OAK RIDGE National Laboratory

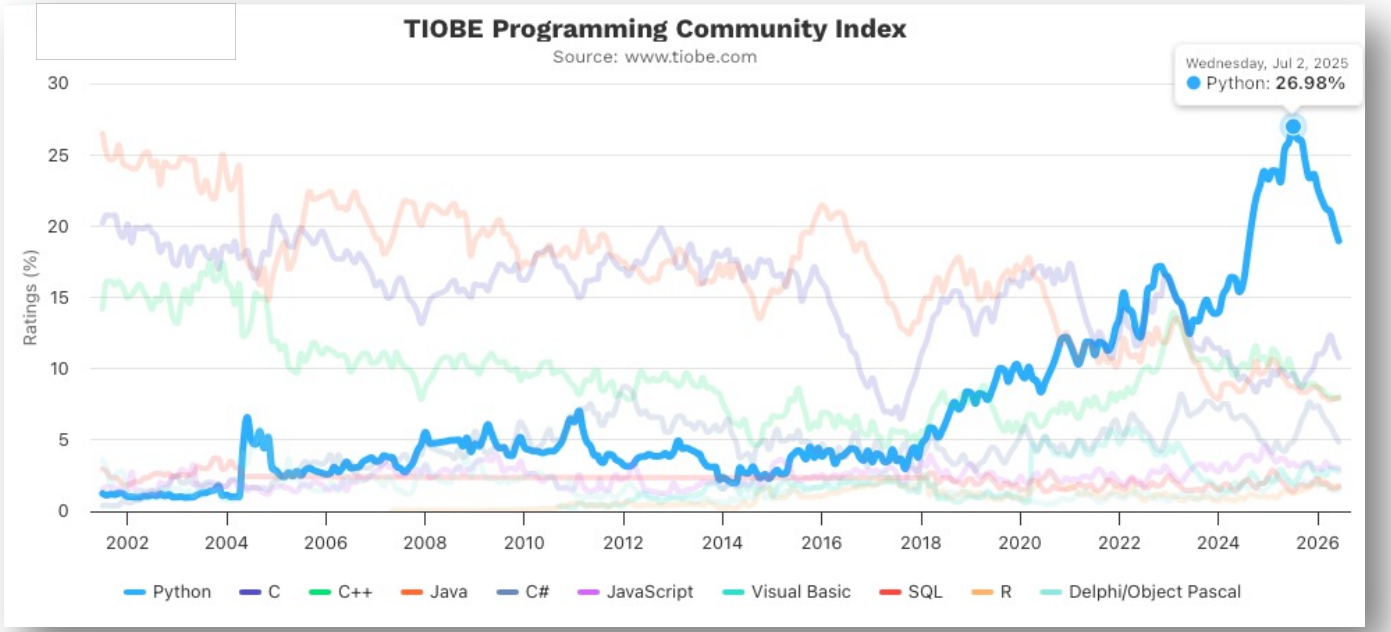
Python

Kay Kasemir

July 2026

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY



- Is very popular
- Can be very powerful
- Is comparably easy to learn
- Is a scripting language
  - Great for quick scripts
  - No type safety
  - No compilation & linking of the complete program
  - Terrible for larger code that is to be maintained for a long time

Python and EPICS

	Channel Access	PV Access
Client	PyEPICS	P4P
Server	PCASPY, PyDevice	P4P, PyDevice

PyEPICS: Channel Access Client

- Easiest 'caput', 'caget' type of client
  - Keeps all used channels open
  - 'caget' will internally 'monitor'

  →Tends to give best performance for simple scripts

- Also offers low level "PV" API
  - https://pyepics.github.io/pyepics

```python
# Example for interacting with the 'fishtank' IOC
from time import sleep
from epics import caget, caput

sp = caget('training:setpoint')
rb = caget('training:tank')
print("Temperature setpoint is at %f and tank temperature is %f" % (sp, rb))

if sp < 40:
    caput('training:setpoint', 40)
    print("Changed setpoint to 40...")
    while True:
        rb = caget('training:tank')
        print("Tank temperature is %f" % rb)
        if rb > 39.5:
            break
        sleep(1)
else:
    caput('training:setpoint', 30)
    print("Changed setpoint to 30...")
    while True:
        rb = caget('training:tank')
        print("Tank temperature is %f" % rb)
        if rb < 30.5:
            break
        sleep(1)
print("Got there, I think")
```

```python
# The first call actually creates the PV and establishes a 'monitor'
print("Tank temperature is %f" % caget(user + ':tank'))

# Calling it again just returns the last received value
print("Tank temperature is %f" % caget(user + ':tank'))

# Consider automation! After
#    caput("something", 42)
# we will very soon receive an update where "something" is 42,
# but by default, caget will return what we've received so far,
# and 'now' will not be 42, yet:
#    now = caget("something")
# To force a 'get', disable the default behavior:
print("Tank temperature is %f" % caget(user + ':tank', use_monitor=False))
```

See /ics/examples/19_python/ca*.py

P4P: PVA Access Client

- .. and Server
- Provides “PV” type API, no caching or internal monitoring as in PyEPICS
  - https://epics-base.github.io/p4p

```
from p4p.client.thread import Context
from time import sleep

pva = Context('pva')

print("Setpoint:", pva.get('epics-dev:setpoint'))
print("Tank temperature:")
for i in range(10):
    print(pva.get('epics-dev:tank'))
    sleep(1)

pva.put('epics-dev:setpoint', 40)
```

```
pva = Server(providers=[
{
    'demo:value': value
}])
```

See /ics/examples/15_pvaccess/*.py

PCASPY: Channel Access Server

- Creates Channel Access channels
  - Support DBR_DOUBLE, DBR_CTRL_DOUBLE, ..
  - GUIs can show value with time stamp, units, precision, …
- There is no IOC, no "records"
  - No SCAN, DESC, EGU, PREC, CALC, INP, … fields
- https://pcaspy.readthedocs.io

```
prefix = 'MTEST:'
pvdb = {
    'RAND' : {
        'prec' : 3,
    },
}
```

```
import random
class myDriver(Driver):
    def __init__(self):
        super(myDriver, self).__init__()

    def read(self, reason):
        if reason == 'RAND':
            value = random.random()
        else:
            value = self.getParam(reason)
        return value
```

```
server.createPV(prefix, pvdb)
driver = myDriver()
```

PyDevice

- Python interface for records on an IOC
- IOC may be accessed via CA or PVA

→ Now best option for a CA or PVA server, see next presentation

Summary

- There are CA and PVA client and server modules for Python
- For dependable automation, remember get/put-callback
  .. See busy record