INTRODUCING `friendly`

PyCon Education Summit - May 12, 2021

ANDRÉ ROBERGE

He/him

https://github.com/aroberge/friendly
https://github.com/aroberge/talks

WHO AM I?

Python hobbyist since 2004

- RUR-PLE
- Reeborg's World
- Crunchy
- docpicture
- AvantPy
- **friendly**
- *etc.*

WHO ARE YOU?

Python educators or mentors.

WHAT IS `friendly`?

A tool to help your students with Python tracebacks.

DIFFERENT PROGRAMMING ENVIRONMENTS

```python
# terminal, VS Code, etc.
from friendly         import ...

# Special cases
from friendly.idle    import ...
from friendly.mu      import ...
from friendly.ipython import ...
from friendly.jupyter import ...
```

OBSERVATION

Running a program with
IDLE, Mu, Thonny, ...

```
python   -i    my_program.py
```

```python
# my_program.py

answer = 42
print('Hello World!')
```

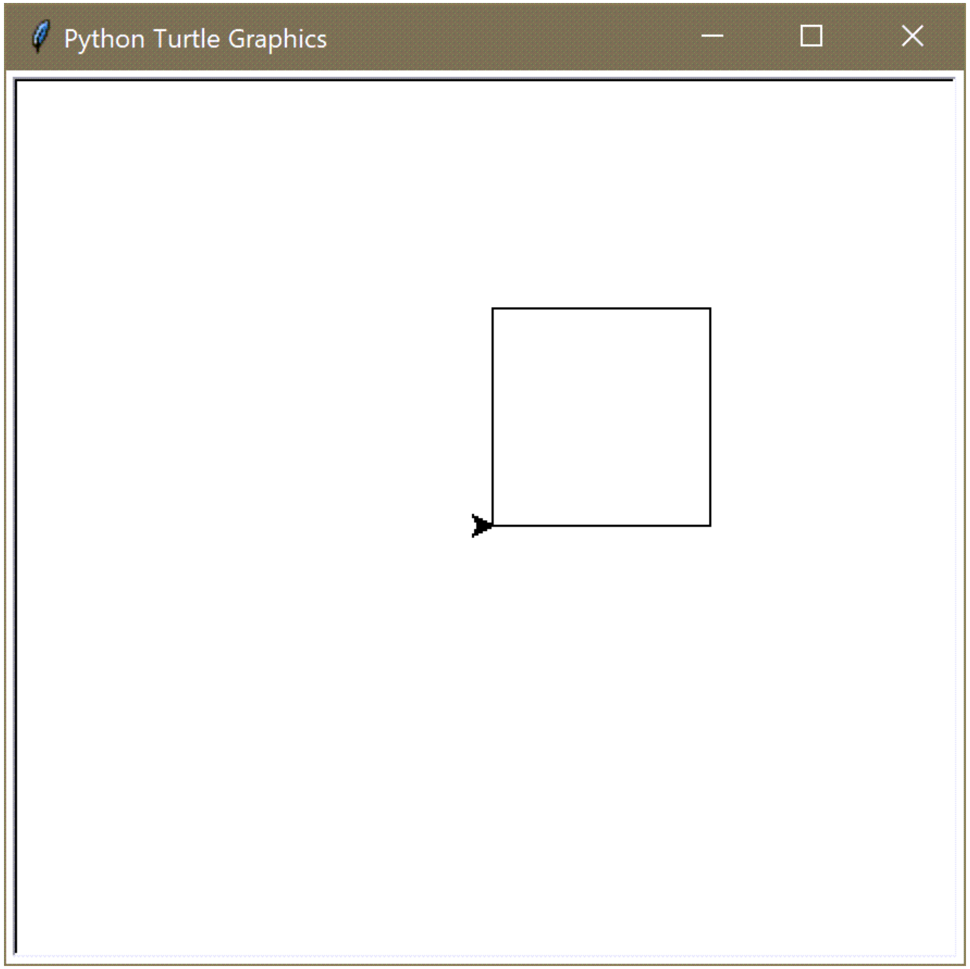
"Run" my_program.py

```
Hello World!
>>> answer
42
>>>
```

DEMO 1

```python
import turtle as t

for i in range(4):
    t.forward(100)
    t.left(90)
```

DEMO 1: EXPECTED



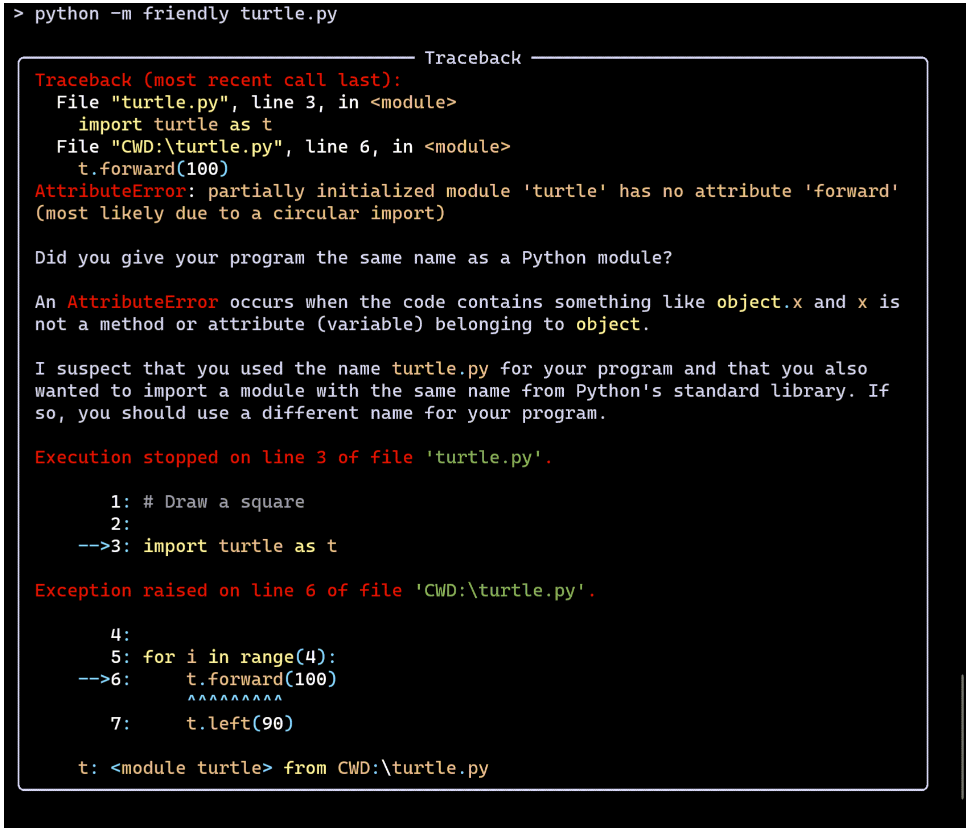
If the demo does not work ...

DEMO 1: RESULT

```
> python turtle.py

Traceback (most recent call last):
  File "turtle.py", line 3, in <module>
    import turtle as t
  File "...\turtle.py", line 6, in <module>
    t.forward(100)
AttributeError: partially initialized module
 'turtle' has no attribute 'forward'
(most likely due to a circular import)
```

DEMO 1: RUNNING WITH `friendly`

```
python -m friendly turtle.py
```



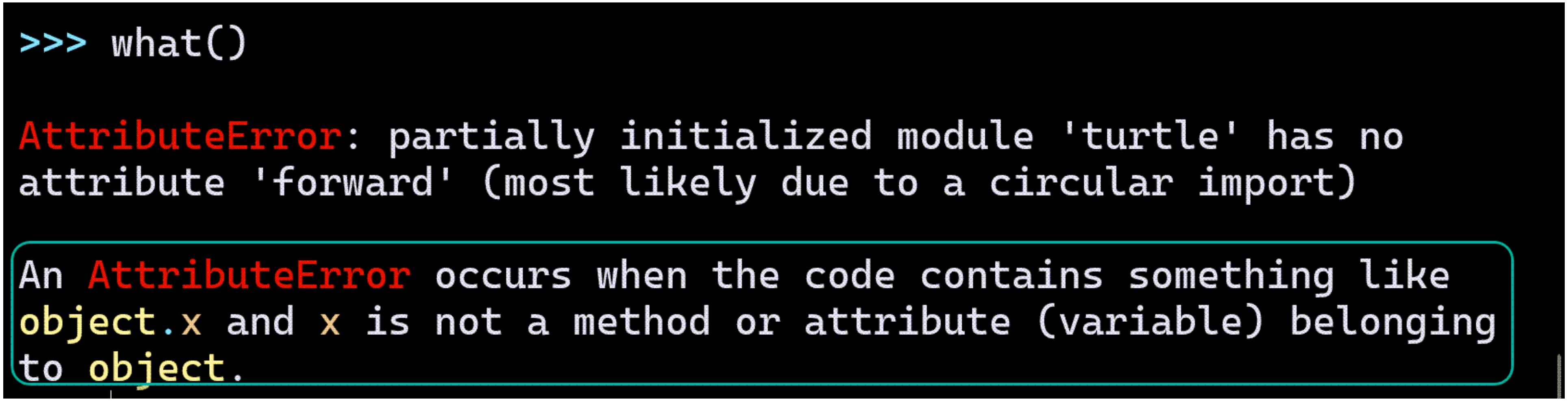
```
> python -m friendly turtle.py

                              Traceback
Traceback (most recent call last):
  File "turtle.py", line 3, in <module>
    import turtle as t
  File "CWD:\turtle.py", line 6, in <module>
    t.forward(100)
AttributeError: partially initialized module 'turtle' has no attribute 'forward'
(most likely due to a circular import)

Did you give your program the same name as a Python module?

An AttributeError occurs when the code contains something like object.x and x is
not a method or attribute (variable) belonging to object.

I suspect that you used the name turtle.py for your program and that you also
wanted to import a module with the same name from Python's standard library. If
so, you should use a different name for your program.

Execution stopped on line 3 of file 'turtle.py'.

       1: # Draw a square
       2:
    -->3: import turtle as t

Exception raised on line 6 of file 'CWD:\turtle.py'.

       4:
       5: for i in range(4):
    -->6:     t.forward(100)
              ^^^^^^^^^
       7:     t.left(90)

    t: <module turtle> from CWD:\turtle.py
```

