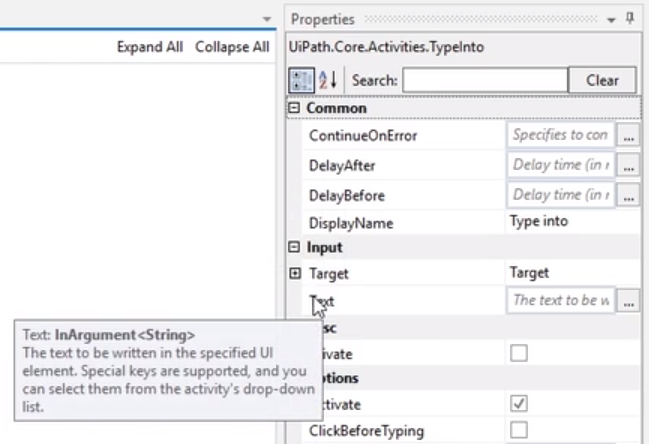
**学习大纲**

* 如何拆分字符串
* 如何改变部分字符串的格式
* 如何在表格中根据条件选中特定的行

**1. 标量型变量，集合，表格**

活动的属性都有预定义的数据类型。鼠标悬停在属性面板的某个属性上，就会出现相应的提示。

* 使用右键菜单或Ctrl+K在属性栏创建的变量，会直接设置为属性预定义的类型。



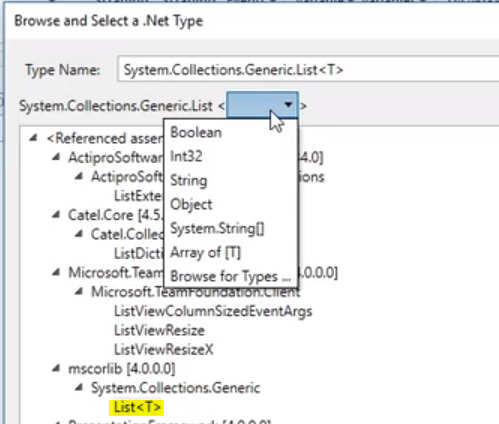
* **标量型变量 (Scalar Variables)**：一个单独的固定类型的数据。
  + e.g. 字符 (Characters)，布尔值，数字，日期和时间
* **集合 (Collections)**:
  + Arrays, Lists, Queues
  + **特殊类型 - 字符串 (Strings)**：字符串可以看作多个字符的集合。
  + **特殊类型 - 字典 (Dictionaries)**：字典包含两个相互关联的集合：名称和值。字典通过名称而不是索引号来引用值。
    - 可以使用字典从Orchestrator队列中提取数据。
* **表格 (Tables)**：将数据保存在二维的结构中，通过行和列来索引数据。

**详细解析：**

1) **Generic Value**：字符串型、布尔型、数字型、日期时间型。使用起来灵活方便，支持多种字符串方法。

2) **Array和List**

* 元素的数量
  + Array：固定数量
  + List：可以通过添加或删除元素来增大或缩小
* **定义方式**
  + Array：在变量面板中选择数组元素的数据类型。定义默认值的语法为：{"Value1", "Value2"}。
  + List：在.NET数据类型中，搜索并选中List<T>，再选择数据类型。定义默认值的语法为：new List(of String) from {"Value1","Value2"}。必须重申数据类型 (of 数据类型)，from加数组的部分可以省略。



* For Each活动：Array和List都适用。**使用For Each时，必须把属性面板中的TypeArgument设置为与Array或List的元素一致的类型。**
* **添加删除元素**：仅适用于List。使用**Invoke Method**活动，在TargetObject中填写List的变量名，在MethodName中填入Add，在属性面板中的Parameters属性里添加新元素，注意Type一定要与List本身的数据类型一致。

3) **字典 (Dictionaries)**

* 作用：
  + 提取Orchestrator队列中的数据
  + 将多个数据作为单个变量传递
* **定义字典**：
  + 在.NET数据类型中，搜索并选中Dictionary<TKey, TValue>，在选择Key和Value的数据类型，一般Key是String型，Value是String型或对象。
  + 定义默认值：new Dictionary(of String, String) from {{"Key1","Value1"}, {"Key2","Value2"}}
* **添加Key和Value**：
  + 加入一个Assign活动，等号左边是字典变量的名称(Name)和新的关键字(Key)，等号右边是新的值(Value)。比如：config("Key3") = "Value3"。

