

This is what I want to share about my experience with Python on IBM i.

For a long time, every time I was trying to learn a new language, the training was targeting how to bounce red balls on the screen, and just the very last chapter of the course was mentioning manipulating data, I remember that when I was training for a Java certification...

In this case, with Python, I tried just from the beginning to retrieve data from the data base and start to work just on data, I think this is the real word for Python.

Python like other object oriented technologies, use SQL scripts to manage the data bases, I was trying with MySQL and with IBM I (DB2), and the differences are just a couple of lines where you define the connection, the rest is the same.

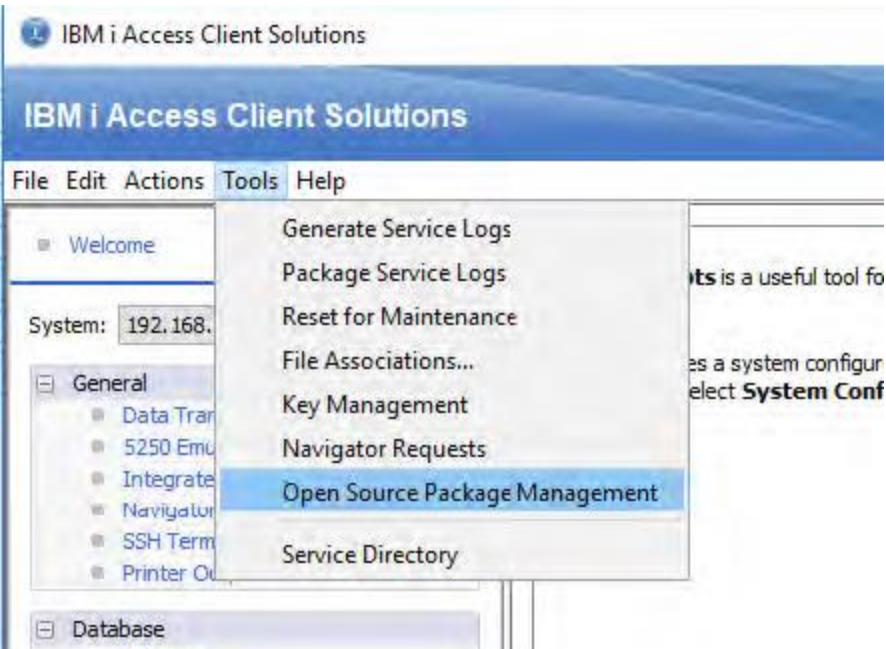
When I was using java years ago when I was in Borders, we were running java just to send our books inventory information to Google Clouds every week, the coding was very simple, just defining the JDBC connections with the IBM i and Google data base and send it, just using SQL scripts.

With Python definitely you can do the same, and in a more simple way, you don't need to define fields or long classes to accomplish this target.

One of the challenges I had using Python in my company, was justifying the use, my boss question was, why we need to use Python if we could accomplish the same with RPG coding?...and this was a fair question, at the end I found one reason (at least for me on that time), with Python I will create full customized excel files including Charts, something that you cannot do with RPG at all, this was my starting point, defining a real target, I wanted to "expand" my resources, not replace them.

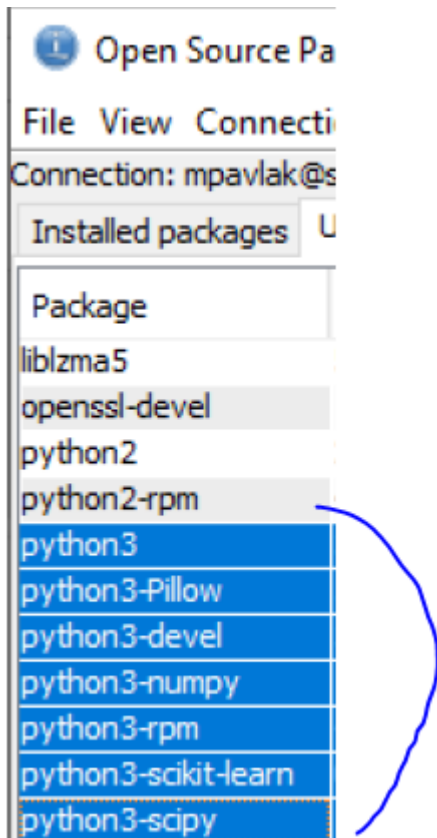
The first challenge was installing Python on the IBM i, I will not get in details here, but in black and white, you need to have functional your IBM I Access Client Solution, if you run this URL, you will see the login screen: <http://your.server.name:2001/>  
If this is not happen, you will need to contact your network administration to make the right corrections.

