

Debugging

Nun lernen wir das Gegenmittel zu fehlern

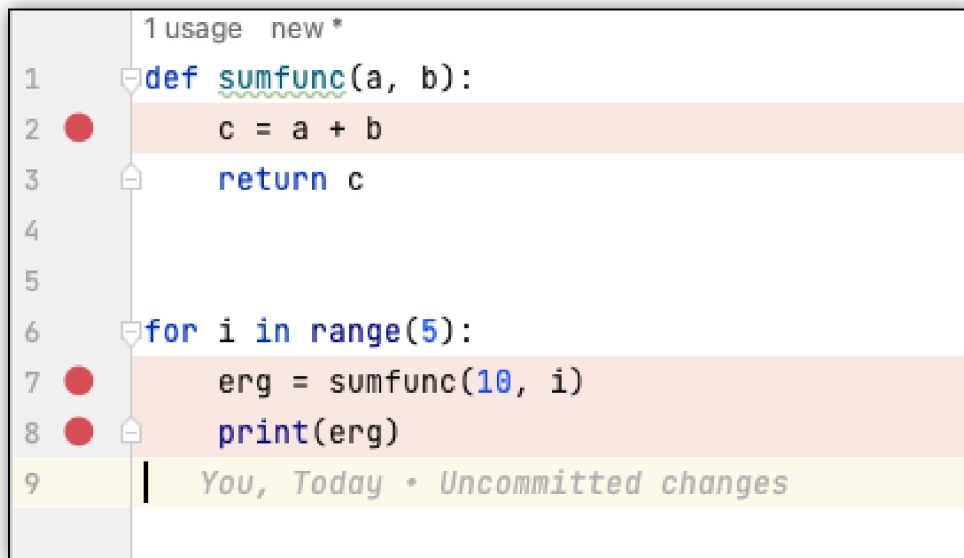
Kurzer recap

Wir erinnern uns nochmals kurz an den logischen Fehler des letzten Kapitels. Diesem möchten wir nun genauer auf den Grund gehen.

```
1 # 16-single-step.py
2
3 def sumfunc(a, b):
4     c = a + b
5     return c
6
7
8 for i in range(5):
9     erg = sumfunc(10, i)
10    print(erg)
```

Breakpoints

Ein wichtiges Element für das Debugging sind Breakpoints. Diese **unterbrechen** das Programm in seiner Ausführung und erlauben ein genaues **untersuchen**.



```
1 usage new *
2 def sumfunc(a, b):
3     c = a + b
4     return c
5
6 for i in range(5):
7     erg = sumfunc(10, i)
8     print(erg)
9 | You, Today • Uncommitted changes
```

