# 沪深上市公司年报解析（表格不含有背景颜色的）

## 代码说明

总共含有4个py文件，除了Main主函数以外，

xsb\_parsing\_cbfx.py 代表解析成本分析

xsb\_parsing\_zyyw.py 代表解析主营业务

xsb\_parsing\_cwfx.py 代表解析财务分析

Main函数的设计思路：

遍历某一个文件夹下所有的年报PDF，并根据年报名，创建以公司名字、股票代码以及报告日期为名的文件夹，通过以上3个py解析文件，将解析的结果保存成对应的3个json文件，json文件以公司名字、股票代码、报告日期和表名字命名（cwfx表示财务分析结果、zyyw表示主营业务、cbfx表示成本分析）

如何解析信息？（以财务分析为例）具体的函数说明见代码注释。

1.根据财务分析三张表的字段，先定位各个表所在的pdf中的页码位置。

如下：

# 资产负债  
for i in range(len(new\_table)):  
 if len(list\_page\_zcfz) == 2:  
 break  
  
 if table\_header\_cwfx[0][1] in new\_table[i]:  
 list\_page\_zcfz.append(pages)  
 if len(new\_table[i]) > 0:  
 # print(new\_table[i])  
 for j in range(len(new\_table[i])):  
 if table\_header\_cwfx[0][2] in new\_table[i][j]:  
 list\_page\_zcfz.append(pages)

2.根据定位的页码以及各个表格表头字段获取每个表的重要信息，对应一个表跨页，需要每一页的分析单独做处理。

如：

def get\_information\_1(pages,list\_page,tables,map):  
 if pages >= list\_page[0] and pages <= list\_page[len(list\_page) - 1]:  
  
 if pages == list\_page[0]:  
  
 for index, p in enumerate(tables):  
  
 new\_p = select\_and\_delete(p)  
  
 if index == len(tables) - 1:  
 # print(new\_p)  
 for k in range(len(new\_p)):  
 if k == 0:  
 if len(new\_p[k]) == 4:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][2], new\_p[k][3])  
 else:  
 map['项目'] = ('期末余额', '期初余额')  
 else:  
 # print(new\_p)  
 if len(new\_p[k]) == 4:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][2], new\_p[k][3])  
 elif len(new\_p[k]) == 3:  
 if boolenzh(new\_p[k][1]):  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][2], '')  
 else:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][1], new\_p[k][2])  
 elif len(new\_p[k]) == 2:  
 if boolenzh(new\_p[k][1]):  
 map[split\_kongge(new\_p[k][0])] = ('', '')  
 else:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][1], '')  
 elif len(new\_p[k]) == 1:  
 map[split\_kongge(new\_p[k][0])] = ('', '')  
 else:  
 continue  
  
  
  
  
  
 elif (list\_page[-1] - list\_page[0]) > 1 and pages == list\_page[len(list\_page) - 1]:  
 for index, p in enumerate(tables):  
  
 new\_p = select\_and\_delete(p)  
 if index == 0:  
  
 for k in range(len(new\_p)):  
  
 if len(new\_p[k]) == 4:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][2], new\_p[k][3])  
 elif len(new\_p[k]) == 3:  
 if boolenzh(new\_p[k][1]):  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][2], '')  
 else:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][1], new\_p[k][2])  
 elif len(new\_p[k]) == 2:  
 if boolenzh(new\_p[k][1]):  
 map[split\_kongge(new\_p[k][0])] = ('', '')  
 else:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][1], '')  
 elif len(new\_p[k]) == 1:  
 map[split\_kongge(new\_p[k][0])] = ('', '')  
 else:  
 continue  
  
  
 elif (list\_page[-1] - list\_page[0]) == 1 and pages == list\_page[len(list\_page) - 1]:  
 for index, p in enumerate(tables):  
 new\_p = select\_and\_delete(p)  
 if index == 0:  
 for k in range(len(new\_p)):  
  
 if len(new\_p[k]) == 4:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][2], new\_p[k][3])  
 elif len(new\_p[k]) == 3:  
 if boolenzh(new\_p[k][1]):  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][2], '')  
 else:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][1], new\_p[k][2])  
 elif len(new\_p[k]) == 2:  
 if boolenzh(new\_p[k][1]):  
 map[split\_kongge(new\_p[k][0])] = ('', '')  
 else:  
 map[new\_p[k][0]] = (new\_p[k][1], '')  
 elif len(new\_p[k]) == 1:  
 map[split\_kongge(new\_p[k][0])] = ('', '')  
 else:  
 continue  
  