After you have it active, you will need to go to Tools/Open Source Package Management to start working downloading the python module into the system, don't use other methodologies, follow the instructions from the IBM I Access menus (you will need qsecofr profile to accomplish this task).



You will be entering the SSH connection parameters like IP system address, user and password

Then you will be selecting the python3 version from the Open Source Package management:



After the installation, you have to make sure to define the path where you want to point the python language:

- 1) Login with your QSECOFR
- 2) Run the IBM PASE, you have to execute:

```
call qp2term
```

```
echo 'PATH=/QOpenSys/pkgs/bin:$PATH' >> $HOME/.profile
```

```
export PATH >> $HOME/.profile
```

After you set the path, you will be able to execute any python program from your IFS:

For example:

```
python3 /home/tdvitale/python/hello.py
```

Note: Every time you run the command 'python3', you have to be in the qshell interactive session running just QSH or call the object qp2term

The program "hello.py" must be in your Home/UserID/python IFS folder to be executed.

Now the magic line to run a Python program from a CL or RPG program:

```
QSH CMD('python3 /home/tdvitale/python/hello.py')
```

If for some reason, the program is not found because the path definition I mentioned before failed, you still can use the full path like this one:

```
QSH CMD('/QOpenSys/pkgs/bin/python3 /home/tdvitale/python/hello.py')
```

One important point to how we connect with the IBM i, use the old ODBC driver from the iSeries Access, this is free if you already have it in your installation.

I will not go explaining Python syntax since you can find them in millions of documents and website on line.

In python, you will need to download the module "pyodbc", in QSH session you will need to run the command "pip3"

```
Pip3 install pyodbc
```

Now, if you create your first Python program to access IBM I and retrieve information from your DB, the program will look like this one:

```
Import pyodbc

Connection = pyodbc.connect(

    Driver='{iSeries Access ODBC Driver}',

    System='10.10.2.21',

    UID='UserID'

    PWD='password')

My cursor = connection.execute('select * from lib.file')

Myresult = mycursor.fetchall()

For x in myresult:

    Print(x)

Print("I retrieved data from the IBM i !!!")
```

In this simple program, you define the connection with the driver, IP system you are connecting, UserID and Password, and then you create the cursor to execute an SQL script captured on the object "Myresult".

Again, I will not go deep on Python syntax, just pointing the steps to make Python works for business.

Note: when you define the connection, you could define the connection to point MySQL if you want it (you have to download with PIP3 first the right module).

Now, the big challenge is to create an excel file with charts and send it to the user, the python program will be called from a CL program.

PGM

```
QSH CMD('python3 /home/tdvitale/python/prfcharts.py')
```

```
SNDDST TYPE(*DOC) TOINTNET((email@yahoo.com *PRI)) DSTD('Charts')
MSG('This is an excel file with charts generated with Python')
CDMDEL(*NO) SENSITIV(*NONE) IMPORTANCE(*HIGH) DOC(FlashRptCharts.xlsx)
FLR(EMAIL)
```

The first command is just to call the python program prfcharts.py from the IFS.

The second command is just to send the document generated FlashRptCharts.xlsx to the user.

To create the excel file, you need to download the module xlswriter, here you will find how this work.

