

Excel 相关分享

Excel 相对于 SQL、python，适用场景更多，日常使用频率也较高，且更易上手。而对于还未接触、熟悉 SQL、python 或其他数据分析工具者而言，用其来预处理少量数据，无疑是较方便的。

其主要功能包括基础操作、图表与图形、公式与函数、数据透视表、宏与 VBA，五大部分。本篇主要介绍下函数部分，因其贯穿其他，且若想发挥 Excel 更多功能，函数是必须掌握的部分。本着从需求、解决问题出发，能更加促进学习动力、熟悉程度，仅先介绍使用频率较高的重点基础函数。同时，Github 上也备份了本笔记资源，[见链接](#)，仅供参考。

当然，若内置函数已无法满足需求，可通过宏与 VBA 自建函数。VBA 作为 Excel 内置的编程模块，语法与 SQL、python 虽有区别，但结构相对简单，有编程基础者，易融会贯通。作为 Excel 技能进阶版，宏与 VBA 可让数据处理的过程更简便。

一、公式与函数

如上所述，作为数据分析分享系列中的 Excel 篇，之所以会重点在讲函数，在于，其可用于任意其他模块，无论是透视表、条件格式、还是数据验证、宏、VBA，甚至是图表，均需夯实的函数基础。

而将 Excel 的基础操作、易入门的数据透视表等学会后，附之强大的函数，也会让你的 Excel 技能显著提升，同时能对数据类型、语法逻辑等有更深的认识。

1、注意事项

- Excel 函数中，仅支持半角双引号，即英文状态下双引号，即，如用单引号输入文本字符串会报错；

- 不等号用"<>"表示，而不是"!=";
- 使用函数时，均需切为英文字符状态，哪怕是输入数值、日期等数据类型;
- 注意对绝对引用、相对引用、混合使用的切换，快捷键为 F4;
- 函数参数间，用英文字符逗号相隔;
- 其内置函数，仅为唯一名称且不区分大小写，如 if 和 IF 一样;
- 学会看函数语法，比如可选参数在语法中会用一对中括号标注等;
- 学会使用嵌套，一个复杂的问题，也许用多层嵌套即可解决;
- 能解决问题的不只一种方法，如，公式过冗长，可读性差，考虑替换方法;

2、逻辑函数

逻辑函数可以对单个或多个表达式进行判断，最终返回 True 或 False 的逻辑值。且与一般语言类似，仅支持 True、False 两值判断，即，不同于 SQL，不会返回 Unknown 第三种判断。

(1) AND、OR、NOT、XOR 函数

函数	详解
AND(logical1,[logical2],...)	支持多个条件参数判断，当且仅当所有参数均为 True 时，返回 True，支持最多 255 个参数;
OR(logical1,[logical2],...)	支持多个条件参数判断，只要有一个参数为 True 时，就返回 True，支持最多 255 个参数;
NOT(logical)	仅有一个参数，返回与参数结果相反的逻辑值;
XOR(logical1,[logical2],...)	当且仅当参数结果中仅一个为 True 时，返回 True; (该函数使用几率不高，且易出错; 若判断不准时，可换其他替代方法);

(2) IF 函数

语法结构: IF(logical_test,[value_if_true],[value_if_false])

详解:

- 若条件 logical_test 为 True，返回指定值 value_if_true，其余为 False

的, 返回他值 value_if_false;

- 如上所说, 中括号为可选参数, 其他为必需参数;
- IF 函数常以嵌套形式, 即多层判断存在;
- 其最多支持 64 个 IF 嵌套, 若嵌套过长, 考虑替代方法;

案例一:

- 与 AND、OR 联用, 判断某产品是否合格:
- 当产品类型为"uxid", 且重量大于 20 时, 或者, 产品类型为"Tpef", 且重量大于 40 时, 产品合格:

- 如 C2 单元格即为,

=IF(OR(AND(A2="uxid",B2>20),AND(A2="Tpef",B2>40)),"合格","不合格")

C2				=IF(OR(AND(A2="uxid",B2>20),AND(A2="Tpef",B2>40)),"合格","不合格")								
	A	B	C	D	E	F	G	H	I	J	K	L
1	product	weight	quality									
2	uxid	64	合格									
3	uxid	13	不合格									
4	uxid	40	合格									
5	Tpef	35	不合格									
6	Tpef	74	合格									
7	Tpef	82	合格									
8												

案例二:

- IF 嵌套使用: 将分数分级: 将分数与各等级界线依次比较:
- 如 C2 单元格即为, =IF(B2>=80,"优秀",IF(B2>=70,"良好",IF(B2>=60,"及格",IF(B2<60,"不及格"))))

C2			=IF(B2>=80,"优秀",IF(B2>=70,"良好",IF(B2>=60,"及格","不及格")))									
	A	B	C	D	E	F	G	H	I	J	K	
1	姓名	得分	评级									
2	luo	26	不及格									
3	zhao	22	不及格									
4	li	78	良好									
5	zhang	8	不及格									
6	feng	41	不及格									
7	yagn	68	及格									
8	sun	56	不及格									
9	hua	67	及格									

(3) IFERROR 函数

语法结构: IFERROR(value,value_if_error)

详解:

- 屏蔽公式返回的错误值;
- 若 value 为错误值, 返回 value_if_error;

案例:

- 如除数为 0, 或空缺时, 不可除, 返回"不可除";
- 如下 C2 单元格, =IFERROR(A2/B2,"/")

C2						
	A	B	C	D	E	F
1	产品营收	数量	平均营收			
2	77336		/			
3	50788	879	57.779			
4	93869	53	1771.1			
5	58586	558	104.99			
6	78125	466	167.65			
7						

3、查找和引用函数

查找与引用函数即是在 指定区域查找并返回特定内容的函数。

(1) ROW、ROWS、COLUMN、COLUMNS、ADDRESS

该系列函数为基础查找与引用函数, 常作为其他函数的参数嵌套使用。

函数	详解
ROW([reference])	返回引用的行号; 如, ROW(A8), 返回8, 即A8单元格的行号; 若省略参数, ROW()即返回公式所在单元格的行号;
ROWS(array)	返回引用或数组的行数; 如, ROWS(A1:A7), 返回7, 即, A1:A7单元格区域的行数;
COLUMN([reference])	返回引用的列号; 如, COLUMN(B6), 返回2, 即B6单元格的列号; 若省略参数, COLUMN()即返回公式所在单元格的列号;
COLUMNS(array)	返回引用或数组的列数; 如, COLUMNS(A1:H7), 返回8, 即, A1:H7单元格区域的列数;
ADDRESS(row_num,column_num,[abs_num],[a1],[sheet_text])	根据指定行号、列号, 返回某个单元格地址; 其中前两个为必需参数, 代表行列号, abs_num代表引用类型, 若空缺, 代表行列均绝对引用, 如, ADDRESS(2,3), 返回"\$C\$2"; 而, ADDRESS(2,3,4), 返回"C2";

关于 ADDRESS 的案例如下：

D3						
	A	B	C	D	E	F
1						
2						
3				\$C\$2		
4				C2		
5						
6						

(2) VLOOKUP 函数

作为 Excel 中使用频率极高的查询函数之一，其可以返回查找值在指定区域对应的其他字段数据。

语法结构：VLOOKUP(lookup_value,table_array,col_index_num,[range_lookup])

详解：

- lookup_value 代表查找值，table_array 代表指定区域（为防止报错，建议将查找值所在的列，放置该区域第一列），col_index 代表返回值所在指定区域的第几列；
- 0 或 False 为精确匹配，True 或省略为模糊查询；

案例：查找学生成绩：如，F2=VLOOKUP(E2,\$A:\$B,2,0)

F2						
	A	B	C	D	E	F
1	姓名	得分			姓名	应得分
2	luo	26			sheng	73
3	zhao	22			nan	65
4	li	78			chen	60
5	zhang	8			wu	87
6	feng	41			jiagn	83

(3) HLOOKUP 函数

与 VLOOKUP 函数极相似，用法也类似。不过，VLOOKUP 中的，V 代表 Vertical 垂直的，H 代表 Horizontal 水平的。

语法结构：HLOOKUP(lookup_value,table_array,row_index_num,[range_lookup])

详解：参数用法与 VLOOKUP 类似，略；

案例：返回产品质量：如，E7=HLOOKUP(C7,1:3,3,0)

	A	B	C	D	E	F	G
1		产品线	A	C	F	H	K
2		产品号	1000938	1000950	1000962	1000974	1000986
3		产品质量	合格	合格	不合格	不合格	合格
4							
5							
6			产品线	产品号	检测所得质量		
7			k	1000986	合格		
8							
9							

(4) MATCH 函数

返回查询项在一行或列的相对位置，若有多符合条件者，仅返回第一次出现的位置。

语法结构：MATCH(lookup_value,lookup_array,[match_type])

详解：

- 第一个参数代表查找对象，接着是指定查找区域，仅支持单行/列，第三个参数代表匹配度；
- 其中，第三个参数中，0 为精确查找（若无完全匹配，返回#N/A），1 或省略（若无，返回小于等于查找值的区域中最大值所在位置），-1（若无，返回大于等于查找值的区域中最小值所在位置）；

案例一：

- 精确查找：如，C5==MATCH(B5,B2:F2,0)

- 在 B2 到 F2 区域中，查找 B5 单元格中值所在的位置：

C5							=MATCH(B5,B2:F2,0)
	A	B	C	D	E		
1							
2	指定区域	a	p	e	t	g	
3							
4		查找值	为0的精确查找				
5		t					4
6							

案例二：

- 精确查找：如，C5=MATCH(B5,C2:F2,0)
- 在 C2 到 F2 区域中，查找 B5 单元格中值所在的位置：

C5							=MATCH(B5,C2:F2,0)
	A	B	C	D	E		
1							
2	指定区域	a	p	e	t	g	
3							
4		查找值	为0的精确查找				
5		a					#N/A
6							

案例三：

- 1 或省略的模糊查找：如，C5=MATCH(B5,B2:F2,1)
- 在 B2 到 F2 区域中，查找 B5 单元格小于等于指定区域最大值所在位置：
- 如，下面指定区域无数值 6，即返回小于等于 6 的最大值 5，所在的序列位置，即 3；