DEMO 1: USING FRIENDLY

```
python -im friendly turtle.py
...

Friendly Console ...

>>>
```

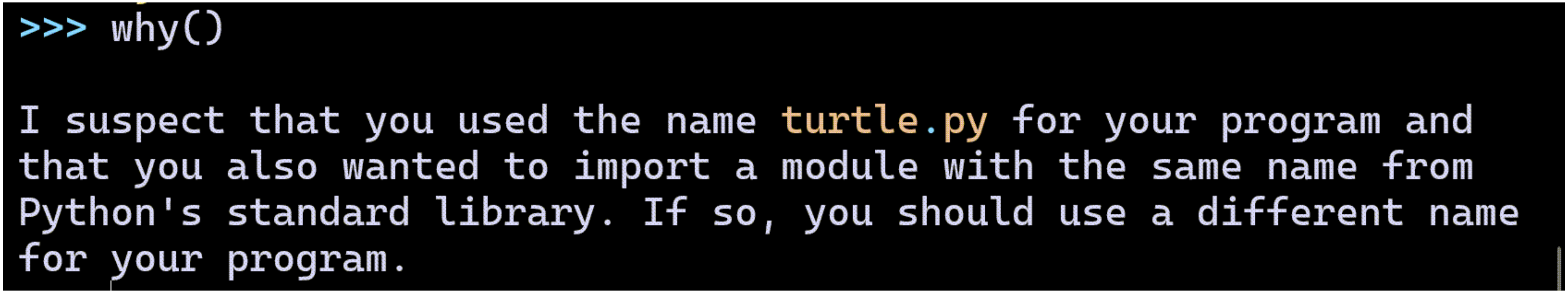
DEMO 1: TRACEBACK AND `hint()`

DEMO 1: what()



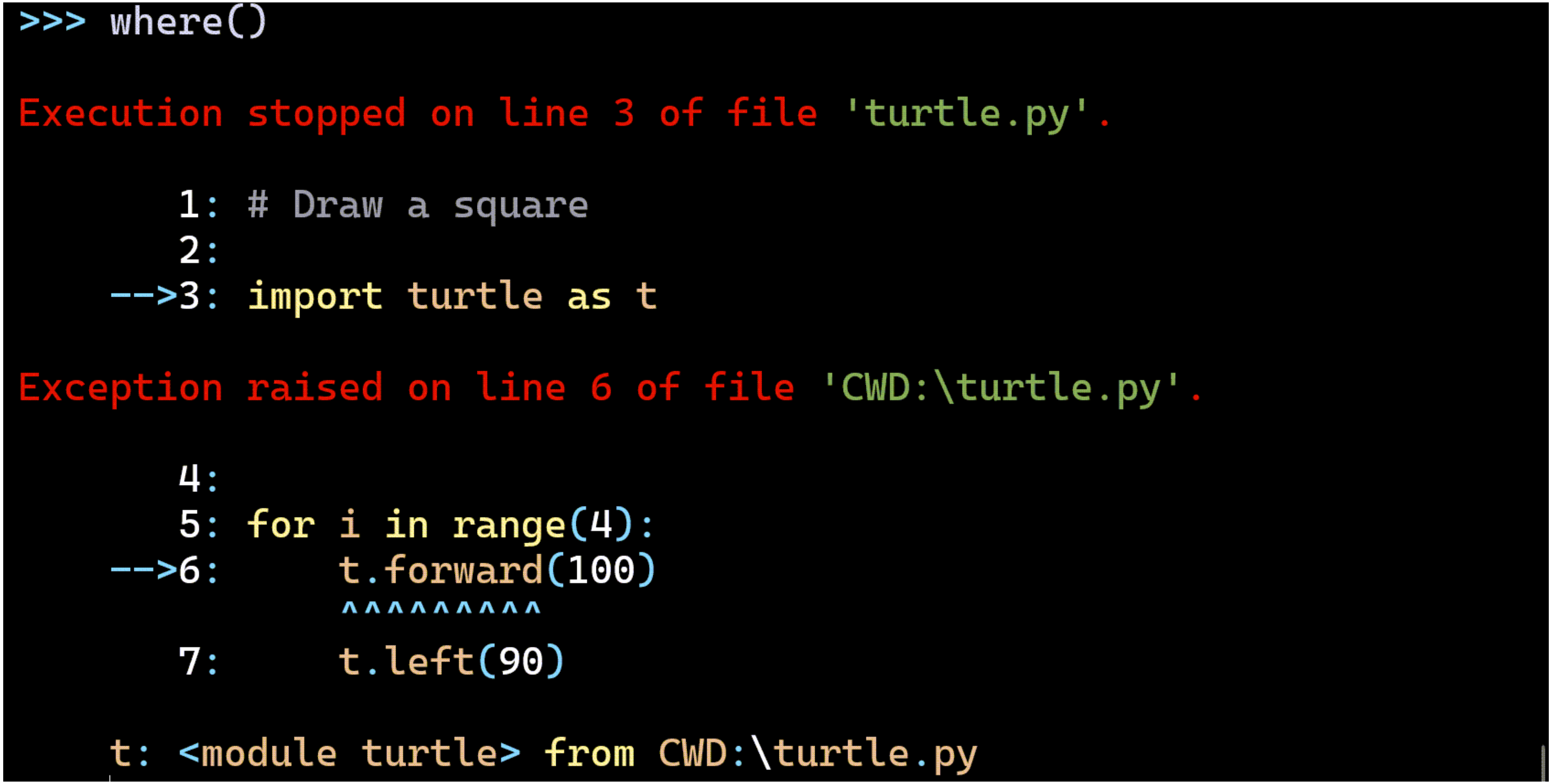
```
>>> what()

AttributeError: partially initialized module 'turtle' has no
attribute 'forward' (most likely due to a circular import)

An AttributeError occurs when the code contains something like
object.x and x is not a method or attribute (variable) belonging
to object.
```

DEMO 1: why()



```
>>> why()

I suspect that you used the name turtle.py for your program and
that you also wanted to import a module with the same name from
Python's standard library. If so, you should use a different name
for your program.
```

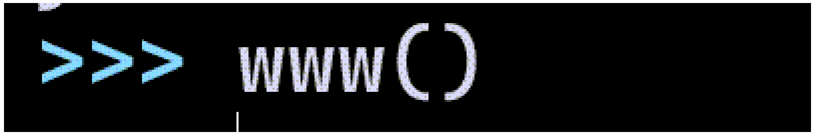
DEMO 1: where()



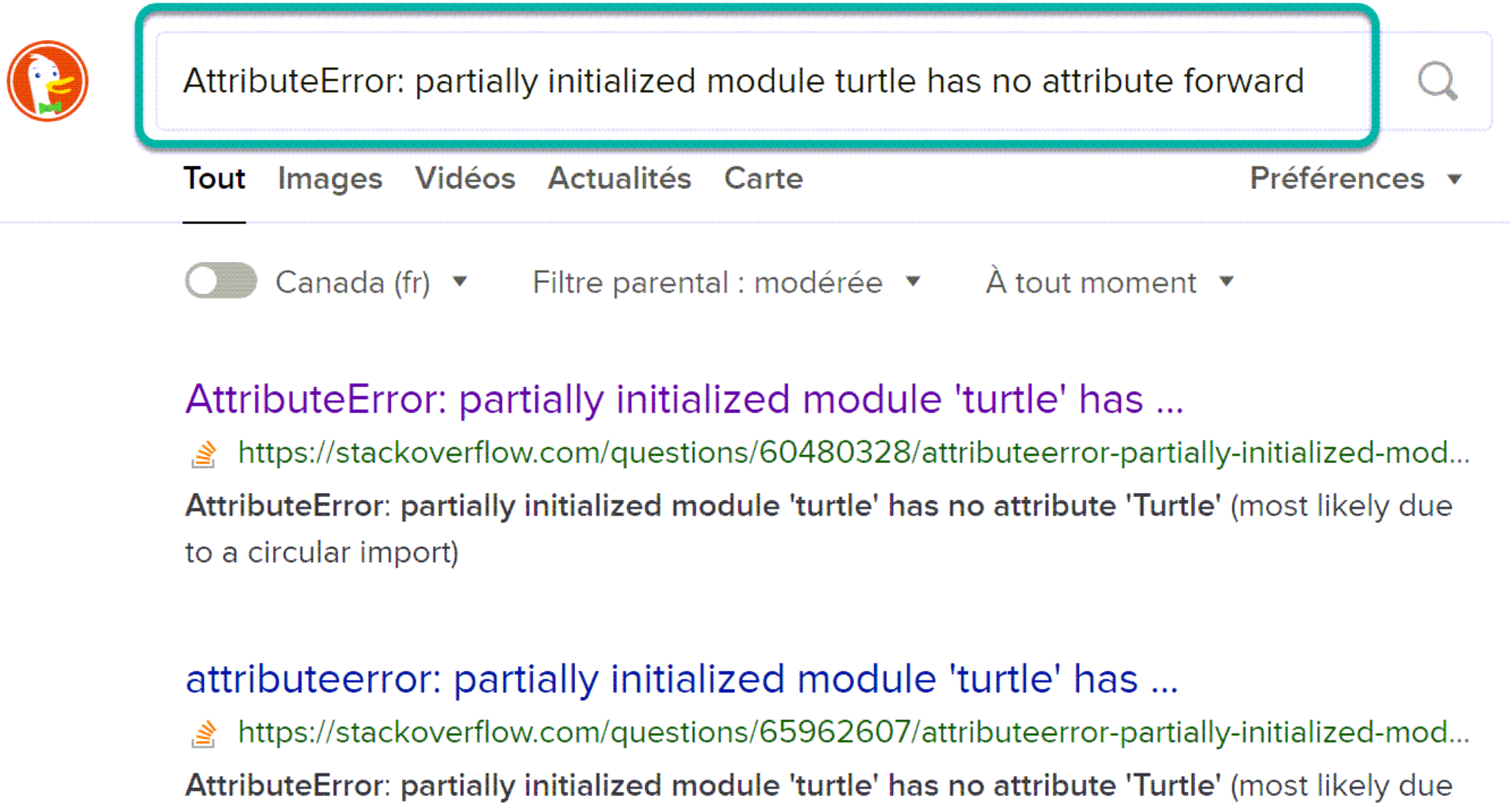
```
>>> where()

Execution stopped on line 3 of file 'turtle.py'.

       1: # Draw a square
       2:
    -->3: import turtle as t

Exception raised on line 6 of file 'CWD:\turtle.py'.

       4:
       5: for i in range(4):
    -->6:     t.forward(100)
              ^^^^^^^^^
       7:     t.left(90)

    t: <module turtle> from CWD:\turtle.py
```

Only one or two frames are shown.