<https://xlswriter.readthedocs.io/>

And this is the entire program I wrote to generate the excel file and the charts

---

Note: In this case I was using the `ibm_db` driver

<https://www.ibm.com/support/pages/python-script-accessing-db2-data-simple-example>

```
$ pip3 install PTable
```

Code to test it

```
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))
```

---

```
#####
# Program Name: prfcharts.py #
# Author: Daniel Vitale #
# Date Written: 09/24/2018 #
#####
# F U N C T I O N S #
#####
# This program will print chars for Flash Report for the last #
# 12 Months. #
#####
```

```
# LOG OF PROGRAM MODIFICATIONS #
```

```
#####
```

```
# Date Programmer Description #
```

```
# -----
```

```
# 11/29/2019 Daniel Vitale Date Written #
```

```
#####
```

```
# Import module to Excel files
```

```
import xlswriter
```

```
# Import module to connect with DB with the core and set connection
```

```
import ibm_db_dbi as db2
```

```
conn = db2.connect()
```

```
cur = conn.cursor()
```

```
# Create a workbook and add a worksheet.
```

```
workbook =  
xlswriter.Workbook('/home/TDVITALE/PythonDev/FlashRptCharts.xlsx')
```

```
worksheet2 = workbook.add_worksheet('Charts')
```

```
worksheet1 = workbook.add_worksheet('Data')
```

```
# Protect Sheets
```

```
worksheet2.protect()
```

```
worksheet1.protect()
```

```
# Add a bold format to use to highlight cells.
```

```

bold = workbook.add_format({'bold': True})

colorBlue = workbook.add_format({'color': 'blue'})

# Define Currency format

currency_format = workbook.add_format({'num_format': '$#,###,##0.00'})

# Set the width column (col start, col end, width)

worksheet1.set_column(0,11,18) # Data

worksheet2.set_column(0,0,90) # Charts, first column

worksheet2.set_column(2,2,90) # Charts, third column

# Get the Last Period of the Report

cur.execute("select month(max(ds_date)) || '/' || year(max(ds_date)) from
ssl.lib.prfp001 ")

data=cur.fetchone()

for PeriodT in data:

    Period = PeriodT

# Insert Title of the Report in the Chart sheet

cell_title = workbook.add_format()

cell_title.set_bold()

cell_title.set_font_color('blue')

cell_title.set_font_size('16')

worksheet2.write_string (0,0,'Monthly Flash Report by Charts Combining the
last 3 years for: ' + PeriodT , cell_title)

```

```

# Write the Headers

worksheet1.write_string (0,0,'Item',bold)

worksheet1.write_string (0,1,'Year',bold)
worksheet1.write_string (0,2,'Month',bold)
worksheet1.write_string (0,3,'Amount',bold)

worksheet1.write_string (0,5,'Year',bold)
worksheet1.write_string (0,6,'Month',bold)
worksheet1.write_string (0,7,'Amount',bold)

worksheet1.write_string (0,9,'Year',bold)
worksheet1.write_string (0,10,'Month',bold)
worksheet1.write_string (0,11,'Amount',bold)

# Create charts objects

chart1 = workbook.add_chart({'type': 'column'}) # For Cash
chart2 = workbook.add_chart({'type': 'column'}) # For Total Loans
chart3 = workbook.add_chart({'type': 'column'}) # For Total Assets
chart4 = workbook.add_chart({'type': 'column'}) # For Total Deposits
chart5 = workbook.add_chart({'type': 'column'}) # For Customers Visits
chart6 = workbook.add_chart({'type': 'column'}) # Active Customers

worksheet1.write_string (1,0,'Cash',colorBlue)

worksheet1.write_string (14,0,'Total Loans',colorBlue)

