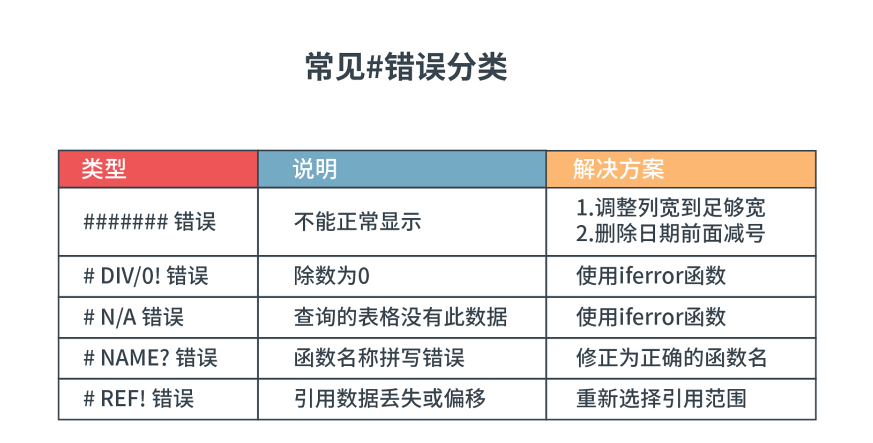
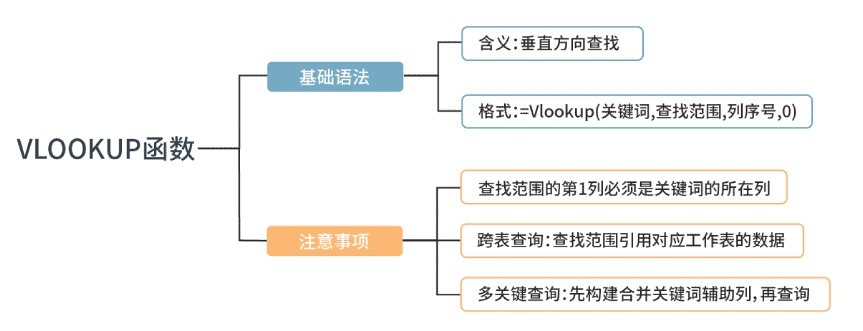
**计算机等级考试（二级）函数**

1. **vlookup函数（精确匹配）、int函数**



**1、ROUNDUP：向上舍入的任意实数。**

|  |  |  |
| --- | --- | --- |
| **1** | **公式** | **说明(结果)** |
| **2** | **=ROUNDUP(3.2,0)** | **将 3.2 向上舍入，小数位为 0 (4)** |
| **3** | **=ROUNDUP(76.9,0)** | **将 76.9 向上舍入，小数位为 0 (77)** |
| **4** | **=ROUNDUP(3.14159, 3)** | **将 3.14159 向上舍入，保留三位小数 (3.142)** |
| **5** | **=ROUNDUP(-3.14159, 1)** | **将 -3.14159 向上舍入，保留一位小数 (-3.2)** |
| **6** | **=ROUNDUP(31415.92654, -2)** | **将 31415.92654 向上舍入到小数点左侧两位 (31500)** |

**1、Rounddown函数是指靠近零值，向下(绝对值减小的方向)舍入数字。**

1. **=ROUNDDOWN(3.2, 0)将 3.2 向下舍入，小数位为 0 (3)**
2. **=ROUNDDOWN(76.9,0)将 76.9 向下舍入，小数位为 0 (76)**
3. **=ROUNDDOWN(3.14159, 3)将 3.14159 向下舍入，保留三位小数 (3.141)**
4. **=ROUNDDOWN(-3.14159, 1)将 -3.14159 向下舍入，保留一位小数 (-3.1)**
5. **=ROUNDDOWN(31415.92654, -2)将 31415.92654 向下舍入到小数点左侧两位 (31400)**

**2、sum函数 、rank函数 、countif函数**

**LARGE**[**函数**](https://baike.so.com/doc/5368923.html)**：返回数据集中的第K个最大值。**

**格式：LARGE(array,k)**

**Array： 为需要找到第 k 个最大值的数据区域。**

**k ：为返回的数据数据区域里的位置(从大到小)。**

**3、vlookup函数（精确匹配）、if函数 、or条件**

**Weekday函数：返回代表一周中第几天的数值，是一个1到7(或0到6)之间的整数。**

**语法格式 ：WEEKDAY(date,type)，**

**Date：日期**

**Type： 数字1 或省略：则1 至7 代表星期天到星期六；**

**数字2 ：则1 至7 代表星期一到星期天；**

**数字3：则0至6代表星期一到星期日。**

如：

=WEEKDAY("2001/8/28"，2) 返回2(星期二)