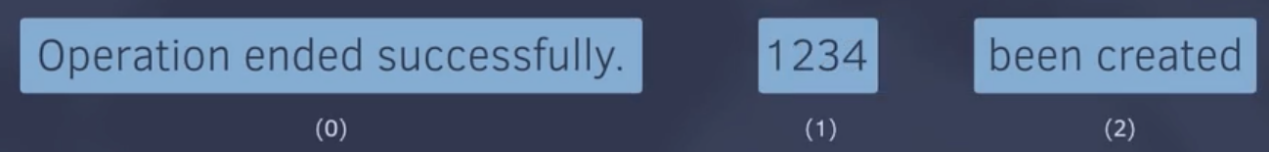
**2. 文本的拆分**

* 文本都要用引号括起来。**如果要表示引号本身，需要输入两次引号**。例如："David says ""Hello"""显示为David says "Hello"。
* **Split方法**：通过分隔符拆分文本，生成一个数组。**数组的索引号从0开始**。
  + 将变量message(message="Operation ended successfully. Record 1234 has been created.") 以空格作为分隔符拆分，写作message.Split(**" "c**)。拆分后将得到一个拥有8个元素的数组，如下图所示：



* + 将光标停在.Split()的括号中间，同时按下**Ctrl+Shift+Space**，会显示Split的更多使用方法和相关的参数信息，上下键可以翻页。
  + **分隔符可以是一个字符串，可以不止有一个**。message.Split({"Record ", " has"},StringSplitOptions.None)可以得到下图所示的数组：





* + 注意：{"Record ", " has"}中，Record的后面和has的前面要加空格。
  + Split方法的第二个参数StringSplittOptions.None指定是否支持空元素。

**3. 文本的组合**

1) **连接文本**：

* **连接字符串型的数据**：直接使用**加号**连接，比如"This " + "that" = "This that"。
* **连接混合类型的数据**：先把非字符串型的数据转换成字符串型，再用加号连接。比如”This “ + myNumber.**ToString** + Now.**ToString** = "This 4210/17/2018 16:15:30"
  + 改变时间/日期格式：例如Now.ToString("MMMM yyyy") 会显示为October 2018。
  + [更多格式的表达参考](https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings)

2) **String.Forma**t**方法**：适用于更长的、更复杂的数据组合，并且允许自定义格式。

* String.Format("Hello {0} {1}", myNumber, Now)
* Hello是静态文本；{0}和{1}是占位符，表示引号右边的数据在整个字符串中显示的位置，从0开始索引。
* 更改日期和时间的格式可以写作：String.Format("Hello {0} {1:MMMM yyyy}", myNumber, Now)

3) **常用的String方法**：filePath = "Downloads/{0}/output.txt"，itemID = "ASD123"

* GenericValues也可以使用除了EndsWith，StartsWith和Format以外的方法。
* [详情参考](https://docs.microsoft.com/en-us/dotnet/api/system.string?view=netframework-4.7.2)

|  |  |  |
| --- | --- | --- |
| **方法** | **语法示例** | **运行结果** |
| Contains | filePath.Contains(".") | True |
| EndsWith/StartsWith | filePath.EndsWith(".txt") | True |
| Format | String.Format(filePath,itemID) | "Downloads/ASD123/output.txt" |
| Replace | filePath.Replace("/","\") | "Downloads\{0}\output.txt" |
| Split | filePath.Split("/".ToCharArray) | {"Downloads", "{0}", "output.txt"} |
| Substring | filePath.Substring(10) | "{0}\output.txt" |
| ToLower/ToUpper | filePath.ToUpper | "DOWNLOADS/{0}/OUTPUT.TXT" |
| Trim | random.Trim | "ASD" |

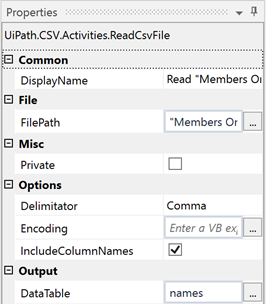
 4) **Data Tables**

常见的数据表格有**Excel文件**或**CSV文件**。CSV（Comma-Separated Values，逗号分隔值也称字符分隔值）文件以纯文本形式存储表格数据。CSV文件由任意数目的记录组成，记录间以某种换行符分隔。每条记录由字段组成，字段间的分隔符是其它字符或字符串，最常见的是逗号或制表符。