```



```
worksheet1.write_string (27,0,'Total Assets',colorBlue)

worksheet1.write_string (40,0,'Total Deposits',colorBlue)

worksheet1.write_string (53,0,'Customer Visits',colorBlue)

worksheet1.write_string (66,0,'Active Customers',colorBlue)
```

```
# Current Year ---
```

```
cur.execute("SELECT distinct year(a.DS_DATE), month(a.DS_DATE), " +

            "(SELECT b.ds_cash from SSLIB.PRFP001 b where year(a.DS_DATE) =
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE
desc fetch first row only), " +

            "(SELECT b.ds_tot from SSLIB.PRFP001 b where year(a.DS_DATE) =
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE
desc fetch first row only), " +

            "(SELECT b.ds_totas from SSLIB.PRFP001 b where year(a.DS_DATE) =
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE
desc fetch first row only), " +

            "(SELECT b.ds_totd * -1 from SSLIB.PRFP001 b where
year(a.DS_DATE) = year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE)
order by b.DS_DATE desc fetch first row only), " +

            "(SELECT sum(b.ds_cusv) from SSLIB.PRFP001 b where
year(a.DS_DATE) = year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE)), "
+

            "(SELECT b.ds_cus# from SSLIB.PRFP001 b where year(a.DS_DATE) =
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE
desc fetch first row only) " +

            "FROM SSLIB.PRFP001 a " +

            "WHERE year(a.ds_date) = year(current date) " +

            "order by month(a.DS_DATE)")
```

```
# Retrieve the Data
```

```
data = cur.fetchall()
```

```
row = 1
```

```
for Year, Month, Amount_Cash, Amount_Loans, Amount_Assets, Amount_Dep,  
Customers_Visits, Active_Customers in data:
```

```
# Cash
```

```
worksheet1.write_number (row, 1, Year)
```

```
worksheet1.write_number (row, 2, Month)
```

```
worksheet1.write_number (row, 3, Amount_Cash, currency_format)
```

```
# Loans Total
```

```
worksheet1.write_number (row + 13, 1, Year)
```

```
worksheet1.write_number (row + 13, 2, Month)
```

```
worksheet1.write_number (row + 13, 3, Amount_Loans, currency_format)
```

```
# Assets
```

```
worksheet1.write_number (row + 26, 1, Year)
```

```
worksheet1.write_number (row + 26, 2, Month)
```

```
worksheet1.write_number (row + 26, 3, Amount_Assets, currency_format)
```

```
# Deposits
```

```
worksheet1.write_number (row + 39, 1, Year)
```

```
worksheet1.write_number (row + 39, 2, Month)
```

```
worksheet1.write_number (row + 39, 3, Amount_Dep, currency_format)
```

```
# Branch Visits
```

```
worksheet1.write_number (row + 52, 1, Year)
```

```
worksheet1.write_number (row + 52, 2, Month)
```

```
worksheet1.write_number (row + 52, 3, Customers_Visits)
```

```
# Active Customers
```

```
worksheet1.write_number (row + 65, 1, Year)
```

```
worksheet1.write_number (row + 65, 2, Month)
```

```
worksheet1.write_number (row + 65, 3, Active_Customers)
```

```
row +=1
```

```
# One Year Before ---
```

```
cur.execute("SELECT distinct year(a.DS_DATE), month(a.DS_DATE), " +
```

```
        "(SELECT b.ds_cash from SSLIB.PRFP001 b where year(a.DS_DATE) =  
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE  
desc fetch first row only), " +
```

```
        "(SELECT b.ds_tot from SSLIB.PRFP001 b where year(a.DS_DATE) =  