=WEEKDAY("2003/02/23"，3) 返回6(星期日)

**4、DATEDIF函数：**

**（1）功能：**是Excel隐藏函数，其在帮助和插入公式里面没有。 返回两个日期之间的年\月\日间隔数。常使用DATEDIF函数计算两日期之差。

**（2）语法：**DATEDIF(起始日期,结束日期,类型)

**注：结束日期必须大于起始日期**

**类型：** Y ：时间段中的整年数。

M：时间段中的整月数。

D：时间段中的天数。

MD： start\_date 与 end\_date 日期中天数的差。忽略日期中的年和月。

YM： start\_date 与 end\_date 日期中月数的差。忽略日期中的年。

YD： start\_date 与 end\_date 日期中天数的差。忽略日期中的年。

**1、计算出生日期为1973-4-1人的年龄；**

公式： =DATEDIF("1973-4-1","2023-2-8","Y") **结果：49**（**计算出来的是周岁**）

**2、计算日期为1973-4-1和当前日期的间隔月份数.**

公式： =DATEDIF("1973-4-1","2023-2-8","M") **结果：598**

**3、计算日期为1973-4-1和当前日期的间隔天数；**

公式： =DATEDIF("1973-4-1","2023-2-8","D") **结果：18210**

**4、计算日期为1973-4-1和当前日期的不计年数的间隔天数；**

公式： =DATEDIF("1973-4-1","2023-2-8","YD") **结果：313**

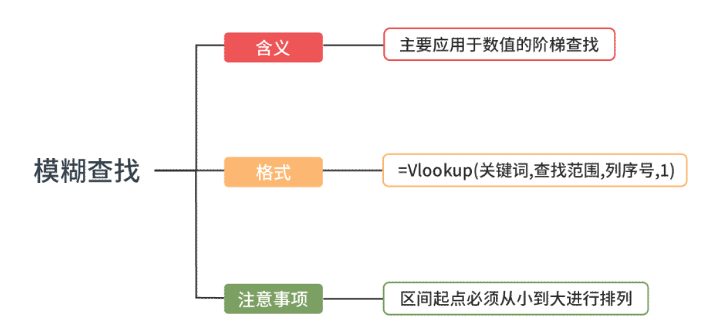
**5、计算日期为1973-4-1和当前日期的不计月份和年份的间隔天数.**

公式： =DATEDIF("1973-4-1","2023-2-8","MD") **结果：7**

**6、计算日期为1973-4-1和当前日期的不计年份的间隔月份数.**

公式： =DATEDIF("1973-4-1","2023-2-8","YM") **结果：10**

**使用vlookup函数进行模糊查询：（构建辅助列）**



**案例1：已知学生成绩分数表，根据分数区间对学生成绩进行等级划分**



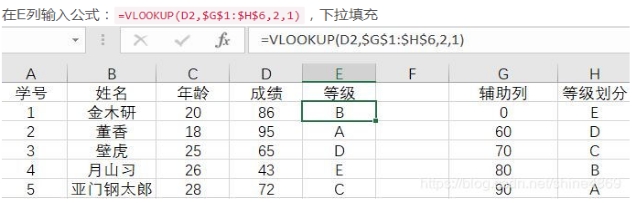


**注意：**

**（1）辅助列构造说明：**只要D列得成绩达到G列中的分数即可对其进行赋值。例如：分数只要超过0分，不超过60分就可赋值为E，以此类推。(这里没有考虑0分的情况;如果考虑0分的情况，把辅助列中的0改为负数即可)

（2）G列和H列位置不能互换！这也对应了"查询值必须在查询区域的首列"的原则。

**所以：辅助列需要放在等级划分的前面！**

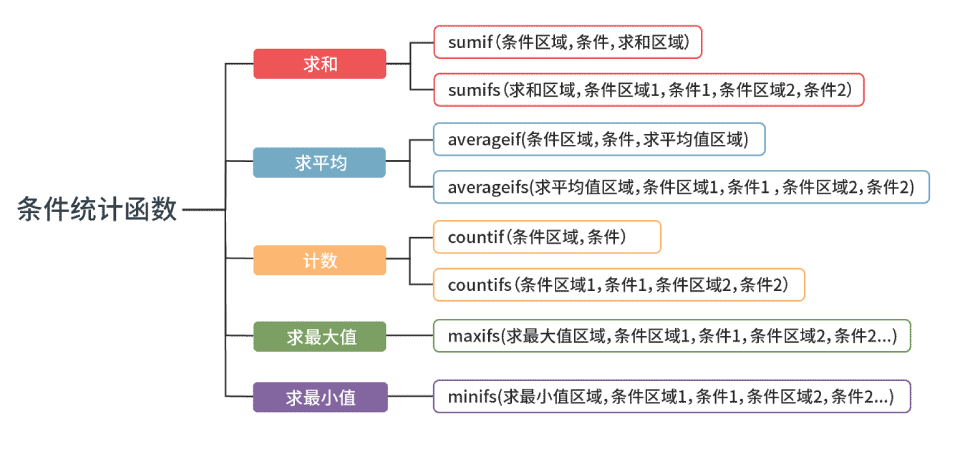


**知识复习：**

**rank函数：求某一个数值在某一区域内的排名。**

**语法：**=rank(要求排名的那个数值或者单元格名称, 排名的区域,[order])

**order的为0和1，默认不用输入，得到的就是从大到小的排名，若是想求倒数第几，order的值请使用1。**

**复制公式注意：选中公式后，按下CTRL+C，然后按下ENTER,最后到目标单元格粘贴。**

**5、vlookup函数、mid函数**

**SEARCH函数：**

**（1）功能：**在第二个文本字符串中查找第一个文本字符串，并返回第一个文本字符串的**起始位置的编号**，该编号从第二个文本字符串的第一个字符算起。

**（2）语法：**SEARCH(find\_text,within\_text,[start\_num])

find\_text:必需，要查找的文本字符串

within\_text:必需，要在哪一个字符串查找

start\_num:可选，从within\_text的第几个字符开始查找。当从第一个字符开始查找时可省略。但实际上，不管你输几，它都是从第一个字符开始查找，只是会跳过从开始到你输入数字中间的字符。

**（3）注意:**

①在find\_text中，可以使用通配符，例如:问号"?"和星号"\*"。其中问号"?"代表任何一个字符，而星号"\*"可代表任何字符串。**如果要查找的字符串就是问号或星号，则必须在这两个符号前加上"~"符号。**

