Python and Excel

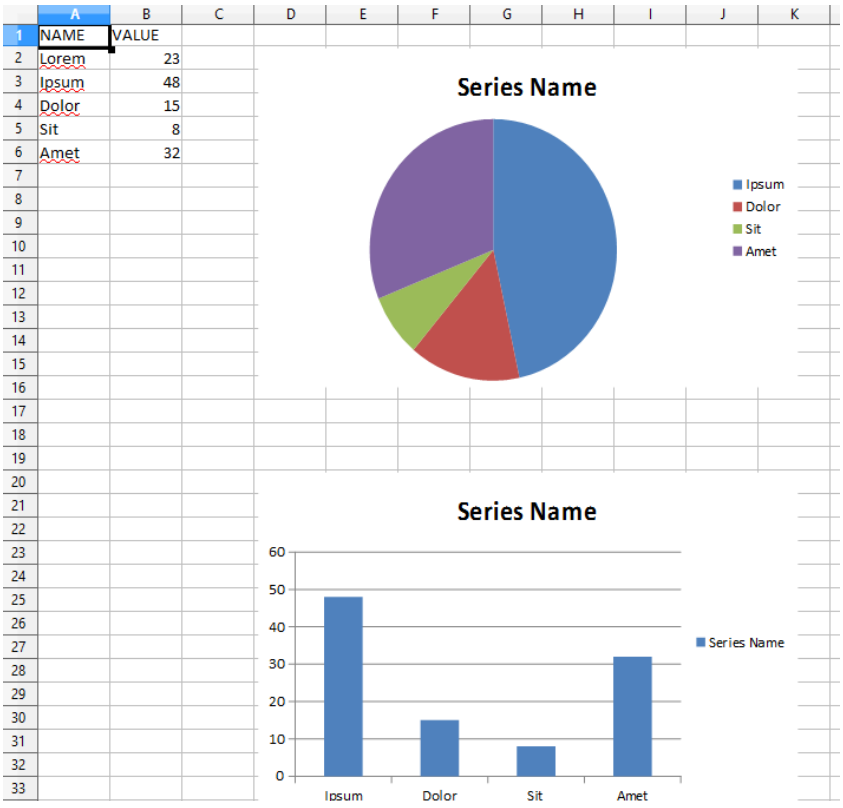
All Versions

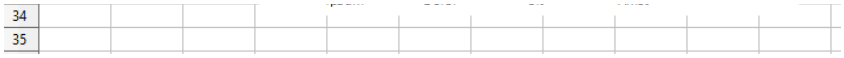
Examples

Create excel charts with xlsxwriter

```
import xlsxwriter

# sample data
chart_data = [
    {'name': 'Lorem', 'value': 23},
    {'name': 'Ipsum', 'value': 48},
    {'name': 'Dolor', 'value': 15},
    {'name': 'Sit', 'value': 8},
    {'name': 'Amet', 'value': 32}
]

# excel file path
xls_file = 'chart.xlsx'

# the workbook
workbook = xlsxwriter.Workbook(xls_file)

# add worksheet to workbook
worksheet = workbook.add_worksheet()

row_ = 0
col_ = 0

# write headers
worksheet.write(row_, col_, 'NAME')
col_ += 1
worksheet.write(row_, col_, 'VALUE')
row_ += 1

# write sample data
for item in chart_data:
    col_ = 0
    worksheet.write(row_, col_, item['name'])
    col_ += 1
    worksheet.write(row_, col_, item['value'])
    row_ += 1
```

Result:





Format Excel files with xlsxwriter

```
import xlsxwriter

# create a new file
workbook = xlsxwriter.Workbook('your_file.xlsx')

# add some new formats to be used by the workbook
percent_format = workbook.add_format({'num_format': '0%'})
percent_with_decimal = workbook.add_format({'num_format': '0.0%'})
bold = workbook.add_format({'bold': True})
red_font = workbook.add_format({'font_color': 'red'})
remove_format = workbook.add_format()

# add a new sheet
worksheet = workbook.add_worksheet()

# set the width of column A
worksheet.set_column('A:A', 30, )

# set column B to 20 and include the percent format we created earlier
worksheet.set_column('B:B', 20, percent_format)

# remove formatting from the first row (change in height=None)
worksheet.set_row('0:0', None, remove_format)

workbook.close()
```

OpenPyXL

OpenPyXL is a module for manipulating an `xlsx` workbook in memory.

```
import openpyxl as opx
workbook = opx.load_workbook(workbook_path)
```

Once you have loaded the workbook into memory, you can access the individual sheets using `workbook.sheets`

```
first_sheet = workbook.worksheets[0]
```

If you want to check the names of the available sheets, you can use `workbook.get_sheet_names()` then use the name to access the sheet.

```
named_sheet = workbook.get_sheet_by_name(sheet_name)
```

Finally, the rows of the sheet can be accessed using `sheet.rows` . To iterate over the rows in a sheet, use:

```
for row in sheet.rows:
    print row[0].value
```

Since each `row` in `rows` is a list of `Cell` s, use `Cell.value` to get the contents of the Cell.

Put list data into a Excel's file.

```
import os, sys
from openpyxl import Workbook
from datetime import datetime

dt = datetime.now()
list_values = [["01/01/2016", "05:00:00", 3], \
               ["01/02/2016", "06:00:00", 4], \
               ["01/03/2016", "07:00:00", 5], \
               ["01/04/2016", "08:00:00", 6], \
               ["01/05/2016", "09:00:00", 7]]

# Create a Workbook on Excel:
wb = Workbook()
sheet = wb.active
sheet.title = 'data'
```

```
# Print the titles into Excel Workbook:
row = 1
sheet['A'+str(row)] = 'Date'
sheet['B'+str(row)] = 'Hour'
sheet['C'+str(row)] = 'Value'

# Populate with data
for item in list_values:
    row += 1
    sheet['A'+str(row)] = item[0]
    sheet['B'+str(row)] = item[1]
    sheet['C'+str(row)] = item[2]

# Save a file by date:
filename = 'data_' + dt.strftime("%Y%m%d_%I%M%S") + '.xlsx'
wb.save(filename)

# Open the file for the user:
os.chdir(sys.path[0])
os.system('start excel.exe "%s\\%s"' % (sys.path[0], filename, ))
```

Read the excel data using xlrd module

Python xlrd library is to extract data from Microsoft Excel (tm) spreadsheet files.

Installation:-

```
pip install xlrd
```

Or you can use setup.py file from pypi

https://pypi.python.org/pypi/xlrd

Reading an excel sheet:- Import xlrd module and open excel file using open_workbook() method.

```
import xlrd
book=xlrd.open_workbook('sample.xlsx')
```

Check number of sheets in the excel

```
print book.nsheets
```

Print the sheet names

```
print book.sheet_names()
```

Get the sheet based on index

```
sheet=book.sheet_by_index(1)
```

Read the contents of a cell

```
cell = sheet.cell(row,col) #where row=row number and col=column number
print cell.value #to print the cell contents
```

Get number of rows and number of columns in an excel sheet

```
num_rows=sheet.nrows
num_col=sheet.ncols
```

Get excel sheet by name

```
sheets = book.sheet_names()
cur_sheet = book.sheet_by_name(sheets[0])
```

Syntax

Parameters

Remarks