 else:  
 for index, p in enumerate(tables):  
 new\_p = select\_and\_delete(p)  
 for k in range(len(new\_p)):  
  
 if len(new\_p[k]) == 4:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][2], new\_p[k][3])  
 elif len(new\_p[k]) == 3:  
 if boolenzh(new\_p[k][1]):  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][2], '')  
 else:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][1], new\_p[k][2])  
 elif len(new\_p[k]) == 2:  
 if boolenzh(new\_p[k][1]):  
 map[new\_p[k][0]] = ('', '')  
 else:  
 map[split\_kongge(new\_p[k][0])] = (new\_p[k][1], '')  
 elif len(new\_p[k]) == 1:  
 map[split\_kongge(new\_p[k][0])] = ('', '')  
 else:  
 continue

3.将获取的信息转换成字典格式。

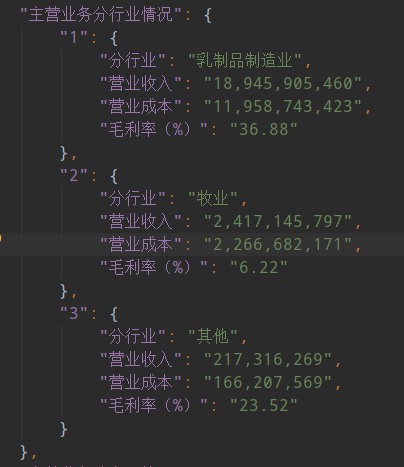
def write\_to\_dict\_3(result\_dir,map1,map2,map3):  
  
 if not os.path.exists(result\_dir):  
 os.makedirs(result\_dir)  
  
