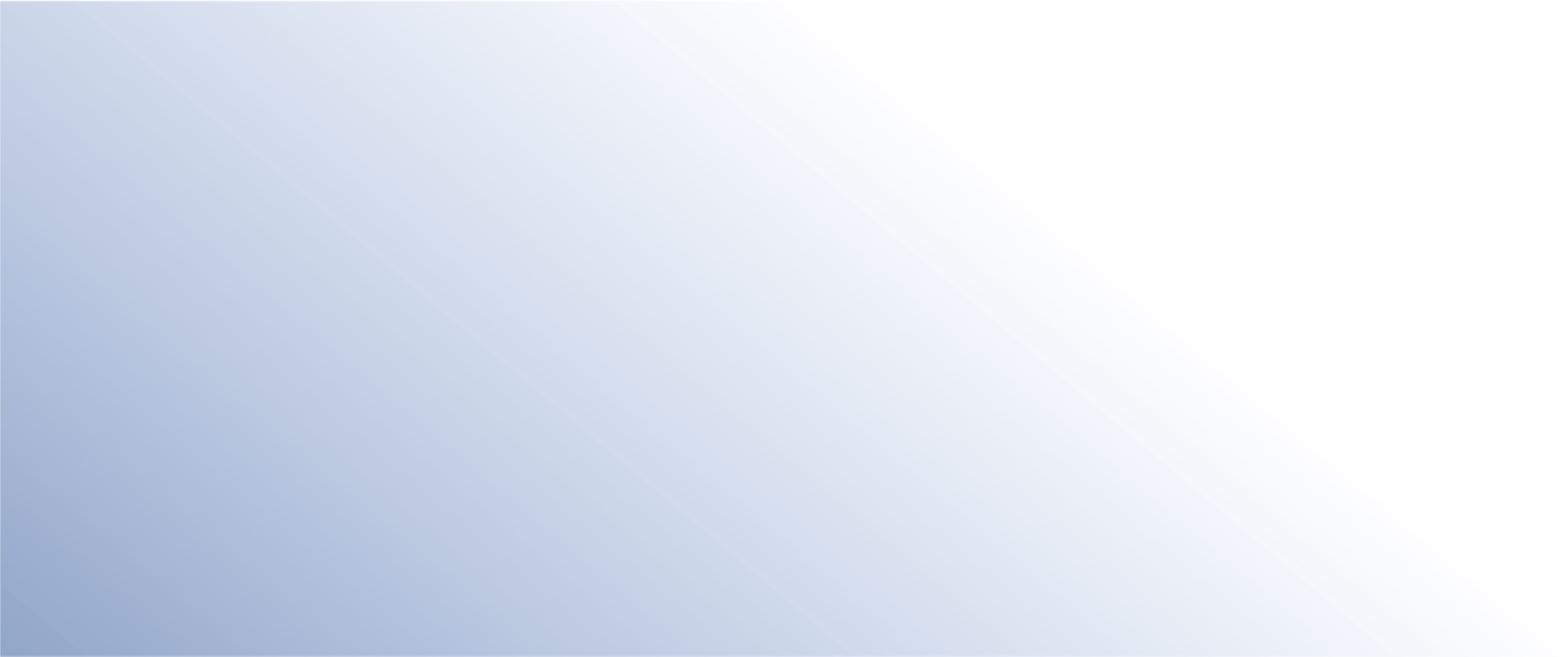
# NAME:NIDHI SINHA ROLL.NO:1DB20CI029 BRANCH:ARTIFICIAL INTELLIGENCE & MACHINE LEARNING SUBJECT:PYTHON PROGRAMMING



## TOPIC:PYTHON FOR SPREADSHEET USERS

• Excel is one of the popular spreadsheets for data management. While it is tedious to write and update the data in a long Excel sheet, Python helps in minimizing this task and helps easy creation, reading, writing of Excel sheet. This can be done by various Python libraries,3 of which will be discussed in this article.

Python for

Spreadsheet

Users

# Using Openpyxl Library

• penpyxl is a Python library for reading and writing Excel (with extension xlsx/xlsm/xltx/xltm) files. The openpyxl module allows

Python program to read and modify Excel files.

*pip install*

*openpyxl*

INSTALLATION

Creating or Writing a

Workbook

# import modules from openpyxl import Workbook import datetime

wb = Workbook()

|  |  |  |
| --- | --- | --- |
| • | We write the code in our code editor and save the code | ws = wb.active  # assign sheet ws['A1'] = 75 |

# add data ws.append([1, 2, 3])

ws.append([4, 5, 6])

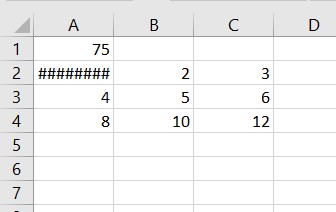
ws.append([8, 10, 12])

ws['A2'] = datetime.datetime.now()

# save spreadsheet

wb.save("test.xlsx")

Output:



Explanation:

* We first import the library and the Spreadsheet is given a workbook instance. Afterward, we assign random values corresponding to columns A. Here we have appended in A1 and A2. We can also override the original values as 1 in A1 and 4 in A2 has been overwritten as respective dates and time. Lastly, we save the sheet, where it’s saved as “test”.