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE  
desc fetch first row only), " +
```

```
        "(SELECT b.ds_totas from SSLIB.PRFP001 b where year(a.DS_DATE) =  
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE  
desc fetch first row only), " +
```

```
        "(SELECT b.ds_totd * -1 from SSLIB.PRFP001 b where  
year(a.DS_DATE) = year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE)  
order by b.DS_DATE desc fetch first row only), " +
```

```
        "(SELECT sum(b.ds_cusv) from SSLIB.PRFP001 b where  
year(a.DS_DATE) = year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE)), "  
+
```

```
        "(SELECT b.ds_cus# from SSLIB.PRFP001 b where year(a.DS_DATE) =  
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE  
desc fetch first row only) " +
```

```
"FROM SSLIB.PRFP001 a " +
```

```
"WHERE year(a.ds_date) = year(current date) - 1 " +
```

```
"order by month(a.DS_DATE)")
```

```
# Retrieve the Data
```

```
data = cur.fetchall()
```

```
row = 1
```

```
for Year, Month, Amount_Cash, Amount_Loans, Amount_Assets, Amount_Dep,  
Customers_Visits, Active_Customers in data:
```

```
    # Cash
```

```
    worksheet1.write_number (row, 5, Year)
```

```
    worksheet1.write_number (row, 6, Month)
```

```
    worksheet1.write_number (row, 7, Amount_Cash, currency_format)
```

```
    row +=1
```

```
    # Loans Total
```

```
    worksheet1.write_number (row + 12, 5, Year)
```

```
    worksheet1.write_number (row + 12, 6, Month)
```

```
    worksheet1.write_number (row + 12, 7, Amount_Loans, currency_format)
```

```
    # Assets
```

```
    worksheet1.write_number (row + 25, 5, Year)
```

```
    worksheet1.write_number (row + 25, 6, Month)
```

```
    worksheet1.write_number (row + 25, 7, Amount_Assets, currency_format)
```

```
    # Deposits
```

```

worksheet1.write_number (row + 38, 5, Year)

worksheet1.write_number (row + 38, 6, Month)

worksheet1.write_number (row + 38, 7, Amount_Dep, currency_format)

# Branch Visits

worksheet1.write_number (row + 51, 5, Year)

worksheet1.write_number (row + 51, 6, Month)

worksheet1.write_number (row + 51, 7, Customers_Visits)

# Active Customers

worksheet1.write_number (row + 64, 5, Year)

worksheet1.write_number (row + 64, 6, Month)

worksheet1.write_number (row + 64, 7, Active_Customers)

# Two Years Before ---

cur.execute("SELECT distinct year(a.DS_DATE), month(a.DS_DATE), " +

            "(SELECT b.ds_cash from SSLIB.PRFP001 b where year(a.DS_DATE) =
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE
desc fetch first row only), " +

            "(SELECT b.ds_tot from SSLIB.PRFP001 b where year(a.DS_DATE) =
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE
desc fetch first row only), " +

            "(SELECT b.ds_totas from SSLIB.PRFP001 b where year(a.DS_DATE) =
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE
desc fetch first row only), " +

            "(SELECT b.ds_totd * -1 from SSLIB.PRFP001 b where
year(a.DS_DATE) = year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE)
order by b.DS_DATE desc fetch first row only), " +

            "(SELECT sum(b.ds_cusv) from SSLIB.PRFP001 b where
year(a.DS_DATE) = year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE)) , "
+