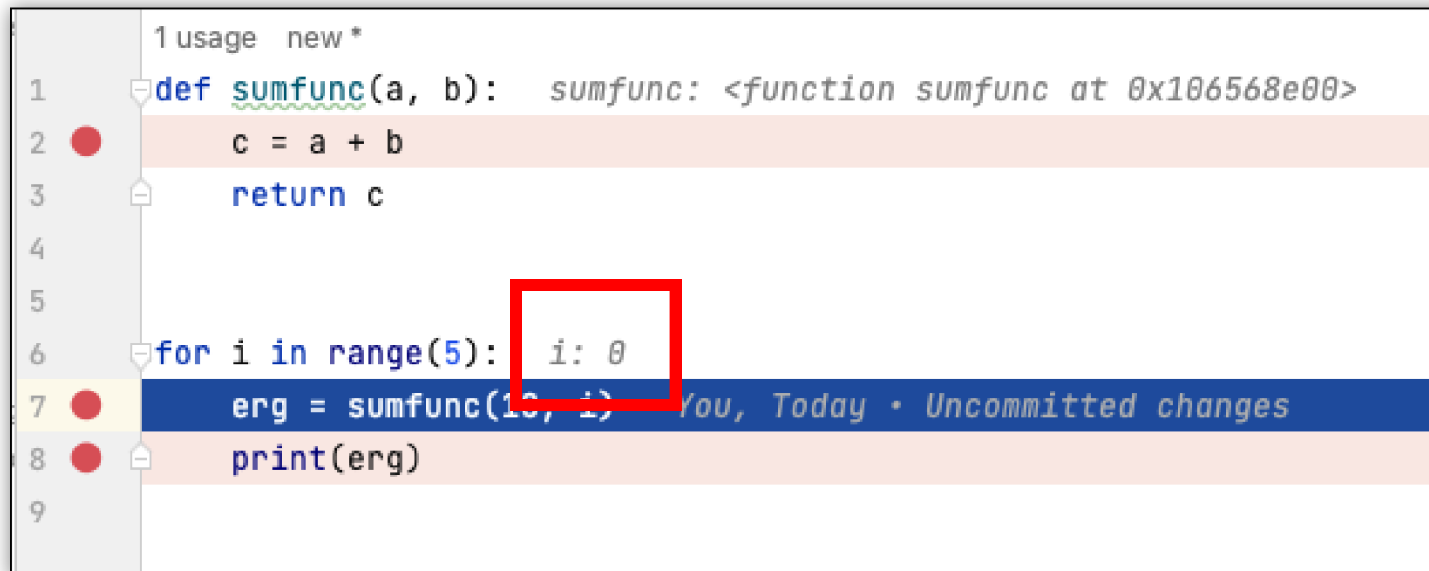
The image shows a code editor with Python code. Line 2 has a red dot breakpoint. Line 7 has a red dot breakpoint. Line 8 has a red dot breakpoint. The status bar at the bottom indicates 'You, Today • Uncommitted changes'.



Als Nächstes muss das Programm diesmal im Debug-Modus ausgeführt werden.

Debug Modus

Wir sehen nun auf den ersten Blick eine wichtige Änderung. Es werden uns sofort in Grau die Variablen zur Laufzeit angezeigt:



```
1 usage new *
2 def sumfunc(a, b):  sumfunc: <function sumfunc at 0x106568e00>
3     c = a + b
4     return c
5
6 for i in range(5):  i: 0
7     erg = sumfunc(10, i)
8     print(erg)
```