②如果找不到find\_text的值，则返回 错误值 #VALUE!。

③如果省略了start\_num参数，则假设其值为 1。

④如果start\_num不大于0(零)或大于within\_text参数的长度，则返回 错误值 #VALUE!。

**（4）实例：**



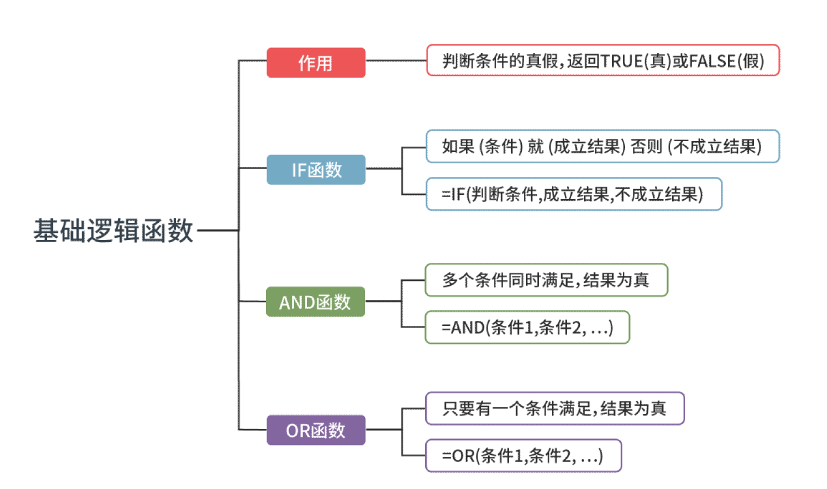
**iferror函数：**

# （1）功能：如公式的计算结果为错误，则返回您指定的值;否则将返回公式的结果。

**（2）语法：IFERROR(value, value\_if\_error)**

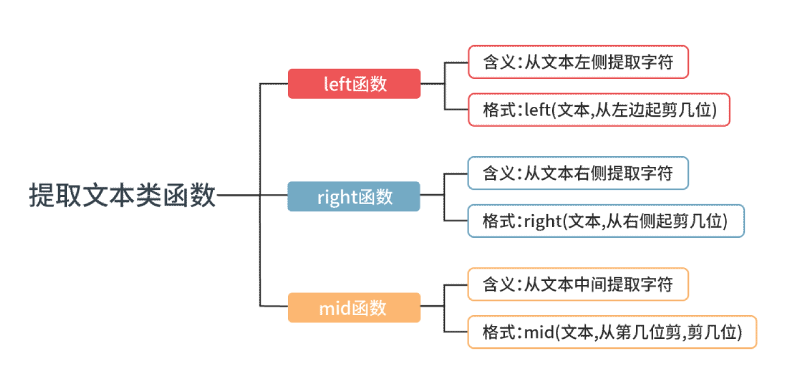
**6、vlookup函数、sumif 函数、sum 函数**

**7、if函数、 or条件、 average函数**



**8、vlookup函数、 mid函数、 left 函数、 if 函数**

**countifs 函数、 min函数**



**9、sqrt函数、模拟分析运算**

**10、left 函数、lenb函数、 len函数、 sum 函数、rank函数**

**INDEX函数：是很常用的查找引用函数。通常情况下，嵌套其余函数一起使用，比如index+match。**

**功能：返回特定行和列交叉处单元格的引用。**

**格式：INDEX(数据区域，行标，列标)**

**match函数：匹配函数,确定列表中某个值的相对位置。**

**功能：返回指定值在指定数据区域中的相对位置，引用区域是一行则返回列标，引用区域是一列则返回行标。**

**格式：match(指定值，指定数据区域，匹配方式)**

**11、vlookup函数、sumif 函数、mid函数、len函数**

**clean函数：删除文本中所有不可见字符（不可打印字符）；**

**格式：clean(文本)**

**trim函数：删除字符串中多余的空格；**

**格式：trim(文本)**

**12、sum函数、iferror函数、vlookup函数**

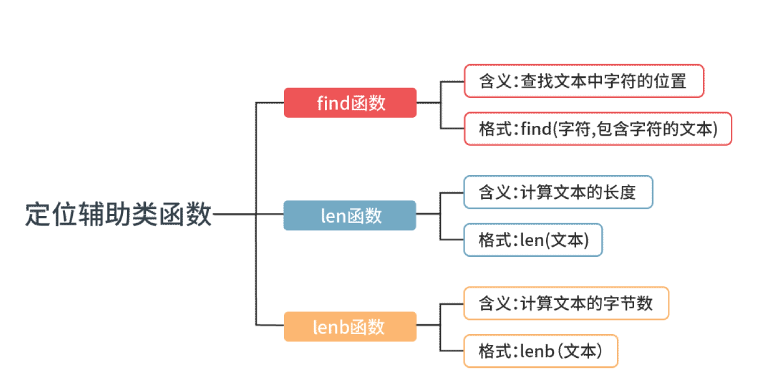
