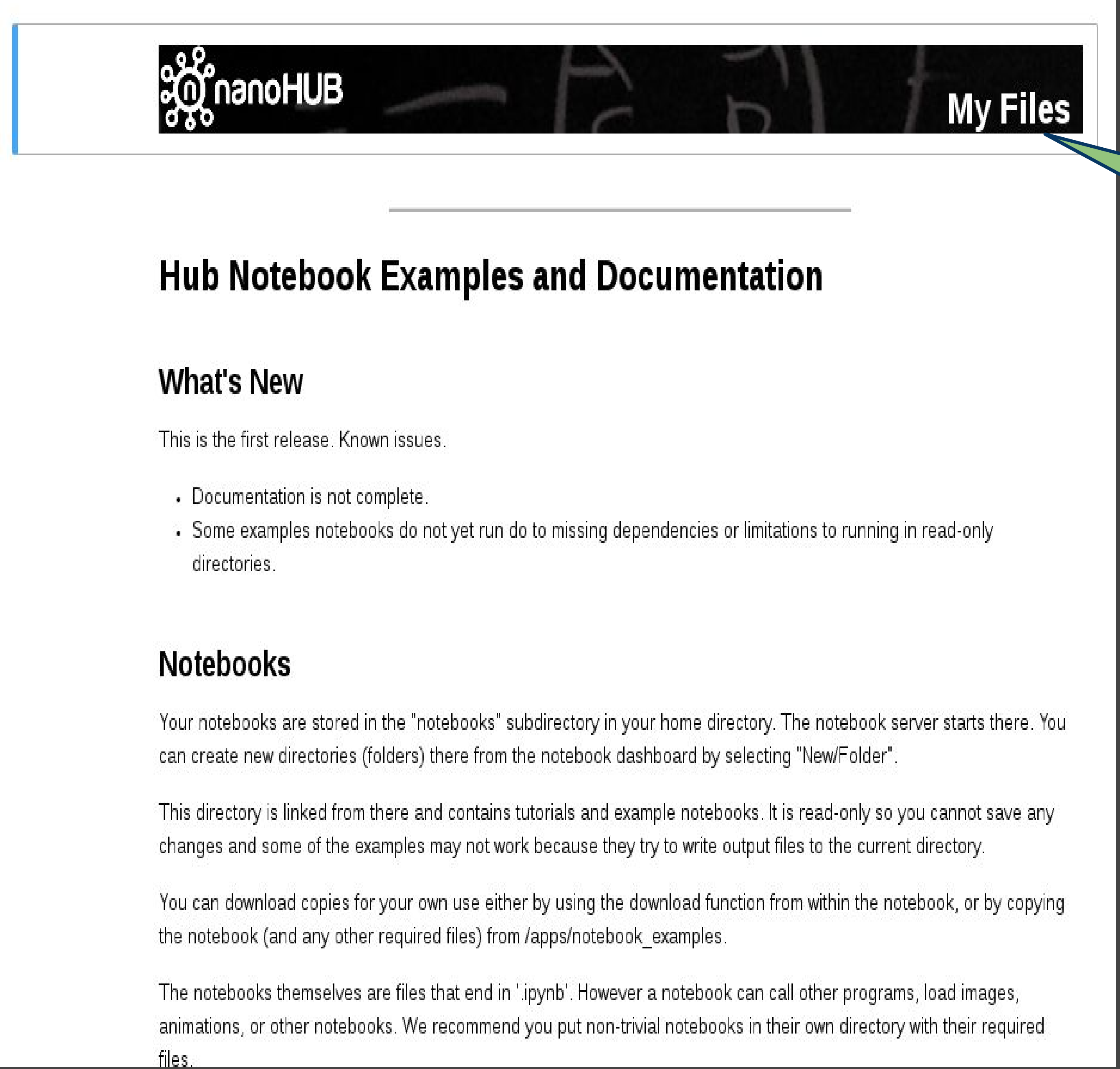
Required hosts:

sessions

A comma-separated list of default required session host types

Ignore

Now start the jupyter notebook tool.
https://nanohub.org/tools/jupyter



## Hub Notebook Examples and Documentation

### What's New

This is the first release. Known issues.

- Documentation is not complete.
- Some examples notebooks do not yet run do to missing dependencies or limitations to running in read-only directories.

### Notebooks

Your notebooks are stored in the "notebooks" subdirectory in your home directory. The notebook server starts there. You can create new directories (folders) there from the notebook dashboard by selecting "New/Folder".