DEMO 1: `www()`



```
>>> www()
```



AttributeError: partially initialized module turtle has no attribute forward

AttributeError: partially initialized module 'turtle' has ...

https://stackoverflow.com/questions/60480328/attributeerror-partially-initialized-mod...

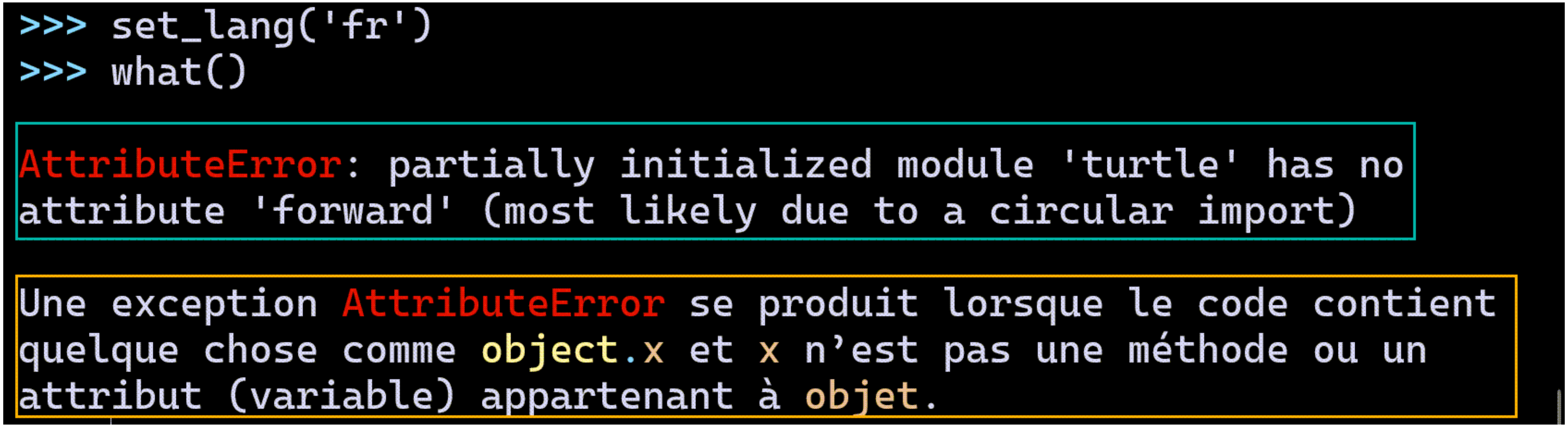
AttributeError: partially initialized module 'turtle' has no attribute 'Turtle' (most likely due to a circular import)

attributeerror: partially initialized module 'turtle' has ...

https://stackoverflow.com/questions/65962607/attributeerror-partially-initialized-mod...

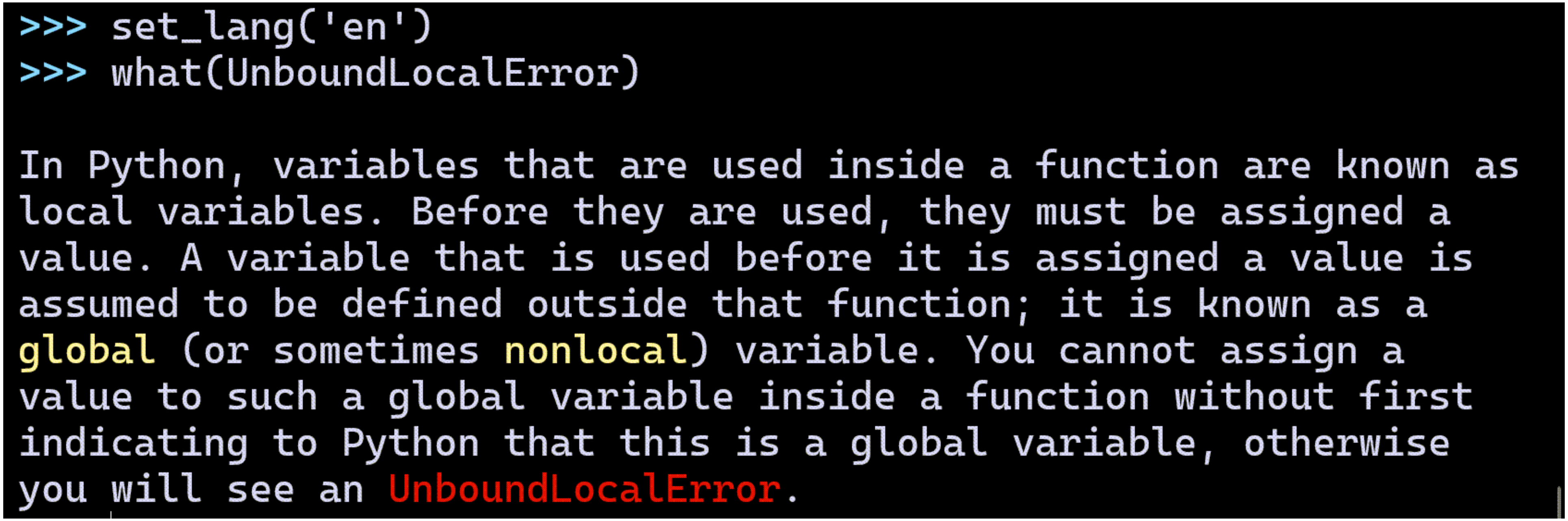
AttributeError: partially initialized module 'turtle' has no attribute 'Turtle' (most likely due

DEMO 1: what() IN FRENCH



```
>>> set_lang('fr')
>>> what()

AttributeError: partially initialized module 'turtle' has no
attribute 'forward' (most likely due to a circular import)

Une exception AttributeError se produit lorsque le code contient
quelque chose comme object.x et x n’est pas une méthode ou un
attribut (variable) appartenant à objet.
```

DEMO 1: what(...)



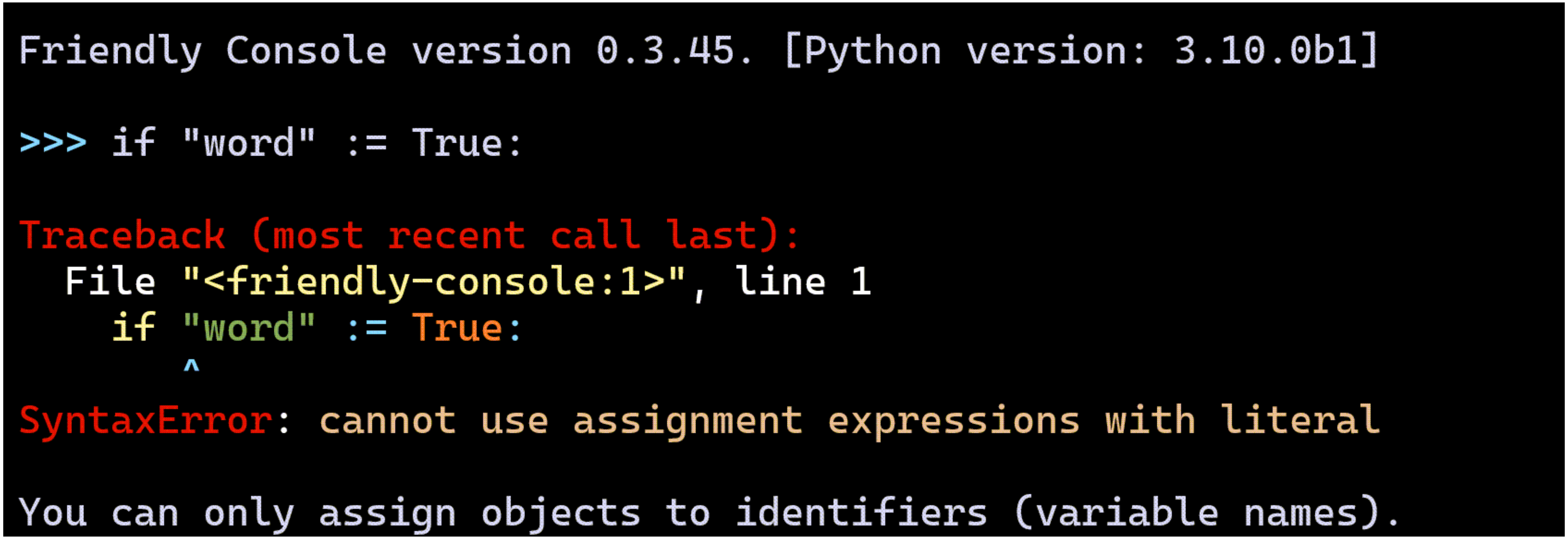
```
>>> set_lang('en')
>>> what(UnboundLocalError)

In Python, variables that are used inside a function are known as
local variables. Before they are used, they must be assigned a
value. A variable that is used before it is assigned a value is
assumed to be defined outside that function; it is known as a
global (or sometimes nonlocal) variable. You cannot assign a
value to such a global variable inside a function without first
indicating to Python that this is a global variable, otherwise
you will see an UnboundLocalError.
```