Reading a Workbook:

# import modules

from openpyxl import

load\_workbook

We can also read a from openpyxl import Workbook Workbook from this

program # load spreadsheet

* This opens the same wb =

load\_workbook(filename = file which we had 'test.xlsx') opened earlier

# import modules from openpyxl import Workbook

Inserting Images in the Workbook: from from openpyxlopenpyxl.drawing.imageimport Workbookimport Image

Spreadsheet = Workbook()

worksheet = Spreadsheet.active

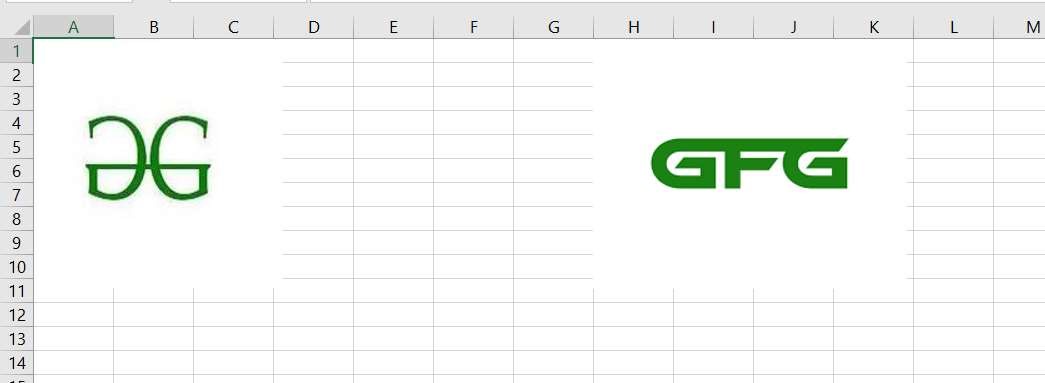
• We can also insert a few images wb = Workbook() in our Excel sheet as ws = wb.active demonstrated in the code # assign title

ws['A1'] = 'Two logos demonstrated'

# create an image img = Image('gfg.png') img2=Image('gfg2.png')

# add image ws.add\_image(img, 'A1') ws.add\_image(img2, 'H1') wb.save('logo.xlsx')

Output



Explanation:

•

We have to download the pillow

library for dependency and import

the sub

-

libraries as shown. We

now download two images we like

and just give them the designated

row, and we have our image

inserted!

## Using XLWT Library

oThis is a library for developers to use to generate

spreadsheet files compatible with Microsoft Excel versions 95 to 2003. The package itself is pure Python with no dependencies on modules or packages outside the standard Python distribution.

oWe can make values bold, colored, italic, etc. Also, the font size can be controlled.

Install the python package by the following command

*pip3 install xlwt*

# Installation *Or*

*pip install xlwt*

# import module

Creating a Workbook import xlwt

from datetime import datetime

• Given code which gives explore the formatting value,something not explored in

# assign attributes

style0 = xlwt.easyxf('font: name Verdana, color-index green, bold on',)

style1 = xlwt.easyxf(num\_format\_str='D-MMM-YY')

style2 = xlwt.easyxf(

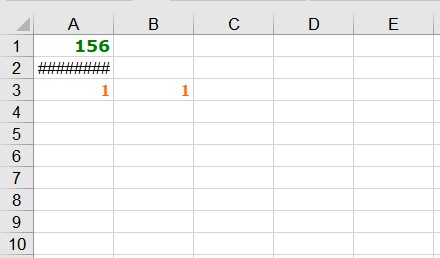
'font: name Times New Roman, color-index orange, bold on',) wb = xlwt.Workbook() ws = wb.add\_sheet('A Test Sheet')

# add data ws.write(0, 0, 156, style0) ws.write(1, 0,

datetime.now(), style1) ws.write(2, 0, 1, style2) ws.write(2, 1, 1, style2) ws.write(2, 2, xlwt.Formula("A3+B3"))

# printing results wb.save('example.xls')

**Output:**



Explanation:

* We import the library and then declare 3 different styles to apply to the values. Style 1 explores one family and color green, while the third one is Times New Roman, and the color is orange with styling as bold. Stye 1 is the format of dates if mentioned, and it’s D-

MM-YY

* We write the values this time in respective rows and columns both starting with 0 indexes. Hence, ws.write(row, column, value, style). Style is optional here.

