Power Systems

IBM

Practical Guide to Using Python

Kevin Adler
kadler@us.ibm.com

Why use Python

- High level language
  - powerful tools for manipulating data: tuples, dicts, list comprehensions
  - regular expressions
  - no compiling needed
  - easy to build web applications
- Lots of packages
  - your problem probably already solved
  - rich standard library and extras on Python Package Index (PyPI)
  - data parsing (CSV, Excel, JSON, ...)
  - web services hooks (reddit, twitter, facebook, dropbox, …)
- Simple, straightforward language
- People know it!
  - used heavily in the industry
  - taught in Academia

Example: Printing file contents from arguments

```
from sys import argv
import re

for arg in argv[1:]:
    if re.match(r'^.*[.](txt|csv)$', arg):
        with open(arg) as file:
            print(file.read())

    elif arg[-4:] == '.bin':
        print("%s is binary, skipping" % arg)

    else:
        print("Sorry, can't handle %s" % arg)
```

Example: Sending files as email

```
from sys import argv
import smtplib
from email.mime.text import MIMEText

smtp = smtplib.SMTP('smtp.example.com')

for arg in argv[1:]:
    with open(arg) as file:
        msg = MimeText(file.read())
        msg['Subject'] = arg
        msg['From'] = 'kadler@us.ibm.com'
        msg['To'] = 'kadler@us.ibm.com'

        smtp.send_message(msg)

smtp.quit()
```

Example: Replacing CPYTOIMPF

```
import ibm_db_dbi as db2
import csv

conn = db2.connect()
cursor = conn.cursor()
cursor.execute("select cusnum, lstnam, init, cdtlmt
from qiws.qcustcdt where cdtlmt > 100")

def trim_col(s):
    try:
        return s.rstrip()
    except AttributeError:
        return s

with open('qcustcdt.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile, \
                        quoting=csv.QUOTE_NONNUMERIC)
    for row in cursor:
        writer.writerow([trim_col(col) for col in row])
```

Example: Replacing CPYTOIMPF

```
938472,"Henning","G K",5000
839283,"Jones","B D",400
392859,"Vine","S S",700
938485,"Johnson","J A",9999
397267,"Tyron","W E",1000
389572,"Stevens","K L",400
846283,"Alison","J S",5000
475938,"Doe","J W",700
693829,"Thomas","A N",9999
593029,"Williams","E D",200
192837,"Lee","F L",700
583990,"Abraham","M T",9999
```

Package Management

- Python has a package manager, pip
- Use pip to install packages from the internet
  - Automatically determines dependencies needed
  - Downloads needed packages from the Python Package Index (pypi.python.org)
  - Installs the packages
- upgrade and uninstall packages as well
- pip can also install local packages (wheels)
- No internet on IBM i? No problem! Check out devpi

Example: Making a text table

```
kadler@wernstrom:~>pip3 install ptable
Successfully installed ptable-0.9.2

kadler@wernstrom:~>cat table.py
from prettytable import PrettyTable
x = PrettyTable()

x.add_column("City name",["Adelaide", "Brisbane",
"Darwin", "Hobart", "Sydney"])

x.add_column("Area", [1295, 5905, 112, 1357, 2058])

x.add_column("Annual Rainfall",[600.5, 1146.4, 1714.7,
619.5, 1214.8])

print(x)
```

https://pypi.python.org/pypi/PTable/0.9.0

Example: Making a text table

```
+------------+------+-----------------+
| City name  | Area | Annual Rainfall |
+------------+------+-----------------+
|  Adelaide  | 1295 |      600.5      |
|  Brisbane  | 5905 |      1146.4     |
|   Darwin   | 112  |      1714.7     |
|   Hobart   | 1357 |      619.5      |
|   Sydney   | 2058 |      1214.8     |
+------------+------+-----------------+
```

Example: Converting database table to text table

```
from prettytable import from_db_cursor
import ibm_db_dbi as db2

conn = db2.connect()

cur = conn.cursor()
cur.execute("select * from qiws.qcustcdt")

print(from_db_cursor(cur))
```