**13、vlookup函数、min函数、if函数、or条件、sum函数**

**HYPERLINK函数:** **创建一个快捷方式(跳转),可以用来打开存储在网络服务器、Intranet或internet中的文件。**

**格式：HYPERLINK（”链接位置”,”显示文本”）**

**注意：链接位置是工作表单元格的话，前面必须加上#**

**14、datedif函数、if函数、len函数、切片器（筛选器）**



**15、round函数、数据模型（2016以上版本自带，低版本需要安装插件）**

**16、vlookup函数、mod函数**

**row函数：返回一个引用的**[**行号**](https://baike.so.com/doc/2178969-2305661.html)**。**

|  |  |
| --- | --- |
| **=ROW(C10)** | **引用所在行的行号(10)** |
| **=ROW(D4:E6)** | **引用中的第一行的行号(4)** |

**17、vlookup函数、 index 函数、match函数、 sumifs函数**

**Indirect函数：对引用进行计算，并显示其内容。当需要更改公式中**[**单元格**](https://baike.so.com/doc/6733917-6948278.html)**的引用，而不更改公式本身，请使用此函数，INDIRECT为间接引用。**

|  |  |
| --- | --- |
|  | **A** |
| **1** | **数据** | **数据** |
| **2** | B2 | 1.333 |
| **3** | B3 | 45 |
| **4** | George | 10 |
| **5** | 5 | 62 |

|  |  |
| --- | --- |
| **公式** | **说明(结果)** |
| =INDIRECT("$A$2") | 单元格A2中的引用值(B2) |
| =INDIRECT("$A$3") | 单元格A3中的引用值(B3) |
| =INDIRECT("$A$4") | 如果单元格B4有定义名"George"，则返回定义名的值(George) |
| =INDIRECT("B"&$A$5) | 单元格B5中的引用值(62) |