"DEMO 1": SUMMARY

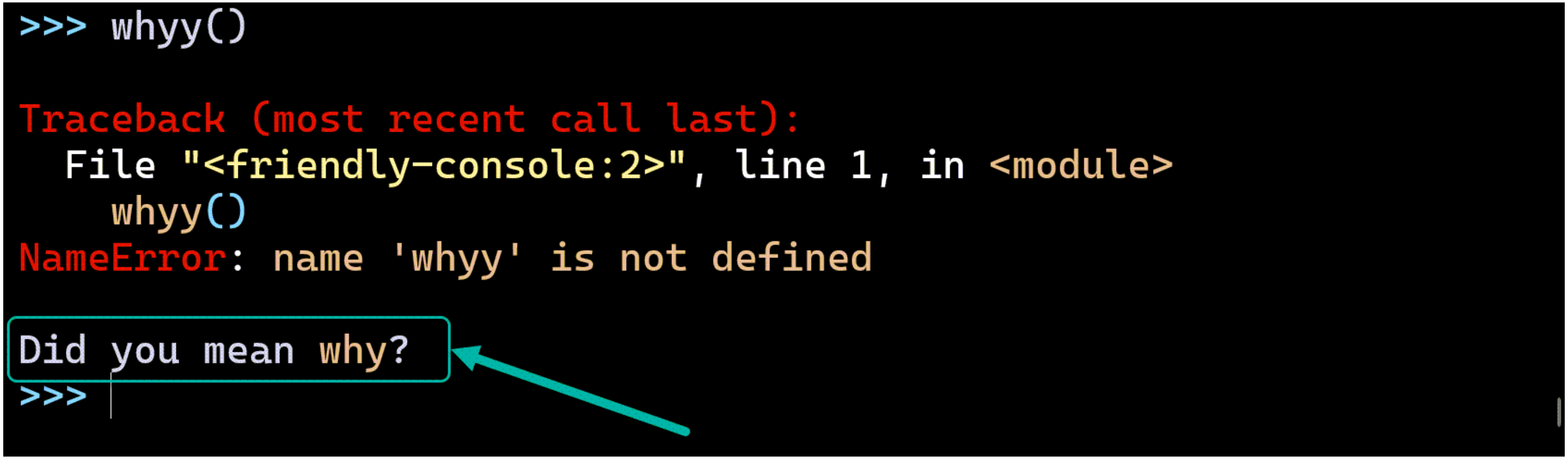
We can use **friendly** to ask questions and obtain answers helping us understand what caused a given traceback.

TYPOS?

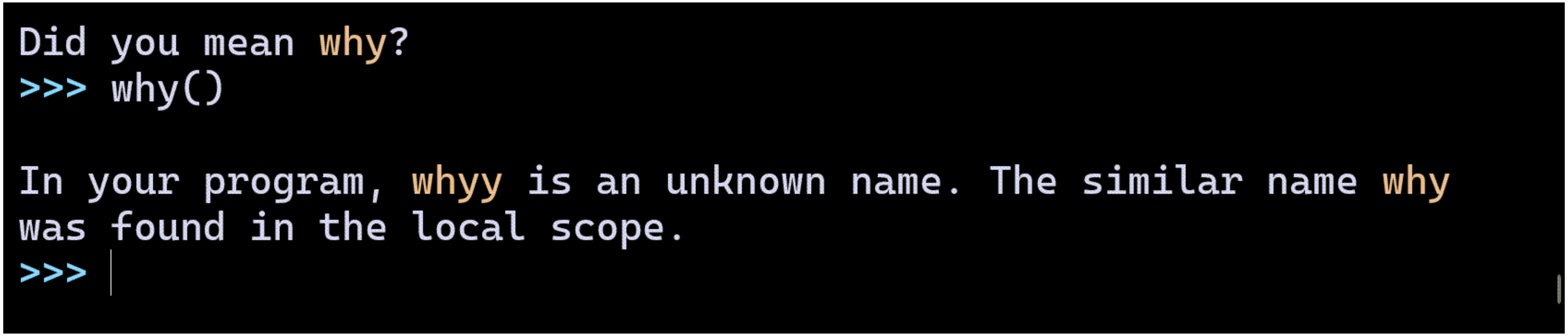
Can a REPL be friendly?



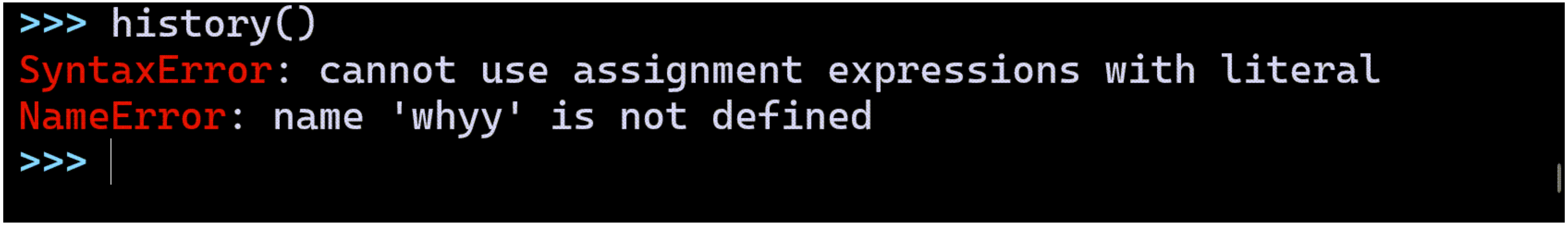
```
Friendly Console version 0.3.45. [Python version: 3.10.0b1]

>>> if "word" := True:

Traceback (most recent call last):
  File "<friendly-console:1>", line 1
    if "word" := True:
       ^
SyntaxError: cannot use assignment expressions with literal

You can only assign objects to identifiers (variable names).
```



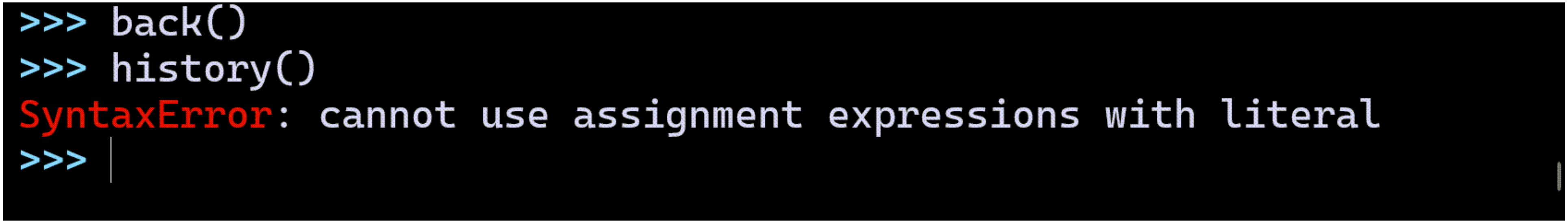
```
>>> whyy()

Traceback (most recent call last):
  File "<friendly-console:2>", line 1, in <module>
    whyy()
NameError: name 'whyy' is not defined

Did you mean why?
>>> 
```



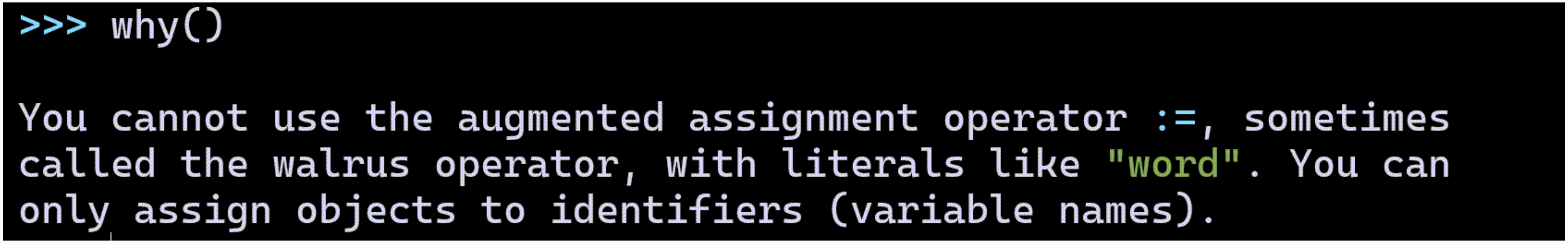
```
Did you mean why?
>>> why()

In your program, whyy is an unknown name. The similar name why
was found in the local scope.
>>> 
```



```
>>> history()
SyntaxError: cannot use assignment expressions with literal
NameError: name 'whyy' is not defined
>>> 
```

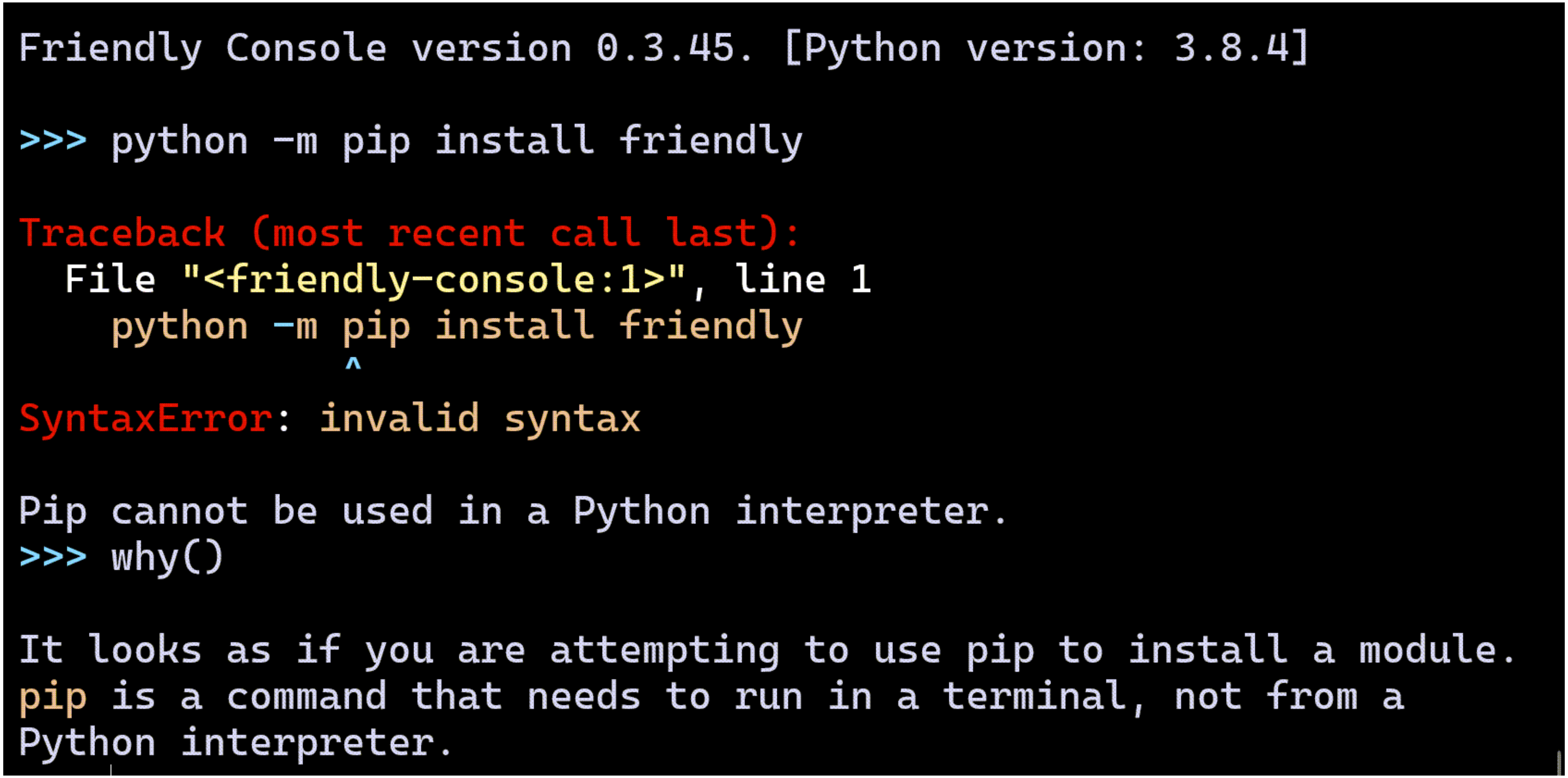


```
>>> back()
>>> history()
SyntaxError: cannot use assignment expressions with literal
>>> 
```