Example: Converting database table to text table

CUSNUM	LSTNAM	INIT	STREET	CITY	STATE	ZIPCOD	CDTLMT	CHGCOD	BALDUE	CDTDUE
938472	Henning	G K	4859 Elm Ave	Dallas	TX	75217	5000	3	37.00	0.00
839283	Jones	B D	21B NW 135 St	Clay	NY	13041	400	1	100.00	0.00
392859	Vine	S S	PO Box 79	Broton	VT	5046	700	1	439.00	0.00
938485	Johnson	J A	3 Alpine Way	Helen	GA	30545	9999	2	3987.50	33.50
397267	Tyron	W E	13 Myrtle Dr	Hector	NY	14841	1000	1	0.00	0.00
389572	Stevens	K L	208 Snow Pass	Denver	CO	80226	400	1	58.75	1.50
846283	Alison	J S	787 Lake Dr	Isle	MN	56342	5000	3	10.00	0.00
475938	Doe	J W	59 Archer Rd	Sutter	CA	95685	700	2	250.00	100.00
693829	Thomas	A N	3 Dove Circle	Casper	WY	82609	9999	2	0.00	0.00
593029	Williams	E D	485 SE 2 Ave	Dallas	TX	75218	200	1	25.00	0.00
192837	Lee	F L	5963 Oak St	Hector	NY	14841	700	2	489.50	0.50
583990	Abraham	M T	392 Mill St	Isle	MN	56342	9999	3	500.00	0.00

Example: Converting database table to Excel spreadsheet

```
from xlsxwriter import Workbook
import ibm_db_dbi as db2

conn = db2.connect()

cur = conn.cursor()
cur.execute("select * from qiws.qcustcdt")
headers = [descr[0] for descr in cur.description]

with Workbook('qcustcdt.xlsx') as workbook:
    worksheet = workbook.add_worksheet()
    worksheet.write_row('A1', headers)
    for rownum, row in enumerate(cur, start=1):
        worksheet.write_row(rownum, 0, row)
```

