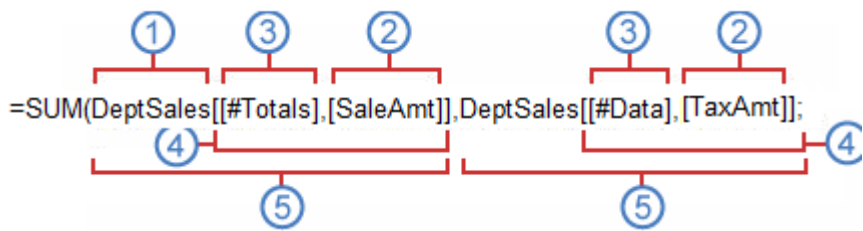


使用结构化引用

Spread. Sheets 支持在表格中使用结构化引用。

结构化引用使用关键字和列名来引用表格中的单元格区域。

结构化引用可以分为以下几个部分，如图所示。图中的公式表示了将税和销售额相加的结果。



- 1. 表名称：表名称是一个有实际意义的名称，它指的是表格的数据区域（不带任何页眉和总计行）
- 2. 列说明符：使用列说明符来引用列的数据。列说明符指代了列的数据（不包含任何列标题行获汇总行）。使用列说明符时，需要用方括号括起来。
- 3. 项目说明符：用于引用表格中特殊的位置，比如总计和汇总等。
- 4. 表说明符：表说明符是由列说明符和项目说明符组成的，需要将其使用方括号括起来。
- 5. 结构化引用：从表名称开始，到表说明符结束的部分称为一个结构化引用。

你可以使用一些特殊说明符以简化公式的编写。

你可以用这些特殊的说明符来方便地引用表格中的任意部分。

下表说明了你可以使用的特殊说明符：

特殊项目	例子	引用	单元格区域
#All	=DeptSales[#All]	整个表格，包括标题，数据和总计。	A1:E8
#Data	=DeptSales[#Data]	数据	A2:E7
#Headers	=DeptSales[#Headers]	标题行	A1:E1
#Totals	=DeptSales[#Totals]	总计行	A8:E8
#This Row	=DeptSales[#This Row]	当前行的一部分	A5:E5 (current row is 5)

"#This Row" "@" . =DeptSales[#This Row] =DeptSales[@]
"#Data" "#Headers" "#Totals":

特殊项目	例子	引用	单元格区域
#Headers and #Data	=DeptSales[#Headers], [#Data]	列标题和数据	A1:E7
#Data and #Totals	=DeptSales[#Data], [#Totals]	数据和合计行	A2:E8

如果未指定特殊项目，输入的公式将会被认为是数据显示在表格中。

特殊项目	例子	引用	单元格区域
None	=DeptSales[] or =DeptSales	数据区域	A2:E7

列说明符表示一列数据，不包括标题行和合计行。

下表列举了列说明符的用法：

列说明符	例子	引用	单元格区域
单列	=DeptSales[SalesAmt]	指代 "SalesAmt" 列的所有数据	C2:C7
多列	=DeptSales[[SalesAmt]:[TaxAmt]]	从 "SalesAmt" 列到 "TaxAmt" 列的数据	C2:E7
无	=DeptSales[#Data]	所有列的数据	A2:E8

所有表，列，和特殊项目说明符需要使用方括号[]括起来。使用嵌套说明符时，需要使用方括号（[]）括起来。包含其他说明符说明符需要外部匹配方括号内的其他说明符方括号括起。如果策略有冲突的话，优先级是从高到低的。

方括号遵循以下规则：

- 1. 如果没有给出任何说明符，表说明符可以省略。比如：公式"=DeptSales[]" 等同于公式"=DeptSales"。
- 2. 如果只有一个特殊说明符或者只有一个列说明符，表说明符可以省去。比如：公式"=DeptSales[#Data]" 等同于公式"=DeptSales[#Data]"，公式"=DeptSales[[TaxAmt]]" 等同于 公式"=DeptSales[TaxAmt]"。
- 3. 如果列说明符和特殊项目说明符一起引用，特殊项目说明符必须使用方括号括起来。比如：公式"=DeptSales[#Data, [TaxAm]]" 和公式"=DeptSales[Data, #Totals]" 是错误的。