```

```
" (SELECT b.ds_cus# from SSLIB.PRFP001 b where year(a.DS_DATE) =  
year(b.DS_DATE) and month(a.DS_DATE) = month(b.DS_DATE) order by b.DS_DATE  
desc fetch first row only) " +
```

```
"FROM SSLIB.PRFP001 a " +
```

```
"WHERE year(a.ds_date) = year(current date) - 2 " +
```

```
"order by month(a.DS_DATE)"
```

```
# Retrieve the Data
```

```
data = cur.fetchall()
```

```
row = 1
```

```
for Year, Month, Amount_Cash, Amount_Loans, Amount_Assets, Amount_Dep,  
Customers_Visits, Active_Customers in data:
```

```
# Cash
```

```
worksheet1.write_number (row, 9, Year)
```

```
worksheet1.write_number (row, 10, Month)
```

```
worksheet1.write_number (row, 11, Amount_Cash, currency_format)
```

```
# Loans Total
```

```
worksheet1.write_number (row + 13, 9, Year)
```

```
worksheet1.write_number (row + 13, 10, Month)
```

```
worksheet1.write_number (row + 13, 11, Amount_Loans, currency_format)
```

```
# Assets
```

```
worksheet1.write_number (row + 26, 9, Year)
```

```
worksheet1.write_number (row + 26, 10, Month)
```

```
worksheet1.write_number (row + 26, 11, Amount_Assets, currency_format)
```

```
# Deposits
```

```
worksheet1.write_number (row + 39, 9, Year)
```

```
worksheet1.write_number (row + 39, 10, Month)
```

```
worksheet1.write_number (row + 39, 11, Amount_Dep, currency_format)
```

```
# Branch Visits
```

```
worksheet1.write_number (row + 52, 9, Year)
```

```
worksheet1.write_number (row + 52, 10, Month)
```

```
worksheet1.write_number (row + 52, 11, Customers_Visits)
```

```
# Active Customers
```

```
worksheet1.write_number (row + 65, 9, Year)
```

```
worksheet1.write_number (row + 65, 10, Month)
```

```
worksheet1.write_number (row + 65, 11, Active_Customers)
```

```
row +=1
```

```
# Generate the final Charts
```

```
# Cash *****
```

```
chart1.set_title ({'name': 'Cash'})
```

```
chart1.set_x_axis ({'name': 'Months'})
```

```
chart1.set_y_axis ({'name': 'Amounts'})
```

```
chart1.set_size({'width': 620, 'height': 350})
```

```
# Set Current Year
```

```
chart1.add_series({  
    'categories':    '=Data!$C$2:$C$13',  
    'values':       '=Data!$D$2:$D$13',  
    'line':         {'color': 'green'},  
    'name':         '=Data!$B$2:$B$2'  
})
```

```
# Set One Year Before
```

```
chart1.add_series({  
    'categories':    '=Data!$G$2:$G$13',  
    'values':       '=Data!$H$2:$H$13',  
    'line':         {'color': 'blue'},  
    'name':         '=Data!$F$2:$F$2'  
})
```

```
# Set Two Years Before
```

```
chart1.add_series({  
    'categories':    '=Data!$K$2:$K$13',  
    'values':       '=Data!$L$2:$L$13',  
    'line':         {'color': 'red'},  
    'name':         '=Data!$J$2:$J$2'  
})
```



```
# Total Loans *****
```

```
chart2.set_title({'name': 'Total Loans'})
```

```
chart2.set_x_axis({'name': 'Months'})
```

```
chart2.set_y_axis({'name': 'Amounts'})
```

```
chart2.set_size({'width': 620, 'height': 350})
```

```
# Set Current Year
```

```
chart2.add_series({  
    'categories':    '=Data!$C$15:$C$26',  
    'values':       '=Data!$D$15:$D$26',  
    'line':         {'color': 'green'},  
    'name':         '=Data!$B$15:$B$15'  
})
```

```
# Set One Year Before
```

```
chart2.add_series({  
    'categories':    '=Data!$G$15:$G$26',  
    'values':       '=Data!$H$15:$H$26',  
    'line':         {'color': 'blue'},  
    'name':         '=Data!$F$15:$F$15'  
})
```

```
# Set Two Years Before
```

```
chart2.add_series({  
    'categories':    '=Data!$K$15:$K$26',
```

```
'values': '=Data!$L$15:$L$26',
'line':   {'color': 'red'},
'name':   '=Data!$J$15:$J$15'
})
```

```
# Total Assets *****
```

```
chart3.set_title({'name': 'Total Assets'})
chart3.set_x_axis({'name': 'Months'})