Example: Converting database table to Excel spreadsheet

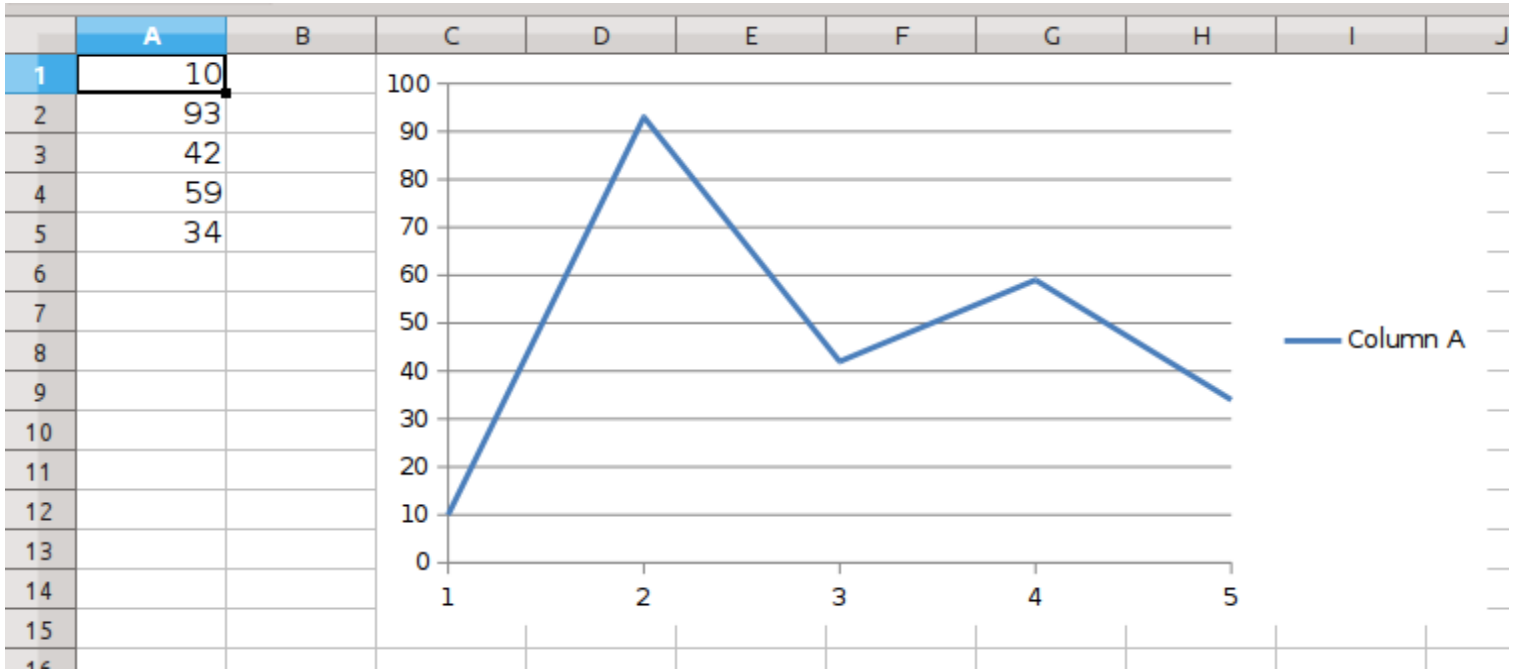
	A	B	C	D	E	F	G	H	I	J	K
1	CUSNUM	LSTNAM	INIT	STREET	CITY	STATE	ZIPCOD	CDTLMT	CHGCOD	BALDUE	CDTDUE
2	938472	Henning	G K	4859 Elm	Dallas	TX	75217	5000	3	37	0
3	839283	Jones	B D	21B NW	Clay	NY	13041	400	1	100	0
4	392859	Vine	S S	PO Box 7	Broton	VT	5046	700	1	439	0
5	938485	Johnson	J A	3 Alpine W	Helen	GA	30545	9999	2	3987.5	33.5
6	397267	Tyron	W E	13 Myrtle	Hector	NY	14841	1000	1	0	0
7	389572	Stevens	K L	208 Snow	Denver	CO	80226	400	1	58.75	1.5
8	846283	Alison	J S	787 Lake	Isle	MN	56342	5000	3	10	0
9	475938	Doe	J W	59 Archer	Sutter	CA	95685	700	2	250	100
10	693829	Thomas	A N	3 Dove Ct	Casper	WY	82609	9999	2	0	0
11	593029	Williams	E D	485 SE 2	Dallas	TX	75218	200	1	25	0
12	192837	Lee	F L	5963 Oak	Hector	NY	14841	700	2	489.5	0.5
13	583990	Abraham	M T	392 Mill St	Isle	MN	56342	9999	3	500	0
14											
15											

Example: Creating a spreadsheet

```
kadler@wernstrom:~>pip3 install xlsxwriter
Successfully installed xlsxwriter-0.8.6

kadler@wernstrom:~>cat excel.py

from xlsxwriter import Workbook

with Workbook('test.xlsx') as workbook:
    worksheet = workbook.add_worksheet()

    worksheet.write_column('A1', [10, 93, 42, 59, 34])

    chart = workbook.add_chart({'type': 'line'})
    chart.add_series({'values': '=Sheet1!$A$1:$A$5'})
    worksheet.insert_chart('C1', chart)
```

http://xlsxwriter.readthedocs.io/index.html

Example: Creating a spreadsheet



Active Job Dashboard

Using Bottle

- Framework for building simple lightweight web applications
- Includes a templating engine
- Self-hosting web server included or use with flipflop (also included in OPS) in FastCGI mode
- Need PTF SI60566 or superseding

- … also need ibm_db – SI60563 or superseding
- … and itoolkit too – SI60564 or superseding
- See https://ibm.biz/installpythonpackages for more info to install

Application Skeleton

```
from bottle import route, run

@route('/')
def root():
    return "<!-- insert html here -->"

run(host='0.0.0.0', port=3333, debug=True,
reloader=True)
```

Let's Get Some Job Info

```
@route('/')
def root():
    itransport = iLibCall()
    itool = iToolKit()

    itool.add(iCmd5250('actjob', 'WRKACTJOB'))
    itool.call(itransport)
    wrkactjob = itool.dict_out('actjob')
    data = wrkactjob['actjob']
```

You Got Your Job Info in My HTML

```
    return """
<html>
<head><title>Active Job Dashboard</title></head>
<body>
<pre>%s</pre>
</body>
</html>
""" % data
```

