**UiPath手把手教程八-参数的使用**

大家好这里是Python知识学堂。

作者在RPA领域工作已经两年多，看到UiPath产品也在不断的完善，有必要分享一下学习和使用UiPath的一些经验，帮助小伙伴踩坑。文章中有什么不对的地方还请指出，一起学习一起进步。接下来我们开始学习：

本次我们学习的内容是关于工作流的相关知识，一些小伙伴说自己的Uipath界面语言是英文的，不习惯中文的，因此在今后的教程中我们也直接使用英文界面的Uipath。

上期文章我们介绍了UiPath中数据类型相关的知识，具体文章可从以下链接获取（点击红色字体即可获取）:

XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

此次文章我们一起来学习一下UiPath中参数的相关知识。

**一、UiPtah运算符**

在介绍参数之前，我们先介绍一下UiPath中一些常用运算符，与其他的编程语言一样，编程语言都有一些常用的运算符：

|  |  |
| --- | --- |
| 符号 | 具体含义 |
| = | 等于 |
| <> | 不等于 |
| + | 加法运算 |
| - | 减法运算 |
| \* | 乘法运算 |
| / | 除法运算 |
| < | 小于运算 |
| > | 大于运算 |
| <= | 小于等于（多用于判断） |
| >= | 大于等于（多用于判断） |

在逻辑运算符方面跟其他的语言也是一致的，只是名称上存在不同而已，使用逻辑运算符可以很好的判断下一动作是否需要执行,常见的逻辑运算符如下：

|  |  |
| --- | --- |
| 符号 | 具体含义 |
| and | 与 |
| or | 或 |
| xor | 亦或 |
| not | 非 |
| andAlso | 短路与 |
| orElse | 短路或 |

我们对其中的andAlso和orElse逻辑运算符号进行说明一下：

andAlso和orElse是其他语言暂时没有的，UiPath中提供了这个方法。假设我们有两个表达式：表达式1和表达式2。

1. 对于andAlso来说，当有语句：“表达式1 andAlso 表达式2”时，如果表达式1和表达式2均为True时，整个语句为True,否则为False;特别的当表达式1为False的时候，后面的表达式2将不在进行计算。
2. 对于orElse来说，当有语句：“表达式1 andAlso 表达式2”时，如果表达式1和表达式2只要有一个为True时，整个语句为True,否则为False;特别的当表达式1为True的时候，后面的表达式2将不在进行计算。

可以看出在Uipath的运算符与其他语言的运算符有很多的相同之处的。

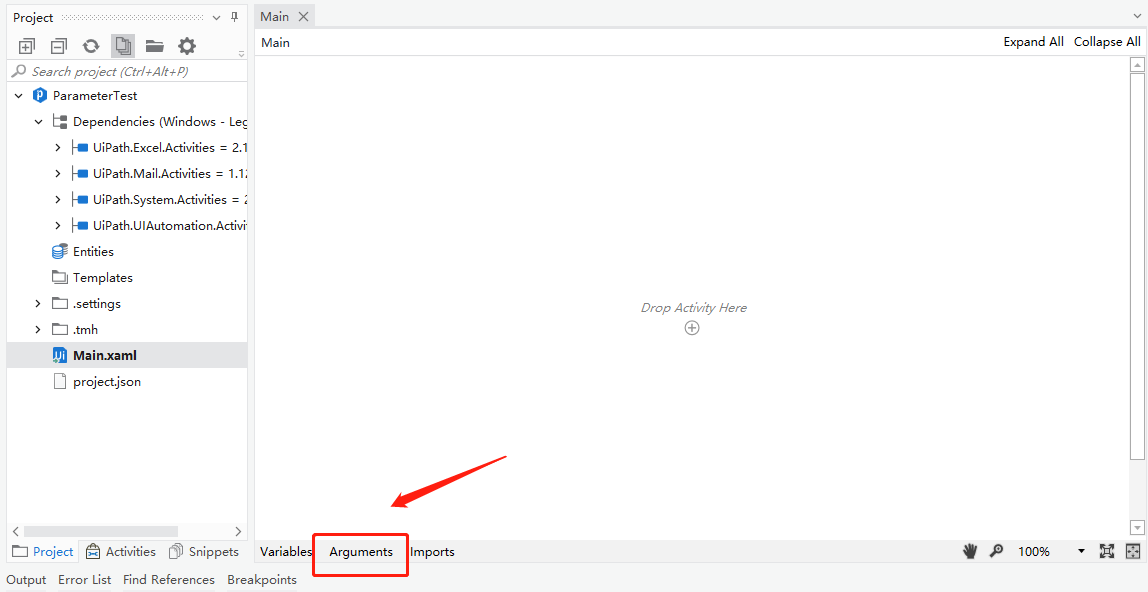
**二、参数**

在之前的文章中我们讲解到了UiPath中的数据结构，诸如Dict,List等等。举例当中，有一些小伙伴可能已经发现，当我们要去编写一个流程的时候，就要使用到一些变量，通过参数去接收流程中的一些数值。