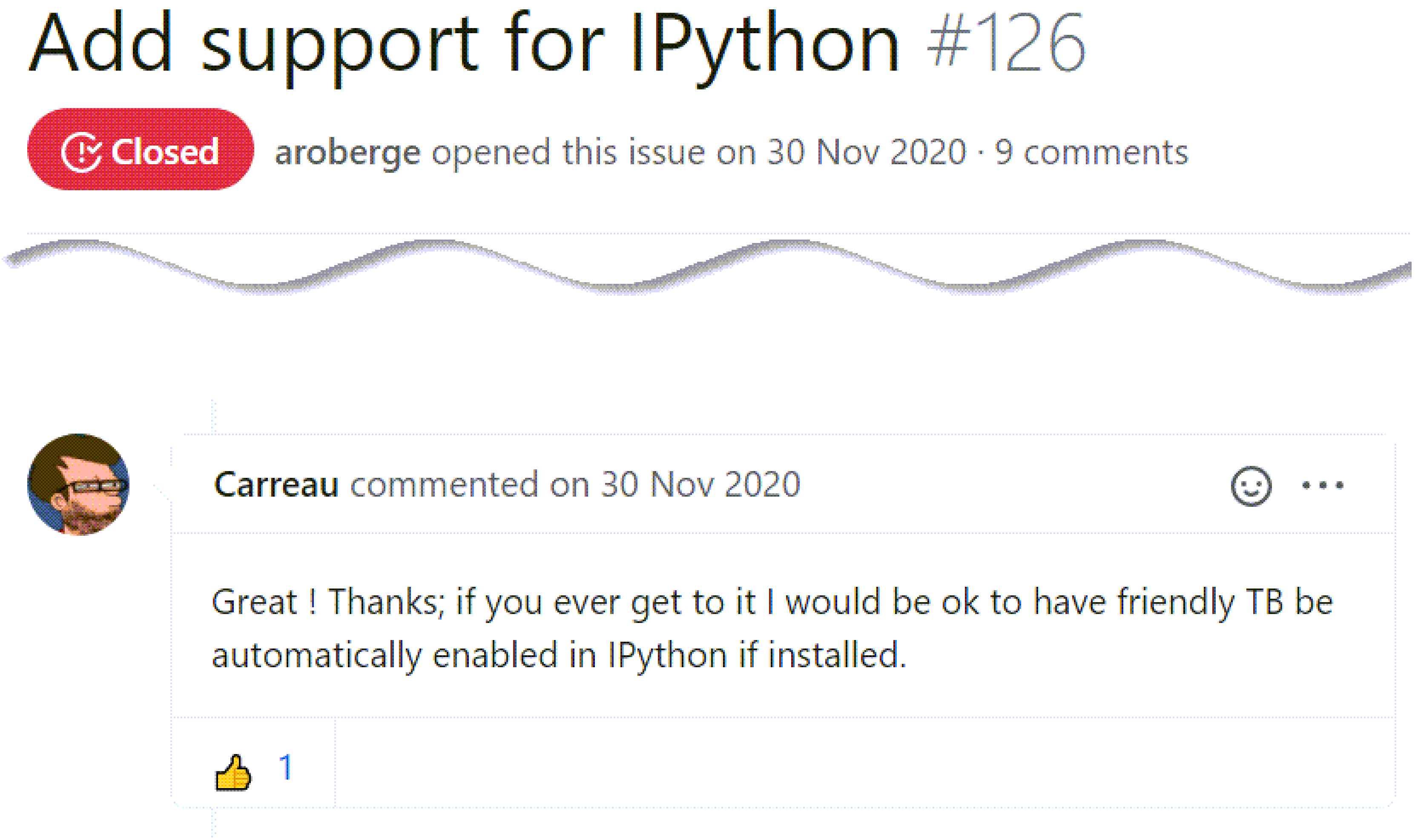


```
>>> why()

You cannot use the augmented assignment operator :=, sometimes
called the walrus operator, with literals like "word". You can
only assign objects to identifiers (variable names).
```

HOW TO INSTALL friendly?



```
Friendly Console version 0.3.45. [Python version: 3.8.4]

>>> python -m pip install friendly

Traceback (most recent call last):
  File "<friendly-console:1>", line 1
    python -m pip install friendly
              ^
SyntaxError: invalid syntax

Pip cannot be used in a Python interpreter.
>>> why()

It looks as if you are attempting to use pip to install a module.
pip is a command that needs to run in a terminal, not from a
Python interpreter.
```



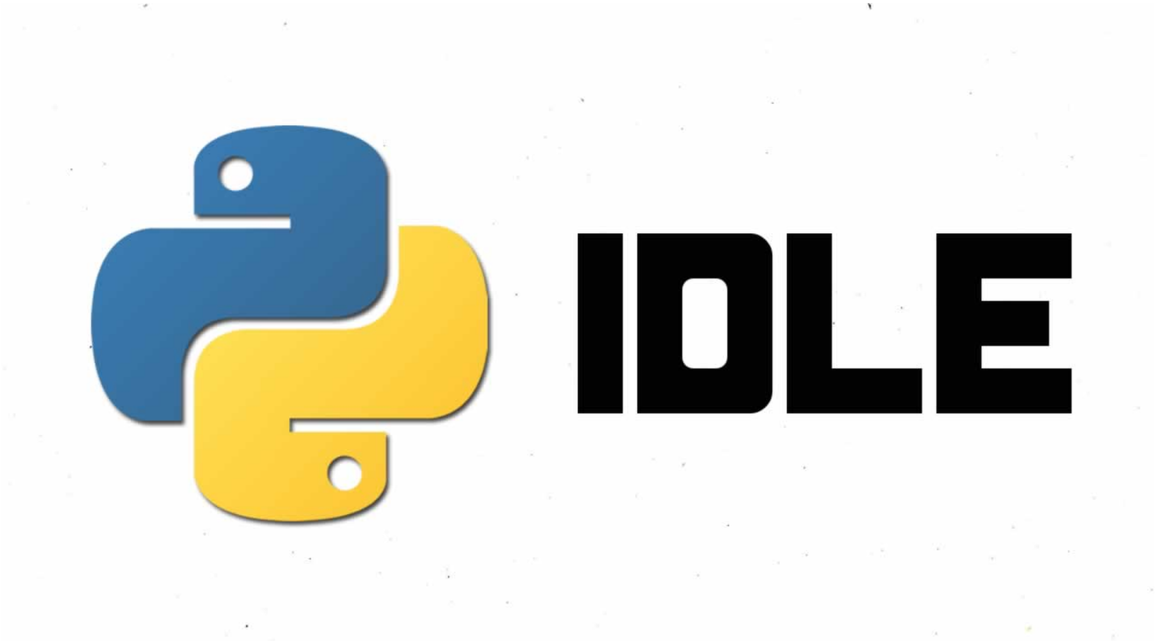
Add support for IPython #126

aroberge opened this issue on 30 Nov 2020 · 9 comments

Carreau commented on 30 Nov 2020

Great ! Thanks; if you ever get to it I would be ok to have friendly TB be automatically enabled in IPython if installed.