You Got Your Job Info in My HTML

Job Info

lp13ut28:3333

```
                                   Work with Active Jobs
5770SS1 V7R2M0 140418                                                                    LP13UT28  11/30/15  21:
Reset . . . . . . . . . . . . . . . . . :   *NO
Subsystems  . . . . . . . . . . . . . . :   *ALL
CPU Percent Limit . . . . . . . . . . . :   *NONE
Response Time Limit . . . . . . . . . . :   *NONE
Sequence  . . . . . . . . . . . . . . . :   *SBS
Job name  . . . . . . . . . . . . . . . :   *ALL
CPU %  . . . :      .0          Elapsed time . . . . . . . :   00:00:00          Active jobs . . . . . . :    265
                                  Current                                 --------Elapsed---------
Subsystem/Job  User        Number   User        Type Pool Pty     CPU   Int    Rsp  AuxIO   CPU%  Function       Status
QBATCH         QSYS        676430   QSYS        SBS    2   0       .0                  0     .0                   DEQW
QCMN           QSYS        676431   QSYS        SBS    2   0       .2                  0     .0                   DEQW
  QACSOTP      QUSER       676465   QUSER       PJ     2  20       .0                  0     .0                   PSRW
  QLZPSERV     QUSER       676478   QUSER       PJ     2  20       .0                  0     .0                   PSRW
  QNMAPINGD    QUSER       676457   QUSER       PJ     2  25       .0                  0     .0                   PSRW
  QNMAREXECD   QUSER       676461   QUSER       PJ     2  25       .0                  0     .0                   PSRW
  QNPSERVR     QUSER       676475   QUSER       PJ     2  20       .0                  0     .0                   PSRW
  QZRCSRVR     QUSER       676468   QUSER       PJ     2  20       .0                  0     .0                   PSRW
  QZSCSRVR     QUSER       676472   QUSER       PJ     2  20       .0                  0     .0                   PSRW
QCTL           QSYS        676405   QSYS        SBS    2   0      4.9                  0     .0                   DEQW
  QSYSSCD      QPGMR       676428   QPGMR       BCH    2  10       .2                  0     .0  PGM-QEZSCNEP     EVTW
QHTTPSVR       QSYS        801660   QSYS        SBS    2   0       .0                  0     .0                   DEQW
  ADMIN        QTMHHTTP    801891   QTMHHTTP    BCH    2  25       .1                  0     .0  PGM-QZHBMAIN     SIGW
  ADMIN        QTMHHTTP    801904   QTMHHTTP    BCI    2  25       .6                  0     .0  PGM-QZSRLOG      SIGW
  ADMIN        QTMHHTTP    801921   QTMHHTTP    BCI    2  25       .9                  0     .0  PGM-QZSRHTTP     SIGW
  ADMIN        QTMHHTTP    802163   SMKELLEY    BCI    2  25       .0                  0     .0  PGM-QZSRCGI      TIMW
  ADMIN1       QLWISVR     801915   QLWISVR     BCI    2  25     14.5                  0     .0  JVM-/qibm/prod   THDW
  ADMIN2       QLWISVR     801920   QLWISVR     BCI    2  25    228.7                  0     .0  JVM-/qibm/prod   THDW
```

Job Done, or is it?

- Basic dashboard works
- No elapsed period, though
- No elapsed statistics
- Plain text formatting limits options, not mobile friendly
- Can we do better?

Move Over WRKACTJOB

- QSYS2.ACTIVE_JOB_INFO
  - Like WRKACTJOB, but in a table format
  - Added in 7.2 TR2, 7.1 TR10
  - ELAPSED_TIME column added in 7.2 TR3, 7.1 TR11

Move Over WRKACTJOB

```
import ibm_db_dbi as db2

conn = db2.connect()

query = "select subsystem, job_name,
function_type, function, elapsed_time from
table(qsys2.active_job_info()) x"

cur = conn.cursor()
cur.execute(query)
return template('root', rows=cur)
```

Root template

```
<html>
<head>
<title>Active Job Dashboard</title>
</head>
<body>
<table>
% for row in rows:
    % include('row', values=row)
% end
</table>
</body>
</html>
```