**18、index 函数、match函数、left函数、if函数、count 函数、**

**countif函数**

**19、vlookup函数、sumif函数、rank函数、mid函数、len函数**

**20、mid函数、vlookup函数、mod函数、if函数、int函数、or条件**

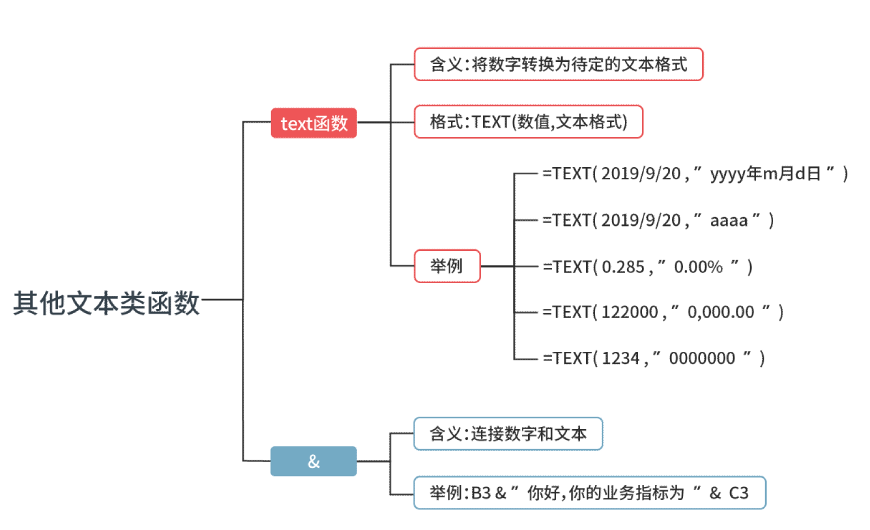
**Column函数: 返回一引用的列号**

**格式：COLUMN(reference)**

**Reference为需要得到其列标的单元格或[单元格区域](https://baike.so.com/doc/4291046-4494512.html" \t "_blank)。Reference 不能引用多个区域，如果省略reference，则假定为是对函数COLUMN所在单元格的[引用](https://baike.so.com/doc/1036451-1096210.html" \t "_blank)。**

|  |  |
| --- | --- |
|  | **A** |
| **1** | **公式** | **说明(结果)** |
| **2** | =COLUMN() | 公式所在的列(1) |
| **3** | =COLUMN(A10) | 引用的列(1) |
| 4 | =COLUMN(C3:D10) | 引用中的第一列的列号 (3) |

**Text函数**



**Sumproduct函数：返回相应的数组或区域乘积的和。**

**注意：数组参数必须具有相同的维数，否则，SUMPRODUCT将返回错误值 #VALUE!。SUMPRODUCT 将非数值型的数组元素作为0处理。**

**格式：SUMPRODUCT(array1,array2,array3, ...)**

**Array1,array2,array3, ... 为 2 到 30 个数组，其相应元素需要进行相乘并求和。**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **B** | **C** | **D** | **E** |
| **1** | **Array1** | **Array1** | **Array2** | **Array2** |
| **2** | 3 | 4 | 2 | 7 |
| **3** | 8 | 6 | 6 | 7 |
| **4** | 1 | 9 | 5 | 3 |

**=SUMPRODUCT(B2:C4,D2:E4)**

**两个数组的所有元素对应相乘，然后把乘积相加，即**

**3\*2+4\*7+8\*6+6\*7+1\*5+9\*3。(156)**

**Date函数：构建日期**

****

**Ceiling函数：将参数Number向上舍入；**

**格式：CEILING(number,significance)**

**Number 待舍入的数值，Significance 基数。**

**如果参数number为非数值型，CEILING 返回**[**错误值**](https://baike.so.com/doc/5600653-5813256.html)**#VALUE!。**

**如果 number 和 significance 符号相同，则对值按远离0的方向进行舍入。如果 number和significance符号相反，则返回错误值#NUM!。**

|  |  |
| --- | --- |
| 公式 | 实际值 |
| =CEILING(-2.5,2) | #NUM! |
| =CEILING(-2.5,-2) | -4 |
| =CEILING(2.5,-2) | #NUM! |
| =CEILING(2.5,2) | 4 |
| =CEILING(2.5,1) | 3 |
| =CEILING(2.5,-1) | #NUM! |
| =CEILING(-2.5,1) | #NUM! |
| =CEILING(-2.5,-1) | -3 |