C5							=MATCH(B5,B2:F2,1)
	A	B	C	D	E	F	
1							
2	指定区域	1	3	5	7	23	
3							
4		查找值	为0的精确查找				
5		6					3
6							

案例四：

- -1 的模糊查找：如，C5=MATCH(B5,B2:F2,1)
- 在 B2 到 F2 区域中，查找 B5 单元格中值大于等于其的指定区域中最小值所在的位置：
- 如，下面指定区域无数值 8，即返回大于等于 8 的最小值 9，其所在序列位置，即 2；

C5 =MATCH(B5,B2:F2,1)						
	A	B	C	D	E	F
1						
2	指定区域	11	9	6	5	3
3						
4		查找值	为0的精确查找			
5		8	2			

(5) INDEX 函数

该函数即根据指定区域，指定行列号返回引用。其经常与 MATCHA 函数结合使用，能够实现类似 VLOOKUP 功能，且更为灵活。

语法结构：

- 引用形式：INDEX(reference,row_num,[column_num],[area_num])
- 数组形式：INDEX(array,row_num,[column_num])

详解：以引用形式为例，第一参数为指定区域，第二三参数为行列号；若第一参数为多个非连续区域，需加上第四参数来指定引用的区域；

案例一：返回 B2:F6 区域中第三行第四列的值，即 E4 单元格的值：公式为：

=INDEX(B2:F6,3,4)

C9 =INDEX(B2:F6,3,4)						
	A	B	C	D	E	F
1						
2		89	31	34	36	20
3		12	89	43	21	33
4		48	57	31	62	93
5		97	53	42	0	18
6		58	68	94	32	51
7						
8						
9			62			
10						

案例二：

- 与 MATCH 函数嵌套联用：查找学生成绩；
- 公式，如，单元格 F2==INDEX(\$A\$1:\$C\$22,MATCH(E2,\$A\$1:\$A\$22,0),3)
- 该公式，即表示从指定区域 A1:C22 中查找，行号为
MATCH(E2,\$A\$1:\$A\$22,0)返回的值，即查找该学生在指定区域的行号，列号
为第 3 列，即评级所在列；
- 其联用，看似比 VLOOKUP 更冗长，但在逆向查询等场景更为灵活；

F2									
	A	B	C	D	E	F	G	H	
1	姓名	得分	评级		姓名	应返回评级			
2	luo	26	不及格		miao	不及格			
3	zhao	22	不及格		sheng	良好			
4	li	78	良好		ytan	及格			
5	zhang	8	不及格		wu	优秀			
6	feng	41	不及格		quand	优秀			
7	yagn	68	及格		lve	良好			
8	sun	56	不及格						

(6) OFFSET 函数

OFFSET 函数极其强大，在数据动态及多维引用等场景中都有用到，且在构建动态的引用中起到关键性作用。

以指定的引用为参照，通过给定偏移量得到新引用。返回的结果可以是单元格或区域，或者是指定的行列数。

语法结构：OFFSET(reference,rows,cols,[height],[width])

详解：

- 第一个参数为起始参照区域；
- 第二个为相对参照区的左上角单元格，向下（正数）或上偏移的行数；
- 第三个和第二个，为向右（正数）或左偏移的列数；

案例一：以单个单元格作参照系：

- 公式，单元格 F7==OFFSET(A4,4,2);
- 即代表从参照单元格 A4 出发，向下偏移 4 行，至 A8 单元格，再向右偏移 2 列，至 C8 单元格；

	A	B	C	D	E	F
1	姓名	班级	得分	评级		
2	luo	10075	26	不及格		
3	zhao	10075	22	不及格		
4	li	10075	78	良好		
5	zhang	10076	8	不及格		
6	feng	10076	41	不及格		
7	yagn	10076	68	及格		
8	sun	10078	56	不及格		
9	hua	10078	67	及格		
10	hong	10078	55	不及格		

案例二：以多个单元格作为的参照区域：

- 公式，单元格 F8==OFFSET(\$A\$4:\$C\$8,3,2,1,1)
- 即代表从参照区域 A4:C8 的左上角单元格 A4 出发，向下偏移 3 行至 A7，再向右偏移 2 列至 C7；
- 仅取了 1 行 1 列，即 C7 单元格；

	A	B	C	D	E	F
1	姓名	班级	得分	评级		
2	luo	10075	26	不及格		
3	zhao	10075	22	不及格		
4	li	10075	78	良好		
5	zhang	10076	8	不及格		
6	feng	10076	41	不及格		
7	yagn	10076	68	及格		
8	sun	10078	56	不及格		
9	hua	10078	67	及格		

案例三：参照区域和返回区域均为多个单元格：

- 公式，选中 F5:G6 单元格区域=OFFSET(\$A\$4:\$C\$8,3,2,2,2)
- 和上个案例类似的是，从 A4:C8 的 A4 偏移至相关区域；
- 因返回的 2 行 2 列引用，需提前选中符合的行列数，并 ctrl+shift+enter，来返回引用，此时公式会加上{}大括号；

	A	B	C	D	E	F	G
1	姓名	班级	得分	评级			
2	luo	10075	26	不及格			
3	zhao	10075	22	不及格			
4	li	10075	78	良好			
5	zhang	10076	8	不及格		68	及格
6	feng	10076	41	不及格		56	不及格
7	yagn	10076	68	及格			
8	sun	10078	56	不及格			
9	hua	10078	67	及格			

(7) INDIRECT 函数

该函数可以根据第一参数的文本字符串生产具体的单元格引用。其常用于对静态名称的引用、工作表行列信息等场景。

语法结构: INDIRECT(ref test,[a1])

详解:

- 第一个参数为表示单元格地址的文本，可以是直接的单元格，如 A1，也可以是字符串，如"A1";
- 注意这两者的区别;

案例一：第一参数为单元格

- 公式：单元格 A4=INDIRECT(A1)
- 先指向 A1 单元格，再将其单元格存储的文本字符串"C1"进行实际引用，返回

C1 单元格存储的内容;

A4				fx		=INDIRECT(A1)	
	A	B	C	D	E		
1	C1		test				
2							
3							
4	test						
5							

案例二：第一参数为文本

- 公式：单元格 A4=INDIRECT("A1")
- 因为第一参数为文本字符串"A1"，而非单元格 A1 本身，因此直接返回 A1

单元格存储的内容，即字符串"C1";

A4				fx		=INDIRECT("A1")	
	A	B	C	D	E	F	
1	C1		test				
2							
3							
4	C1						
5							

(8) CHOOSE 函数

该函数可以根据指定的数字序号，返回与其对应的数据值、区域引用或嵌套结果等。

语法结构：CHOOSE(index_num,value1,[value2],...)

详解：

- 第一参数为数值型数据，或单元格为数值型类型；
- 若第一参数为小数，则直接取整数部分；

案例：

- 如下，公式：B2=CHOOSE(2.6,"A","M","B","D","Y");

- 直接返回数组中的第 2 个值;

Figure 10-10: The CHOOSE function. The CHOOSE function takes a number and a list of values and returns the value at the specified position. In this example, the formula in cell B2 is =CHOOSE(2, "A", "M", "B", "D", "Y"). The value 2 is the position of the value to return, and the list of values is "A", "M", "B", "D", "Y". The result of the formula is "M".

4、统计函数

统计与求和函数在工作中被广泛应用，需重点掌握。

(1) 基础统计- -SUM、 AVERAGE、 COUNT、 MAX 等

函数	详解
SUM(number1,[number2],...)	返回所有参数总和;
AVERAGE(number1,[number2],...)	返回所有参数均值;
COUNT(value1,[value2],...)	返回所有数值型参数的个数;
COUNTA(value1,[value2],...)	返回所有不为空的单元格个数;
COUNTBLANK(value1,[value2],...)	返回区域中空格数;
MAX(number1,[number2],...)	返回所有参数最大值;
MIN(number1,[number2],...)	返回所有参数最小值;

这里着重对 COUNT、COUNTA、COUNTBLANK 作比较案例:

案例：公式分别为：

- 单元格 D2=COUNT(A2:A18)
- 单元格 D3=COUNTA(A2:A18)
- 单元格 D4=COUNTBLANK(A2:A18)

D2 fx =COUNT(A2:A18)				
	A	B	C	D
1	数据		函数	返回值
2	luo		COUNT	7
3	zhao		COUNTA	14
4	li		COUNTBLANK	3
5	zhang			
6	feng			
7	yagn			
8	8			
9	41			
10	68			
11	56			
12	67			
13				
14				
15				
16	73			
17	52			
18	ytan			

(2) MODE.SNGL 与 MODE.MULT 函数

这些均为众数函数，较早版本的直接为 MODE 函数。

语法结构：这两个函数结构类似：

- MODE.SNGL(number1,[number2],...)
- MODE.mult(number1,[number2],...)

详解：

- MODE.SNGL 仅返回 1 个最多值，若多个频率最高者，取第一个出现数字；
- MODE.MULT 返回出现频率最高数组，即存在多个频率最高者，返回数组；

案例：公式如下：

- 单元格 D2=MODE.SNGL(A2:A17)
- 单元格 D4:D9, {=MODE.MULT(A2:A17)}
- 需注意的是，当函数返回的是数组形式的是，需选中多个单元格作为数组返回区域，如这里选择了 D4:D9，作为可能返回的多个频率最高者，若个数不足，

即返回#N/A;

D4 {=MODE.MULT(A2:A17)}				
	A	B	C	D
1	数据		D列公式为	返回值
2		3	"=MODE.SNGL(A2:A17)"	2
3		2		
4		2	"{=MODE.MULT(A2:A17)}"	2
5		1	"{=MODE.MULT(A2:A17)}"	5
6		3	"{=MODE.MULT(A2:A17)}"	#N/A
7		2	"{=MODE.MULT(A2:A17)}"	#N/A
8		5	"{=MODE.MULT(A2:A17)}"	#N/A
9		5	"{=MODE.MULT(A2:A17)}"	#N/A
10		5		
11		2		
12		3		
13		4		
14		6		
15		1		
16		5		
17		0		

(3) MEDIAN 函数

返回数组中的中位数。

语法结构：MEDIAN(number1,[[number2],...])