- 如果表格列标题包含特殊字符，在写表说明符的时候需要使用方括号括起来。特殊字符包括：Tab、换行符、回车符、逗号（,）、冒号（:）、句号（.）、左中括号（[）、右中括号（]）、井号（#）、单引号（'）、双引号（"）、左大括号（{）、右大括号（}）、美元符号（\$）、脱字符号（^）、与号（&）、星号（*）、加号（+）、等于号（=）、减号（-）、大于符号（>）、小于符号（<）、除号（/）。比如，如果有一个列的列标题叫“#column1”，公式“=Table1[#column1]”这个公式就是错误的，应该写为公式“=Table1['#column1]”。
- 如果在列说明符中有区域运算符，必须至少有列说明符使用方括号括起来，比如：公式“=Table1[column1:column2]”是错误的，但是它可以被写成这样：公式“=Table1[[column1]:column2]”或者公式“=Table1[column1:[column2]]”或者公式“=Table1[[column1]:[column2]]”。
- 当使用'@'特殊说明符时，可以将方括号省略去。比如：公式“=DeptSales[@, TaxAmt]”可以简写为：公式“=DeptSales[[@], TaxAmt]”。

以下通用规则同样适用：

- 所有的表名，列名，特殊项目说明符均不区分大小写。
- 下列字符具有特殊含义，需要使用转义字符（^）：左中括号（[）、右中括号（]）、井号（#）、单引号（'）。比如：公式公式“=Table1[colu#mn1]”是错误的，应该写为 公式“=Table1[colu'^#mn1]”。
- 使用逗号（,）将列说明符和特殊项目说明符分割开，比如： “=DeptSales[[#Data], [#Totals], TaxAmt]”。
- 列说明符和特殊项目说明符的顺序是不受依赖的。比如：公式“=DeptSales[[#Data], [#Totals], TaxAmt]”等同于公式 “=DeptSales[[#Totals], TaxAmt, [#Data]]”。
- 在'@' 特殊项目分隔符后面的逗号（,）可以被省略，比如：公式“=DeptSales[@, TaxAmt]” 等同于公式 “=DeptSales[@TaxAmt]”。

你可以使用 `setColumnDataFormula` 和 `setColumnFormula` 方法来为公式添加结构化引用。你也可以在表格区域以外引用表中的数据，比如：

`activeSheet.getCell(7,1).formula("SUM(Table1[SubTotal])")`；。注意，当设置了多个公式时，最后一个设置的公式优先生效。

列说明符在表头和表页脚中不会生效。

	A	B	C	D
1	Value1	Value2	SubTotal	
2	1	5	5	
3	2	5	10	
4	3	5	15	
5	Total		30	
6				

参考代码

以下代码创建了一个表，并在公式中使用了结构化引用来创建总计栏。

JavaScript

```
activeSheet.tables.add("Table1", 0, 0, 4, 3,
GC.Spread.Sheets.Tables.TableThemes.dark1);
activeSheet.getCell(0,0).text("Value1");
activeSheet.getCell(0,1).text("Value2");
activeSheet.getCell(0,2).text("SubTotal");
activeSheet.getCell(1,0).text("1");
activeSheet.getCell(2,0).text("2");
activeSheet.getCell(3,0).text("3");
activeSheet.getCell(1,1).text("5");
activeSheet.getCell(2,1).text("5");
activeSheet.getCell(3,1).text("5");
var sTable =
activeSheet.tables.findByName("Table1").setColumnDataFormula(2,
"=[Value1]*[Value2]");
sTable.showFooter(true);
//set footer value
sTable.setColumnValue(0, "Total");
sTable.setColumnFormula(2, "SUM(Table1[SubTotal])");
activeSheet.getColumn(0).width(80);
activeSheet.getColumn(1).width(80);
activeSheet.getColumn(2).width(80);
//activeSheet.getCell(7,1).formula("SUM(Table1[SubTotal])");
```