**21、vlookup函数、if函数、or条件、left函数**

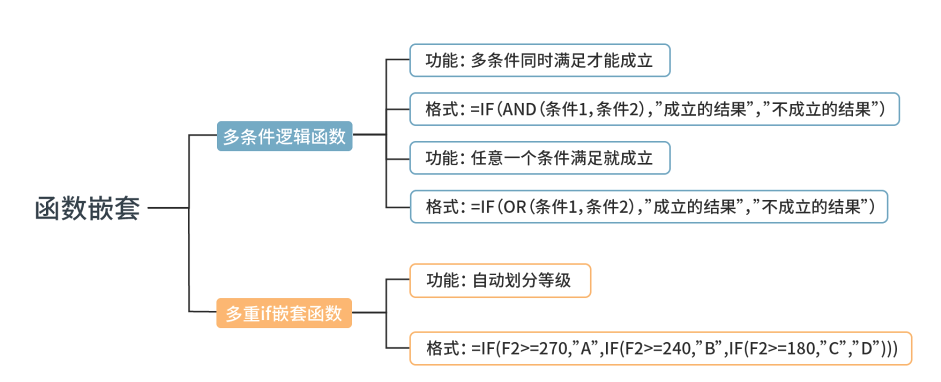
**22、sumif函数、sum函数**

**N函数：返回转化为**[**数值**](https://baike.so.com/doc/3847737-4040005.html)**后的值。**

**23、n函数、sumifs函数、iferror函数**

**24、and条件、if嵌套、countif函数、countifs函数、**

**sumif函数、sum函数**

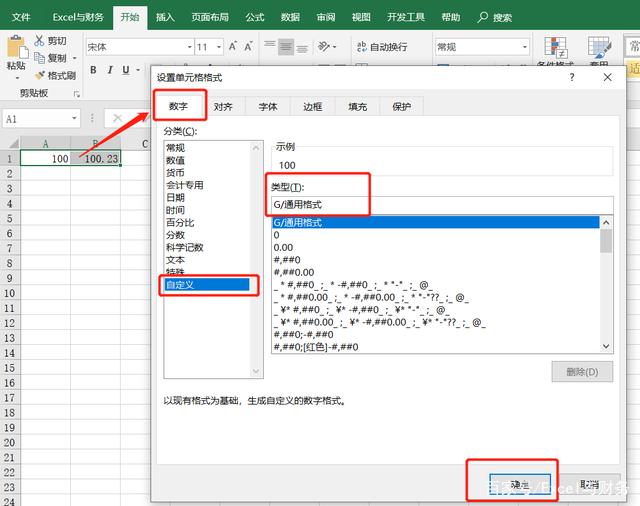


**这9种Excel表格问题都可以用自定义单元格格式解决！**

**一、【G/通用格式】**

以常规的数据显示，输入的数字多少就显示多少，如100显示100，100.23显示100.23

鼠标右键点击单元格，选择【设置单元格格式】（或按Ctrl+1键），分类选择【自定义】，类型文本框中输入：G/通用格式 最后点击【确定】，设置完成。

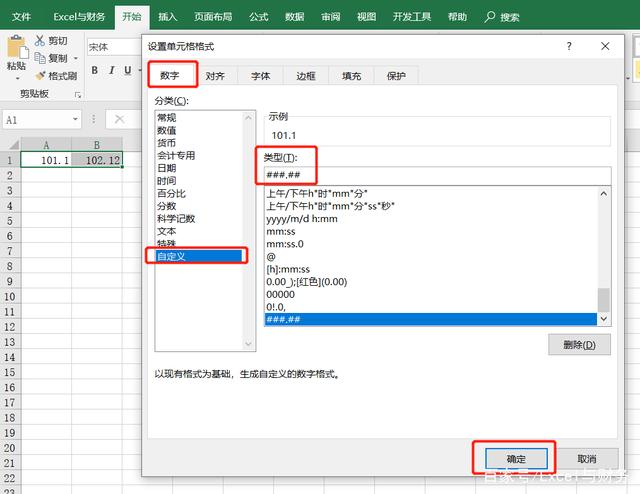


**二、【#】：数字占位符**

只显有意义的零而不显示无意义的零。小数点后数字如大于"#"的数量,则按"#"的位数四舍五入。

类型文本框中输入：###.## 则101.1 显示为 101.1, 102.123 显示为 102.12