* **Read CSV活动**：将CSV文件中的数据存入DataTable类型的变量中。勾选IncludeColumnNames，表示第一行是列标题。
* **Output Data Table活动**：将DataTable型数据转换成字符串型，可以通过Message Box活动或Write Line活动显示该字符串。适合抓取数据或Debug。
* **For Each Row活动**：在DataTable的每一行中循环。
* 如果要取得某一列的数据，可以直接**指定列标题**。例如：在For Each Row活动中，**row("Name")**.ToString将返回该DataTable的**Name**一列下所有的数据。
* 如果要获得某一列某一行的数据，还要加上行号，行号从0开始且不计算标题行。例如：sampleData.Rows**(0)**("Name").ToString。
  + 可以指定列的索引号代替列标题，**第一列的索引号是0**。如果Name是第2列，可以写作sampleData.Rows(0)**(1)**.ToString
* **Select方法**：搜索整个DataTable的行，找到符合指定条件的行，并返回一个数组。
* **实例：**提取DataTable变量sampleData中年龄小于40且Income大于40k的行。
  + 将sampleData.**Select**("Age<40 and Income>**'**40k**'**")赋值给变量filteredRows。filteredRows的数据类型是**DataRow[]**（数组）。
  + DataRow[]通过**For Each活动**循环：Foreach **row**in filteredRows。注意将**TypeArgument**的类型改为**DataRow**（不是数组）。
  + 在For Each活动中添加Write Line活动：row("Name").ToString + " " + row("Age").ToString + " " + row("Income").ToString。
  + 运行程序会在Output面板中写入符合条件的行的Name，Age和Income信息。

除了Select方法外，DataTable类型还有很多手写的方法。

**4. 练习：为俱乐部成员设置昵称**

* 要求：为俱乐部成员取一个昵称，昵称由名字的前三个字母全大写和姓氏的前三个字母全小写组成，如Marcella Knipp的昵称为MARkni。
* 源文件：CVS文件。第一行是列标题(First, Last, Club Number)。从第二行开始记录名字、姓氏、是否为俱乐部成员(Yes/No)。
* 思路：
  + 添加**Read CSV活动**：读取源文件，赋值给一个DataTable类型的变量**names**。确保IncludeColumnNames属性是被勾选的状态。
  + 
    - 如果需要读取的文件存放在项目文件夹下，则可以直接输入文件名，而不需要完整的文件路径。
  + 新建一个变量**clubMembers**，变量类型选择**Array of [T]**，元素类型选择**Browse for Types...**，搜索**DataRow**。这样就创建了一个DataRow的数组。
  + 添加**Assign活动**：To=**clubMembers**；Value=**names.Select("[Club Member]= ‘Yes’ ")**
    - 如果列标题包含空格，则要使用中括号**[ ]**将它括起来。
    - 文本内容要用单引号**''**括起来。
  + 添加**For Each活动**：修改为Foreach **row** in **clubMembers**，将**TypeArgument**属性设置为**DataRow**。
  + 新建三个String型变量，**firstName**，**lastName**和**nickName**。
  + 在For Each里添加两个**Assign活动**：firstName=**row("First").ToString**；lastName=**row("Last").ToString**。
  + 生成nickName需要用到：**Substring方法**提取名和姓的前三个字母；**ToUpper方法**将前三个字母大写；**ToLower方法**将前三个字母小写；**+号**连接字符串。
  + 在For Each里添加**Assign活动**：nickName=**firstName.Substring(0,3).ToUpper + lastName.Substring(0,3).ToLower**。
    - Substring(0,3)表示从第一个字符开始(Start Index)，提取三个字符(Length)。
  + 在For Each里添加**Wirte Line活动**：Text=**nickName**。运行后在Output面板中会显示俱乐部成员的昵称。

**\*本课使用过的新活动、方法、函数等：**

* **Invoke Method**
* **Read CSV**
* **Output Data Table**
* **For Each Row**
* .Contains()：根据文本中是否包含指定的字符串，返回True或False。
* .Split()
* String.Format()