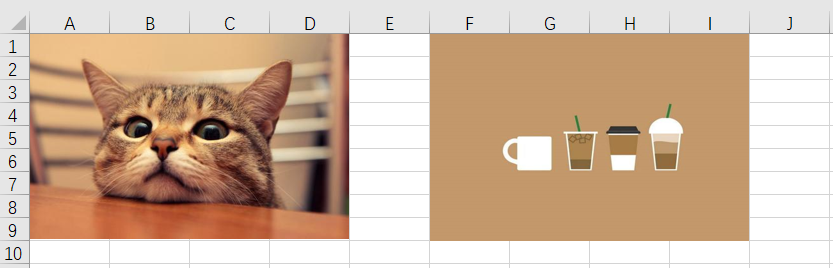
在上篇推文《操作Excel的Xlwings教程（四）》（添加链接）中重点介绍了Xlwings中的Shape和Chart等API知识点。本次推文将继续上次的知识点，将继续介绍Xlwings中其他API相关的知识，并展示一些小例子，让大家可以跟着教程进行学习：

1. **Pictures**

图片类的操作在Excel中是非常常见的，那么在xlwings中对图片的操作有哪些常用的方法和属性呢？接下来我将介绍其中的一些常用的操作，假设我们现在有一个文件1.xlsx，其sheet页的有两张图片：



我们来看一些代码：

**import** xlwings **as** xw  
wb = xw.Book(**'1.xlsx'**)  
wb.sheets[0].pictures *# 查看引用的sheet页中图片的对象*wb.sheets[0].pictures.count *# 统计引用的sheet页中图片对象的数量，次数输出2*

那么我们如何增加新的图片呢,我们可以这样进行操作：

wb.sheets[0].pictures.add(**r'C:\Users\LEGION\Desktop\1.jpg'**)

当然了增加的图片会有默认的位置，具体的位置大家一试便知，值的说明的是这个add()方法它有几个参数，总结如下：

|  |  |
| --- | --- |
| image | 文件路径或Matplotlib图形对象。 |
| left | 以磅为单位距离左侧边缘的位置，默认为0。 |
| top | 以磅为单位距离上侧边缘的位置，默认为0。 |
| width | 设置图度。 |
| height | 设置图高。 |
| name | Excel图片名称。如果未提供，则默认为Excel标准名称，例如“图片1”。 |
| update | 替换更新图片。 |
| scale | 缩放尺度。 |

值的一提的是，xlwings支持对Matpoltlib对象的操作，那么在xlwings中怎么插入Matplotlib的图像呢？

**import** matplotlib.pyplot **as** plt  
**import** numpy **as** np  
x = [-4,-3,-2,-1,0,1,2,3,4]  
figure = plt.figure()  
plt.plot(np.cos(x)/2,np.sin(x)/3)  
wb.sheets[0].pictures.add(figure, name=**'sin#cos'**, update=**True**)

当然，可同样使用上述介绍的参数对图片的位置和大小进行调整。

1. **Pictures**

老话：picture对象是pictures集合的成员

**方法和属性：**

wb.sheets[0].pictures[0] *#引用图片，或者wb.sheets[0].charts['PictureName']*

wb.sheets[0].pictures[0].delete() *# 删除图片*

wb.sheets[0].pictures[0].height *# 返回或设置代表图片高度*

wb.sheets[0].pictures[0].left *# 返回或设置图片水平位置*

wb.sheets[0].pictures[0].width *# 返回或设置图片宽度。*

wb.sheets[0].pictures[0].top *# 返回或设置图片垂直位置。*

wb.sheets[0].pictures[0].name *#返回或设置图片的名称。*

wb.sheets[0].pictures[0].parent *# 返回图片的父级,输出<Sheet [1.xlsx]Sheet1>*

wb.sheets[0].pictures[0].update(**'图片路径'**) *#用新图片替换现有图片*

以上就是picture对象的相关方法，注意在使用这些方法的时候一定要引用正确的sheet页的picture，不要引用错对象。

前几篇推文和今天介绍的picture就是xlwings中常见的一些对象知识了。

1. **题外分析**

有人有可能有这样的一个疑问，Excel的VBA和xlwings在处理数据方面速度是怎样的呢？这里不在进行写代码的方式来说明，我们来分析一下看看，以便分析一下VBA的在这方面的优势和不足，假设我们现在要计算上十万行Excel文件，操作其中两列数据的和：

**方法1：**使用xlwings读Range的数据进行操作，显然这种方式要使用for循环反复的引用range对象，使得在计算的过程中这是非常耗时的。

**方法2：**使用xlwings将数据一次性读入python的List中，在List中进行计算返回结果，这样的操作方式避免了大量引用range对象，减少了耗时，当然了数量量很大的时候也就对设备的内存提出了新的要求。

**方法3：**使用VBA进行操作。VBA是Visual Basic的一种宏语言，是在其桌面应用程序中执行通用的自动化(OLE)任务的编程语言。跟使用使用第三方库不一样的是，VBA操作的时候

是基于更加底层的接口来实现的，所以从这个角度来看，这种方式是肯定比方法1快的。

**方法4：**当然了，我们知道xlwings是支持pandas的对象的，因此我们也可以使用pandas来进行处理，毕竟pandas是处理数据的神器，而且pandas处理大量的数据也有其独有的优势，这是值得我们注意的。

不管哪种方式，都是解决问题的方式，都值得我们在平时的学习中进行总结和归纳，感兴趣的小伙伴可以生成数据实时这几种方式到底哪种更优哦。

下次的文章内容还没想好写什么，不过也都是跟python，机器学习，办公自动化等相关的内容啦！因为有一些小伙伴不懂python嘛，所以本人可能更新python的基础教程，当然在每次的基础教程后面，本人会出题提供大家练习，也会根据所写的内容进行拓展哦！

最后感谢大家的关注！