详解：

- 若数组参数个数为奇数个，即直接返回中间的数字；
- 若参数个数为偶数个，即返回中间两个数的均值；

案例：公式如下：

- 单元格 D2=MEDIAN(A2:A6)
- 单元格 D3=MEDIAN(A2:A7)

	A	B	C	D
1	数据		公式	结果
2	1		=MEDIAN(A2:A6)	4
3	4		=MEDIAN(A2:A7)	5.25
4	7			
5	6.5			
6	2			
7	9			

(4) LARGE 与 SMALL 函数

其分别返回数据集中第 K 个最大值/最小值。

语法结构：

LARGE(array,k)

SMALL(array,k)

案例：公式如下：即返回第二个最大值/最小值

- 单元格 D2>=LARGE(A2:A7,2)
- 单元格 D3=SMALL(A2:A7,2)

	A	B	C	D
1	数据		公式	结果
2	1		=LARGE(A2:A7,2)	7
3	4		=SMALL(A2:A7,2)	2
4	7			
5	6.5			
6	2			
7	9			

(5) RANK、RANK.AVG、RANK.EQ 函数

这三个均为排序函数，返回数据不同的排名方式，但均为并列排名占用名次式排序。

语法结构：

- RANK.EQ(number,ref,[order])
- RANK.AVG(number,ref,[order])
- RANK(number,ref,[order])

详解：

- 该三个函数结构类似；
- 第一个参数均为需排名的数据；
- 第二个参数为所处的列表；

- 第三个参数可略，若为 0 或略，降序排列，否则升序；
- RANK.EQ 按最佳排名排序；
- RANK.AVG 返回平均排位；
- RANK 与 RANK.EQ 类似，其为 EXCEL 早期版本的兼容模式；

案例：公式如下：

- 单元格 E2=RANK.EQ(A6,A2:A8)
- 单元格 E3=RANK.AVG(A6,A2:A8)
- 单元格 E4=RANK(A6,A2:A8)
- 单元格 E5=RANK.EQ(A6,A2:A8,1)
- 单元格 E6=RANK.AVG(A6,A2:A8,1)
- 单元格 E7=RANK(A6,A2:A8,1)
- 单元格 E8=RANK.EQ(A4,A2:A8)
- 单元格 E9=RANK.AVG(A4,A2:A8)
- 以上公式进一步表明，并列排名时，RANK.EQ 和 RANK 都会取靠前排名；
- 且三种排名方式，并列排名者均会相应地占用下列名次；

	A	B	C	D	E
1	数据		公式	详解	结果
2	1		=RANK.EQ(A6,A2:A8)	6按照RANK.EQ降序排列	3
3	2		=RANK.AVG(A6,A2:A8)	6按照RANK.AVG排列	3.5
4	3		=RANK(A6,A2:A8)	6按照RANK降序排列	3
5	6		=RANK.EQ(A6,A2:A8,1)	6按照RANK.EQ升序排列	4
6	6		=RANK.AVG(A6,A2:A8,1)	6按照RANK.AVG升序排列	4.5
7	7		=RANK(A6,A2:A8,1)	6按照RANK升序排列	4
8	9		=RANK.EQ(A4,A2:A8)	3按照RANK.EQ降序排列	5
9			=RANK.AVG(A4,A2:A8)	3按照RANK.AVG降序排列	5
10					

(6) COUNTIFS 函数

相对 COUNTIF 函数，更为强大，这里直接介绍 COUNTIFS 函数。

语法结构： COUNTIFS(criteria_range1,criteria1,[criteria_range2,criteria2]...)

详解： 即，计算满足多个条件的所有单元格个数；

案例：

- 单元格 G2=COUNTIFS(B:B,F2)
- 单元格 H2=COUNTIFS(B:B,F2,C:C,">=60")

H2								=COUNTIFS(B:B,F2,C:C,">=60")
	A	B	C	D	E	F	G	H
1	姓名	班级	得分	评级		班级	学生人数	及格人数
2	luo	10075	26	不及格		10075	4	3
3	zhao	10078	22	不及格		10076	1	1
4	li	10075	78	良好		10077	5	2
5	zhang	10078	8	不及格		10078	5	2
6	feng	10079	41	不及格		10079	2	1
7	yagn	10077	68	及格		10080	4	4
8	sun	10077	56	不及格				
9	hua	10078	67	及格				
10	hong	10078	55	不及格				
11	nan	10080	65	及格				

(7) SUMIFS 函数

语法结构：

SUMIFS(sum_range,criteria_range1,criteria1,[criteria_range2,criteria2]...)

详解： 按多个条件对指定区域求和；

案例：

- 单元格 G2=SUMIFS(D:D,A:A,F2)
- 单元格 H2=SUMIFS(D:D,A:A,F2,B:B,"A")

H2									=SUMIFS(D:D,A:A,F2,B:B,"A")
	A	B	C	D	E	F	G	H	
1	水果	厂家	品类	营收		水果	总营收	A厂家营收	
2	葡萄	A	巨峰	27760		葡萄	186178	60448	
3	葡萄	B	玫瑰香	47710		苹果	177826	54789	
4	葡萄	B	夏黑	47235		香蕉	179318	88983	
5	葡萄	B	蓝宝石	30785		橘子	139611	93378	
6	葡萄	A	茉莉香	32688					
7	苹果	A	红富士	41287					
8	苹果	C	花牛	11745					
9	苹果	D	嘎啦	49888					
10	苹果	A	国光	13502					
11	苹果	B	黑卡	25465					
12	苹果	B	红星	35939					
13	香蕉	C	皇帝蕉	48808					
14	香蕉	A	金手指	49296					
15	香蕉	A	小米蕉	39687					
16	香蕉	C	红皮蕉	41527					

(8) AVERAGEIFS 函数

语法结构：

AVERAGEIFS(average_range,criteria_range1,criteria1,[criteria_range2,criteria2]..)

详解：同 SUMIFS 类似，返回符合多个条件的均值；

案例：

- 单元格 H2=AVERAGEIFS(D:D,A:A,G2)
- 单元格 I2=AVERAGEIFS(D:D,A:A,G2,B:B,"A")

I2									=AVERAGEIFS(D:D,A:A,G2,B:B,"A")
	A	B	C	D	E	F	G	H	I
1	水果	厂家	品类	营收			水果	各品类平均营收	A厂家各品类营
2	葡萄	A	巨峰	27760			葡萄	37235.6	30224
3	葡萄	B	玫瑰香	47710			苹果	29637.66667	27394.5
4	葡萄	B	夏黑	47235			香蕉	44829.5	44491.5
5	葡萄	B	蓝宝石	30785			橘子	27922.2	31126
6	葡萄	A	茉莉香	32688					
7	苹果	A	红富士	41287					
8	苹果	C	花牛	11745					
9	苹果	D	嘎啦	49888					
10	苹果	A	国光	13502					
11	苹果	B	黑卡	25465					
12	苹果	B	红星	35939					
13	香蕉	C	皇帝蕉	48808					
14	香蕉	A	金手指	49296					
15	香蕉	A	小米蕉	39687					
16	香蕉	C	红皮蕉	41527					
17	橘子	B	砂糖橘	23009					

(9) SUMPRODUCT 函数

语法结构：SUMPRODUCT(array1,[array2],[array3],...)

详解：对给定的数组，将其对应的数据相乘后，返回总和；

案例：

单元格 D2=SUMPRODUCT(A2:A7,B2:B7)

D2					=SUMPRODUCT(A2:A7,B2:B7)
	A	B	C	D	E
1	包裹数	每个包裹重量/kg		包裹总重量	
2	76	5		2451	
3	45	10			
4	64	10			
5	18	6			
6	37	6			
7	93	7			
8					

5、数学和三角函数

(1) INT、TRUNC、ROUND、CEILINIG、ROUNDUP 等数值取舍函数

函数	详解	案例	返回结果
INT(number)	向下取整函数；无法指定精度，仅保留整数；	INT(5.28);	5;
		INT(5.98);	5;
		INT(-5.28);	-6;
		INT(-5.98);	-6;
TRUNC(number,[num_digits])	截断取整；直接对指定数值截位，无关正负符号；	TRUNC(5.28,1);	5.2;
		TRUNC(-5.28,1);	-5.2;
ROUND(number,num_digits)	四舍五入函数，精度为必需参数；	ROUND(5.98,0);	6;
		ROUND(-5.98,0);	-6;
ROUNDUP(number,num_digits)	向上取数，远离0的方向，即负数时，会比原数小；	ROUNDUP(5.28,0);	6;
		ROUNDUP(-5.28,0);	-6;
ROUNDDOWN(number,num_digits)	向下取数，靠近零的方向；	ROUNDDOWN(5.98,0);	5;
		ROUNDDOWN(-5.98,0);	-5;
CEILING(number,significance)	将数据向上取数，取为最接近基数的整数倍；一般基数用1即可；另有CEILING.MATH可用于对负数的取舍；	CEILING(5.28,1);	6;
		CEILING(5.98,1);	6;
		CEILING(-5.28,1);	-5;
		CEILING(-5.98,1);	-5;
FLOOR(number,significance)	将数据向下取数，取为最接近基数的整数倍；一般基数用1即可；另有FLOOR.MATH可用于对负数的取舍；	FLOOR(5.28,1);	5;
		FLOOR(5.98,1);	5;
		FLOOR(-5.28,1);	-6;
		FLOOR(-5.98,1);	-6;

对 CEILING 函数、CEILING.MATH 函数、Floor 函数、

FLOOR.MATH 函数，可以在入门的基础上，尝试其他参数的探索。但如同之前所

说，函数不在于多，而在于精。抓住基础，将其频繁运用。同样地，工作或项目中遇

到需高阶函数的场景时，再去熟悉即可。

(2) MOD 函数

其为求余函数。

语法结构: MOD(number,divisor)

详解:

- 返回两数相除后的余数;
- 第一参数为被除数, 第二参数为除数;