Row template

```
<tr>
% for value in values:
<td>{{value}}</td>
% end
</tr>
```

Move Over WRKACTJOB

Active Job Dashboard

lp13ut28:3333

QBATCH	676430/QSYS/QBATCH	None	None	5.382
QCMN	676431/QSYS/QCMN	None	None	5.382
QCMN	676465/QUSER/QACSOTP	None	None	5.382
QCMN	676478/QUSER/QLZPSERV	None	None	5.382
QCMN	676457/QUSER/QNMAPINGD	None	None	5.382
QCMN	676461/QUSER/QNMAREXECD	None	None	5.382
QCMN	676475/QUSER/QNPSERVR	None	None	5.382
QCMN	676468/QUSER/QZRCSRVR	None	None	5.382
QCMN	676472/QUSER/QZSCSRVR	None	None	5.382
QCTL	676405/QSYS/QCTL	None	None	5.382
QCTL	676428/QPGMR/QSYSSCD	PGM	QEZSCNEP	5.382
QHTTPSVR	801660/QSYS/QHTTPSVR	None	None	5.382
QHTTPSVR	801891/QTMHHTTP/ADMIN	PGM	QZHBMAIN	5.382
QHTTPSVR	801904/QTMHHTTP/ADMIN	PGM	QZSRLOG	5.382
QHTTPSVR	801921/QTMHHTTP/ADMIN	PGM	QZSRHTTP	5.382
QHTTPSVR	802163/QTMHHTTP/ADMIN	PGM	QZSRCGI	5.382
QHTTPSVR	801915/QLWISVR/ADMIN1	JVM	/qibm/prod	5.382

Titled Accordingly

```
from string import capwords

show_cols = ( 'SUBSYSTEM', 'JOB_NAME',
'AUTHORIZATION_NAME', 'JOB_TYPE',
'FUNCTION_TYPE', ... )

headers = [ capwords(col.replace('_', ' ')) \
            for col in show_cols ]

# headers = ( 'Subsystem', 'Job Name',
'Authorization Name', Job Type', 'Function Type',
... )
```

So much better

Active Job Dashboard

lp13ut28:3333

Elapsed time: 5.495

Job Name	Authorization Name	Job Type	Function Type	Function	Job Status	Elapsed Interaction Count	Elaps Tota Respo Tim
676430/QSYS /QBATCH	QSYS	SBS	None	None	DEQW	None	Non
676431/QSYS/QCMN	QSYS	SBS	None	None	DEQW	None	Non
676465/QUSER /QACSOTP	QUSER	PJ	None	None	PSRW	None	Non
676478/QUSER /QLZPSERV	QUSER	PJ	None	None	PSRW	None	Non
676457/QUSER /QNMAPINGD	QUSER	PJ	None	None	PSRW	None	Non
676461/QUSER /QNMAREXECD	QUSER	PJ	None	None	PSRW	None	Non
676475/QUSER /QNPSERVR	QUSER	PJ	None	None	PSRW	None	Non
676468/QUSER /QZRCSRVR	QUSER	PJ	None	None	PSRW	None	Non

Let's Snaz it Up a Bit

- jQuery
  - open source javascript library
  - extendable with plugins
  - massages over brower differences
  - https://jquery.com
- tablesorter jQuery plugin
  - sort tables client side
  - can sort text, integers, floats, currency, IP addresses, dates, times, and more
  - extendable to understand different column formats
  - http://tablesorter.com/docs/

Snazzy!

Active Job Dashboard

lp13ut28:3333

Elapsed time: 14.150 seconds

Refresh Reset

Job Name	Authorization Name	Job Type	Function Type	Function	Job Status	Elapsed Interaction Count	Elapsed Total Response Time	Elapsed T Disk Io Co
676465/QUSER/QACSOTP	QUSER	PJ			PSRW	0	0	0
676478/QUSER /QLZPSERV	QUSER	PJ			PSRW	0	0	0
676457/QUSER /QNMAPINGD	QUSER	PJ			PSRW	0	0	0
676461/QUSER /QNMAREXECD	QUSER	PJ			PSRW	0	0	0
676475/QUSER /QNPSERVR	QUSER	PJ			PSRW	0	0	0
676468/QUSER /QZRCSRVR	QUSER	PJ			PSRW	0	0	0
676472/QUSER /QZSCSRVR	QUSER	PJ			PSRW	0	0	0
676428/QPGMR /QSYSSCD	QPGMR	BCH	PGM	QEZSCNEP	EVTW	0	0	0
801891/QTMHHTTP /ADMIN	QTMHHTTP	BCH	PGM	QZHBMAIN	SIGW	0	0	0

Snazzy!

