# AutoTest Studio入门系列8：代码调试

AutoTest Studio内置一个强大的代码调试引擎，这在测试用例开发过程中非常有用。

代码调试方式多种多样，但总的来说，常用的就三个步骤，断点，监视变量，修改变量。本节以一个简单的demo为例介绍AutoTest Studio调试功能的使用。

新建一个名为“debugtest.py”的测试用例，输入以下内容。

from autotest import \*

a=1

b=2

c=3

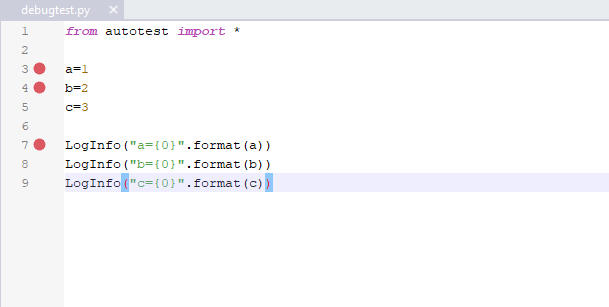
LogInfo("a={0}".format(a))

LogInfo("b={0}".format(b))

LogInfo("c={0}".format(c))

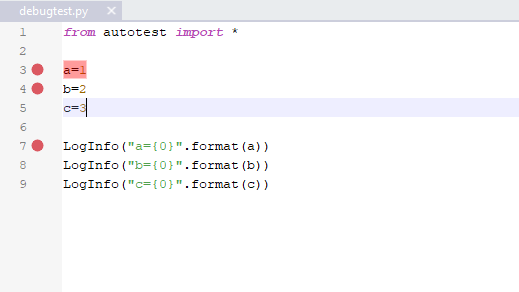
**断点**

在代码运行前，我们设置一些断点。



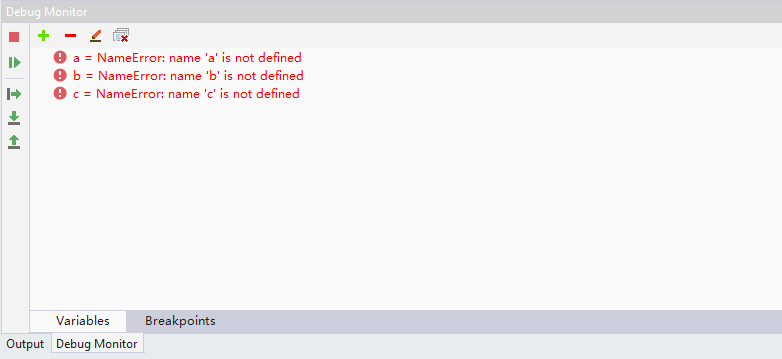
**运行代码**

以“Debug Run”方式运行代码到第一个断点处。

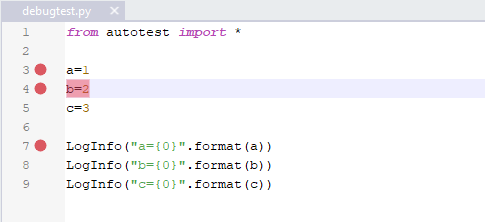


**监视变量**

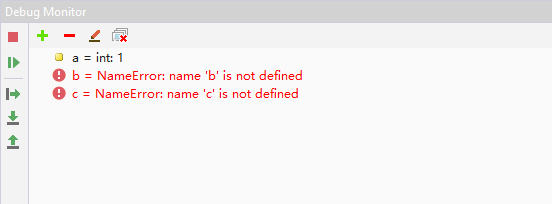
运行代码到第一个断点处时，我们新建三个监视变量分别为a，b，c，当然你也可以在代码运行前新建监视变量。



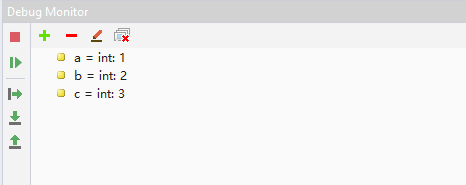
新建变量后，调试器会立即获取当前变量的值，但本示例中此时提示了“NameError”错误，这是因为代码还没有运行对a，b，c赋值操作。这里继续运行代码到第二个断点处。



在Debug Monitor中可以看到a已经有正确的值。

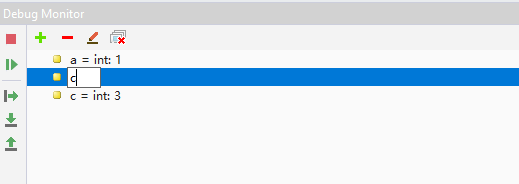


继续运行代码至下一个断点处，a，b，c完成赋值操作，Debug Monitor可以看到a，b，c的值。

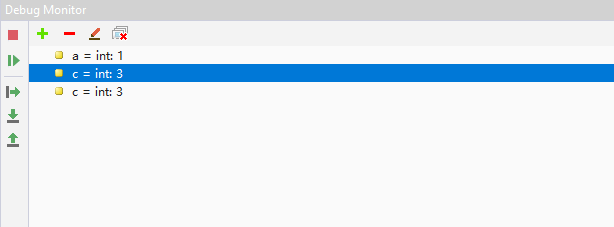


**修改变量**

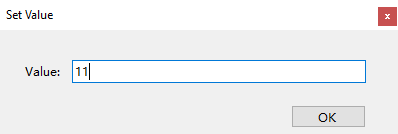
调试器允许用户在代码运行期间修改监视变量的名称和值，接着上面的示例，点击右键菜单“Edit Name”，在编辑状态下，将b的名称修改为c。



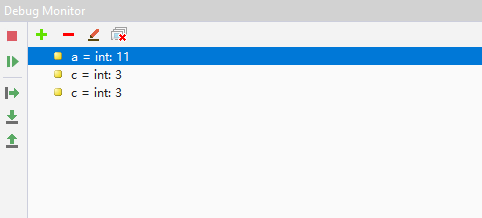
修改完成后，鼠标双击空白区域，让修改生效，生效后，调试器会立即获取新变量的值。



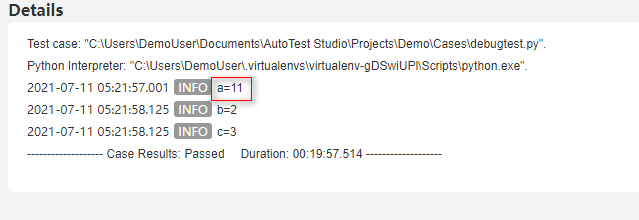
除了修改变量名称外，调试引擎也允许用户修改变量的值，选中a变量，点击右键菜单“Set Value”，设置a变量的值为“11”。



点击OK后，Debug Monitor显示，a的值已经变成“11“了。



为了验证修改的变量值是否生效，我们将代码运行至结束，查看日志打印。



从日志中可以看到，代码执行到“LogInfo("a={0}".format(a)) ”时已经是修改后的值了。

代码调试是开发过程中非常重要的一项技能，本文只是简单介绍了AutoTest Studio调试功能的基本使用，更复杂的使用技巧，大家可以在使用过程中自行探索。