

HTCondor Python Bindings Tutorial

Brian Bockelman
HTCondor Week 2019

HTCondor Clients in 2012

Command Line Clients

Fully Featured!

Requires fork/exec and process handling

Outputs in multiple formats

Something
Missing
In
The
Middle

SOAP Clients

Features! (Some)

Language agnostic (everyone hates XML equally?)

Caveats with respect to scalability, security.

Design Philosophy

- **ClassAds**: Everything based on ClassAds; make these the “core” of the bindings.
- **pythonic**: Semantics and APIs should feel natural to a python programmer.
 - Use iterators, exceptions, guards. ClassAds behave as much like a dict as reasonable.
- **Backward compatible**: APIs are here to stay for as long as possible.
 - When we absolutely must, use standard python `DeprecationWarning` techniques.
 - Yes, this means that we keep even design warts for far longer than we'd like!
- **Native code**: Call same HTCondor library code as CLI; identical in performance.
- **Complete**: If you can do it with the command line tools, you should be able to do it with python.

Pythonic!

- Since *pythonic* is in our design philosophy, I decided the education should use the tools favored by the python community:
- Sphinx-based documentation. Hosted on ReadTheDocs; looks / feels / smells like python documentation.
 - *Hey, this is what HTCondor just adopted!*
- Jupyter-based tutorials. Login with a university credential; spawns a Docker container with a private HTCondor instance. Interact via your browser.

Sphinx Docs

<https://htcondor.readthedocs.io/en/latest/apis/python-bindings>

The screenshot shows a web browser window displaying the Sphinx documentation for HTCondor Python Bindings. The browser's address bar shows the URL `https://htcondor.readthedocs.io/en/latest/apis/python-bindings/`. The page has a dark blue header with the text "HTCondor Manual" and "latest". Below the header is a search bar labeled "Search docs". On the left side, there is a sidebar with a "CONTENTS" section listing various manual sections: Overview, Users' Manual, Administrators' Manual, Miscellaneous Concepts, Grid Computing, and Cloud Computing. Below this, there is a section for "Application Programming Interfaces (APIs)" which is expanded to show "Python Bindings". Under "Python Bindings", there are links for "Introductory Tutorials", "Advanced Tutorials", "htcondor API Reference", and "classad API Reference". The main content area on the right has a breadcrumb trail: "Docs » Application Programming Interfaces (APIs) » Python Bindings", followed by a link to "Edit on GitHub". The main heading is "Python Bindings". The text below the heading states: "The HTCondor Python bindings expose a Pythonic interface to the HTCondor client libraries. They utilize the same C++ libraries as HTCondor itself, meaning they have nearly the same behavior as the command line tools." Below this, there are sections for "Introductory Tutorials" and "Advanced Tutorials". The "Introductory Tutorials" section says: "These tutorials cover the basics of the Python bindings and how to use them through a quick overview of the major components. Each tutorial is meant to be done in sequence. Start here if you've never used the bindings before!". The "Advanced Tutorials" section says: "The advanced tutorials are in-depth looks at specific pieces of the Python modules. Each is meant to be stand-alone and should only require knowledge from the introductory tutorials." Below these, there is a section for "htcondor API Reference" which says: "Documentation for the public API of `htcondor`."

Python Bindings — HTCondor

[https://htcondor.readthedocs.io/en/latest/apis/python-bindings/](#)

HTCondor Manual
latest

Search docs

CONTENTS

- Overview
- Users' Manual
- Administrators' Manual
- Miscellaneous Concepts
- Grid Computing
- Cloud Computing

Application Programming Interfaces (APIs)

- Python Bindings
 - Introductory Tutorials
 - Advanced Tutorials
 - htcondor API Reference
 - classad API Reference

Docs » Application Programming Interfaces (APIs) » Python Bindings [Edit on GitHub](#)

Python Bindings

The HTCondor Python bindings expose a Pythonic interface to the HTCondor client libraries. They utilize the same C++ libraries as HTCondor itself, meaning they have nearly the same behavior as the command line tools.

Introductory Tutorials

These tutorials cover the basics of the Python bindings and how to use them through a quick overview of the major components. Each tutorial is meant to be done in sequence. Start here if you've never used the bindings before!