• XlsxWriter is a Python module for writing files in the Excel 2007+ XLSX file format.XlsxWriter can be used to write text, numbers, formulas, and hyperlinks to multiple worksheets, and it supports features Here is a simple example: Conditional formatting., Worksheet PNG/JPEG/BMP/WMF/EMF images, Rich multi-format strings, Cell comments., Integration with Pandas., Textboxes., Support for adding Macros.

Using Xlsx

writer

•Install the below library from the

commands

*pip install XlsxWriter*



Installation

*Or*

*pip3 install*

*XlsxWriter*

# import module import xlsxwriter

|  |  |
| --- | --- |
| **Example** | # Create an new Excel file and add a |

# worksheet.

This is another way to create an Excel sheet and with a wide range of options as well as images of all extensions supported. Let's take a code example.

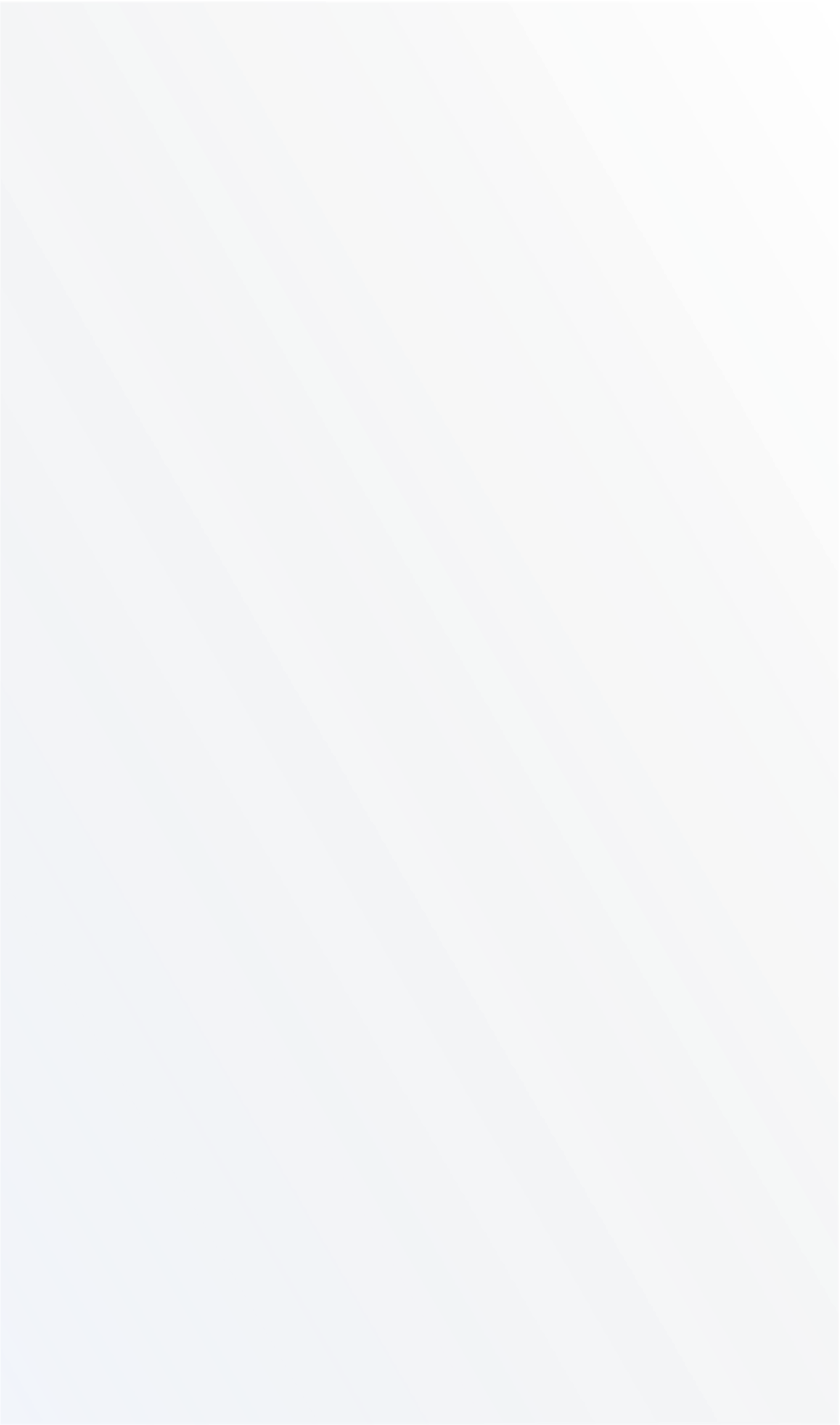
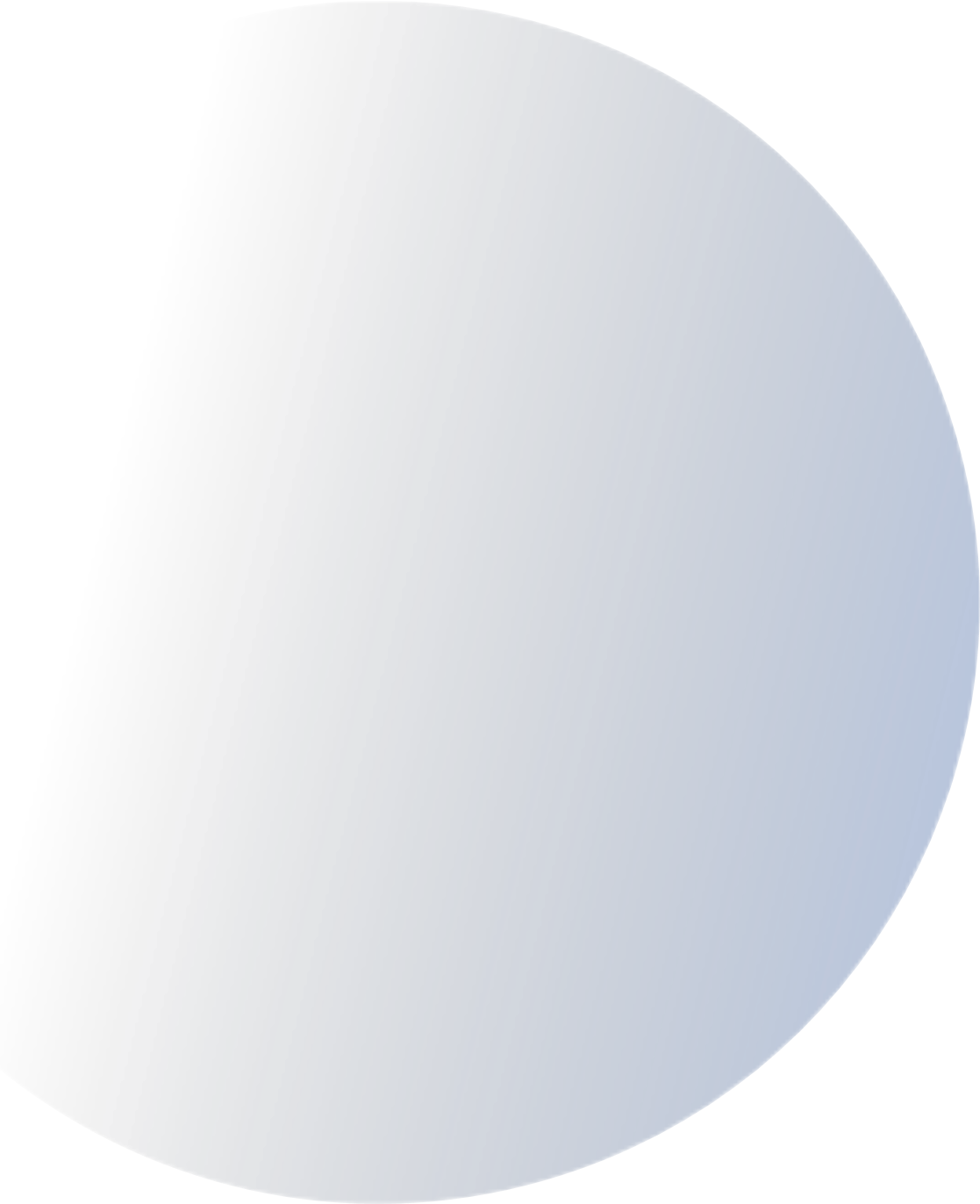
workbook = xlsxwriter.Workbook('demo.xlsx') ws = workbook.add\_worksheet() # Widen the first column to make the # text clearer.

ws.set\_column('A:A', 20) # Adding a bold format to use to highlight # cells.

style = workbook.add\_format({'bold': True})

# add data ws.write('A1', 'Geeks') ws.write('A2', 'for Geeks', style) ws.insert\_image('B5', 'logo.png') workbook.close()

We first import the said library and create a workbook (demo.xlsx) and start by setting a wide column and giving the size as 20. We then declare a style name bold where we set style as true. Next, we write the word “Geeks” with no style but the next word “for Geeks” is applied to the style we have given. Notice as compared to the earlier example we use true instead of on and easyxf function to declare a style. This makes xlsxWriter the most readable and user-free syntax compared to the other two.



**Explanation**

Output:

