

Article

[Oliver Wilms](#) · Jul 25 3m read[Open Exchange](#)

Python for test-data

Hello,

I became aware of Python in the early 2000s when I started automating tasks. Some of our processes utilized python scripts. I never figured it out very well, and we decided to do away with Python because nobody on our team was familiar with it.

Along the way, I heard a lot about Python. My interest was renewed when InterSystems offered bonus points for using Python in contests.

The things I like the most about Python so far is how easy it is to find the mean of the ages of all Titanic passengers. For my test-data app I wanted to find the largest value in the 'Qty' column in a CSV file.

The solution was as simple as this:

```
import numpy as np
import pandas as pd

def source():
    return "ORG"
def target():
    return "GKH"
def maxqty(filename="QuantityOnHandSync.csv"):
    # Read data
    qohs = pd.read_csv(filename, ",")

    # Set Qty to a NumPy array
    qty = qohs['Qty']

    # Use numpy to find the maximum qty
    max_qty = np.nanmax(qty)

    return str(max_qty)
```

[test-data/test-data.py at master · oliverwilms/test-data \(github.com\)](#)

With Embedded Python, IRIS developers who are familiar with ObjectScript and Python developers have equal access to memory and data. This means both developer groups can use the language of their choice to take advantage of IRIS powerful engine.

With Embedded Python, I can create a class with classmethods and it looks like any ObjectScript class. Inside any classmethod I can use ObjectScript syntax for ObjectScript and Python syntax for Python.

This is a line with familiar ObjectScript code:

```
Set file = $System.Util.ManagerDirectory()_"data/QuantityOnHandSync.csv"
```

[https://github.com/oliverwilms/test-data/blob/master/src/dc/python/test.cls#:~:text=Set%20file%20%3D%20%24System.Util.ManagerDirectory\(\)%22data/QuantityOnHandSync.csv%22](https://github.com/oliverwilms/test-data/blob/master/src/dc/python/test.cls#:~:text=Set%20file%20%3D%20%24System.Util.ManagerDirectory()%22data/QuantityOnHandSync.csv%22)

The lines before and after it both use ObjectScript Set commands, but they utilize Python. The first line imports the Python code

```
Set td = ##class(%SYS.Python).Import("test-data")
```

[https://github.com/oliverwilms/test-data/blob/master/src/dc/python/test.cls#:~:text=Try%20%7B-,Set%20td%20%3D%20%23%23class\(%25SYS.Python\).Import\(%22test%2Ddata%22\).-Set%20file%20%3D](https://github.com/oliverwilms/test-data/blob/master/src/dc/python/test.cls#:~:text=Try%20%7B-,Set%20td%20%3D%20%23%23class(%25SYS.Python).Import(%22test%2Ddata%22).-Set%20file%20%3D)

The second line executes the Python code.

```
Set maxq = td.maxqty(file)
```

[https://github.com/oliverwilms/test-data/blob/master/src/dc/python/test....\(file\)](https://github.com/oliverwilms/test-data/blob/master/src/dc/python/test....(file))

Here is an example with familiar ObjectScript commands Set and Write, but what is written is Python syntax

```
Set dt = ##class(%SYS.Python).Import("datetime")
```

```
write dt.date.today().isoformat()
```

[https://github.com/oliverwilms/test-data/blob/master/src/dc/python/test.cls#:~:text=Set%20dt%20%3D\(\).isoformat\(\)](https://github.com/oliverwilms/test-data/blob/master/src/dc/python/test.cls#:~:text=Set%20dt%20%3D().isoformat())

Embedded Python can be used for Interoperability anywhere ObjectScript code can be used.

Here is a BPL example of calling embedded Python in a Business Process:

```
<assign name="tQty" property="tQty" value=
"##class(dc.python.test).MaxQuantity()" action="set" xpos='335' ypos='700' >
```

[https://github.com/oliverwilms/test-data/blob/master/src/dc/iris/streamProcess.cls#:~:text=%3Cassign%20name%3D%22tQty%22%20property%3D%22tQty%22%20value%3D%22%23%23class\(dc.python.test\).MaxQuantity\(\)%22%20action%3D%22set%22%20xpos%3D%27335%27%20ypos%3D%27700%27%20%3E](https://github.com/oliverwilms/test-data/blob/master/src/dc/iris/streamProcess.cls#:~:text=%3Cassign%20name%3D%22tQty%22%20property%3D%22tQty%22%20value%3D%22%23%23class(dc.python.test).MaxQuantity()%22%20action%3D%22set%22%20xpos%3D%27335%27%20ypos%3D%27700%27%20%3E)

I used a DTL and here I used ObjectScript code to get a sequence of Social Security Numbers. Of course we can use Embedded Python just as easily as ObjectScript:

```
<assign value='##class(dc.iris.ssn).GetNextSSN()' property  
='target.SSN' action='set' />
```

[https://github.com/oliverwilms/test-data/blob/master/src/dc/iris/dtl.cls#:~:text=%3Cassign%20value%3D%27%23%23class\(dc.iris.ssn\).GetNextSSN\(\)%27%20property%3D%27target.SSN%27%20action%3D%27set%27%20/%3E](https://github.com/oliverwilms/test-data/blob/master/src/dc/iris/dtl.cls#:~:text=%3Cassign%20value%3D%27%23%23class(dc.iris.ssn).GetNextSSN()%27%20property%3D%27target.SSN%27%20action%3D%27set%27%20/%3E)

I very much want to learn more about using Python. I believe it can create nice graphs and dashboards. I want to explore those options in the future.

[#Embedded Python](#) [#Python](#) [#InterSystems IRIS](#)

[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/python-test-data>