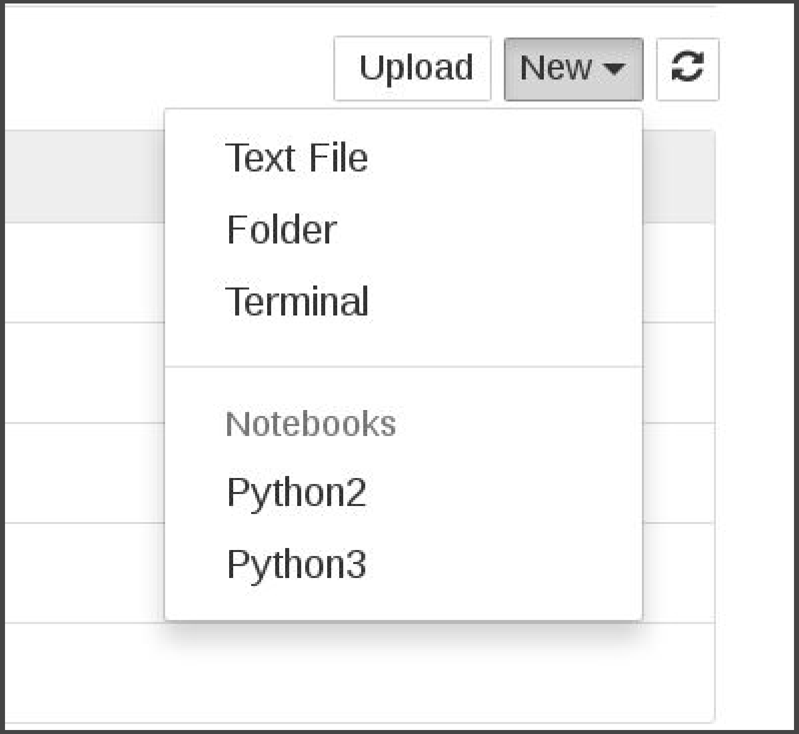
This directory is linked from there and contains tutorials and example notebooks. It is read-only so you cannot save any changes and some of the examples may not work because they try to write output files to the current directory.

You can download copies for your own use either by using the download function from within the notebook, or by copying the notebook (and any other required files) from /apps/notebook_examples.

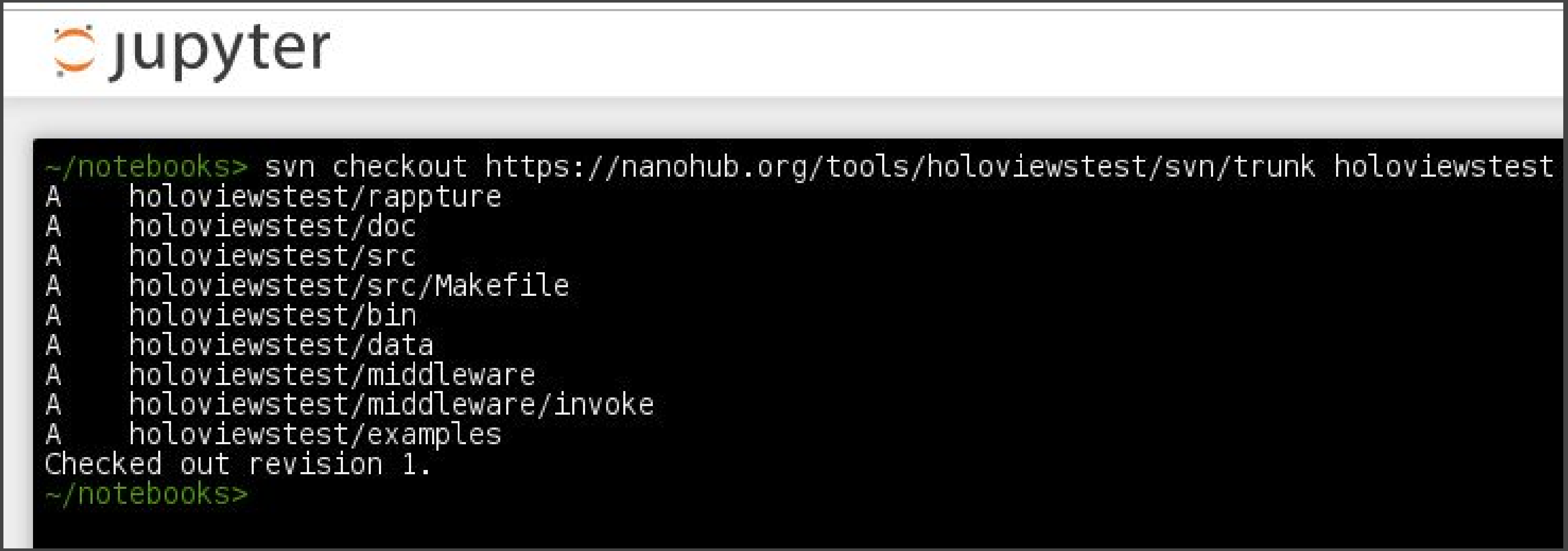
The notebooks themselves are files that end in '.ipynb'. However a notebook can call other programs, load images, animations, or other notebooks. We recommend you put non-trivial notebooks in their own directory with their required files.

Click Here!

You should see a list of folders and files.  Click on the “New” menu pulldown near the upper-right. Select “Terminal”.