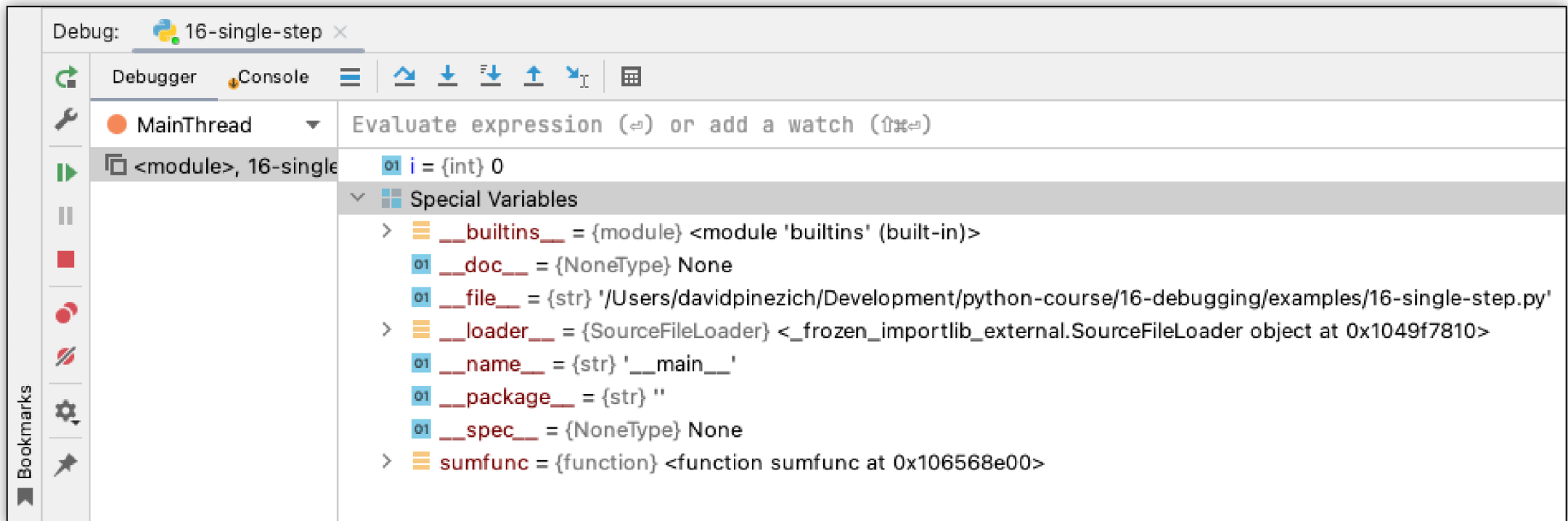
Debug Modus

Anhand der blauen (starken) Untermalung sehen wir auch das das Programm bei Linie 7 gestoppt hat.

```
1 usage new *
2 def sumfunc(a, b):  sumfunc: <function sumfunc at 0x106568e00>
3     c = a + b
4     return c
5
6 for i in range(5):  i: 0
7     erg = sumfunc(10, i)  You, Today • Uncommitted changes
8     print(erg)
9
```

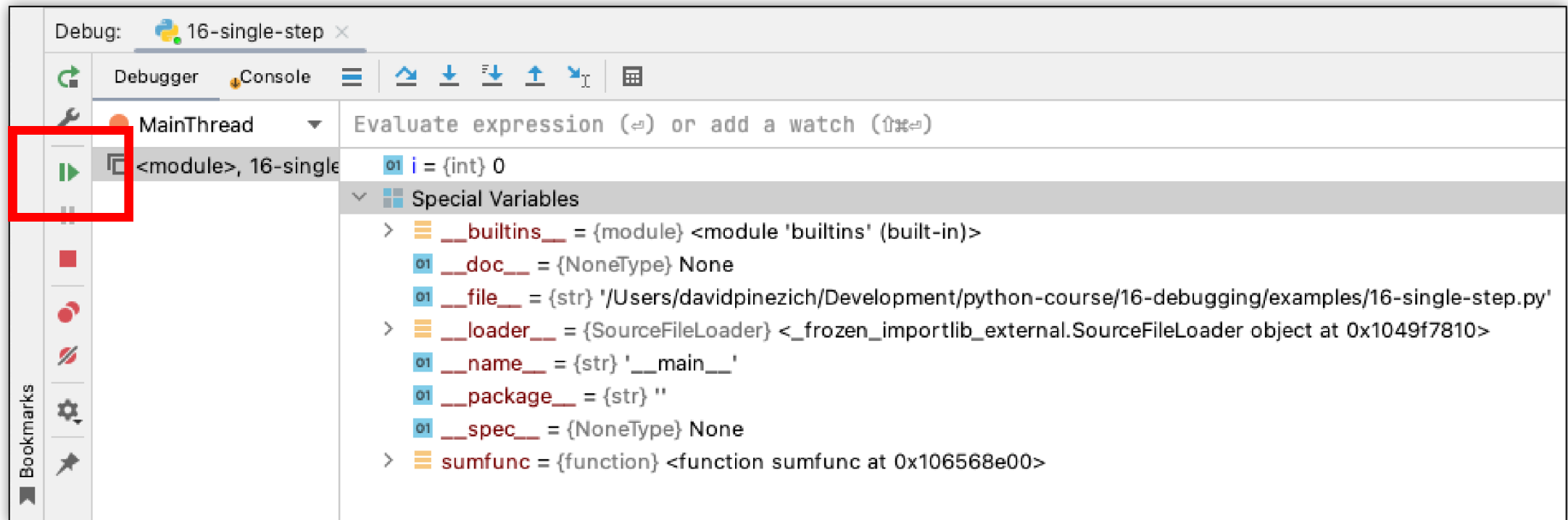
Debug Modus

Im unteren Teil sehen wir nun einerseits die Laufzeit variable `i` und auch alle Steuermethoden.