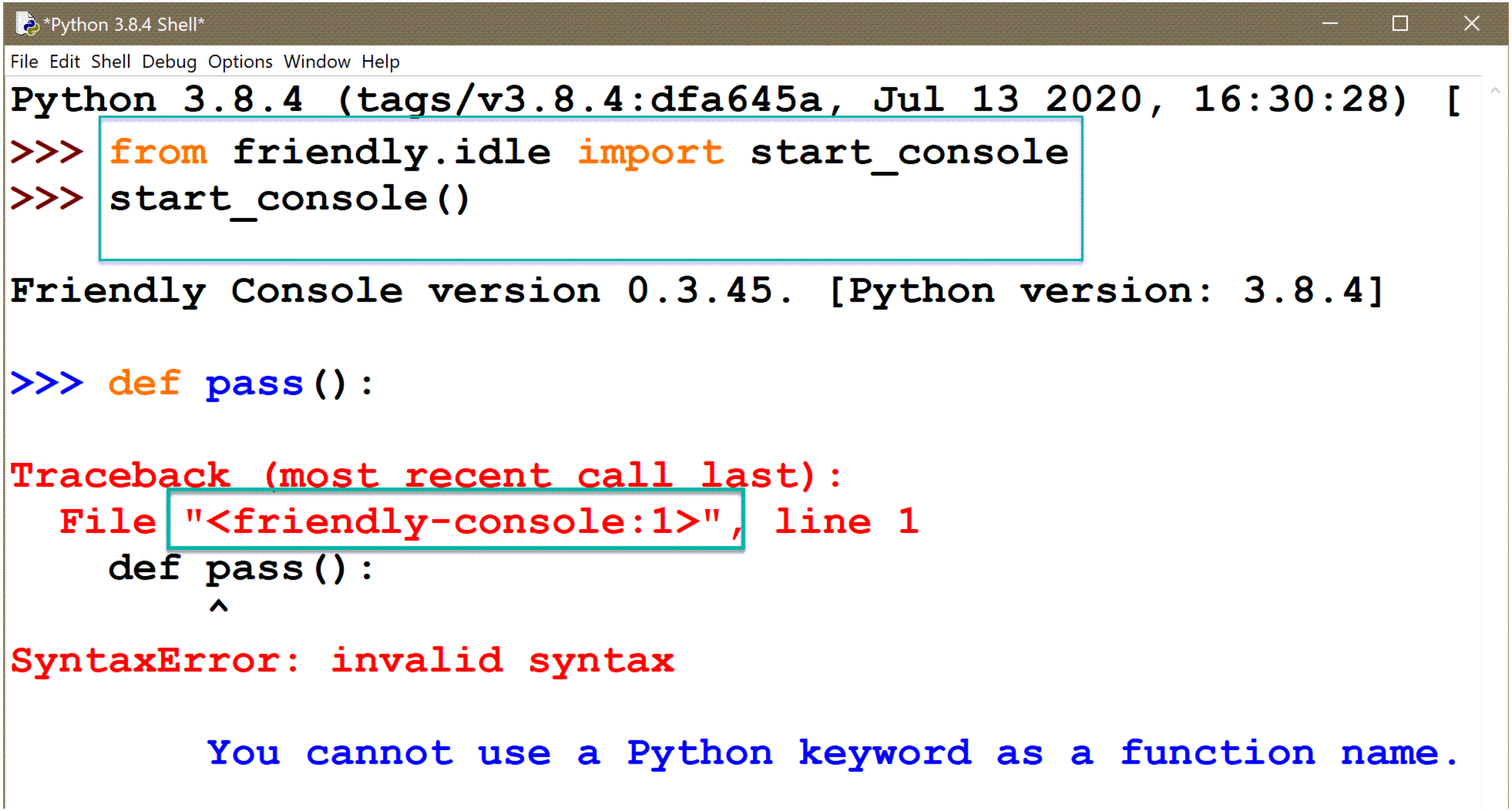
Next: demos with IDLE, Mu (, and Jupyter Lab?)



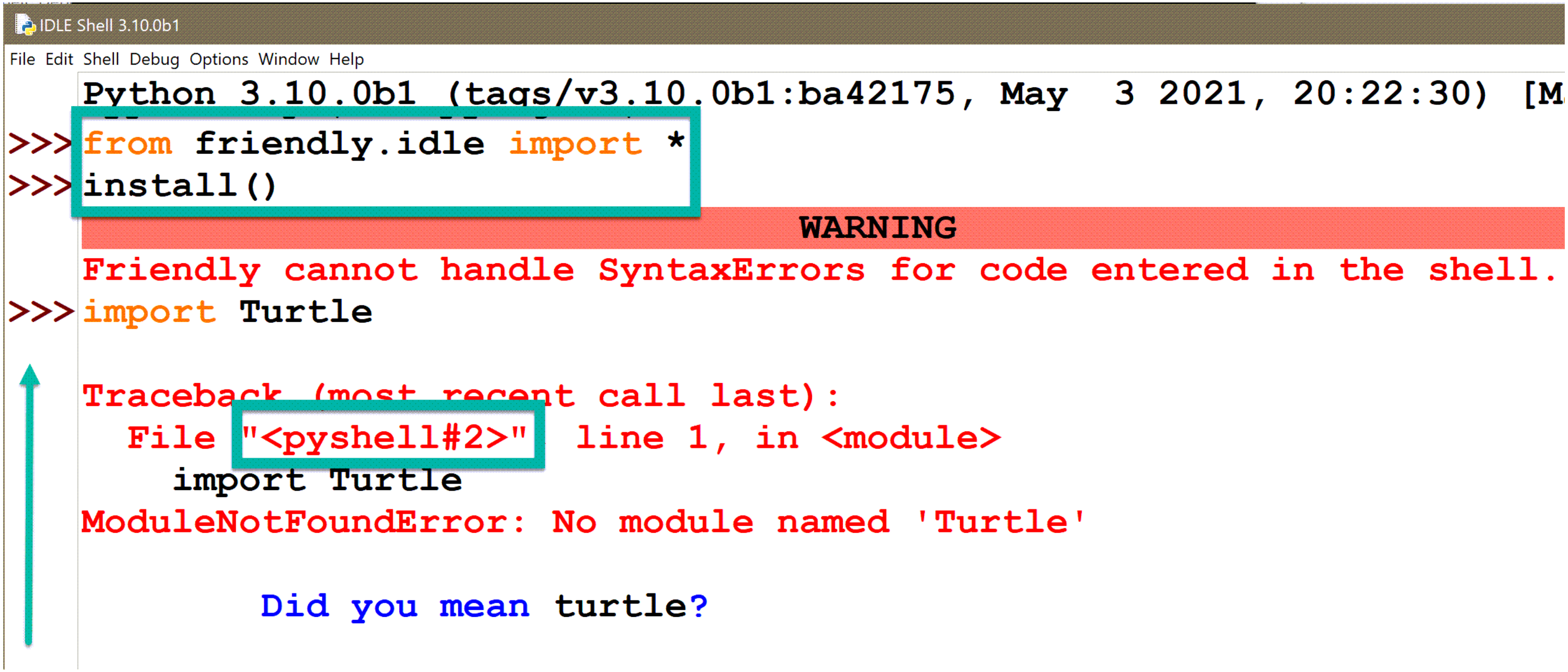
DEMO

```
from friendly.idle import run

run("test.py")
```

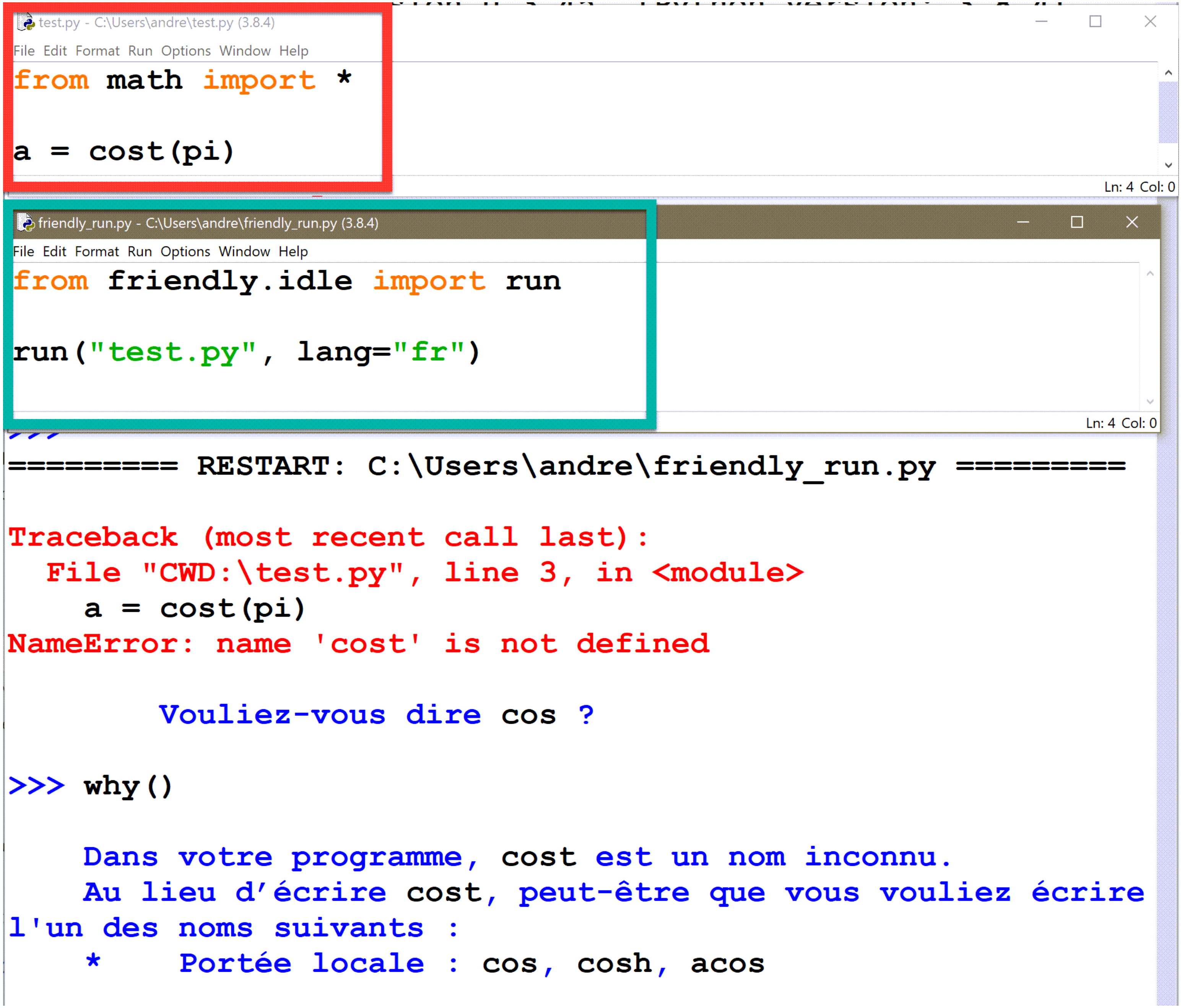
If the demo does not work.



```
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:30:28) [
>>> from friendly.idle import start_console
>>> start_console()

Friendly Console version 0.3.45. [Python version: 3.8.4]

>>> def pass():

Traceback (most recent call last):
  File "<friendly-console:1>", line 1
    def pass():
        ^
SyntaxError: invalid syntax

        You cannot use a Python keyword as a function name.
```



```
Python 3.10.0b1 (tags/v3.10.0b1:ba42175, May  3 2021, 20:22:30) [M
>>> from friendly.idle import *
>>> install()
                                WARNING
Friendly cannot handle SyntaxErrors for code entered in the shell.
>>> import Turtle

Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    import Turtle
ModuleNotFoundError: No module named 'Turtle'

        Did you mean turtle?
```



test.py - C:\Users\andre\test.py (3.8.4)
File Edit Format Run Options Window Help
Ln: 4 Col: 0
friendly_run.py - C:\Users\andre\friendly_run.py (3.8.4)
File Edit Format Run Options Window Help
Ln: 4 Col: 0

```
from math import *

a = cost(pi)
```

