# AutoTest Studio Getting started tutorials 8: Code Debugging

AutoTest Studio has a built-in powerful code debugging engine, which is very useful in the process of test case development.

There are many code debugging methods, but in general, the three commonly used steps are breakpoints, monitor variables, and modify variables. This section takes a simple demo as an example to introduce the use of AutoTest Studio's debugging function.

Create a new test case named "debugtest.py" and enter the following content.

from autotest import \*

a=1

b=2

c=3

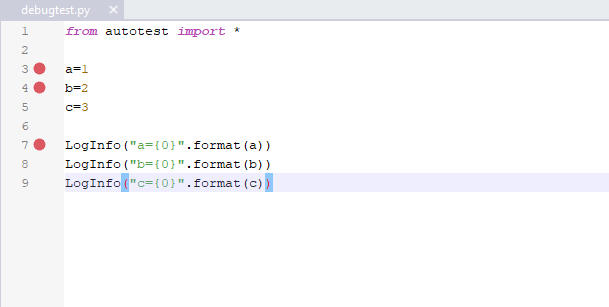
LogInfo("a={0}".format(a))

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

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

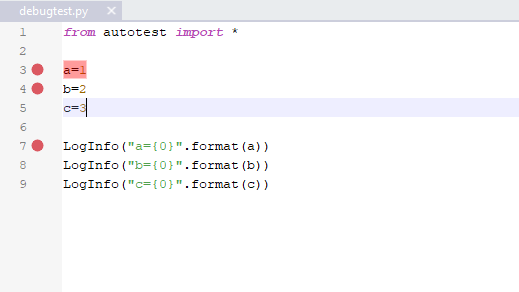
**Breakpoint**

Before the code runs, we set some breakpoints.



**Run the code**

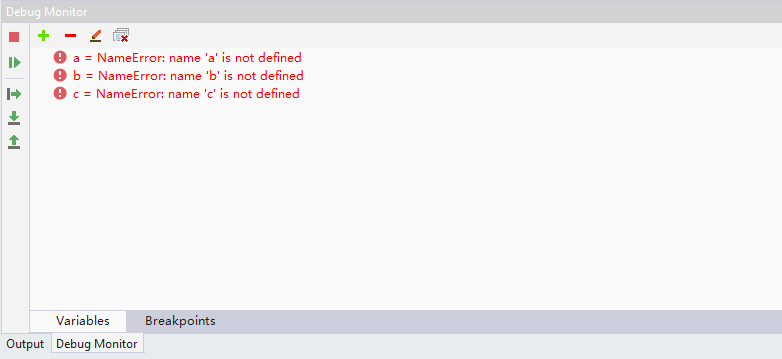
Run the code to the first breakpoint in "**Debug Run**" mode.



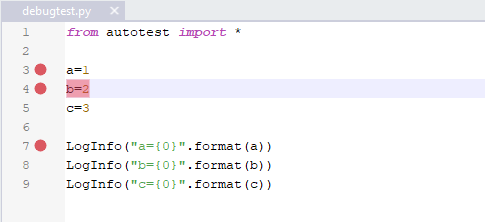
**Monitor variable**

When running the code to the first breakpoint, we create three new monitoring variables, namely a, b, and c. Of course, you can also create new monitoring variables before the code runs.

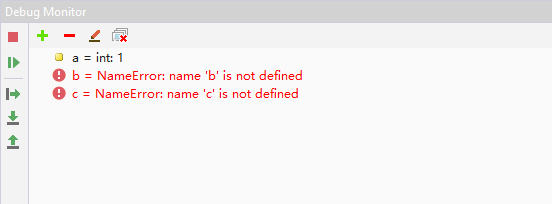
Place.



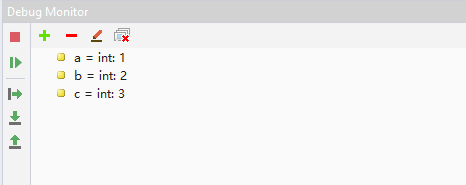
After the new variable is created, the debugger will immediately get the value of the current variable, but in this example, the "**NameError**" error is prompted at this time, because the code has not yet run the assignment operation to a, b, and c. Here continue to run the code to the second breakpoint.



In the Debug Monitor, you can see that a has the correct value.

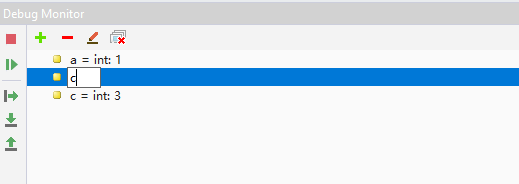


Continue to run the code to the next breakpoint, a, b, c complete the assignment operation, Debug Monitor can see the value of a, b, c.

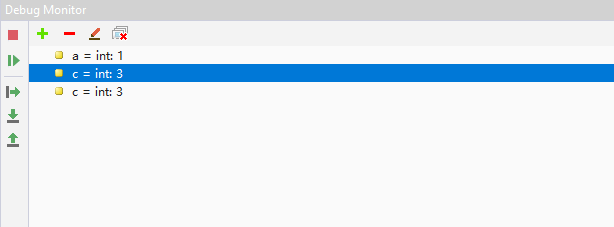


**Modify variables**

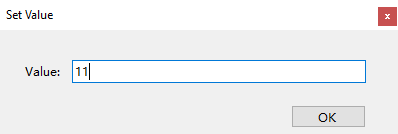
The debugger allows the user to modify the name and value of the monitored variable during the code execution. Following the example above, click the right-click menu "Edit Name", and in the editing state, change the name of b to c.



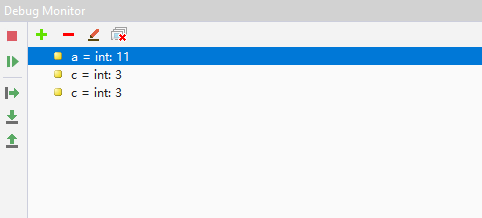
After the modification is completed, double-click the blank area to make the modification take effect. After it takes effect, the debugger will immediately obtain the value of the new variable.



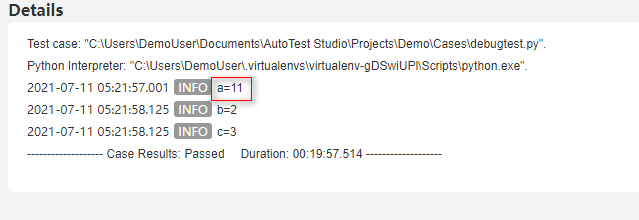
In addition to modifying the variable name, the debugging engine also allows the user to modify the value of the variable, select the a variable, click the right-click menu "Set Value", and set the value of the a variable to "11".



After clicking **OK**, Debug Monitor shows that the value of a has become "**11**".



In order to verify whether the modified variable value takes effect, we run the code to the end and check the log print.



As you can see from the log, the code is already at the modified value when it reaches "LogInfo("a={0}".format(a))".

Code debugging is a very important skill in the development process. This article only briefly introduces the basic use of AutoTest Studio's debugging functions. You can explore more complex skills during use.