Debug Modus

Versuchen Sie nun mal auf den Pfeil zu drücken (rot markiert) und zu sehen was passiert:

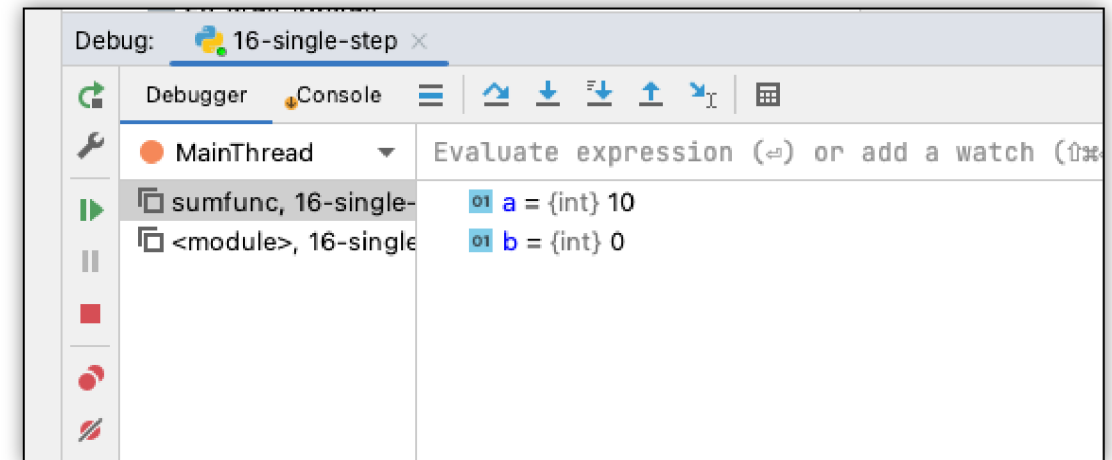


Debug Modus

Wir sehen sofort, dass das Programm bis zum nächsten Breakpoint gesprungen ist. Es werden neu definierte Variablen in Grau angezeigt.

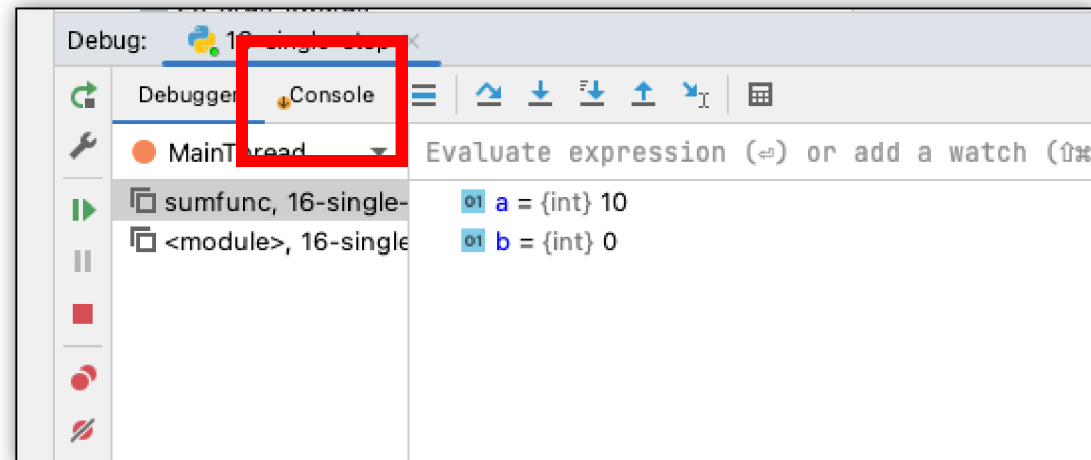
```
1 usage new *
2 def sumfunc(a, b):  a: 10  b: 0
3   c = a + b
4   return c
5
6 for i in range(5):
7   erg = sumfunc(10, i)
8   print(erg)
9
```