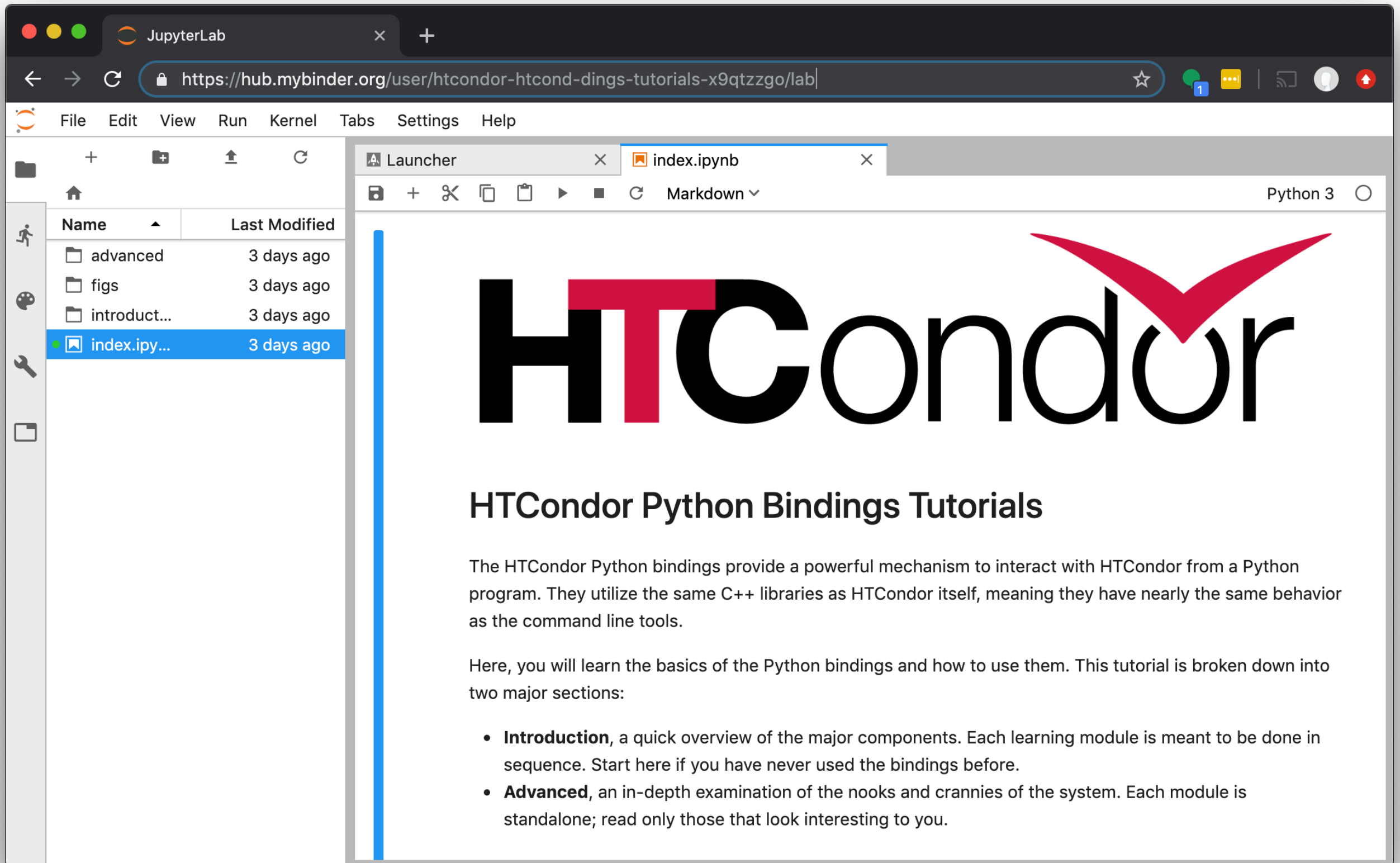
Advanced Tutorials

The advanced tutorials are in-depth looks at specific pieces of the Python modules. Each is meant to be stand-alone and should only require knowledge from the introductory tutorials.

htcondor API Reference

Documentation for the public API of `htcondor`.

Jupyter-based Tutorials



The screenshot shows a JupyterLab web interface in a browser. The address bar displays the URL <https://hub.mybinder.org/user/htcondor-htcond-dings-tutorials-x9qtzzgo/lab>. The interface includes a top menu bar with options: File, Edit, View, Run, Kernel, Tabs, Settings, and Help. On the left, a file browser sidebar shows a directory structure with folders 'advanced', 'figs', and 'introduc...', and a file 'index.ipynb' which is currently selected. The main workspace displays the 'index.ipynb' file, which is a Markdown document. The document features the 'HTCondor' logo, where the 'HT' is in black and the 'Condor' is in black with a large red checkmark above the 'o'. Below the logo, the title 'HTCondor Python Bindings Tutorials' is shown. The text explains that the HTCondor Python bindings provide a powerful mechanism to interact with HTCondor from a Python program, utilizing the same C++ libraries as HTCondor itself. It states that the tutorial is broken down into two major sections:

- **Introduction**, a quick overview of the major components. Each learning module is meant to be done in sequence. Start here if you have never used the bindings before.
- **Advanced**, an in-depth examination of the nooks and crannies of the system. Each module is standalone; read only those that look interesting to you.

