

User Defined Function In python For Excel

By:

Naya Zidan

Oudisho Qattyne

Agathy Homisy

Shahd Alallaf

4th Year

Class: 5

من الاحصائيات التي من الصعب تحقيقها في Excel هي :

1- الوسيط

2- المنوال

لذلك سنقوم باكمال ما تعلمناه في الجلسة الخامسة من تابع الجمع و تابع احضار اقل قيمة و نضيف تابعي الوسيط و المنوال

أولا نقوم بإضافة أسماء التوابع الى القائمة ليتحقق منها:

```
if function_name_token not in ['MEDIAN' , 'MODE']:  
    raise ValueError('Undefined Function !')
```

ثم نقوم ببناء التوابع الجديدة

1- الوسيط:

نبدأ ببناء تابع يأخذ متحول عبارة عن قائمة و يرتبها و يعيد الوسيط مع مراعاة اذا كان عدد الأرقام زوجي او فردي ففي حالة الزوجي يقوم بأخذ الرقمين الذين بالوسط و يحسب المتوسط الحسابي لهما اما الفردي فيأخذ الرقم الوحيد الذي في الوسط:

```
def calc_median(values):  
    index = len(values)/2  
    values.sort()  
    if(len(values)%2 == 0):  
        first_number = values[int(index)-1]  
        second_number = values[int(index)]  
        median = (first_number + second_number)/2  
    else:  
        median = values[math.floor(index)]  
  
    return median
```

ثم نقوم ببناء التابع الذي يأتي بقيم الخلايا بحسب اذا كانت مجال ام تم التصريح عن أسماء الخلايا المطلوبة بشكل صريح و يقوم باستدعاء تابع الحساب الذي بنيناه مسبقا و يعيد ما يعيده أي المتوسط:

```
119 def MEDIAN(tokens):
120     if tokens[0] != '(' or tokens[-1] != ')':
121         return -1
122
123     # empty range => return 0
124     if len (tokens) == 2 and tokens[0] == '(' and tokens[-1] == ')':
125         print('State 1')
126         return 0
127     # range => sum the values
128     # TODO : = SUM(A1;A3;A4:A5)
129     if len (tokens) > 2 and tokens[0] == '(' and tokens[-1] == ')':
130
131
132         if re.match(r'[A-Za-z]\d+',str(tokens[1])) and tokens[2] == ':' and re.match(r'[A-Za-z]\d+',str(tokens[-2])):
133
134             range_end_ref_num = re.findall(r'\d+',str(tokens[-2]))
135             range_start_ref_num = re.findall(r'\d+',str(tokens[1]))
136             range_ref_name = re.findall(r'[A-Za-z]',str(tokens[1]))
137
138
139             range_cells_names = []
140             for i in range(int(range_start_ref_num[0]),int(range_end_ref_num[0])+1):
141                 range_cells_names.append(str(range_ref_name[0])+str(i))
142
143
144             range_values = []
145             for cell in range_cells_names:
146                 temp = read_excel_cell(file_path,sheet_name,cell)
147                 range_values.append(temp)
148
149
150         else:
151
152             range_values = []
153             for tok in tokens:
154                 if re.match(r'[A-Za-z]\d+',str(tok)):
155                     temp = read_excel_cell(file_path,sheet_name,str(tok))
156                     range_values.append(temp)
157
158             median = calc_median(range_values)
159
160         return median
```

2- المنوال:

نبدأ ببناء تابع يأخذ متحول عبارة عن قائمة يمر على الأرقام فيها و يضيفها الى قاموس في حال لم تكن موجودة فيه من قبل و يزيد قيمتها اذا كانت موجودة

ثم يأتي بأكثر تكرار موجود و يقارن فيه كل قيم القاموس و يعود بقائمة فيا اكثر الأرقام تكرارا

و بعدها يتأكد من ان عدد القيم المتكررة لا يساوي عدد الأرقام المدخلة و الا فيعيد انه لايجد منوال لان الأرقام كلها تكون مكررة بنفس القيمة

اما اذا كانت الأرقام الأكثر تكرارا اقل من الأرقام المدخلة فيتم اعادتها على انها المنوال

```
76
77 def calc_modes(range_values):
78     dict = {}
79     for value in range_values:
80         if value in dict.keys():
81             dict[value] += 1
82         else:
83             dict[value] = 1
84
85     max_frequency = max(dict.values())
86     modes = [key for key, count in dict.items() if count == max_frequency]
87
88     if len(modes) == len(dict.values()):
89         return 'no modes'
90
91     else:
92         return modes
93
```

ثم نقوم ببناء التابع الذي يأتي بقيم الخلايا بحسب اذا كانت مجال ام تم التصريح عن أسماء الخلايا المطلوبة بشكل صريح و يقوم باستدعاء تابع الحساب الذي بنيناه مسبقا و يعيد ما يعيده أي المتوسط:

```
94
95 def MODE(tokens):
96     if tokens[0] != '(' or tokens[-1] != ')':
97         return -1
98
99     if len (tokens) == 2 and tokens[0] == '(' and tokens[-1] == ')':
100         print('State 1')
101         return 0
102
103     if len (tokens) > 2 and tokens[0] == '(' and tokens[-1] == ')':
104
105
106         if re.match(r'[A-Za-z]\d+',str(tokens[1])) and tokens[2] == ':' and re.match(r'[A-Za-z]\d+',str(tokens[-2])):
107             range_end_ref_num = re.findall(r'\d+',str(tokens[-2]))
108             range_start_ref_num = re.findall(r'\d+',str(tokens[1]))
109             range_ref_name = re.findall(r'[A-Za-z]',str(tokens[1]))
110
111
112             range_cells_names = []
113             for i in range(int(range_start_ref_num[0]),int(range_end_ref_num[0])+1):
114                 range_cells_names.append(str(range_ref_name[0])+str(i))
115
116
117             range_values = []
118             for cell in range_cells_names:
119                 temp = read_excel_cell(file_path,sheet_name,cell)
120                 range_values.append(temp)
121
122
123         else:
124
125             range_values = []
126             for tok in tokens:
127                 if re.match(r'[A-Za-z]\d+',str(tok)):
128                     temp = read_excel_cell(file_path,sheet_name,str(tok))
129                     range_values.append(temp)
130
131             modes = calc_modes(range_values)
132
133         return modes
134
```

التنفيذ:

```
179
180 exp = '=MEDIAN(A1;A2)'
181 t = lexical_analyzer(exp)
182 syntax_analyzer(t)
183
184 exp = '=MODE(A1;A2)'
185 t = lexical_analyzer(exp)
186 syntax_analyzer(t)
187
188 exp = '=MEDIAN(A1:A5)'
189 t = lexical_analyzer(exp)
190 syntax_analyzer(t)
191
192 exp = '=MODE(A1:A5)'
193 t = lexical_analyzer(exp)
194 syntax_analyzer(t)
```

OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL

```
PS G:\Year4\2\ProgrammingLanguages\Pract
ttyne_AgathyHomsy_ShahdAlallaf_Y4_C5\MED
MEDIAN result : 20.0
MODE result : no modes
MEDIAN result : 20
MODE result : [20]
PS G:\Year4\2\ProgrammingLanguages\Pract
```