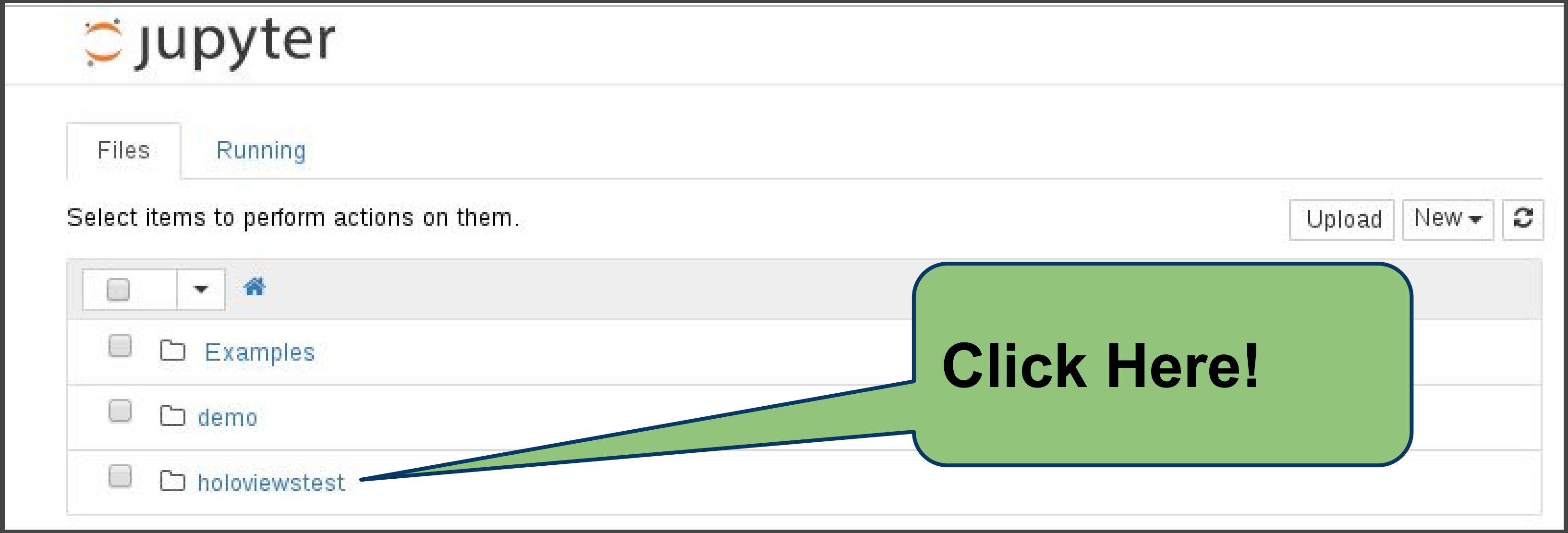


Now check out the subversion repository that was created for you in the tool creation process.



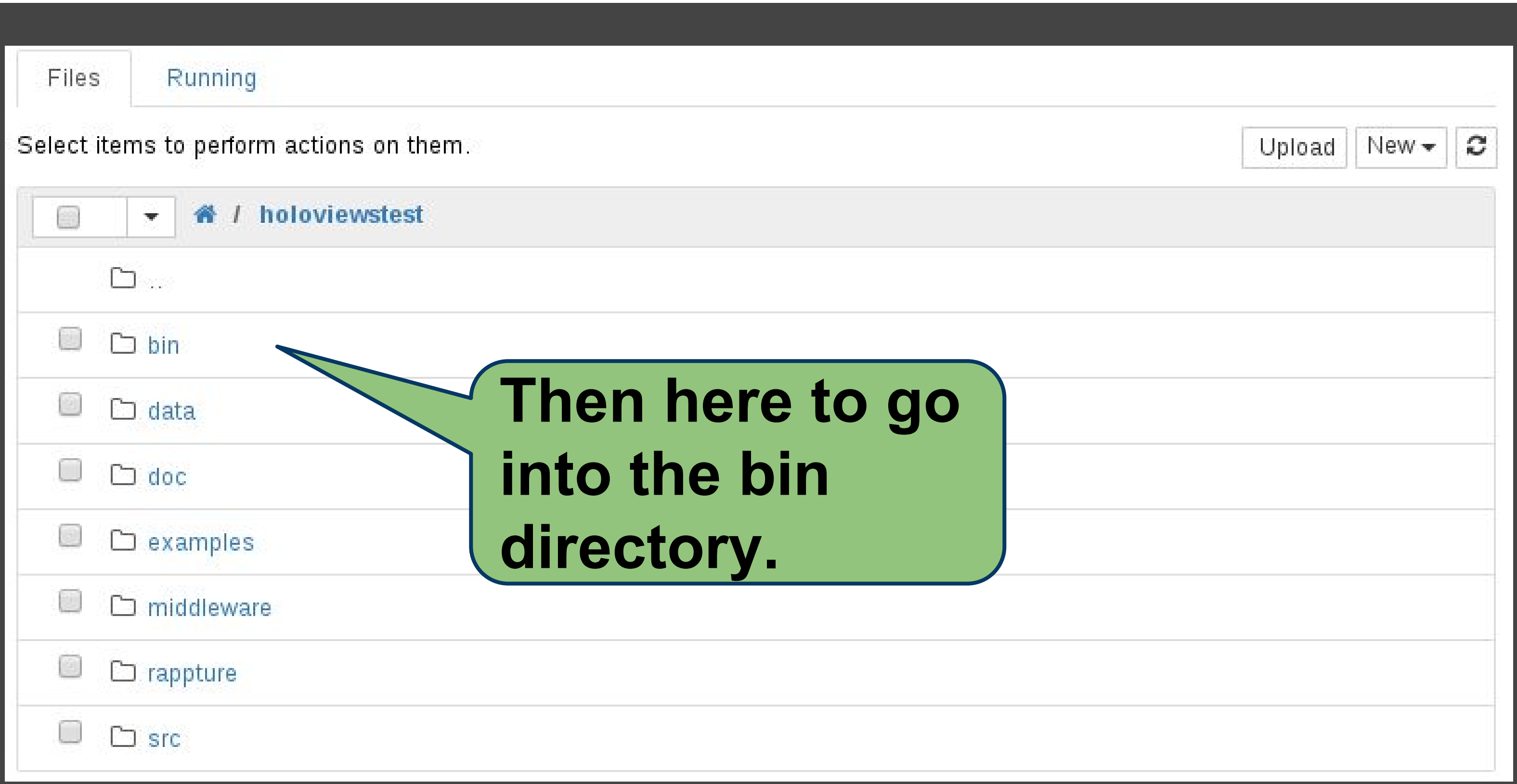
```
~/notebooks> svn checkout https://nanohub.org/tools/holoviewstest/svn/trunk holoviewstest
A    holoviewstest/rappture
A    holoviewstest/doc
A    holoviewstest/src
A    holoviewstest/src/Makefile
A    holoviewstest/bin
A    holoviewstest/data
A    holoviewstest/middleware
A    holoviewstest/middleware/invoke
A    holoviewstest/examples
Checked out revision 1.
~/notebooks>
```