Notebook View

The screenshot displays the JupyterLab interface in a web browser. The browser's address bar shows the URL `https://hub.mybinder.org/user/htcondor-htcond-dings-tutorials-x9qtzzgo/lab`. The JupyterLab window has a menu bar with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tabs', 'Settings', and 'Help'. On the left is a file browser sidebar with a table of files and folders:

Name	Last Modified
advanced	seconds ago
figs	3 days ago
introduc...	3 days ago
index.ipy...	3 days ago

The main area shows a notebook titled 'Interacting-With-Daemons'. The notebook has a toolbar with icons for saving, adding, deleting, and running cells, along with a 'Markdown' dropdown. The notebook content includes:

Interacting With Daemons

In this module, we'll look at how the HTCondor Python bindings can be used to interact with running daemons.

Let's start by importing the correct modules:

```
[ ]: import htcondor
```

Configuration

The HTCondor configuration is exposed to Python in two ways:

- The local process's configuration is available in the module-level `param` object.
- A remote daemon's configuration may be queried using a `RemoteParam`

The `param` object emulates a Python dictionary:

```
[ ]: print(htcondor.param['SCHEDD_LOG']) # Prints the schedd's current log file.
print(htcondor.param.get('TOOL_LOG')) # Print None as TOOL_LOG isn't set by default.
print(htcondor.param.setdefault('TOOL_LOG', '/tmp/log')) # Sets TOOL_LOG to /tmp/log.
print(htcondor.param['TOOL_LOG']) # Prints /tmp/log, as set above.
```

Terminal View

The screenshot displays the JupyterLab web interface in a browser window. The browser's address bar shows the URL `https://hub.mybinder.org/user/htcondor-htcond-dings-tutorials-x9qtzzgo/lab`. The JupyterLab interface includes a top menu bar with options: File, Edit, View, Run, Kernel, Tabs, Settings, and Help. On the left, a file browser sidebar shows a directory structure with folders named 'advanced', 'figs', and 'introduc...', and a file named 'index.ipynb' which is currently selected. The main area of the interface is the 'Terminal View', which shows a terminal session for the user 'jovyan' at the prompt `jovyan@jupyter-htcondor-2dhtcond-2ddings-2dtutorials-2dx9qtzzgo:~/tutorials$`. The terminal output shows the command `condor_q` being executed, followed by a header line: `-- Schedd: jovyan@jupyter-htcondor-2dhtcond-2ddings-2dtutorials-2dx9qtzzgo : <10.12.30.223:9618?... @ 05/19/19 17:02:33`. Below this is a table with columns: OWNER, BATCH_NAME, SUBMITTED, DONE, RUN, IDLE, HOLD, TOTAL, and JOB_IDS. The table content shows 'Total for query: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended' and 'Total for all users: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended'. The terminal session ends with the prompt `jovyan@jupyter-htcondor-2dhtcond-2ddings-2dtutorials-2dx9qtzzgo:~/tutorials$` and a cursor.

JupyterLab

<https://hub.mybinder.org/user/htcondor-htcond-dings-tutorials-x9qtzzgo/lab>

File Edit View Run Kernel Tabs Settings Help

Home

Name	Last Modified
advanced	a minute ago
figs	3 days ago
introduc...	3 days ago
index.ipynb	3 days ago

jovyan@jupyter-htcondor-2dhtcond-2ddings-2dtutorials-2dx9qtzzgo:~/tutorials\$ condor_q

-- Schedd: jovyan@jupyter-htcondor-2dhtcond-2ddings-2dtutorials-2dx9qtzzgo : <10.12.30.223:9618?... @ 05/19/19 17:02:33

OWNER	BATCH_NAME	SUBMITTED	DONE	RUN	IDLE	HOLD	TOTAL	JOB_IDS
Total for query: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended								
Total for all users: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended								

jovyan@jupyter-htcondor-2dhtcond-2ddings-2dtutorials-2dx9qtzzgo:~/tutorials\$

You can help!

- The contents of the tutorials and documentation are kept on GitHub:
 - <https://github.com/htcondor/htcondor-python-bindings-tutorials>
 - Note the new location for 2019! JupyterLab & Binder integration recently overhauled by Josh Karpel.
- Find a bug? Spot some missing content?
 - Simply send a pull request; Travis-CI will test and update the static content once merged.

Let's Proceed!

<http://bit.ly/htcpy-stable>