Debugging your code

Mu can execute your code in steps. You can do that by pressing the `Debug` button with the Bug icon next to it.

Try to type in the following code in Mu and press `Debug` :

```
my_variable = 0
while True:
    my_variable += 1
```

Notice that a few things happen: You get not just one, but two new windows. You also get a number of new buttons!

The window to your right lists all the variables that exists right now.

- `__file__` is the current file that you're running
- `__name__` is the name of the current program

The new buttons can help you navigate your code step by step. We'll focus on the `Step Over` button now.

Try to press the `Step Over` and see what happens.

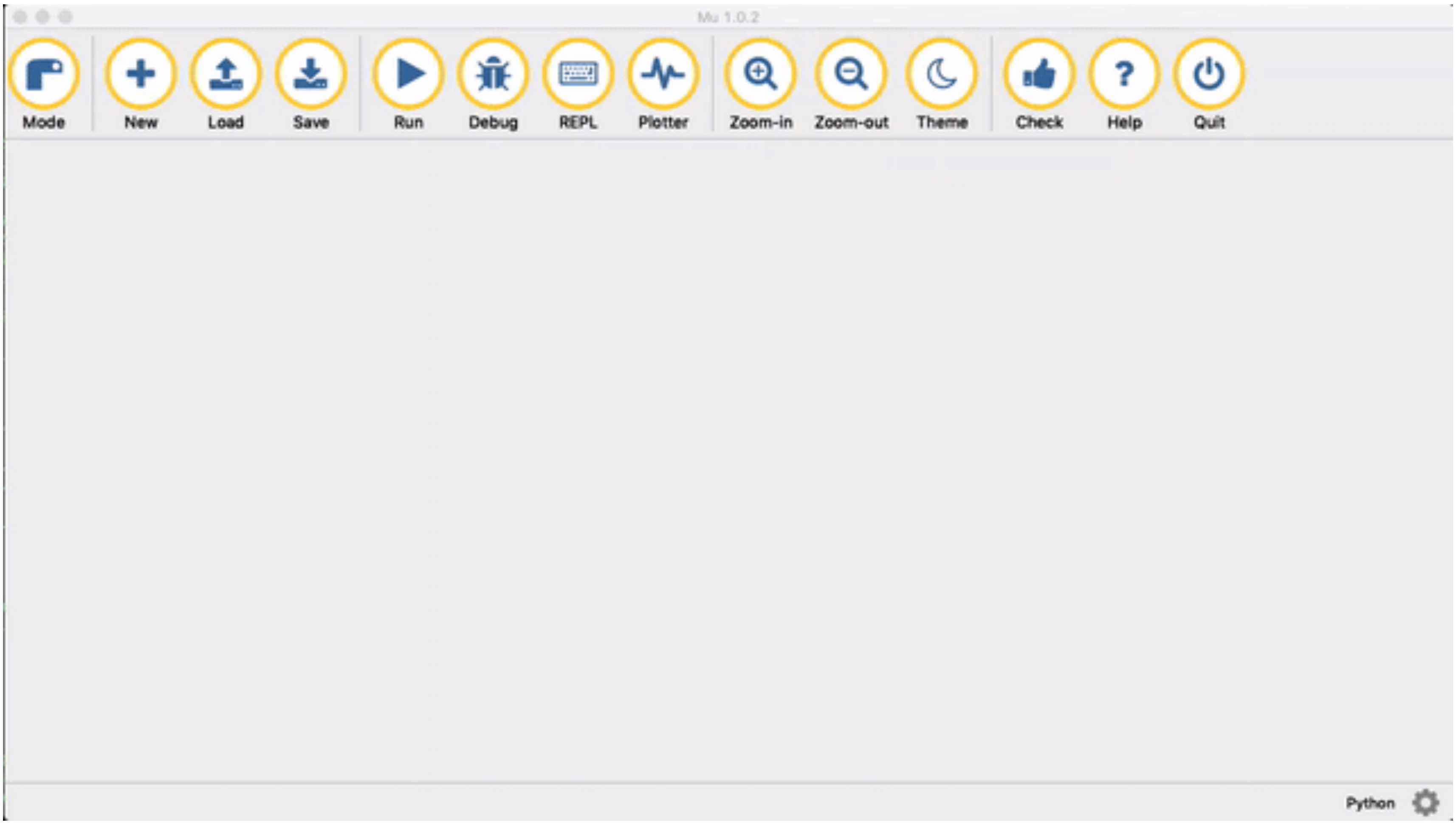
Press it again and again until you get bored.

Breakpoints

Find line number 3 (`my_variable += 1`) and click in the grey area to the left of the line.

A red dot should appear. This is called a **break point**.

Whenever you press the `Continue` button in the debug mode, your program will continue to run, until it reaches the break point.



Working with the debugger

The debugger can help you explain *exactly* what happens, **in your pace**!

Let's try to debug this turtle program:

```
from turtle import forward, left


moves = [100, 10, 100]
for move in moves:
    forward(move)
    left(90)
```

Debugging your code

Mu can execute your code in steps. You can do that by pressing the `Debug` button with the Bug icon next to it.