# Now go back to the file list and you should see your project.



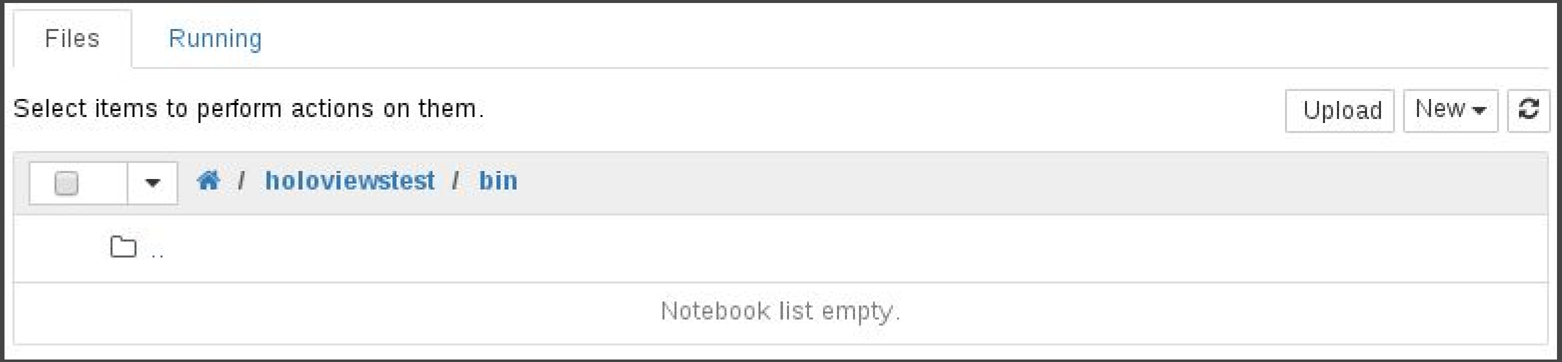
Select items to perform actions on them.

Click Here!



Select items to perform actions on them.

Then here to go into the bin directory.



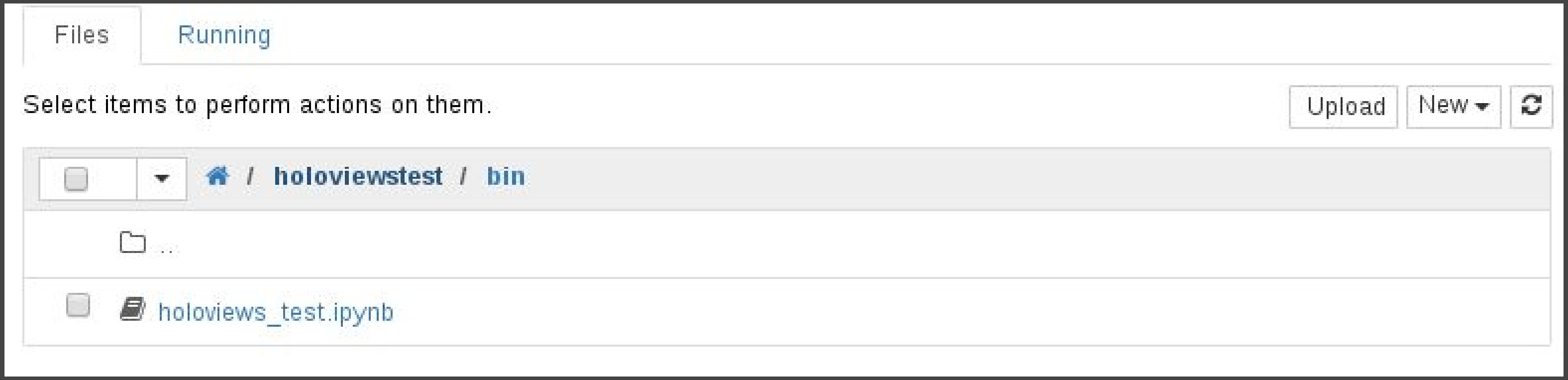
Select items to perform actions on them.