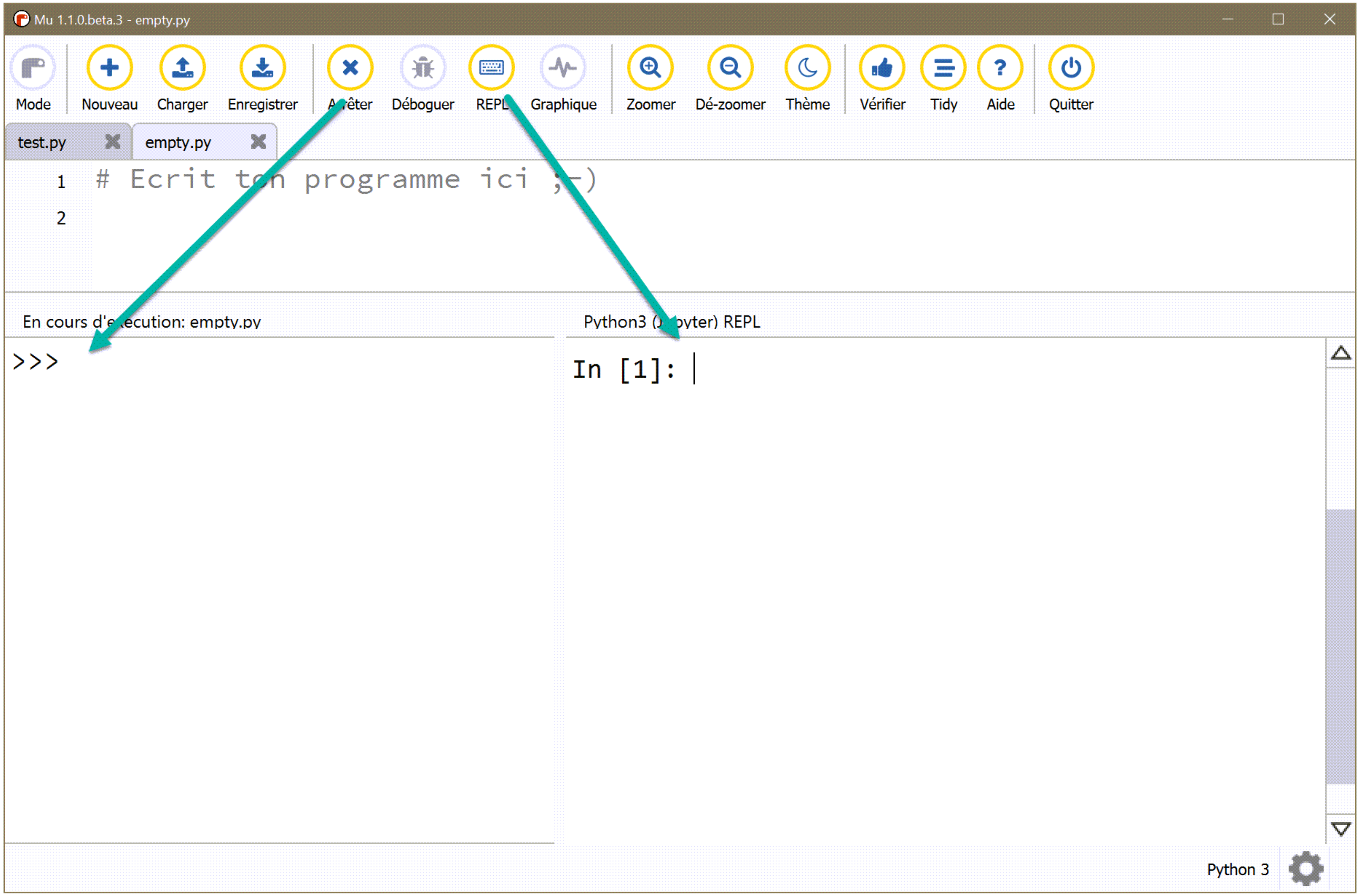
```
from friendly.idle import run

run("test.py", lang="fr")
```

```
========= RESTART: C:\Users\andre\friendly_run.py =========

Traceback (most recent call last):
  File "CWD:\test.py", line 3, in <module>
    a = cost(pi)
NameError: name 'cost' is not defined

        Vouliez-vous dire cos ?

>>> why()

    Dans votre programme, cost est un nom inconnu.
    Au lieu d’écrire cost, peut-être que vous vouliez écrire
l'un des noms suivants :
    *    Portée locale : cos, cosh, acos
```

MU

DEMO

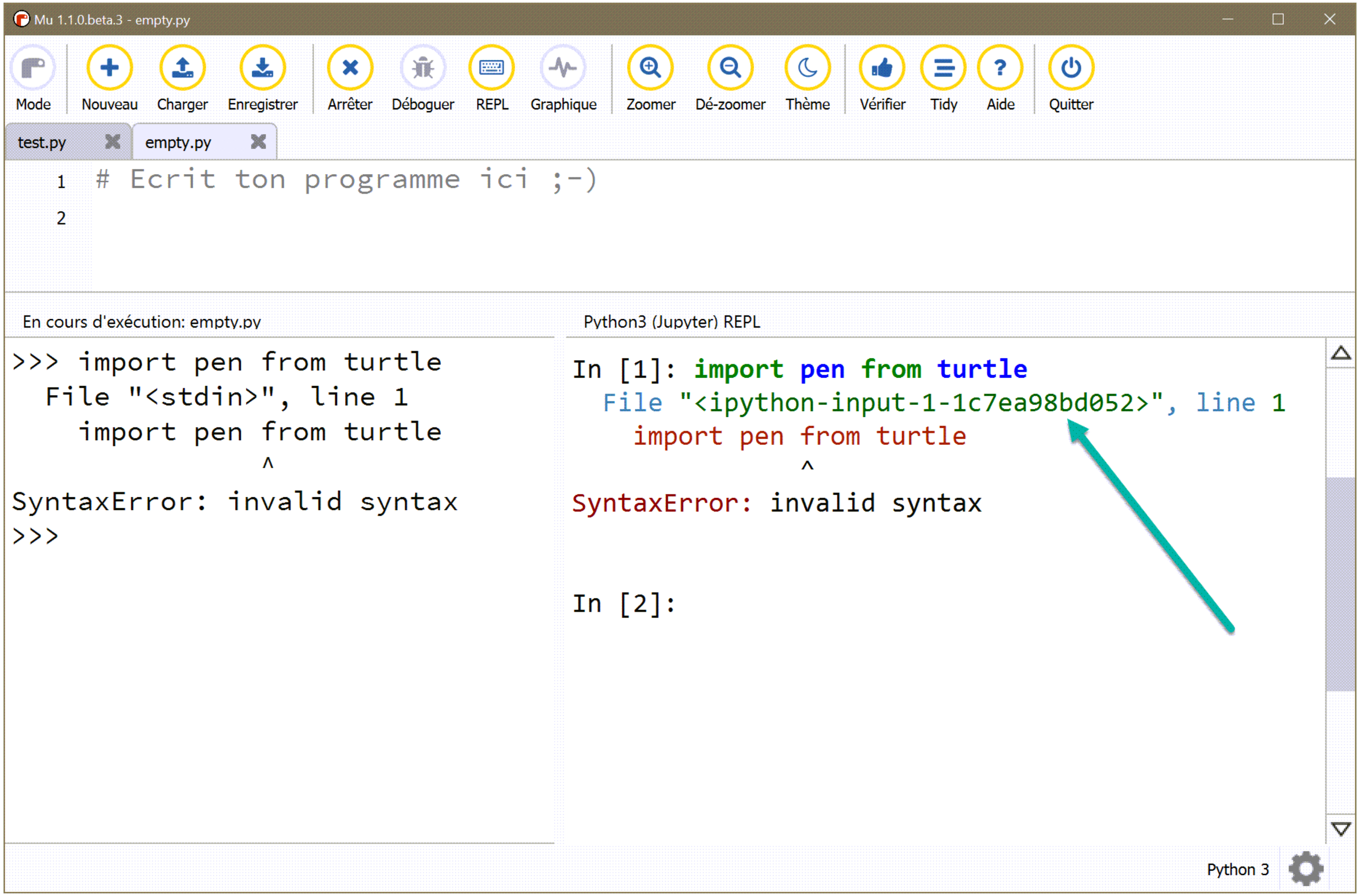
If the demo does not work.



```
# Ecrit ton programme ici ;-)
```

```
>>>
```