Try to type in the following code in Mu and press `Debug` :

```
my_variable = 0
while True:
    my_variable += 1
```

Notice that a few things happen: You get not just one, but two new windows. You also get a number of new buttons!

The window to your right lists all the variables that exists right now.

- `__file__` is the current file that you're running
- `__name__` is the name of the current program

The new buttons can help you navigate your code step by step. We'll focus on the `Step Over` button now.

Try to press the `Step Over` and see what happens.

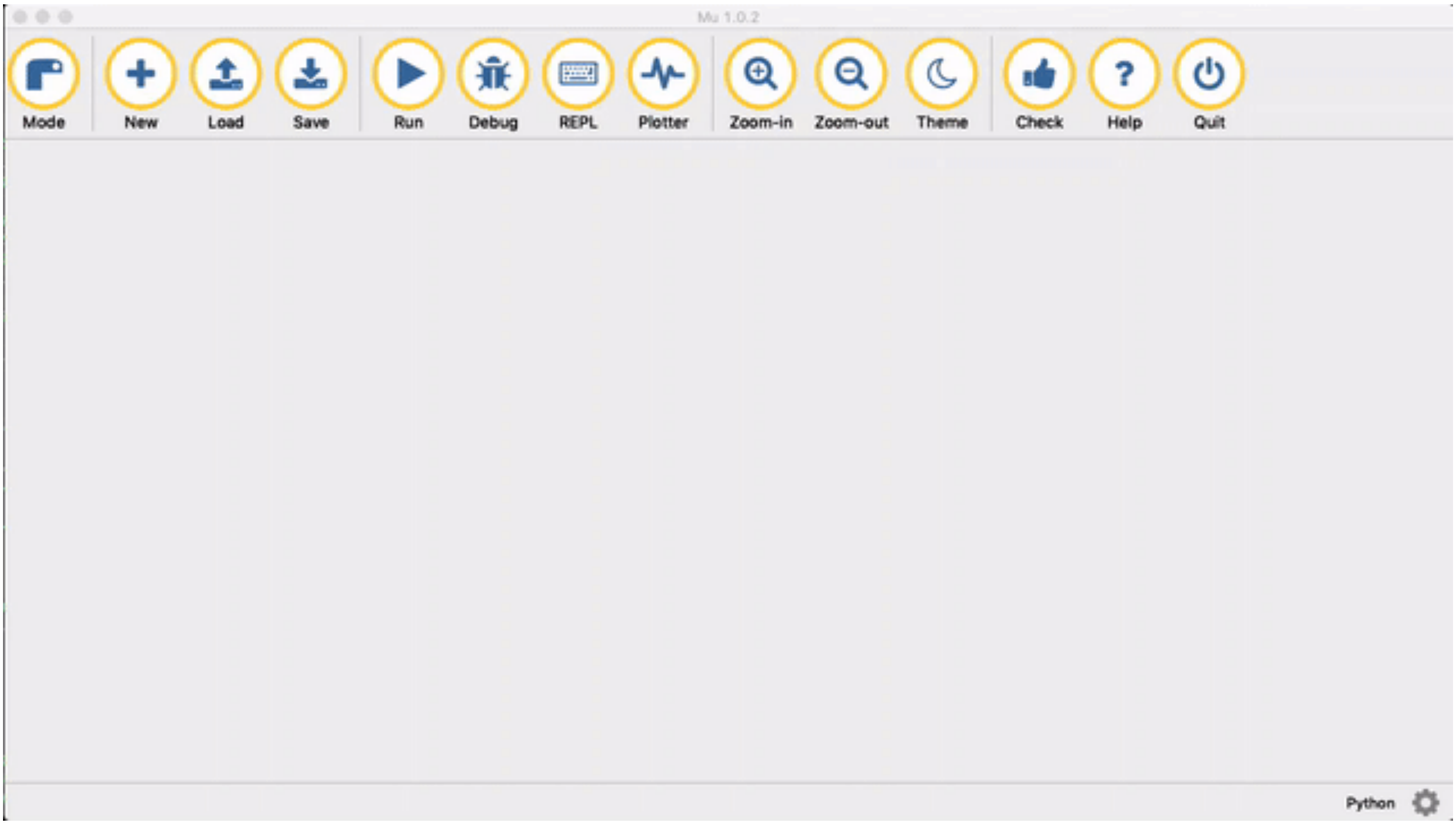
Press it again and again until you get bored.

Breakpoints

Find line number 3 (`my_variable += 1`) and click in the grey area to the left of the line.

A red dot should appear. This is called a **break point.**

Whenever you press the `Continue` button in the debug mode, your program will continue to run, until it reaches the break point.



Working with the debugger

The debugger can help you explain *exactly* what happens, **in your pace**!

Let's try to debug this turtle program:

```
from turtle import forward, left


moves = [100, 10, 100]
for move in moves:
    forward(move)
    left(90)
```