Active Job Dashboard

lp13ut28:3333

Elapsed time: 14.150 seconds

Refresh Reset

Job Name	Authorization Name	Job Type	Function Type	Function	Job Status	Elapsed Interaction Count	Elapsed Total Response Time	Elapsed T Disk Io Co
692469/TIMMR /QPADEV0001	TIMMR	INT	CMD	WRKACTJOB	DSC	0	0	0
802473/QUSER /QZRCSRVS	SMKELLEY	PJ			TIMW	0	0	0
803111/QUSER /QZRCSRVS	SMKELLEY	PJ			TIMW	0	0	0
803276/QUSER /QZRCSRVS	SMKELLEY	PJ			TIMW	0	0	0
802163/QTMHHTTP /ADMIN	SMKELLEY	BCI	PGM	QZSRCGI	TIMW	0	0	0
802123/QSECOFR /QP0ZSPWP	RWATKIN	BCI	PGM	bsh	THDW	0	0	0
802124/QSECOFR /QP0ZSPWP	RWATKIN	BCI	PGM	sftp-serve	SELW	0	0	0
676522/QYPSJSVR /QYPSJSVR	QYPSJSVR	BCH	PGM	jvmStartPa	SIGW	0	0	0
801923/QWEBADMIN /ADMIN4	QWEBADMIN	BCI	JVM	/qibm/prod	THDW	0	0	0

How do I get it?

- 5733-OPS is a “skip-ship” LPO that is licensed for IBM i 7.1+
- Initially only Option 1 was defined (Node.js v.1x) – all the others are placeholders, to be defined later and delivered via PTF
- Python 3 delivered via 5733-OPS Option 2 in June 2015 and Python 2 delivered via 5733-OPS Option 4 in May 2016
- To get Python 3, you must install 5733-OPS *BASE and Option 2, and then install the following (or superseding) PTFs and any requisites:
  - SI59051 – Python 3 runtime

How do I get it?

- 5733-OPS is a “skip-ship” LPO that is licensed for IBM i 7.1+
- Initially only Option 1 was defined (Node.js v.1x) – all the others are placeholders, to be defined later and delivered via PTF
- Python 3 delivered via 5733-OPS Option 2 in June 2015 and Python 2 delivered via 5733-OPS Option 4 in May 2016
- To get Python 3, you must install 5733-OPS *BASE and Option 2, and then install the following (or superseding) PTFs and any requisites:
  - SI59051 – Python 3 runtime
- Or, just get the new OPS group, level 1:
  - 7.3: SF99225
  - 7.2: SF99223
  - 7.1: SF99123

But Wait, There's More

- We also include many optional Python packages:
  - SI60563 – ibm_db package, DB2 interface & Django adapter
  - SI60564 – itoolkit package, IBM i toolkit wrapper for XMLService
  - SI60565 – flipflop package, FastCGI gateway
  - SI60566 – bottle package, lightweight web framework
- Each PTF just lays down the “wheel”, you still need to install it. eg.
  - `cd /QOpenSys/QIBM/ProdData/OPS/Python-pkgs/ibm_db`
  - `pip3 install ibm_db-*cp34*.whl`
- See https://ibm.biz/installpythonpackages for more info

Getting Started Resources

- Official Tutorial: https://docs.python.org/3/tutorial/
- Hitchhiker's Guide to Python: http://docs.python-guide.org/en/latest/
- Learn Python in Y Minutes: https://learnxinyminutes.com/docs/python3/
- Learning Python subreddit: http://www.reddit.com/r/learnpython and their Wiki: https://www.reddit.com/r/learnpython/wiki/index
- Python on IBM i Examples (Coming Soon): http://ibm.biz/pythonexamplesonibmi

Questions?