案例：公式：单元格 A2=MOD(23,9)

A2				=MOD(23,9)			
	A	B	C				
1	公式	返回结果					
2	=MOD(23,9)	5					
3							

(3) RAND、RANDBETWEEN 函数

均为产生随机数的函数。

语法结构:

- RAND()
- RANDBETWEEN(bottom,top)

详解:

- RAND 函数无需参数, 返回 0 到 1 的随机小数, 含 0;

- RANDBETWEEN 返回指定区间的随机整数，含边界值；

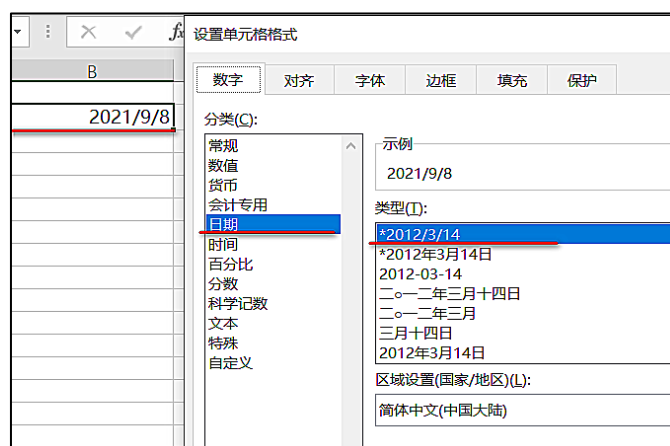
案例：公式如下：

- =RAND()
- =RANDBETWEEN(9.8,29.28)
- 可见，RANDBETWEEN 边界值均为小数，但只会随机产生整数；

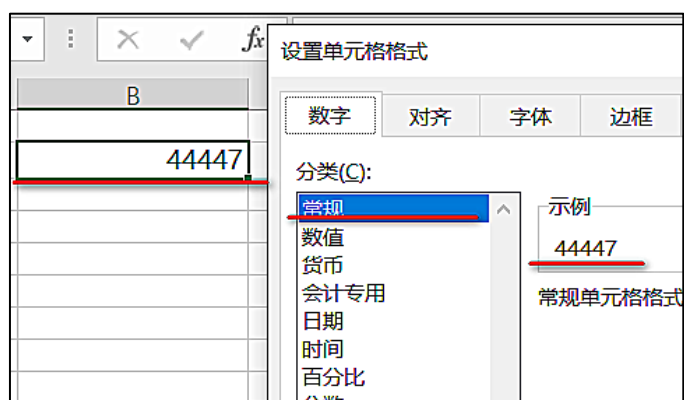
公式	返回结果
=RAND()	0.4967015653
=RANDBETWEEN(9.8,29.28)	20

6、日期和时间函数

Excel 中，日期和时间本质都是以数值型存储。如下，案例中，若单元格 B2 的内容，以日期格式显示，则显示为熟悉的时间形式：



但若以常规形式显示，则为数值型：



其每个日期对应唯一整数序列值，日期区间在 1900-01-01 到 9999-12-31，即，1900-01-01 也对应着序列值 1。为避免报错，Excel 中输入日期时间时，应用英文字符输入完整的数据，如年份输入 4 位。年月份中间可用"/"，或，"-"连接，而时间用":"连接，中间以空格隔开。

(1) TODAY, NOW, YEAR, MONTH, DAY, HOUR, MINUTE,

SECOND 函数

函数	详解	案例	返回结果
TODAY()	无参数，返回系统当前日期；	/	直接返回当前日期，如 "2024-11-21"；
NOW()	返回系统当前时刻；	/	直接返回当前时间，如 "2024-11-21 15:32"；
YEAR(serial_number)	返回某日期年份；	YEAR("2019-08-09")	2019；
MONTH(serial_number)	返回某日期月份；	MONTH("2019-08-09")	8；
DAY(serial_number)	返回某日期具体的一天；	DAY("2019-08-09")	9；
HOUR(serial_number)	返回时间数据中的小时；	HOUR("2019/08/09 15:43:38")	15；
MINUTE(serial_number)	返回时间数据的分钟；	MINUTE("2019/08/09 15:43:38")	43；
SECOND(serial_number)	返回时间中秒；	SECOND("2019/08/09 15:43:38")	38；

(2) DATE 函数

语法结构: DATE(year,month,day)

详解：根据指定年月日返回日期序列；

案例：公式如下：单元格 B2=DATE(2019,8,9)

B2			X	✓	f _x	=DATE(2019,8,9)
	A	B	C	D	E	F
1						
2		2019/8/9				
3						

(3) TIME 函数

语法结构: TIME(hour,minute,second)

详解：根据指定小时、分钟、秒数返回时间：

案例：公式：单元格 B2=TIME(23,13,34)

B2				
	A	B	C	D
1				
2		23:13:34		
3				

当然，以上时间也可通过单元格格式，重新设置，如下，为 AM/PM 形式：

B2				
	A	B		
1				
2		11:13:34 PM		
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

设置单元格格式

数字 对齐 字体 边框 填充 保护

分类(C):

- 常规
- 数值
- 货币
- 会计专用
- 日期
- 时间
- 百分比
- 分数
- 科学记数
- 文本
- 特殊
- 自定义

示例

11:13:34 PM

类型(T):

- h:mm:ss AM/PM
- yyyy"年"m"月"d"日"
- m/d/yy
- d-mmm-yy
- d-mmm
- mmm-yy
- h:mm AM/PM
- h:mm:ss AM/PM
- h:mm
- h:mm:ss
- h"时"mm"分"
- h"时"mm"分"ss"秒"

(4) EDATE 函数

语法结构：EDATE(start_date,months)

详解：

- 返回某个日期相隔指定月份之前之后的日期；
- 第二参数为月份数，负数返回之前日期，正数返回之后日期，小数截断取整；

案例：公式如下：

- 单元格 B2=EDATE("2019-08-09",2)
- 单元格 B2=EDATE("2019-08-09",-2)
- 单元格 B2=EDATE("2019-08-09",2.3)

- 单元格 B2=EDATE("2019-08-09",2.7)

	A	B
1	公式	返回结果
2	EDATE("2019-08-09",2)	2019/10/9
3	EDATE("2019-08-09",-2)	2019/6/9
4	EDATE("2019-08-09",2.3)	2019/10/9
5	EDATE("2019-08-09",2.7)	2019/10/9
6		

(5) EOMONTH 函数

语法结构: EOMONTH(start_date,months)

详解:

- 返回相隔指定月数之前或之后的月底日期;
- 参数的用法, 同 EDATE 函数;

案例: 公式如下:

- 单元格 B2=EOMONTH("2019-08-09",2)
- 单元格 B2=EOMONTH("2019-08-09",-2)
- 单元格 B2=EOMONTH("2019-08-09",2.3)
- 单元格 B2=EOMONTH("2019-08-09",2.7)

公式	返回结果
EOMONTH("2019-08-09",2)	2019/10/31
EOMONTH("2019-08-09",-2)	2019/6/30
EOMONTH("2019-08-09",2.3)	2019/10/31
EOMONTH("2019-08-09",2.7)	2019/10/31

(6) DATEDIF 函数

语法结构: DATEDIF(start_date,end_date,unit)

详解:

- 返回两日期之间的间隔;
- 第一二参数为开始结束日期, 结束日期必须不小于开始日期;

- unit 为返回类型，从年数差数到天数差数等；

案例：公式如下：

- 单元格 E2=DATEDIF(A2,B2,"Y")
- 单元格 E3=DATEDIF(A3,B3,"M")
- 单元格 E4=DATEDIF(A4,B4,"D")
- 单元格 E5=DATEDIF(A5,B5,"MD")
- 单元格 E6=DATEDIF(A6,B6,"YM")
- 单元格 E7=DATEDIF(A7,B7,"YD")

E2 ✖ ✔ fx =DATEDIF(A2,B2,"Y")					
	A	B	C	D	E
1	开始日期	结束日期	unit参数	返回类型	返回结果
2	2018/7/4	2019/8/9	Y	时间段的整年数	1
3	2018/7/4	2019/8/9	M	整月数	13
4	2018/7/4	2019/8/9	D	整天数	401
5	2018/7/4	2019/8/9	MD	忽略月年的天数差	5
6	2018/7/4	2019/8/9	YM	忽略日年的月数差	1
7	2018/7/4	2019/8/9	YD	忽略年的天数差	36

(7) WEEKDAY 函数

语法结构：WEEKDAY(serial_number,[return_type])

详解：

- 返回对应某日期所处的星期值；
- 第二个参数可略，也可指定返回类型；
- 若无特定，返回为 1(星期日)，7(星期六)；

案例：公式：单元格 B2=WEEKDAY(A2)

B2 ✖ ✔ fx =WEEKDAY(A2)					
	A	B	C	D	E
1	日期	返回结果			
2	2024/12/20	6			
3					

(8) WORKDAY 函数

语法结构: WORKDAY(start_date,days,[holidays])

详解:

- 返回距离起始日期之前或之后工作日数的日期;
- 第二参数为不含周末及节假日的天数;
- 第三参数 holidays 可略, 包含需从日历中排除的节假日日期;

案例: 公式: 单元格 C3=WORKDAY(A2,B2)

C2					=WORKDAY(A2,B2)
	A	B	C	D	
1	开始日期	相隔的工作日天数	返回结果		
2	2024/12/20	7	2024/12/31		
3					

(9) NETWORKDAYS

语法结构: NETWORKDAYS(start_date,end_date,[holidays])

详解:

- 返回两日期间的工作日天数;
- 同样, 第三个参数为可选的需排除的节假日天数;

案例: 公式: 单元格 C2=NETWORKDAYS(A2,B2)

C2					=NETWORKDAYS(A2,B2)
	A	B	C	D	
1	开始日期	相隔的工作日天数	返回结果		
2	2024/11/10	2024/11/27	13		
3					

(10) DAYS 函数

语法结构: DAYS(end_date,start_date)