Notebook list empty.

Now select New/Python2 or New/Python3 (preferred) to create your notebook.

Alternatively, click on the “Upload” button and upload a file or notebook.

Or, if the notebook is already in your workspace, use the terminal to copy it to this location.



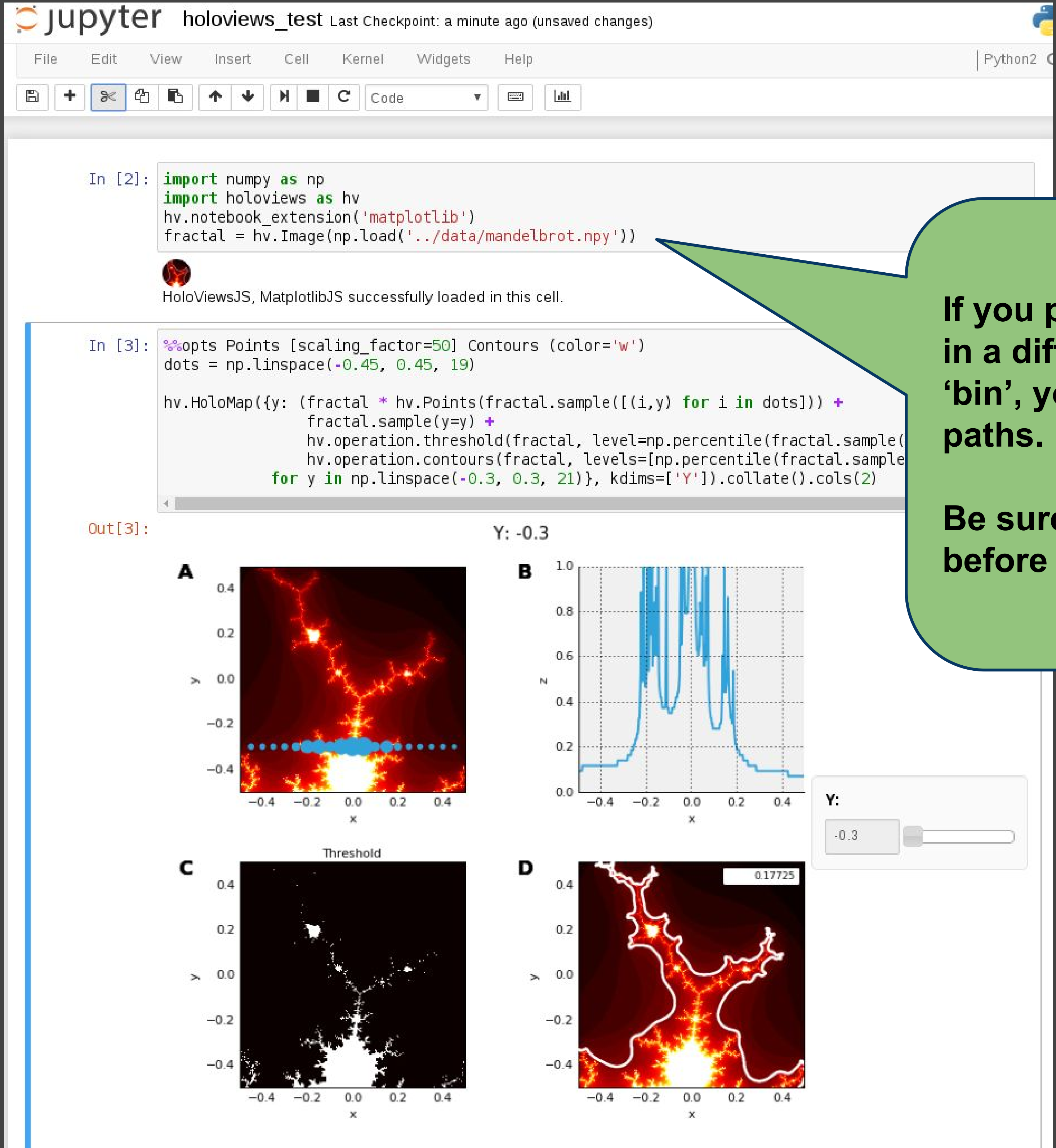
Select items to perform actions on them.

□ / holoviewstest / bin

..

□ holoviews_test.ipynb

We’ve uploaded a notebook, and also uploaded a data file to the data subdirectory.   Then we ran the notebook...



In [2]:

```
import numpy as np
import holoviews as hv
hv.notebook_extension('matplotlib')
fractal = hv.Image(np.load('../data/mandelbrot.npy'))
```

HoloViewsJS, MatplotlibJS successfully loaded in this cell.