chart3.set_y_axis({'name': 'Amounts'})
chart3.set_size({'width': 620, 'height': 350})
```

```
# Set Current Year
```

```
chart3.add_series({
'categories': '=Data!$C$28:$C$39',
'values':     '=Data!$D$28:$D$39',
'line':      {'color': 'green'},
'name':      '=Data!$B$28:$B$28'
})
```

```
# Set One Year Before
```

```
chart3.add_series({
'categories': '=Data!$G$28:$G$39',
'values':     '=Data!$H$28:$H$39',
'line':      {'color': 'blue'},
'name':      '=Data!$F$28:$F$28'
```

```
})
```

```
# Set Two Years Before
```

```
chart3.add_series({  
    'categories':    '=Data!$K$28:$K$39',  
    'values':    '=Data!$L$28:$L$39',  
    'line':    {'color': 'red'},  
    'name':    '=Data!$J$28:$J$28'  
})
```

```
# Total Deposits *****
```

```
chart4.set_title ({'name': 'Total Deposits'})  
chart4.set_x_axis ({'name': 'Months'})  
chart4.set_y_axis ({'name': 'Amounts'})  
chart4.set_size ({'width': 620, 'height': 350})
```

```
# Set Current Year
```

```
chart4.add_series({  
    'categories':    '=Data!$C$41:$C$52',  
    'values':    '=Data!$D$41:$D$52',  
    'line':    {'color': 'green'},  
    'name':    '=Data!$B$41:$B$41'  
})
```

```
# Set One Year Before
```

```
chart4.add_series({
    'categories':    '=Data!$G$41:$G$52',
    'values':        '=Data!$H$41:$H$52',
    'line':          {'color': 'blue'},
    'name':          '=Data!$F$41:$F$41'
})
```

```
# Set Two Years Before
```

```
chart4.add_series({
    'categories':    '=Data!$K$41:$K$52',
    'values':        '=Data!$L$41:$L$52',
    'line':          {'color': 'red'},
    'name':          '=Data!$J$41:$J$41'
})
```

```
# Total Visits *****
```

```
chart5.set_title({'name': 'Total Branch Visits'})
```

```
chart5.set_x_axis({'name': 'Months'})
```

```
chart5.set_y_axis({'name': 'Amounts'})
```

```
chart5.set_size({'width': 620, 'height': 350})
```

```
# Set Current Year
```

```
chart5.add_series({
    'categories':    '=Data!$C$54:$C$65',
    'values':        '=Data!$D$54:$D$65',
```

```
'line': {'color': 'green'},
'name': '=Data!$B$54:$B$54'
})
```

```
# Set One Year Before
```

```
chart5.add_series({
'categories': '=Data!$G$54:$G$65',
'values': '=Data!$H$54:$H$65',
'line': {'color': 'blue'},
'name': '=Data!$F$54:$F$54'
})
```

```
# Set Two Years Before
```

```
chart5.add_series({
'categories': '=Data!$K$54:$K$65',
'values': '=Data!$L$54:$L$65',
'line': {'color': 'red'},
'name': '=Data!$J$54:$J$54'
})
```

```
# Active Customers *****
```

```
chart6.set_title({'name': 'Active Customers'})
```

```
chart6.set_x_axis({'name': 'Months'})
```

```
chart6.set_y_axis({'name': 'Amounts'})
```

```
chart6.set_size({'width': 620, 'height': 350})
```

```
# Set Current Year
```

```
chart6.add_series({  
    'categories':    '=Data!$C$67:$C$78',  
    'values':    '=Data!$D$67:$D$78',  
    'line':    {'color': 'green'},  
    'name':    '=Data!$B$67:$B$67'  
})
```

```
# Set One Year Before
```

```
chart6.add_series({  
    'categories':    '=Data!$G$67:$G$78',  
    'values':    '=Data!$H$67:$H$78',  
    'line':    {'color': 'blue'},  
    'name':    '=Data!$F$67:$F$67'  
})
```

```
# Set Two Years Before
```

```
chart6.add_series({  
    'categories':    '=Data!$K$67:$K$78',  
    'values':    '=Data!$L$67:$L$78',  
    'line':    {'color': 'red'},  
    'name':    '=Data!$J$67:$J$67'  
})
```

```
# Insert the charts into the worksheet2 (for Charts)
```

```
worksheet2.insert_chart('A3', chart1) # Cash
worksheet2.insert_chart('C3', chart2) # Loans
worksheet2.insert_chart('A22', chart3) # Assets
worksheet2.insert_chart('C22', chart4) # Deposits
worksheet2.insert_chart('A41', chart5) # Visits
worksheet2.insert_chart('C41', chart6) # Active Customers

workbook.close()
```