函数	详解	案例	返回结果
ISNUMBER	判断是否为数值类型；是为True，否为False；	ISNUMBER("34");	False;
		ISNUMBER(34);	True;
ISTEXT	判断是否为文本类型；是为True，否为False；	ISTEXT("34");	True;
		ISTEXT(34);	False;
TYPE	直接返回数值类型；1为数字、2为文本、4为逻辑值、16为错误值、64为数组常量；	TYPE(34);	1;
		TYPE("34");	2;
		TYPE(TRUE);	4;

(2) CONCAT、TEXTJOIN 函数

其与&符号，都是用来连接多个区域或字符串。

函数/符号	语法结构	案例	返回结果
CONCAT	CONCAT(text1,[text2],...)	CONCAT("W","O","R","D")	WORD;
TEXTJOIN	TEXTJOIN(delimiter,ignore_empty,text1,[text2],...)	TEXTJOIN(";",0,"W","O","R","D")	W;O;R;D;
&	text1&text2&text3...	"W"&"O"&"R"&"D"	WORD;

(3) EXACT 函数

语法结构：EXACT(text1,text2)

详解：比较字符串是否完全相同，且可区分字母大小写；

案例：公式：单元格 C2=EXACT(A2,B2)

C2				=EXACT(A2,B2)
	A	B	C	D
1	字符串一	字符串二	返回结果	
2	uekpTRQ	uekpTrQ	FALSE	
3	uekpTRQ	uekpTRQ	TRUE	

(4) LOWER、UPPER、PROPER 函数

语法结构：

- LOWER(text)
- UPPER(text)
- PROPER(text)

详解：

- 将所有字母转为小写；
- 将所有字母转为大写；
- 首字母转为大写，其余为小写；

案例：公式如下：

- 单元格 B2==LOWER(A2)
- 单元格 C2=UPPER(A2)
- 单元格 D2=PROPER(A2)

B2 ✕ ✓ fx =LOWER(A2)				
	A	B	C	D
1	字符串	LOWER结果	UPPER结果	PROPER结果
2	uekpTRQ	uekp tr q	UEKPTRQ	Uekptrq
3	BekpTRq	bekptrq	BEKPTRQ	Bekptrq
4				

(5) CODE、CHAR 函数

语法结构：

- CODE(text)
- CHAR(text)

详解：

CODE、CHAR 函数用于处理字符与编码的转换；

CODE 返回文本中第一个字符在 Excel 中的数字编码；

CHAR 则根据 Excel 中的代码数字，返回字符；

这里的字母，分大小写；

案例：公式如下：

A	B
公式	返回结果
=CODE("A")	65
=CODE("a")	97
=CODE("Amn")	65
=CHAR(65)	A
=CHAR(97)	a

(6) LEN、LENB 函数

语法结构：

- LEN(text)
- LENB(text)

详解：

- 均可以计算字符长度；
- LEN 函数对任意单个字符均按一个长度计算；
- LENB 函数对双字节字符，如汉字，按两个长度计算；

案例：公式如下：

A	B
公式	返回结果
=LEN("data分析")	6
=LENB("data分析")	8

(7) LEFT、RIGHT、MID 函数

语法结构：

- LEFT(text,[num_chars])
- RIGHT(text,[num_chars])
- MID(text,start_num,num_chars)

详解：

- 以上均为字符串提取函数；
- LEFT 函数自左往右提取字符串；
- RIGHT 函数自右往左提取字符串；
- MID 函数可从单元格任意位置提取字符串；
- MID 中第一参数为要提取的字符串，第二参数为指定文本中起始的位置，第三参数为需返回的字符个数；

案例：公式如下

字符串	公式	返回结果
egnoqkjwe	=LEFT(A2,3)	egn
egnoqkjwe	=RIGHT(A3,3)	jwe
egnoqkjwe	=MID(A4,5,3)	qkj

(8) FIND、SEARCH 函数

语法结构：

- FIND(find_text,within_text,[start_num])
- SEARCH(find_text,within_text,[start_num])

详解：

- 均为查找指定字符串在单元格的位置，并返回其起始位置值；

- FIND 函数区分大小写，SEARCH 不区分；
- FIND 不支持通配符搜索，SEARCH 支持，更灵活；

案例：公式如下：

字符串	公式	返回结果
egnoQKJYwe	=FIND("k jy",A2)	#VALUE!
egnoQKJYwe	=FIND("KJY",A3)	6
egnoQKJYwe	=SEARCH("k jy",A4)	6
egnoQKJYwe	=SEARCH("KJY",A5)	6

(9) SUBSTITUTE、REPLACE 函数

语法结构：

- SUBSTITUTE(text,old_text,new_text,[instance_num])
- REPLACE(old_text,start_num,num_chars,new_text)

详解：

- 均为替换函数；
- SUBSTITUTE 将指定字符替换为新字符，若原字符串中无指定字符，返回原字符串；
- REPLACE 函数对字符串指定位置直接进行新字符替换；
- SUBSTITUTE 区分大小写；
- REPLACE 不会替换成等量原长度的字符串，只是替换为指定的字符；

案例：公式如下：

字符串	公式	返回结果
egnoQKJYwe	=SUBSTITUTE(A2,"qk jyw","****")	egnoQKJYwe
egnoQKJYwe	=SUBSTITUTE(A3,"QKJYw","****")	egno****e
egnoQKJYwe	=REPLACE(A4,5,4,"*")	egno*we
egnoQKJYwe	=REPLACE(A5,5,4,"****")	egno****we

(10) REPT 函数

语法结构：REPT(text,number_times)

详解：将指定文本重复多次组成字符串；

案例：公式如下：

A	B
公式	结果
=REPT("#",3)	###

(11) CLEAN、TRIM 函数

语法结构：

- CLEAN(text)
- TRIM(text)

详解：

- CLEAN 函数将非打印字符删除；
- TRIM 函数将左右两端空格清除；

案例：公式如下：

A	B	C
字符串	公式	结果
gjeogn ejgn	=TRIM(A2)	gjeogn ejgn

(12) TEXT 函数

该函数可对字符串格式化，是使用频率极高的函数之一。

语法结构：TEXT(value,format_text)

详解：

- 第一参数为要转换为指定格式文本的数值，也可以是文本型数字；

- 第二参数用来指定格式代码；

案例一：用来设置不同数字格式

实现类似 Excel 中自定义格式的功能：

几种常用占位符如下：

- 0，数字占位符，数量不足需补齐，如"000"返回的整数不能小于 3 位数字；
- 若为小数，需改写为"0.00"等相应格式的小数位数，否则四舍五入整数；
- #，数字占位符，数量不足无需补齐；
- @，文本占位符，连续表示重复文本；

源数据	公式	返回结果
24.83479	TEXT(A2,"0")	25
24.83479	TEXT(A3,"0.0")	24.8
24.83479	TEXT(A4,"0000")	0025
24.83479	TEXT(A5,"####")	25
0.2	TEXT(A6,"0.00%")	20.00%
2019/8/9	TEXT(A6,"yyyy年m月d日")	2019年8月9日
2019/8/9	TEXT(A7,"yyyy-mm-dd")	2019-08-09

案例二：实现类似 IF 函数的条件格式化文本

其第二参数，可分为 4 个条件区段，用半角分号隔开，默认情况定义为：

[>0];[<0];[=0];[文本]

公式如下：

A	B	C
增长率	公式	结果
20.00%	TEXT(A2,"升;降;持平;空缺")	升
-18.00%	TEXT(A3,"升;降;持平;空缺")	降
0.00%	TEXT(A4,"升;降;持平;空缺")	持平
null	TEXT(A5,"升;降;持平;空缺")	空缺
35.00%	TEXT(A6,"升;降;持平;空缺")	升

案例三：将默认分区段改为自定义

自定义可改为：[条件 1];[条件 2];[不满足的其他部分]

且自定义条件数量最多为 3 个，包含不满足的部分；

如下，若无不满足的条件部分，或者超过条件个数，均会发生错误：

姓名	得分	公式	结果
luo	26	=TEXT(C2,"[>=90]优秀;[>=80]良好;[>=60]合格;不合格")	#VALUE!
li	78	=TEXT(C3,"[>=90]优秀;[>=80]良好;[>=70]中等")	#VALUE!
zhao	65	=TEXT(C4,"[>=90]优秀;[>=60]及格;不及格")	及格

所以，若想把成绩分级，可采用嵌套函数：其中第一个 TEXT 函数里，不满足部分，

用 0 表示，返回原数据，即上述中的用 0 表示的整数；

若数据为小数，则增加小数位：

姓名	得分	公式	结果
luo	26	=TEXT(TEXT(C2,"[>=90]优秀;[>=80]良好;0"),"[>=70]中等;[>=60]及格;不及格")	不及格
li	78	=TEXT(TEXT(C3,"[>=90]优秀;[>=80]良好;0"),"[>=70]中等;[>=60]及格;不及格")	中等
zhao	65	=TEXT(TEXT(C4,"[>=90]优秀;[>=80]良好;0"),"[>=70]中等;[>=60]及格;不及格")	及格
zhang	86	=TEXT(TEXT(C5,"[>=90]优秀;[>=80]良好;0"),"[>=70]中等;[>=60]及格;不及格")	良好
feng	97	=TEXT(TEXT(C6,"[>=90]优秀;[>=80]良好;0"),"[>=70]中等;[>=60]及格;不及格")	优秀
yagn	54	=TEXT(TEXT(C7,"[>=90]优秀;[>=80]良好;0"),"[>=70]中等;[>=60]及格;不及格")	不及格

8、数组公式

首先明确下概念，Excel 中数组是指按一行、一列或多行多列排列的一组数据元素的集合。

与 Numpy 类似，一行 N 列的一维横向数组大小为 1*N，一列 N 行的一维纵向数组大小为 N*1，M 行 N 列的二维数组大小为 M*N。

对于初学者，最需要注意的就是，数组公式不同于普通公式，以按

<Ctrl+Shift+Enter>组合键完成编辑。也即，若直接 Enter 编辑完，则无法实现该公式功能。

案例：

- 直接计算水果的总销售额：公式如下：
- 单元格 E2 输入=SUM(b2:b7*c2:c7)
- 接着<Ctrl+Shift+Enter>完成编辑，在 fx 函数里，则呈现为：