 dict = {}  
 dict['资产负债表'] = map1  
 dict['利润表'] = map2  
 dict['现金流量表'] = map3  
  
 return dict

4.写入json或者mongodb

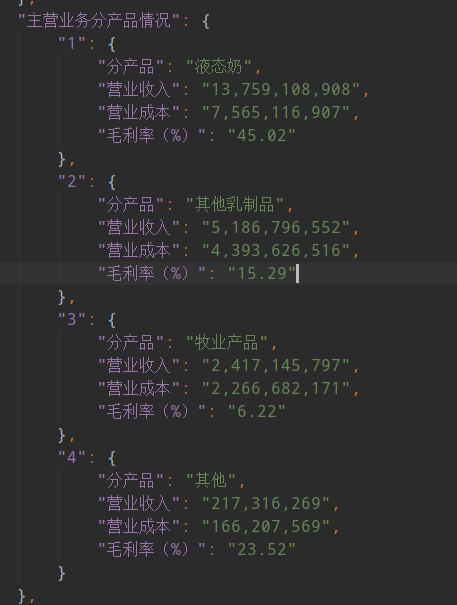
## 解析结果

### 一、主营业务（单位：元）

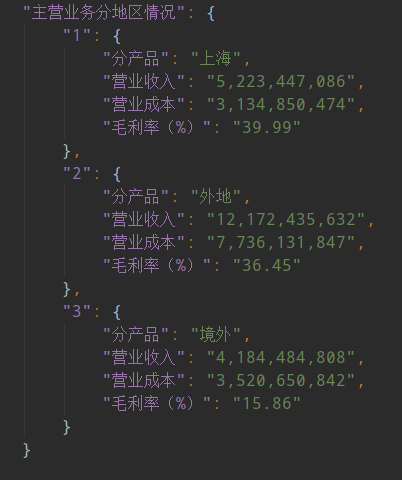
1. 主营业务分行业情况，不同行业包含营业收入、营业成本和毛利率字段



2. 主营业务分产品情况，不同产品包含营业收入、营业成本和毛利率字段

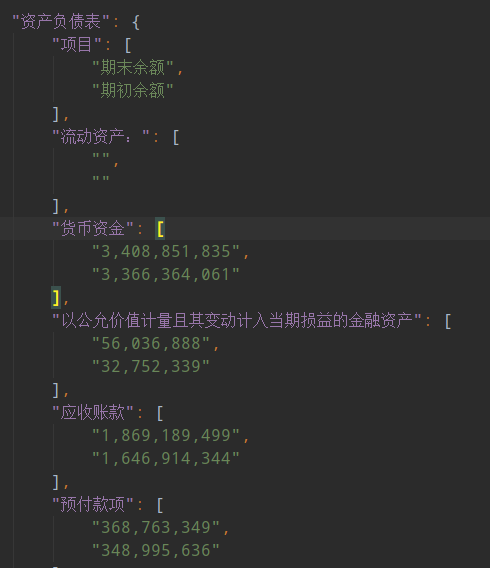


3. 主营业务分地区情况，不同地区包含营业收入、营业成本和毛利率字段

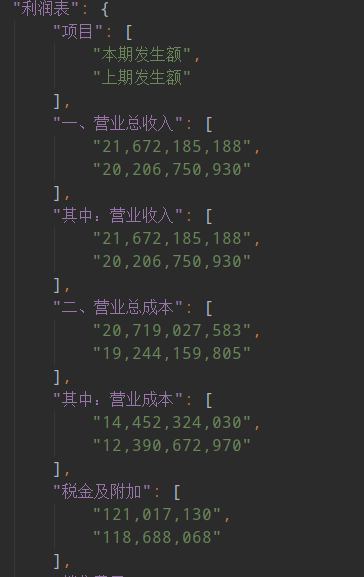


### 二、财务分析表（单位：元）

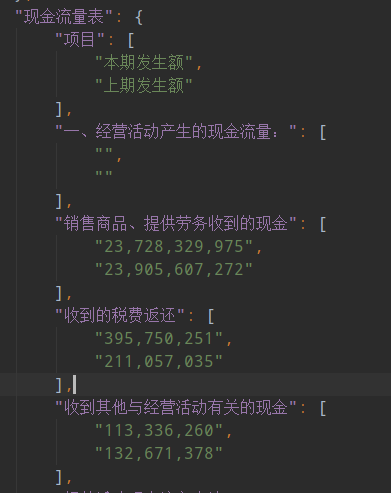
1.资产负债表，json第一个字段为项目、期末余额、期初余额，下面是资产负债表的信息。



2.利润表，json第一个字段为项目、本期发生额和上期发生额，下面是利润表的信息。

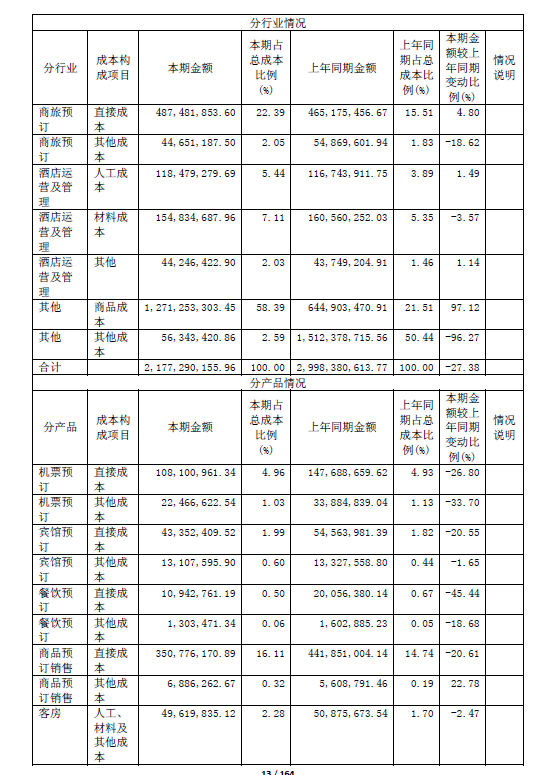


3.现金流量表，json第一个字段为项目、本期发生额和上期发生额，下面是现金流量表的信息。

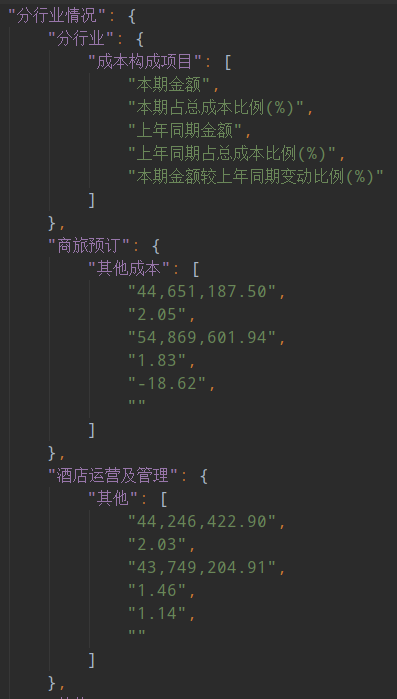


### 三、成本分析

PDF中成本分析表的格式如下：



1. 分行业情况



2. 分产品情况