**此方法适用于设置数字小数位。**

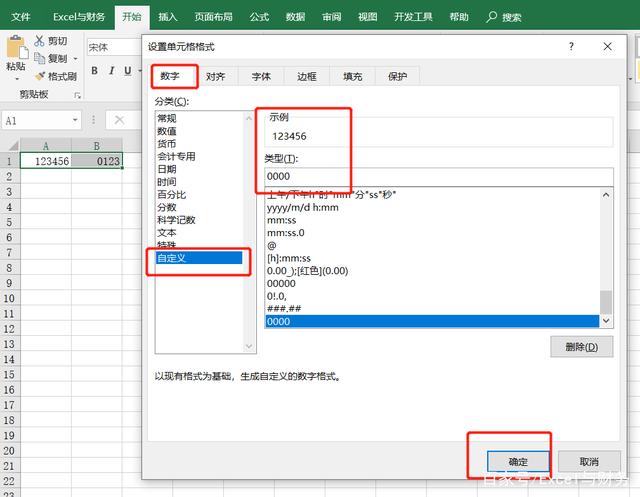


**三、【0】：数字占位符**

这种格式下，如果单元格内数字个数大于0占位符个数,则显示实际数字,如果小于0占位符的数量,则用0补足。

在类型文本框中输入：0000 123456则显示123456，123则显示0123

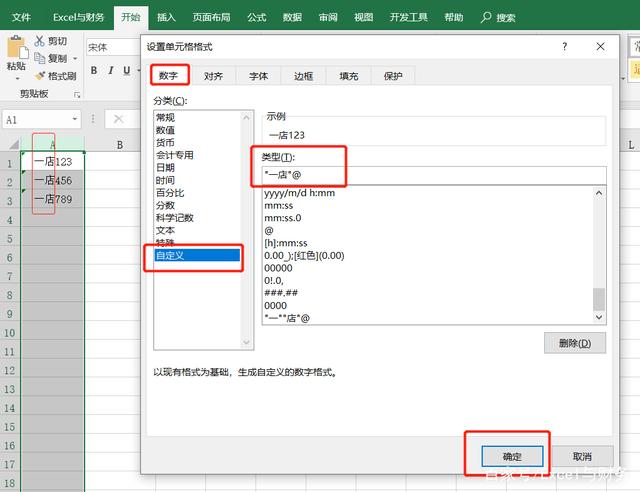
**此方法适用于设置固定数字长度。**



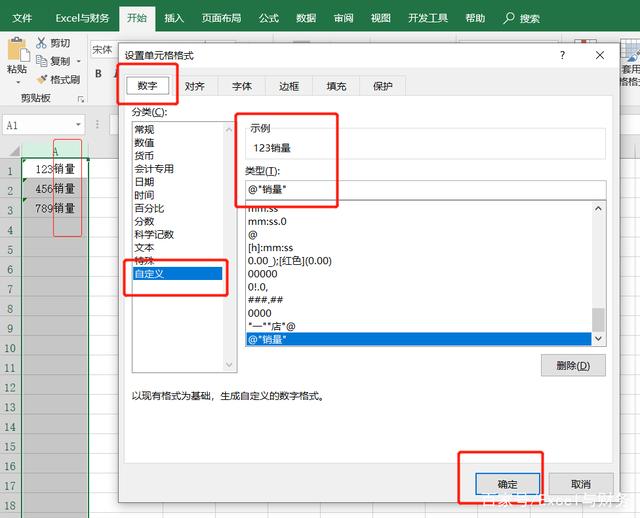
**四、【@】文本占位符**

如果使用单个@, 作用是引用原始文本, 要在输入数字数据之后自动添加文本,

自定义格式为："文本内容"@；要在输入数据之前自动添加文本,



自定义格式为：@"文本内容"。在要输入的数据之后自动添加文本



@符号的位置决定了Excel输入的数字数据相对于添加文本的位置。如果使用多个@, 则可以重复文本。

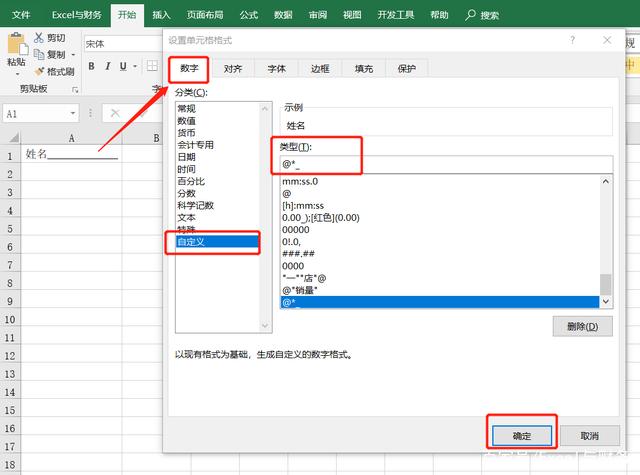
**此方法适用于批量给单元格内容添加前缀、后缀。**

**五、【@\*】：重复字符充满列宽**

如果想让单元格内容充满列宽，用到代码：@\*重复字符

在类型框中输入：@\*\_ 单元格输入：“姓名”则显示“姓名\_\_\_\_\_\_\_”