{=SUM(b2:b7*c2:c7)}

E2						
	A	B	C	D	E	F
1	水果种类	单价	数量		公式	返回结果
2	苹果	5.99	5		=SUM(B2:B7*C2:C7)	422.35
3	香蕉	4.98	8			
4	梨	7.9	6			
5	草莓	20.8	10			
6	橙子	9.89	7			
7	砂糖橘	3.99	7			
8						

其他更复杂的数组公式，可遇到相应项目时，再行探索。

二、宏与 VBA

一个简单的定义是，Excel 中被录制下的一串操作叫作宏，用来录制宏的工具就是宏录制器。

可通过开发工具中的录制宏来实现，这里需要注意绝对引用及相对引用的切换，

1、宏

案例：每行数据间插入空行：

原数据版式为：

	A	B	C	D	E	F	G
	工号	姓名	基本工资	加班工资	应发工资	扣除	实发金额
	A001	luo	8086	953	9039	600	8439
	A002	zhao	14932	1399	16331	766	15565
	A003	li	14701	1404	16105	351	15754
	A004	dew3rf	15343	2346	17689	457	17232
	A005	feng	12343	978	13321	513	12808
	A006	yagn	10458	932	11390	683	10707
	A007	sun	12279	909	13188	465	12723
	A008	hua	6924	592	7516	443	7073

单击 A1-->开发工具-->录制宏-->命名宏与快捷键-->使用相对引用-->在 A2 与 A3 之间插入两空行-->复制 A1 行粘贴至现 A4 行-->单击 A4-->停止录制;

执行该宏的快捷键后，数据版式修改为：

	A	B	C	D	E	F	G
1	工号	姓名	基本工资	加班工资	应发工资	扣除	实发金额
2	A001	luo	8086	953	9039	600	8439
3							
4	工号	姓名	基本工资	加班工资	应发工资	扣除	实发金额
5	A002	zhao	14932	1399	16331	766	15565
6							
7	工号	姓名	基本工资	加班工资	应发工资	扣除	实发金额
8	A003	li	14701	1404	16105	351	15754
9							
10	工号	姓名	基本工资	加班工资	应发工资	扣除	实发金额
11	A004	dew3rf	15343	2346	17689	457	17232
12							
13	工号	姓名	基本工资	加班工资	应发工资	扣除	实发金额
14	A005	feng	12343	978	13321	513	12808
15							
16	工号	姓名	基本工资	加班工资	应发工资	扣除	实发金额
17	A006	yagn	10458	932	11390	683	10707
18							

这就是宏的方便之处。尤其对于需重复执行的操作，创建宏能简便不少流程。

2、VBA

VBA，全称，Visual Basic For Applications，是可供扩展 Excel 的功能的编程语言。录制宏只是 VBA 的一部分。

若拥有其他编程语言，入门 VBA 会更容易，因其循环、逻辑判断等语句存在一定共通性。本章仅介绍几个简单的语法，其他语法及操作可自行探索。

与 python 类似，VBA 中可用 4 个空格代表缩进，来表示代码属于某个代码块。

(1) 声明变量

与 python、SQL 等声明方式均不同，VBA 中可用 Dim 语句来自定义变量。

语法结构：Dim 变量名 As 数据类型

当然，这不是唯一定义变量的方式，其还支持以下定义方式：

- 定义为私有变量: Private 变量名 As 数据类型
- 公有变量: Public 变量名 As 数据类型
- 静态变量: Static 变量名 As 数据类型

若不确定作用域, 用 Dim 即可。

(2) IF 条件判断

语法结构:

```
IF 条件表达式 Then
    返回 True 时需执行的操作和计算
Elseif 条件表达式 Then
    返回 True 时需执行的操作和计算
...
Else
    返回 True 时需执行的操作和计算
End If
```

(3) Case 条件判断

与 SQL 类似的, VBA 中也有 Case 条件判断, 但存在些微差异。

语法结构:

```
Select Case 表达式
    Case Is 条件表达式 1
        返回 True 时执行的操作
    Case Is 条件表达式 2
        返回 True 时执行的操作
    ...
    Case Else
        返回 True 时执行的操作
End Select
```

(4) For...Next 循环

与 python 类似，VBA 支持 For 循环，但也存在些许差别。

- **语法结构一：**若省略 step，默认步长为 1

```
For 循环变量 = 初值 to 终值 step 步长值  
    循环体（要执行的循环操作）  
Next 循环变量名
```

- **语法结构二：**

```
For Each 变量 In 集合名或数组名  
    循环体（要执行的循环操作）  
Next [元素变量]
```

(5) Do While 循环语句

为简化管理，暂时只介绍开头判断式。

语法结构：

```
Do [While 循环条件]  
    循环体（要执行的循环操作）  
Loop
```

(6) Do Until 循环语句

同样的，为简化管理，暂时只介绍开头判断式。

语法结构：

```
Do [Until 循环条件]  
    循环体（要执行的循环操作）  
Loop
```


(7) Sub 过程

VBA 的过程就是完成某个任务所需的代码组合。宏录制器中的宏就是 Sub 过程，可以保存该任务的代码。

语法结构：

```
Sub 过程名称()  
    待执行代码  
End Sub
```

一般将其插入在一个模块中，且通过 Private 或 Public 来声明该过程的作用域。

公共过程 Public 声明，语法如下：

```
Public Sub 过程名称([参数列表])  
    待执行代码  
End Sub
```

(8) Function 函数

和 Sub 过程类似，自定义函数如下：

语法结构：

```
Public Function 函数名([参数列表])  
    待执行代码  
End Function
```

3、VBA 综合案例

结合以上 VBA 中常用的几个语法结构，作简要案例展示。

(1) 案例一：IF 条件判断与 For 循环

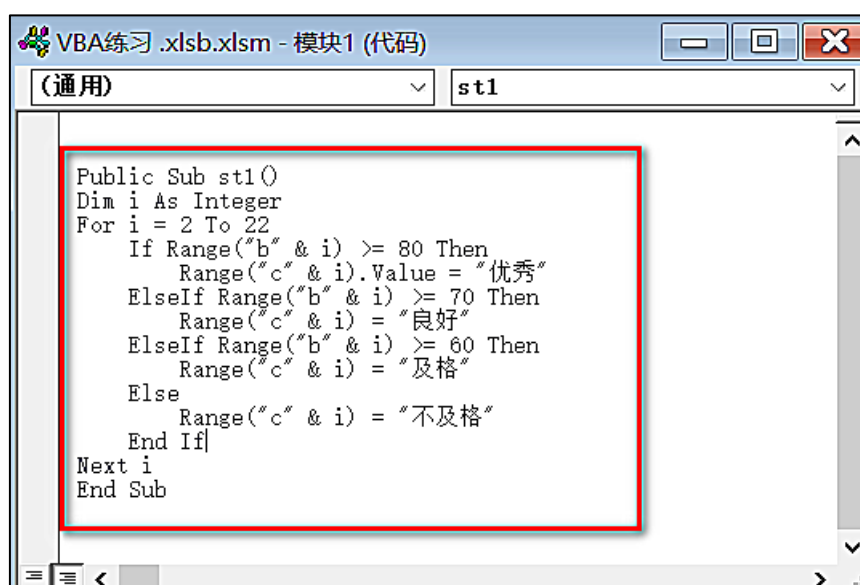
原表：

	A	B
1	姓名	得分
2	luo	26
3	zhao	22
4	li	78
5	zhang	8
6	feng	41
7	yagn	68
8	sun	56
9	hua	67
10	hong	55
11	nan	65

要求：将学生成绩表，按"不低于 80 分为优秀，不低于 70 分为良好，不低于 60 分为及格，其余为不及格"评级：

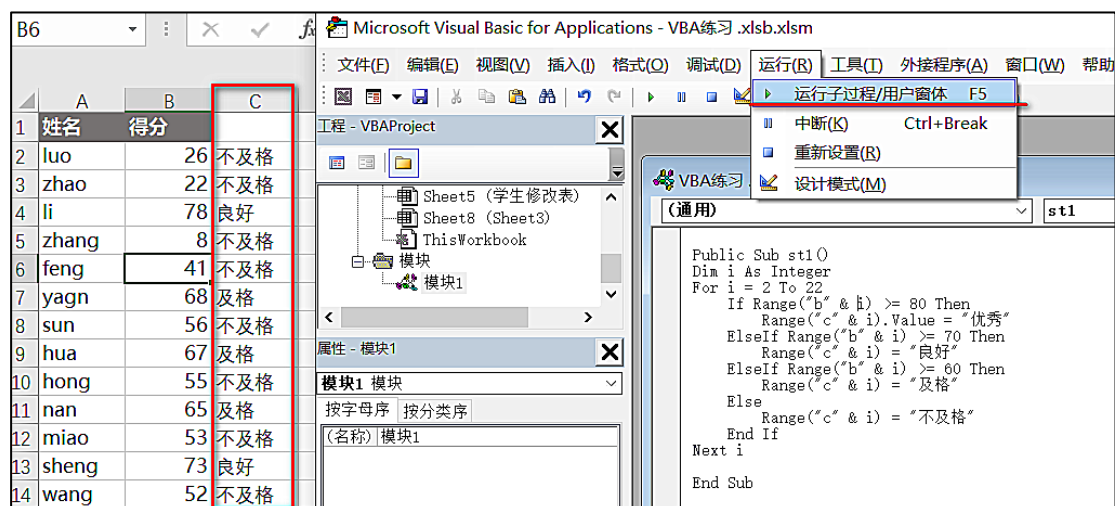
开发工具-->Visual Basic-->模块-->插入过程-->子程序-->命名-->输入相应代码，

如下：



```
Public Sub st1()  
Dim i As Integer  
For i = 2 To 22  
    If Range("b" & i) >= 80 Then  
        Range("c" & i).Value = "优秀"  
    ElseIf Range("b" & i) >= 70 Then  
        Range("c" & i) = "良好"  
    ElseIf Range("b" & i) >= 60 Then  
        Range("c" & i) = "及格"  
    Else  
        Range("c" & i) = "不及格"  
    End If  
Next i  
End Sub
```