In [3]:

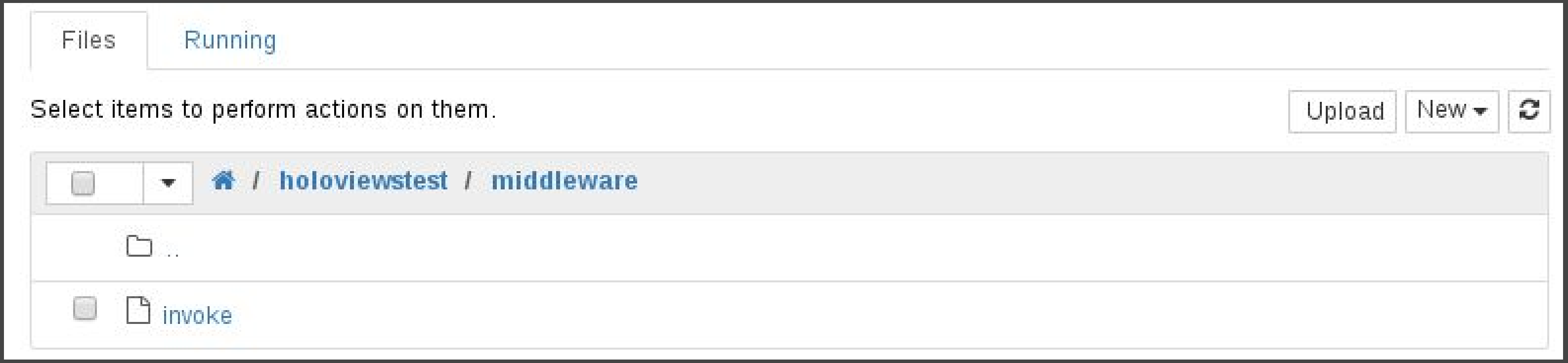
```
%%opts Points [scaling_factor=50] Contours (color='w')
dots = np.linspace(-0.45, 0.45, 19)

hv.HoloMap({y: (fractal * hv.Points(fractal.sample([(i,y) for i in dots])) +
                fractal.sample(y=y) +
                hv.operation.threshold(fractal, level=np.percentile(fractal.sample(
                hv.operation.contours(fractal, levels=[np.percentile(fractal.sample
            for y in np.linspace(-0.3, 0.3, 21)}, kdims=['Y']).collate().cols(2)
```

Out[3]:

Y: -0.3

**If you put data, images or video in a different directory than ‘bin’, you should adjust your paths.**

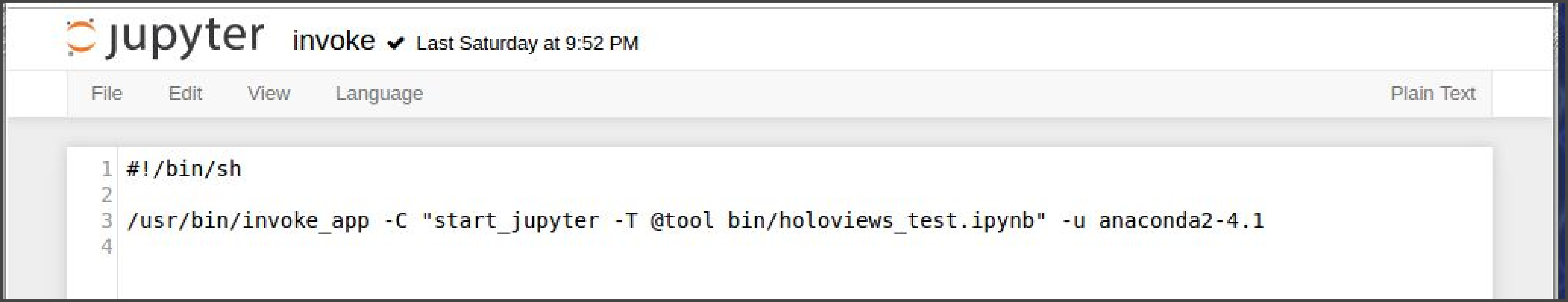
**Be sure to save any changes before you close the notebook.**



Select items to perform actions on them.

invoke

Finally, to tell the hub how to launch the notebook, we need to edit the invoke script that was automatically created.  Just double-click on it to launch the editor.



```
#!/bin/sh

/usr/bin/invoke_app -C "start_jupyter -T @tool bin/holoviews_test.ipynb" -u anaconda2-4.1

```

You should use an invoke script like the above, substituting your notebook name.  If additional modules are needed, include them at the end after “-u”.  Note that the anaconda2-4.1 module is required.

- Check in your notebook and changes.  Then update the tool status.

- Once it is installed, you can try launching it.