此方法适用批量重复内容填充单元格，如：自动添加下划线

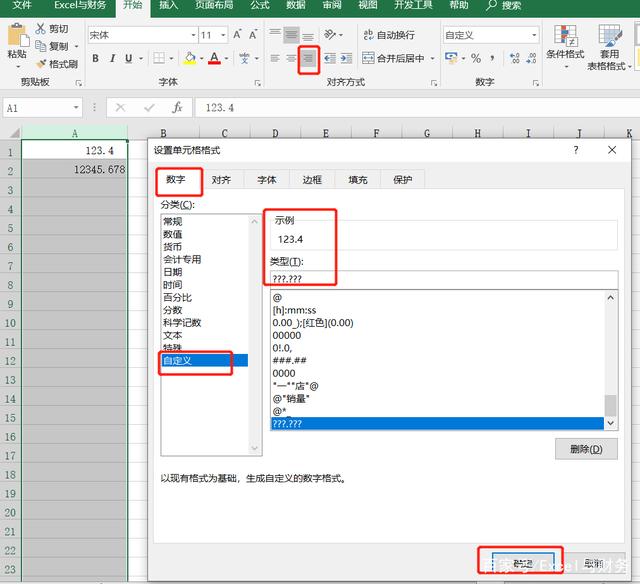


**六、【?】: 数字占位符**

在小数点两边为无意义的零添加空格,以便当按固定宽度时,小数点可对齐,另外还用于对不等到长数字的分数

类型文本框中输入：???.??? 单元格内输入 123.4 显示 123.4； 输入12345.678 显示 12345.678，如果小数位超过 ?占位符个数，则按四舍五入

**此方法适用于按小数点对齐数据。**



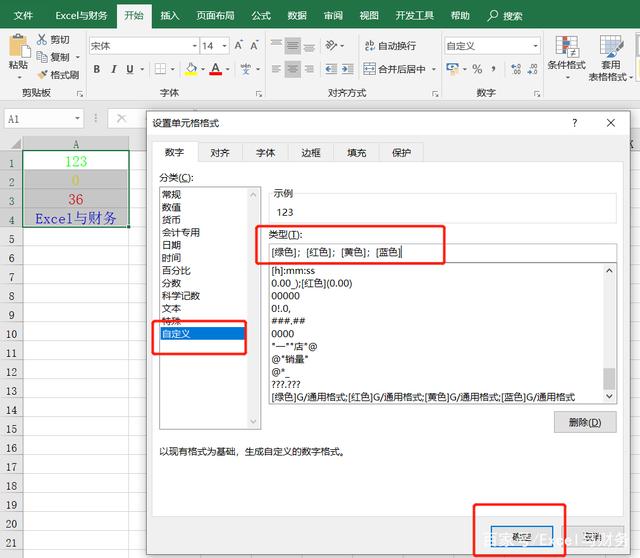
**七、【颜色】：用指定的颜色显示字符**

我们可以用自定义格式显示单元格数据颜色，有八种颜色可选：红色、黑色、黄色、绿色、白色、蓝色、青色和洋红。

在类型文本框中输入：[绿色]；[红色]；[黄色]；[蓝色]

单元格数字大于0时，显示为绿色，小于0时显示为红色，等于0时显示为黄色，单元格内容为文本时则显示绿色

**此方法适合区分数字大小，自动设置颜色方便阅读查看**



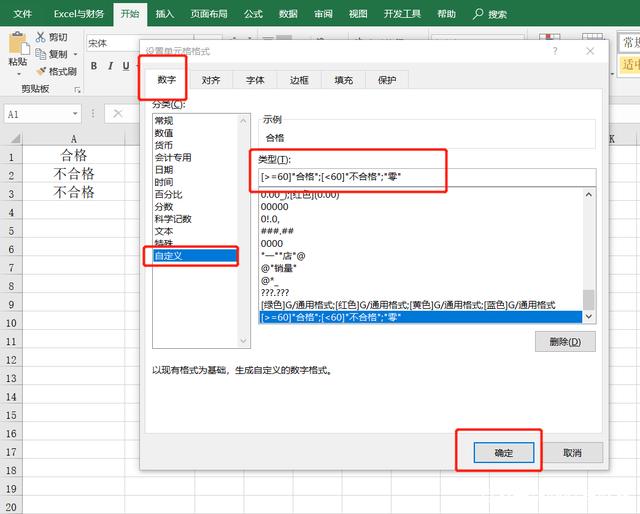
**八、条件：判断数字大小后再按设置显示**

自定义格式也可以按条件显示，但限于使用三个条件, 前两个条件是明确的, 最后就是上两个条件不包含的其他。条件要放到方括号中。

在类型文本框中输入：[>=60]"合格"；[<60]"不合格"；"零"

单元格输入80显示“合格”, 45显示“不合格”

**此方法适用于根据数值大小而显示不同内容，相当于IF函数，如判断学生成绩是否合格**



**九、时间和日期代码**

"YYYY"或"YY"：按四位(1900~9999)或两位(00~99)显示年

"MM"或"M"：以两位(01~12)或一位(1~12)表示月。

"DD"或"D"：以两位(01~31)或一位(1-31)来表示天。

在类型文本框中输入：yyyy"年"m"月"d"日"

单元格内输入2020-8-19显示为 2020年8月19日