运行该子过程，C 列转换如下：

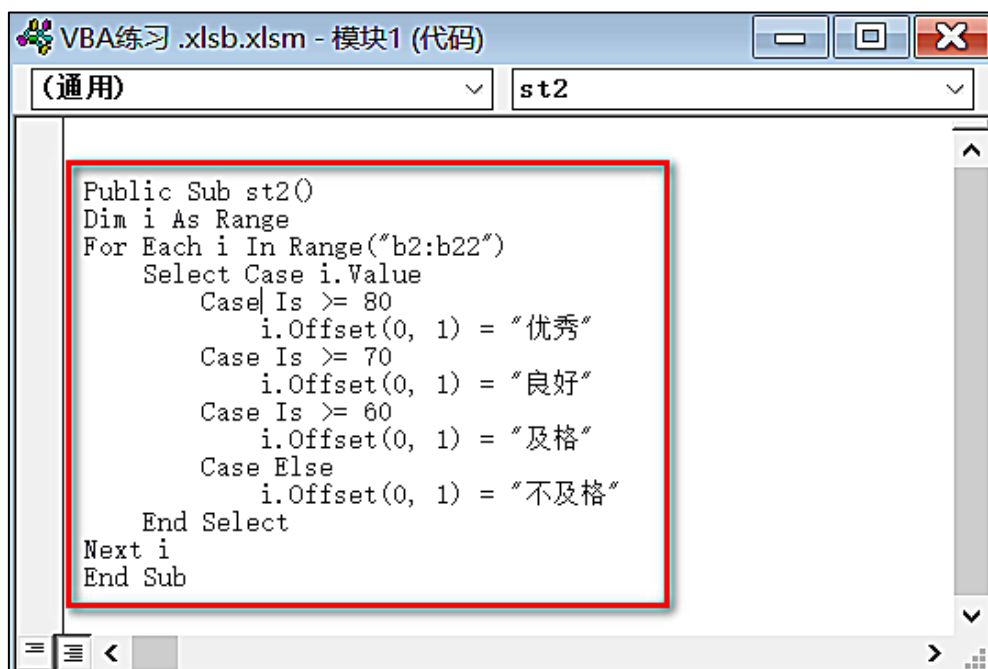


以上，通过 VBA 一键返回整列的结果。若原数据量较大，需求更为复杂，VBA 将比 Excel 其他操作，流程更为简便，效率更高。

(2) 案例二：Case 条件判断与 For 循环

原表仍为学生成绩表，但采用另外的方法，将其评级：

同样地在模块里插入子程序后，输入以下代码：

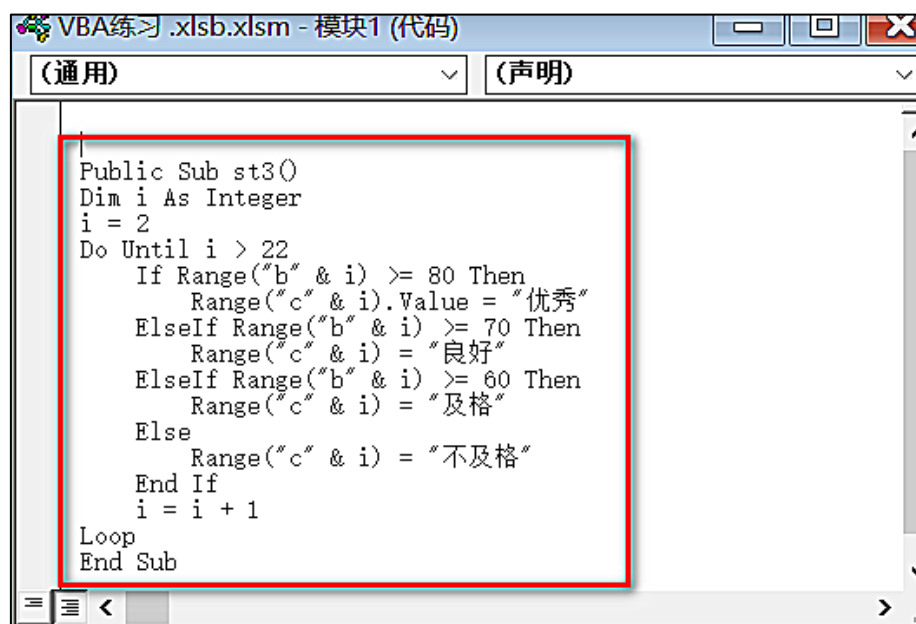


运行子过程，即可。

(3) 案例三：IF 条件判断与 DO UNTIL 循环

同样地，用另一种循环模式实现此需求：

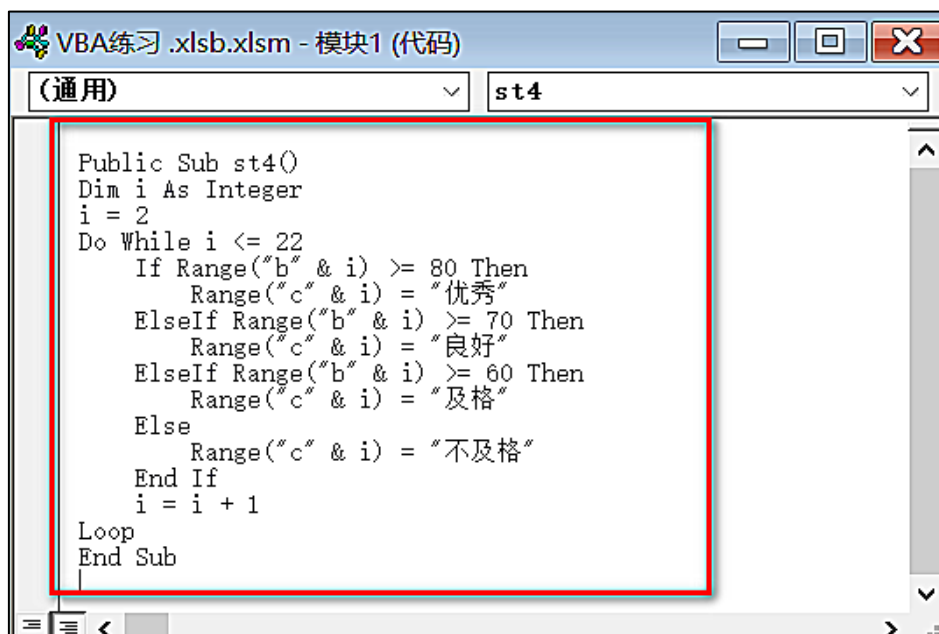
代码如下：



```
Public Sub st3()  
Dim i As Integer  
i = 2  
Do Until i > 22  
    If Range("b" & i) >= 80 Then  
        Range("c" & i).Value = "优秀"  
    ElseIf Range("b" & i) >= 70 Then  
        Range("c" & i) = "良好"  
    ElseIf Range("b" & i) >= 60 Then  
        Range("c" & i) = "及格"  
    Else  
        Range("c" & i) = "不及格"  
    End If  
    i = i + 1  
Loop  
End Sub
```

(4) 案例四：IF 条件判断与 DO WHILE 循环

原表同上，代码改至如下：



```
Public Sub st4()  
Dim i As Integer  
i = 2  
Do While i <= 22  
    If Range("b" & i) >= 80 Then  
        Range("c" & i) = "优秀"  
    ElseIf Range("b" & i) >= 70 Then  
        Range("c" & i) = "良好"  
    ElseIf Range("b" & i) >= 60 Then  
        Range("c" & i) = "及格"  
    Else  
        Range("c" & i) = "不及格"  
    End If  
    i = i + 1  
Loop  
End Sub
```

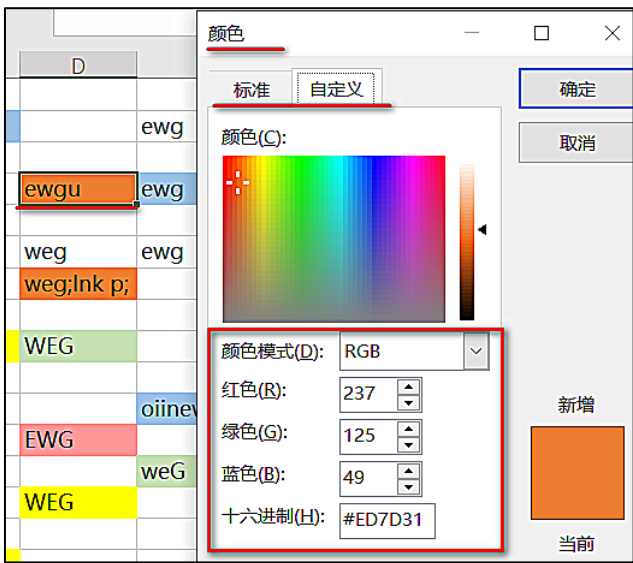
(5) 案例五：创建自定义函数

原表如下：单元格区域内，被填充为不同颜色

	A	B	C	D	E
1	dgn				
2		dga	weg		ewg
3					
4	weg		g	ewgu	ewg
5		GJOEWJG			
6				weg	ewg
7	WEG			weg;lnk p;	
8		adv	wg		
9			ewg	WEG	
0	wegf				
1	gw		Weg		oiinewqg
2		WGE		EWG	
3					weG
4		we	g	WEG	
5					
6	weg		weg		
7					ewg
8		ewg		weg	