通过这样的发现我们可以知晓，参数可以帮助我们进行数据的处理，好处就是使用参数可以使我们的流程变得清晰和灵活。参数的其他的理解大家可以想一下Python等语言中函数的参数。

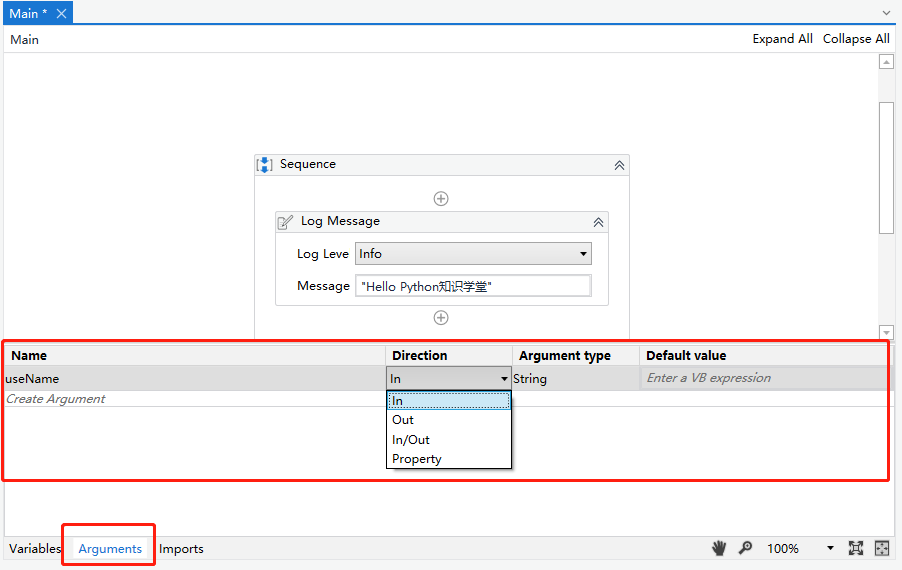
**2.1 参数小例子一**

在UiPath中参数的使用在参数面板(Arguments),在整个界面中如下所示：



旁边是Variables面板也就是变量面板（这里就不在介绍了），那么变量和参数的区别是什么呢？可以这样简单的认为他们的差异是：变量用于一个个活动之间，所以变量有属于活动的这么个属性，而参数用于流程之间的。

以下我们创建了一个参数usename:



可以看出参数有四个属性值：

Name:名称，即参数的名称。

Direction:方向，即参数的传递方向，有in,out,in/out,Property可以选择。

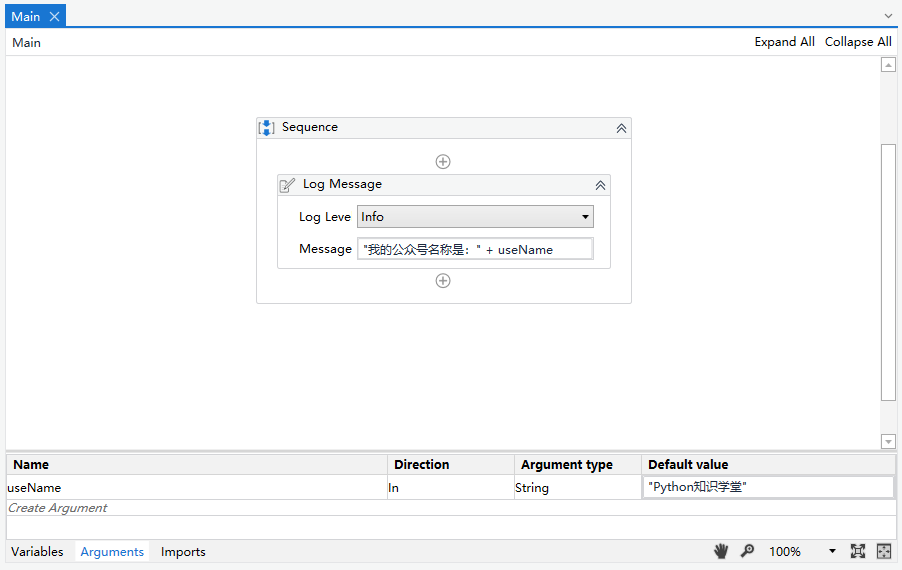
Argument type:参数类型，即规定参数的类型。

Default value:默认值，即参数的默认值。

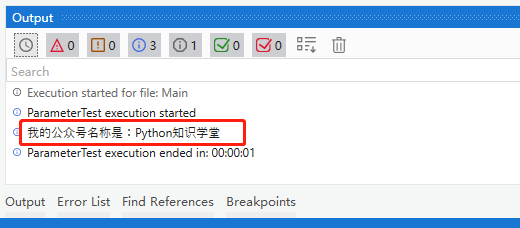
我们来解释一下Direction这个属性的取值：

1. 当Direction为in时，表示将数据输入到流程中，在实际开发流程的时候，如果我们需要将目标数据输入到流程中，就设定Direction=in。
2. 当Direction为out时，表示将数据从流程中输出，在实际开发流程的时候，如果我们需要将目标数据从流程中输出的时候，就设定Direction=out。
3. 上面两者的结合，即都可以满足两者特性。
4. Property不详(我还没发现什么地方用到，大家可以告诉我哈)

接下来我们给上述创建的参数赋予一个值，然后打印出来看看：



输出为：



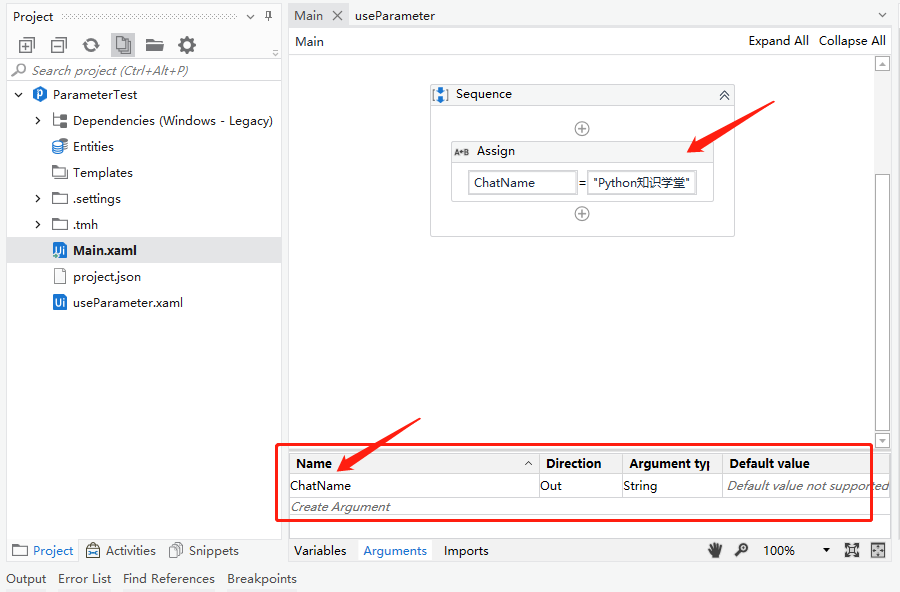
所以可以看出，我们在设置参数值之后是可以进行打印操作的。另外上述参数的Direction我们设置的为in。

