Getting sheets from the workbook

```
my_sheetnames = wb.sheetnames # return list object
```

Charts

```
import openpyxl.chart
wb_chart = openpyxl.Workbook()
sheet_chart = wb_chart.active
for i in range(1, 11):
sheet_chart['A' + str(i)] = i

refObj = openpyxl.chart.Reference(sheet-_chart, min_col=1, min_row=1, max_col=1, max_row=10)
seriesObj = openpyxl.chart.Series(refObj, title='First series')
chartObj = openpyxl.chart.BarChart()
chartObj.title = 'My Chart'
chartObj.append(seriesObj)
sheet_chart.add_chart(chartObj, 'C5')
wb_chart.save('sampleChart.xlsx')
```

Charts

```
# 1. Create a Reference object from a rectangular selection of cells.
# 2. Create a Series object by passing in the Reference object.
# 3. Create a Chart object.
# 4. Append the Series object to the Chart object.
# 5. Add the Chart object to the Worksheet object, optionally specifying which cell should be
# the top-left corner of the chart.
# Ex.: BarChart()
```

Get the row, column, coordinate from the cell

```
cell_A1_row = sheet['A1'].row
cell_A1_column = sheet['A1'].column
cell_A1_coordinate = sheet['A1'].coordinate
area_cells = sheet['A1':'C3'] # tuple of all the cell objects
```

Get the value from the cell

```
cell_A1_value = sheet['A1'].value
```

Get the active sheet

```
anotherSheet = wb.active
```

Opening excel documents with openpyxl

```
import openpyxl
wb = openpyxl.load_workbook('example.x-lsx')
```

Get a sheet from the workbook

```
sheet = wb[my_sheetnames[0]] # sheet3 for example
```

Getting a cell from the sheet

```
cell_A1 = sheet['A1']
```

Get the sheet's title as a string

```
my_titles = sheet.title
```

Moving ranges. The cells will overwrite

```
sheet.move_range("D4:F10", rows=-1, cols=2)
```

Unmerge cells

```
wb_unmerge = openpyxl.load_workbook('-merged.xlsx')
sheet_unmerge = wb_unmerge.active
sheet_unmerge.unmerge_cells('A1:D3')
sheet_unmerge.unmerge_cells('C5:D5')
wb_unmerge.save('unmerged.xlsx')
```

To set the value of these merged cells

```
sheet_merge['A1'] = 'Twelve cells merged together.'
sheet_merge.merge_cells('C5:D5')
sheet_merge['C5'] = 'Two merged cells.'
wb_merge.save('merged.xlsx')
```

Merging and Unmerging Cells

```
wb_merge = openpyxl.Workbook()
sheet_merge = wb_merge.active
sheet_merge.merge_cells('A1:D3')
```

Get the rows, columns

```
# Using the rows return a tuple of tuples. Inner tuples - row.
# Using the columns return a tuple of tuples. Inner tuples - the cell object in a particular column.
# Convert to list with the list() function. Use index in the larger tuple.
# Ex.: to get the tuple that represents row 1
tuple_row_1 = list(sheet.rows)[0]
# Ex.: to get the tuple that represents column B
tuple_column_B = list(sheet.columns)[1]
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Converting between column letters and numbers

```
from openpyxl.utils import get_column_letter, column_index_from_string
col_letter = get_column_letter(1)
col_max_letter = get_column_letter(sheet.max_column)
index_letter = column_index_from_string('A')
# Get A's number
```

Get the highest column number

```
sheet_max_column = sheet.max_column
```

Get the highest row number

```
sheet_max_row = sheet.max_row
```

Getting a cell using row and column

```
cell_B1 = sheet.cell(row=1, column=2) # if add argument 'value=' it'll change the value of cell
```

Insert row

```
sheet.insert_rows(7)
```

Creating and Removing Sheets

```
wb_new.create_sheet() # Add a new sheet
wb_new.create_sheet(index=0, title='First sheet') # Create a new sheet at index 0
wb_new.create_sheet(index=2, title='Middle sheet') # Create a new sheet at index 2
del wb_new['Middle sheet'] # Remember to call the save() method to save changes
```

Writing Values to Cells

```
# Writing values to cells is much like writing values to keys in a dictionary.
sheet_new['A1'] = 'Hello, world!'
print(sheet_new['A1'].value)
```

Module openpyxl.styles

```
# Setting the Font Style of Cells
from openpyxl.styles import Font
wb_style = openpyxl.Workbook()
sheet_style = wb_style['Sheet']
italic24Font = Font(size=24, italic=True, name='Calibri') # Create a font.
sheet_style['A1'].font = italic24Font # Apply the font to A1.
sheet_style['A1'] = 'Hello, world!'
wb_style.save('styles.xlsx')
```

Formulas

```
# Add formulas to cell just like any normal value.
wb_formulas = openpyxl.Workbook()
sheet_formulas = wb_formulas.active
sheet_formulas['A1'] = 200
sheet_formulas['A2'] = 300
sheet_formulas['A3'] = '=SUM(A1:A2)' # Set the formula
wb_formulas.save('writeFormula.xlsx')
```

Setting Row Height and Column Width

```
wb_dimension = openpyxl.Workbook()
sheet_dimension = wb_dimension.active
sheet_dimension['A1'] = 'Tall row'
sheet_dimension['B2'] = 'Wide column'
sheet_dimension.row_dimensions[1].height = 70 # Set the height
sheet_dimension.column_dimensions['B'].width = 20 # Set the width
sheet_dimension.column_dimensions['C'].hidden = True # Hide the column 'C'
wb_dimension.save('dimensions.xlsx')
```

Freezing Panes

```
# All rows above and columns to the left of this cell will be frozen
# To unfreeze all panes, set freez_panes to None or 'A1'
wb_freeze = openpyxl.load_workbook('produceSales.xlsx')
sheet_freeze = wb_freeze.active
sheet_freeze.freeze_panes = 'A2' # Freeze the rows above A2.
wb_freeze.save('freezeExample.xlsx')
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com