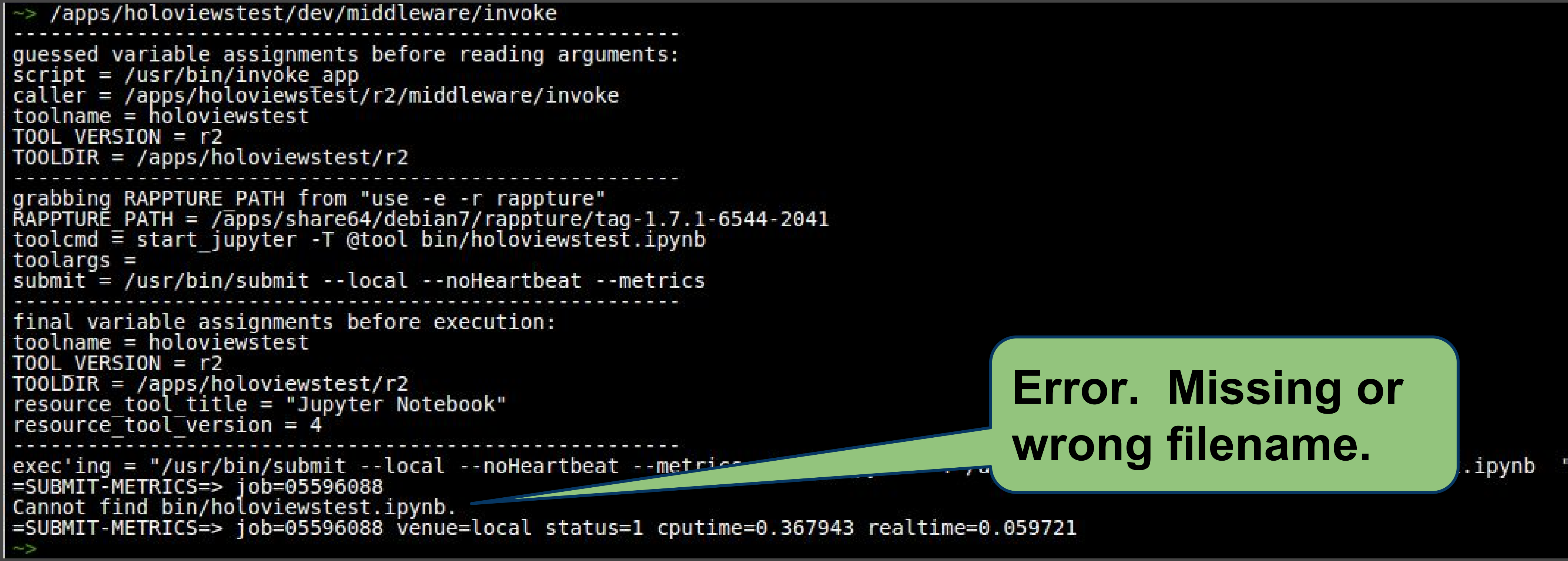
# Troubleshooting

If it does not launch:

From a terminal or workspace,

```
> use -e anaconda2-4.1

> /apps/my_app/dev/middleware/invoke
```

# Troubleshooting

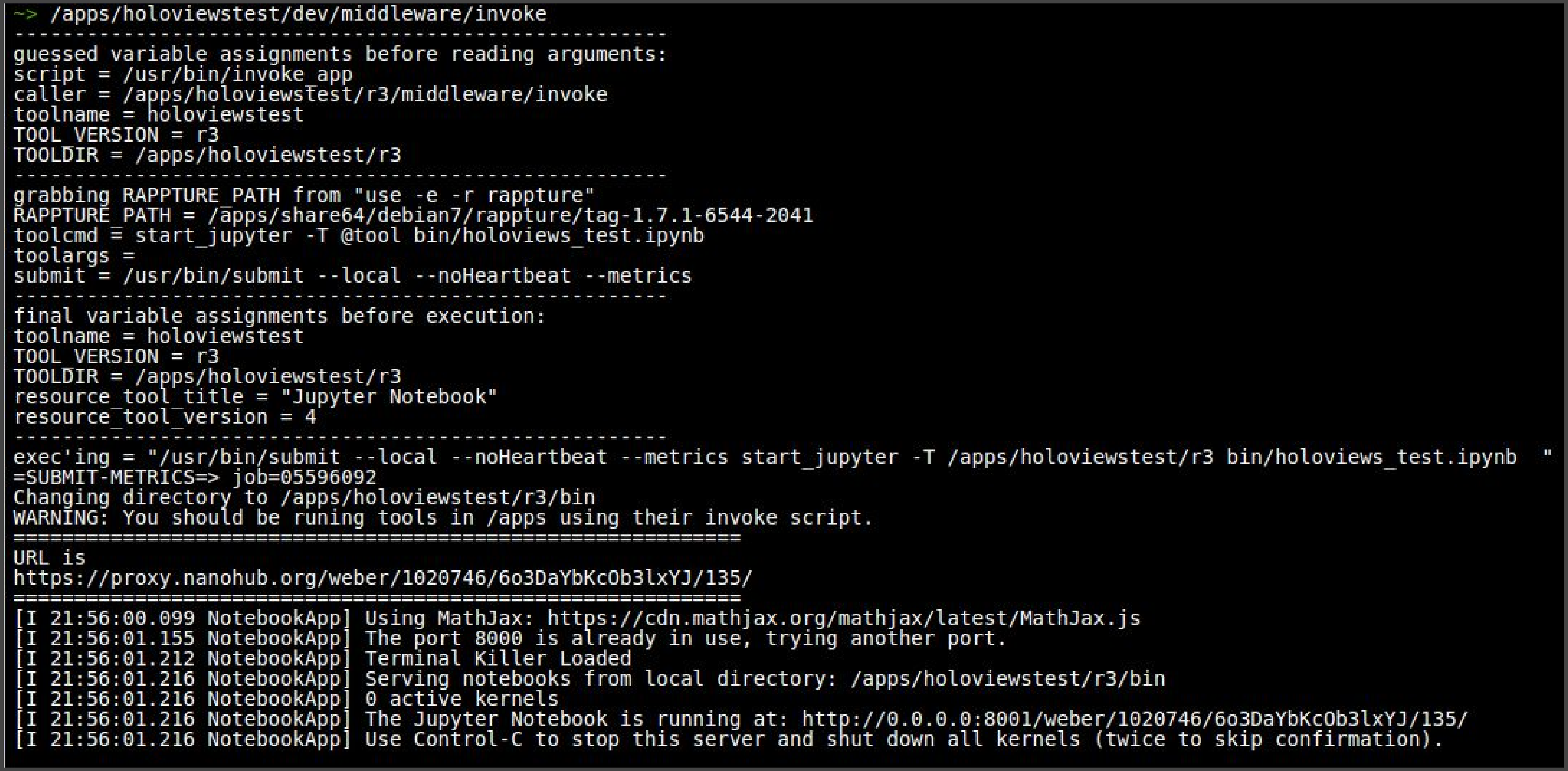


```
~> /apps/holoviewstest/dev/middleware/invoke
------------------------------------------------------
guessed variable assignments before reading arguments:
script = /usr/bin/invoke_app
caller = /apps/holoviewstest/r2/middleware/invoke
toolname = holoviewstest
TOOL_VERSION = r2
TOOLDIR = /apps/holoviewstest/r2
------------------------------------------------------
grabbing RAPPTURE_PATH from "use -e -r rappture"
RAPPTURE_PATH = /apps/share64/debian7/rappture/tag-1.7.1-6544-2041
toolcmd = start_jupyter -T @tool bin/holoviewstest.ipynb
toolargs =
submit = /usr/bin/submit --local --noHeartbeat --metrics
------------------------------------------------------
final variable assignments before execution:
toolname = holoviewstest
TOOL_VERSION = r2
TOOLDIR = /apps/holoviewstest/r2
resource_tool_title = "Jupyter Notebook"
resource_tool_version = 4
------------------------------------------------------
exec'ing = "/usr/bin/submit --local --noHeartbeat --metri                                               .ipynb  "
=SUBMIT-METRICS=> job=05596088
Cannot find bin/holoviewstest.ipynb.
=SUBMIT-METRICS=> job=05596088 venue=local status=1 cputime=0.367943 realtime=0.059721
~>
```

Error. Missing or wrong filename.

The above shows an example of what could go wrong.

# Expected Output



```
~> /apps/holoviewstest/dev/middleware/invoke
------------------------------------------------------
guessed variable assignments before reading arguments:
script = /usr/bin/invoke_app
caller = /apps/holoviewstest/r3/middleware/invoke
toolname = holoviewstest
TOOL_VERSION = r3
TOOLDIR = /apps/holoviewstest/r3
------------------------------------------------------
grabbing RAPPTURE_PATH from "use -e -r rappture"
RAPPTURE_PATH = /apps/share64/debian7/rappture/tag-1.7.1-6544-2041
toolcmd = start_jupyter -T @tool bin/holoviews_test.ipynb
toolargs =
submit = /usr/bin/submit --local --noHeartbeat --metrics
------------------------------------------------------
final variable assignments before execution:
toolname = holoviewstest
TOOL_VERSION = r3
TOOLDIR = /apps/holoviewstest/r3
resource_tool_title = "Jupyter Notebook"
resource_tool_version = 4
------------------------------------------------------
exec'ing = "/usr/bin/submit --local --noHeartbeat --metrics start_jupyter -T /apps/holoviewstest/r3 bin/holoviews_test.ipynb  "
=SUBMIT-METRICS=> job=05596092
Changing directory to /apps/holoviewstest/r3/bin
WARNING: You should be runing tools in /apps using their invoke script.
============================================================
URL is
https://proxy.nanohub.org/weber/1020746/6o3DaYbKcOb3lxYJ/135/
============================================================
[I 21:56:00.099 NotebookApp] Using MathJax: https://cdn.mathjax.org/mathjax/latest/MathJax.js
[I 21:56:01.155 NotebookApp] The port 8000 is already in use, trying another port.
[I 21:56:01.212 NotebookApp] Terminal Killer Loaded
[I 21:56:01.216 NotebookApp] Serving notebooks from local directory: /apps/holoviewstest/r3/bin
[I 21:56:01.216 NotebookApp] 0 active kernels
[I 21:56:01.216 NotebookApp] The Jupyter Notebook is running at: http://0.0.0.0:8001/weber/1020746/6o3DaYbKcOb3lxYJ/135/
[I 21:56:01.216 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

This is the expected output.

NOTE: If you ran this from a workspace, you could open a browser at the URL and try it.  If you ran it from a terminal, it will not open because it is already running inside a notebook server.

# Publishing

Once it launches successfully, be sure to update the status and have it published.