上述我们创建参数的方式是使用参数面板来设置（Arguments面板）的。实际流程开发的时候还可以使用其他方式来创建参数。

**2.2 参数小例子二**

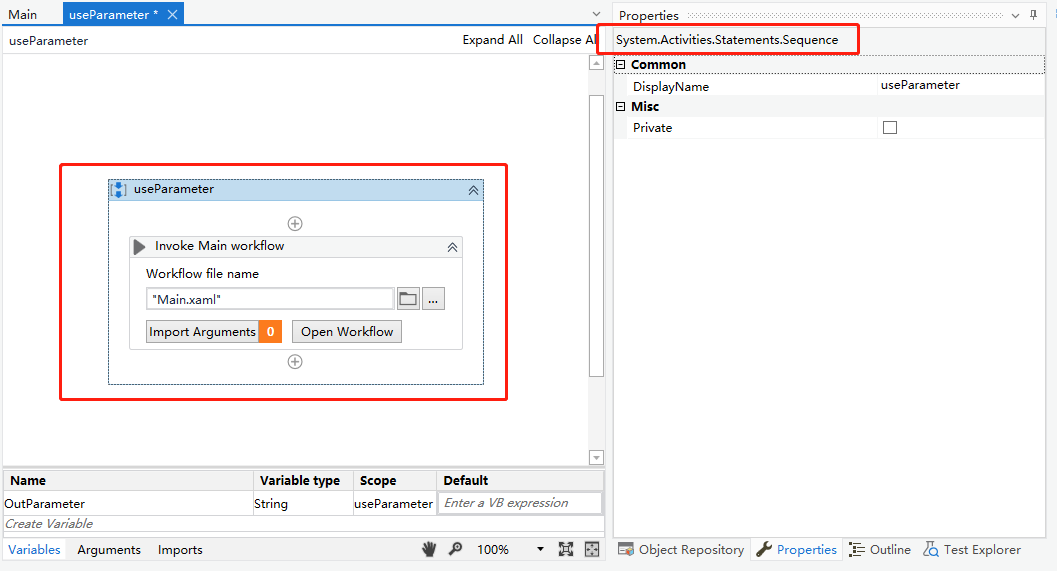
上述使用的参数类型为in,接下来我们介绍一个不同流程中进行参数传递的例子。

1. 我们在主“Main.xaml”中创建有一个流程，如下所示：

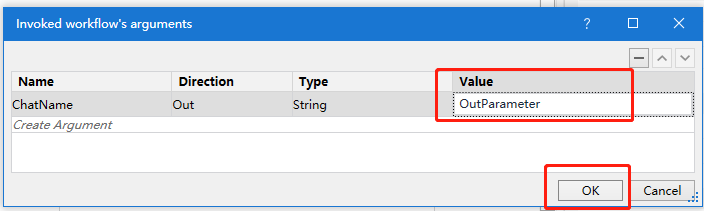


上述参数我们的Direction设置为“Out”类型。

1. 接着我们创建一个“useParamete.xaml”的空白流程，之后将“Main.xaml”拖动到其中。



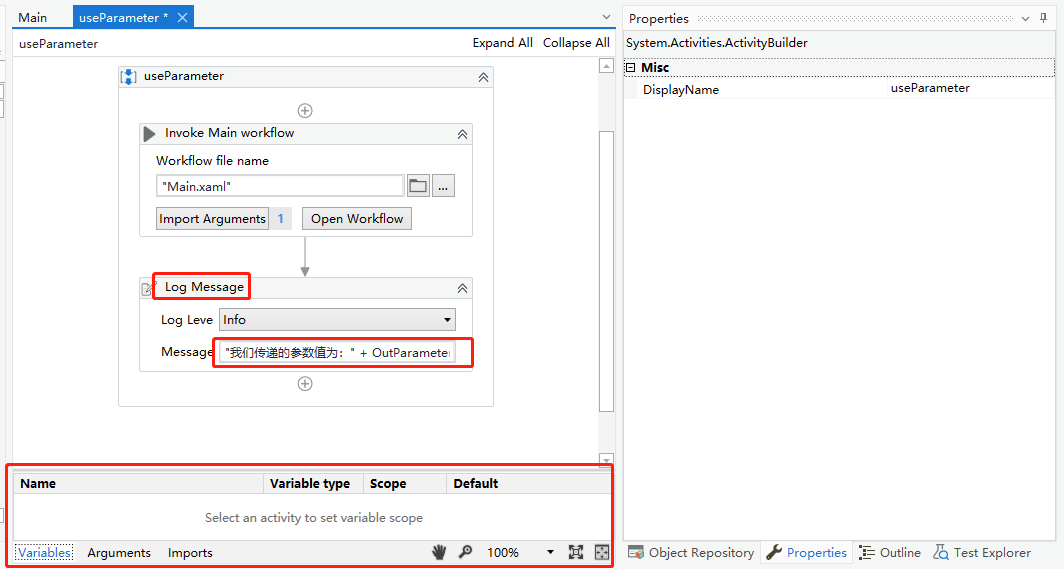
1. 拖动完毕之后，我们进行参数的导入，即上图中显示橙色按钮“0”。点击“Import Arguments”,并进行如下设置，并点击OK。



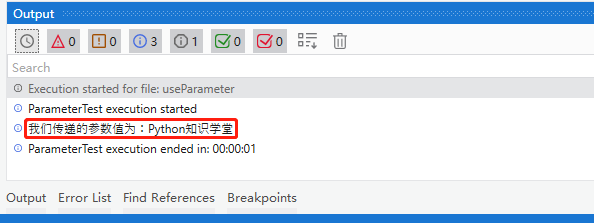
上述设置中，Value值为OutParameter为一个变量值。

1. 在此之后，我们使用活动“Log Message”来进行打印。其中Message栏的内容为：

"我们传递的参数值为：" + OutParameter



1. 接着我们运行，得出的结果如下：



可以肯出我们很好的完成了一个参数从一个流程传递到另外一个流程当中来了。

值得注意的是，我们在“useParamete.xaml”中并没有设置变量OutParameter，而直接进行了使用，这也是可以的。

**三、总结**

通过以上的学习，相比大家已经学会了UiPath中参数的相关知识。我们学会了如何进行参数的创建和使用，也学会了在UiPath中如何将参数的传递用于不同的流程之间。

以上就是本期的全部内容，下期文章我们再见。