You, Today • Uncommitted changes



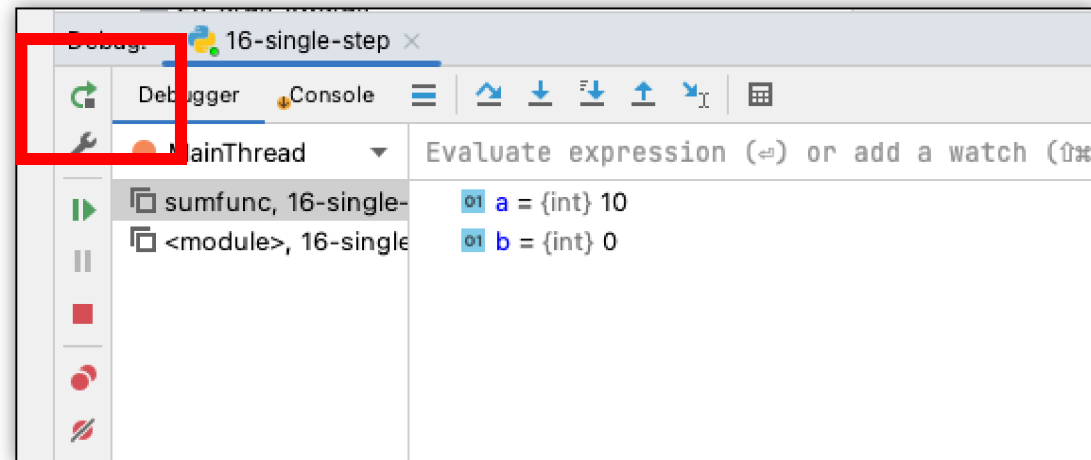
Debug Modus

Wir können jederzeit auch auf die Konsolenausgabe wechseln:



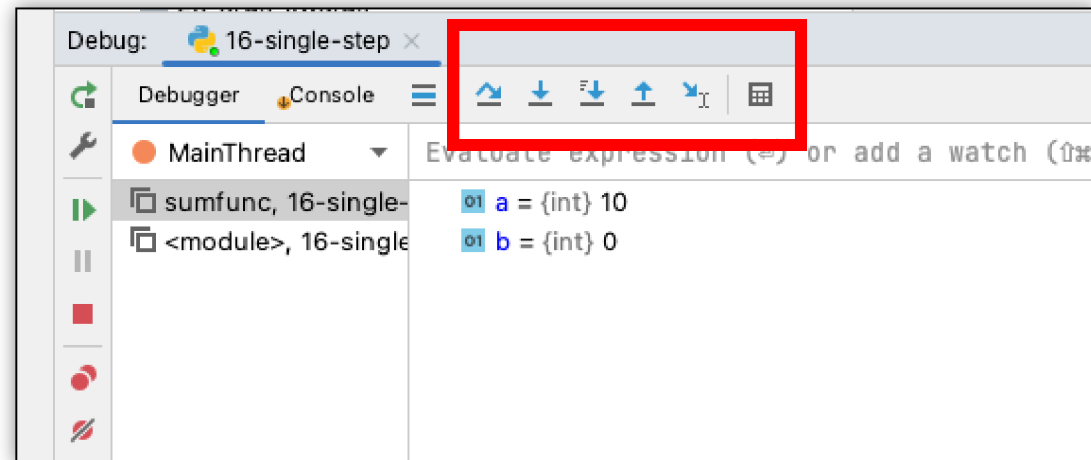
Debug Modus

Das Programm von vorne beginnen:



Debug Modus

In aufwändigeren Programmen in einzelne Funktionen gehen oder dies Überspringen (dasselbe kann auch mit genügend Breakpoints erreicht werden)



Ende

Das war alles für dieses Kapitel