要求：计算该区域内各填充颜色的单元格个数：

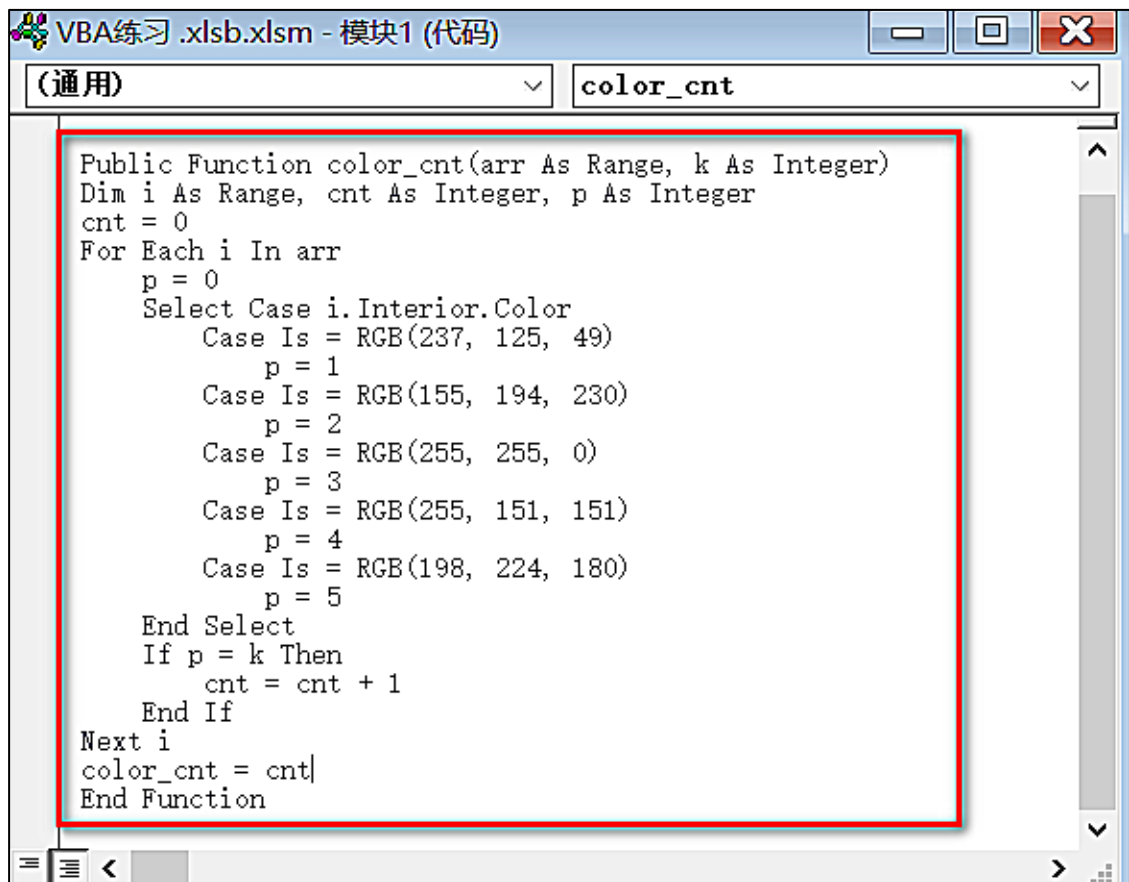
为解决该需求，需提前查询各颜色 RGB 参数，查看单元格填充颜色，即可获得：



建立需求表：

Color	各颜色RGB值	参数	颜色个数
orange	rgb(237,125,49)	1	
blue	rgb(155,194,230)	2	
yellow	rgb(255,255,0)	3	
pink	rgb(255,151,151)	4	
green	rgb(198,224,180)	5	

VBA 模块中，插入过程-->函数-->编写相关代码：



```

Public Function color_cnt(arr As Range, k As Integer)
    Dim i As Range, cnt As Integer, p As Integer
    cnt = 0
    For Each i In arr
        p = 0
        Select Case i.Interior.Color
            Case Is = RGB(237, 125, 49)
                p = 1
            Case Is = RGB(155, 194, 230)
                p = 2
            Case Is = RGB(255, 255, 0)
                p = 3
            Case Is = RGB(255, 151, 151)
                p = 4
            Case Is = RGB(198, 224, 180)
                p = 5
        End Select
        If p = k Then
            cnt = cnt + 1
        End If
    Next i
    color_cnt = cnt
End Function
    
```

该自定义函数代码逻辑为：

命名函数 color_cnt，并要求使用时必含两个参数（需统计的单元格区域，用来参与比较的辅助参数）；

若单元格 RGB 值等于指定颜色值，赋值相应参数；

计算区域内，等于该参数的单元格个数，即符合颜色的个数，如下：

J2	=color_cnt(\$A\$1:\$E\$31,I2)									
	A	B	C	D	E	F	G	H	I	J
1	dgn						Color	各颜色RGB值	参数	颜色个数
2		dga	weg		ewg		orange	rgb(237,125,49)	1	6
3							blue	rgb(155,194,230)	2	8
4	weg		g	ewgu	ewg		yellow	rgb(255,255,0)	3	7
5		GJOEWJG					pink	rgb(255,151,151)	4	6
6				weg	ewg		green	rgb(198,224,180)	5	7
7	WEG			weg;lnk p;						
8		adva	wa							

(6) 案例六：创建自定义函数

仍是统计上表中各颜色的单元格个数：

若对需求表无格式上的特别要求，可将代码简化：

建立需求表如下：

G	H
Color	颜色个数
orange	
blue	
yellow	
pink	
green	

代码直接修改为：

```
Public Function color_cnt(arr As Range, k As Range)
    Dim i As Range, cnt As Integer
    cnt = 0
    For Each i In arr
        If i.Interior.Color = k.Interior.Color Then
            cnt = cnt + 1
        End If
    Next i
    color_cnt = cnt
End Function
```

即，不再附加辅助参数，直接把将单元格颜色填充为指定样式，并应用其自身 RGB

作为可变动参数，返回结果；

公式如下：

H2				=color_cnt(\$A\$1:\$E\$31,G2)				
	A	B	C	D	E	F	G	H
1	dgn						Color	颜色个数
2		dga	weg		ewg		orange	6
3							blue	8
4	weg		g	ewgu	ewg		yellow	7
5		GJOEWJG					pink	6
6				weg	ewg		green	7
7	WEG			weg;lnk p;				
8		adv	wg					
9			ewg	WEG				
10	wegf							
11	gw		Weg		oiinewqg			
12		WGE		EWG				
13					weG			

(7) 案例七：复杂公式与 VBA 应用效率对比




若碰到原数据如下，需提取中间的数值部分：

A	B
源数据	提取数字
苹果Apple消费15元不等	
电费-->105元多	
水费_30元不到?	
咖啡,。消费25元等	
看电影 50元还不到	

- 公式解决如下：

=-LOOKUP(0,-MID(A2,MATCH(0,0/MID(A2,ROW(\$1:\$99),1),0),ROW(\$1:\$15)))

同时按<Ctrl+Shift+Enter>键，将其复制到 B6 单元格：

B2		:	  		{=-LOOKUP(0,-MID(A2,MATCH(0,0/MID(A2,ROW(\$1:\$99),1),0),ROW(\$1:\$15)))}									
	A	B	C	D	E	F	G	H	I	J				
1	源数据	提取数字												
2	苹果Apple消费15元不等	15												
3	电费-->105元多	105												
4	水费_30元不到?	30												
5	咖啡,。消费25元等	25												
6	看电影 50元还不到	50												

以上公式，虽也能解决该需求，但较为冗长，且易出错。

- VBA 解决如下：

代码如下，创建自定义函数：

```
Public Function extract_int(arr As Range)
    Dim cnt As Integer, i As Integer, t As String, j As String
    cnt = Len(arr)
    j = ""
    For i = 1 To cnt
        t = Mid(arr, i, 1)
        If Asc(t) >= 48 And Asc(t) <= 57 Then
            j = j & t
        End If
    Next i
    extract_int = j * 1
End Function
```


该函数说明：

- 先把每个字符串按 MID 函数分割成一个一个字符；
- 再通过 ASC 函数（类似 Excel 里的 CODE 函数）将其转为数字编码，由于 0~9 数字，在 Excel 中的编码为 48~57，符合条件者，即存至辅助变量，连接；
- 最后将文本变量转为数值形式即可；

通过上述两种方法的对比，一定程度上也体现出 VBA 的灵活性。尤其是面对非标准数据、繁琐需求时，其便利程度远超于公式等操作。

需注意的是，Excel 公式里的通用函数，并非完全适用于 VBA，如上述的 CODE 函数，在 VBA 里需换成 ASC。这些特殊情况，待到具体问题解决即可，无需牢记。

三、学习资源

相对 SQL、python、C 语言等，Excel 是相对能较快入门的工具，且由于绝大多数工作均会接触到，并不缺乏项目经验。但也限制于工作场景，平时无需过多掌握如 VBA 等模块，要达到精通也较难。针对需熟悉、提高的模块，专项学习、练习即可。

这里，仅为相关资源的推荐，但不限于此：

1、书籍（选择亟需掌握的模块即可）：

- 《让 Excel 飞》（较薄、很适合的一本入门书，几乎涵盖 Excel 所有模块，看完能对 Excel 有个全方位的了解）
- 《Excel 函数与公式大全》（ExcelHome 编著）
- 《Excel 数据透视表应用大全》（ExcelHome 编著）
- 《别怕，Excel VBA 其实很简单》

2、论坛：

- 对于 Excel 中碰到的难题，着力推荐 [ExcelHome](#) 论坛。该论坛从创立之初到现在，网罗了各种难题，一般通过关键词搜索，就能从类似发帖中获得解决方案。当然，也可以自己发帖寻求帮助。
- 若 Excel 技能已达到一定程度，分享也是促进深入学习的一种途径，如去输出，在论坛里帮助他人解决难题。
- 除此以外，善用 AI，辅助查询，对于解决难题也是一个途径。

3、教学视频：

这里不再做特别推荐，b 站、红薯、快抖等等有诸多讲学视频，都可尝试跟学，比方说秋叶 Excel、Excel 从零到一等均会经常分享短视频教学，无论是对于初入门还是中阶者，都可以温故而知新。

四、其他思考

- Excel 作为接触最频繁的数据整理、分析工具，若能在入门的基础上，掌握些许其他模块，如透视表、函数、query、VBA 等的内容，对于标准、少量的数据清洗，不失为一个利器；
- 善用论坛。相对 python 等出现的难题需要各大网站搜索解决方案来说，专门用来解决 Excel 难题的论坛，可谓是精准速达。基本上，项目上可能遇到的问题，在该网站均能获得解答；
- 和其他工具学习一样，最重要就是动手练习，代码写多了，也就熟能生巧了；