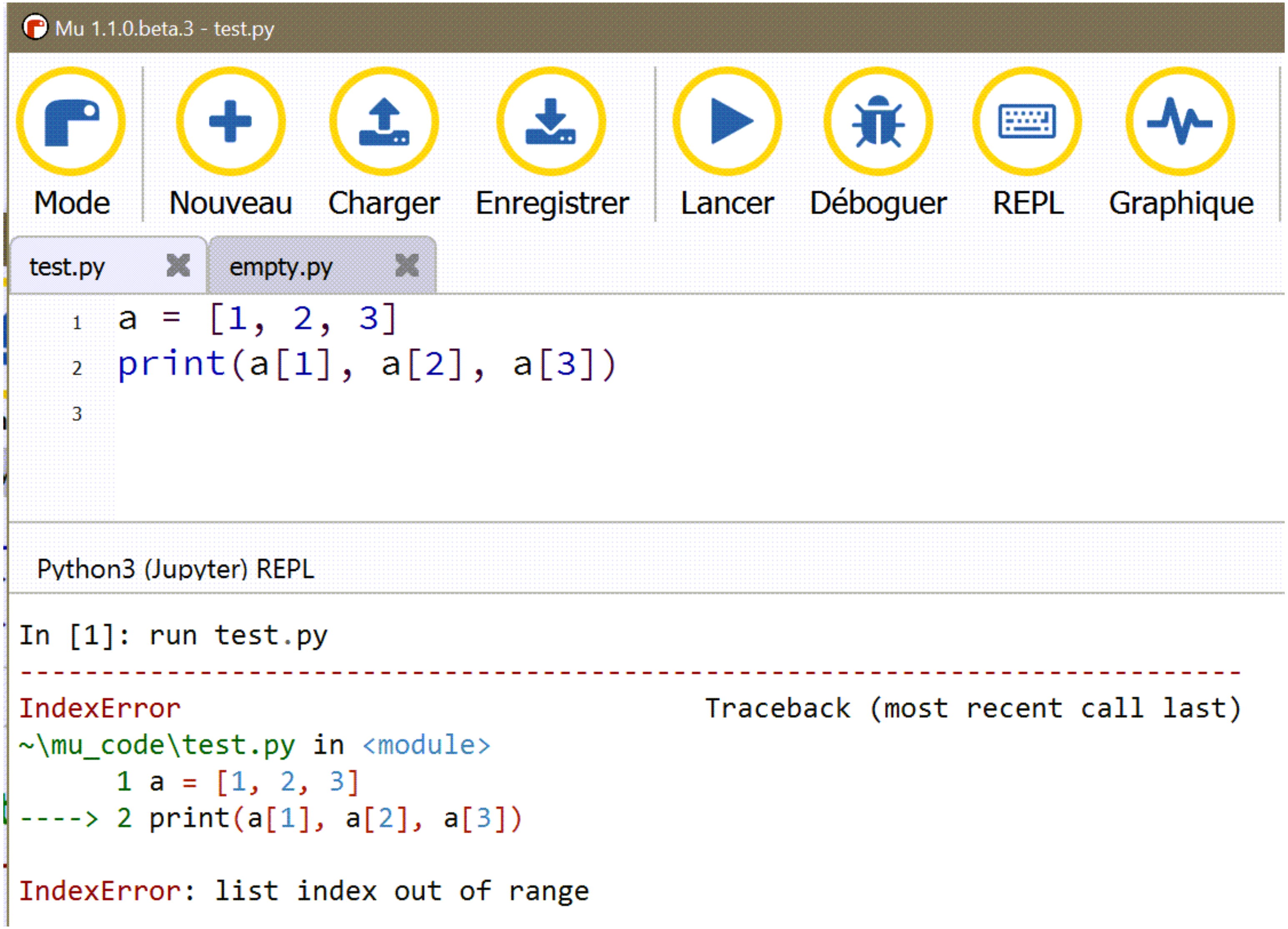
```
In [1]:
```



```
# Ecrit ton programme ici ;-)
```

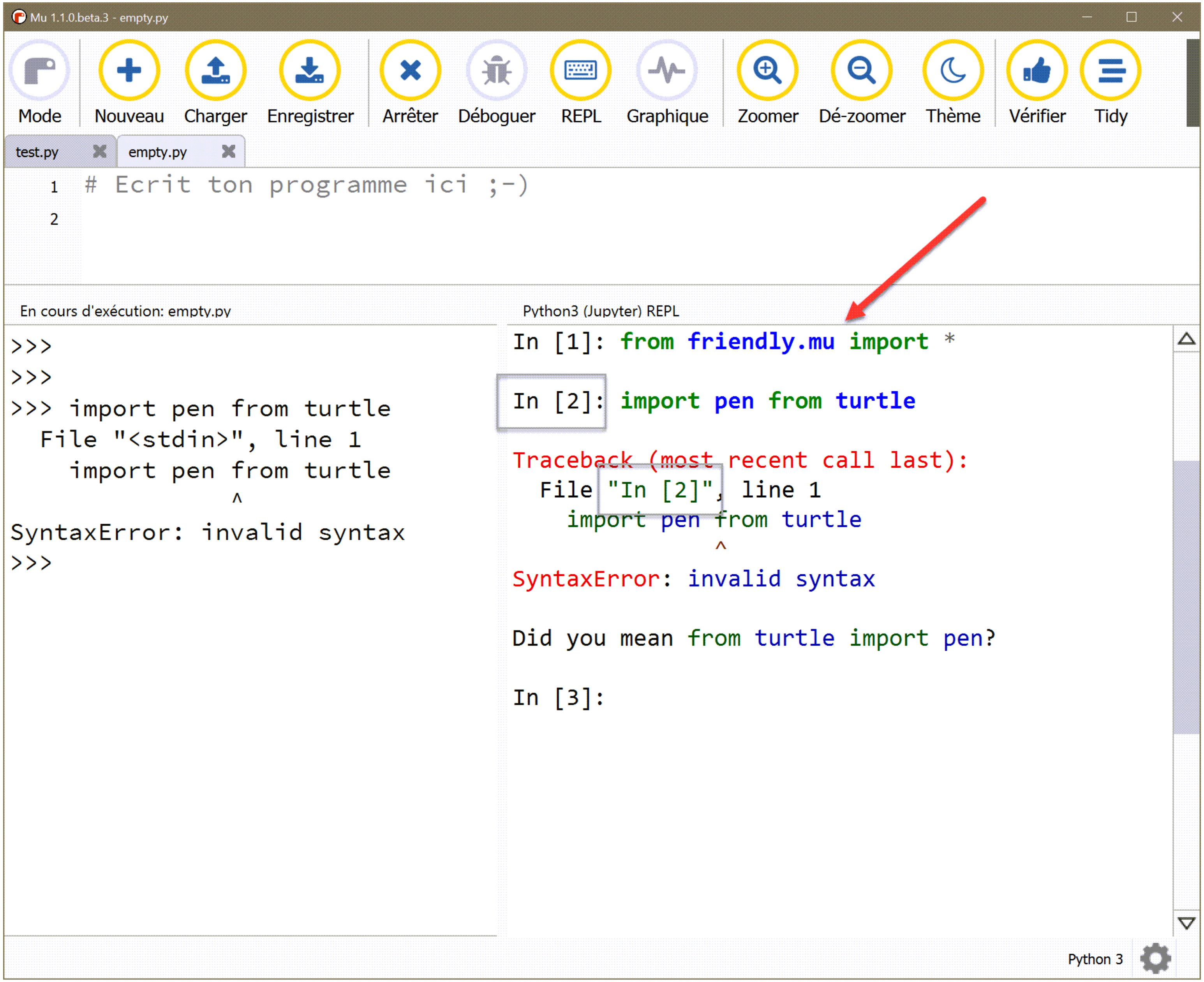
```
>>> import pen from turtle
  File "<stdin>", line 1
    import pen from turtle
                   ^
SyntaxError: invalid syntax
>>>
```

```
In [1]: import pen from turtle
  File "<ipython-input-1-1c7ea98bd052>", line 1
    import pen from turtle
                   ^
SyntaxError: invalid syntax


In [2]:
```



```python
a = [1, 2, 3]
print(a[1], a[2], a[3])
```

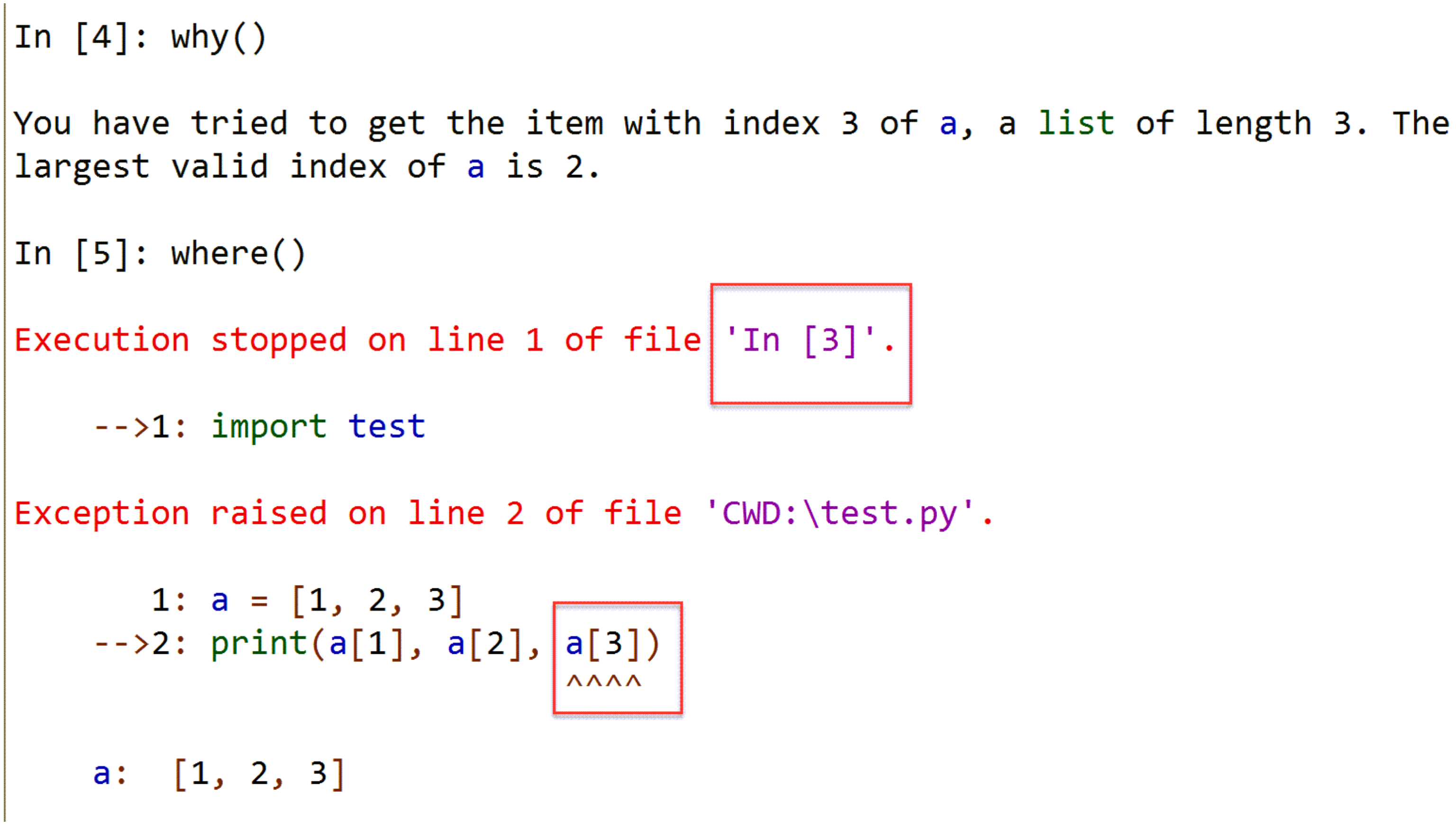
```
In [1]: run test.py
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
~\mu_code\test.py in <module>
      1 a = [1, 2, 3]
----> 2 print(a[1], a[2], a[3])

IndexError: list index out of range
```



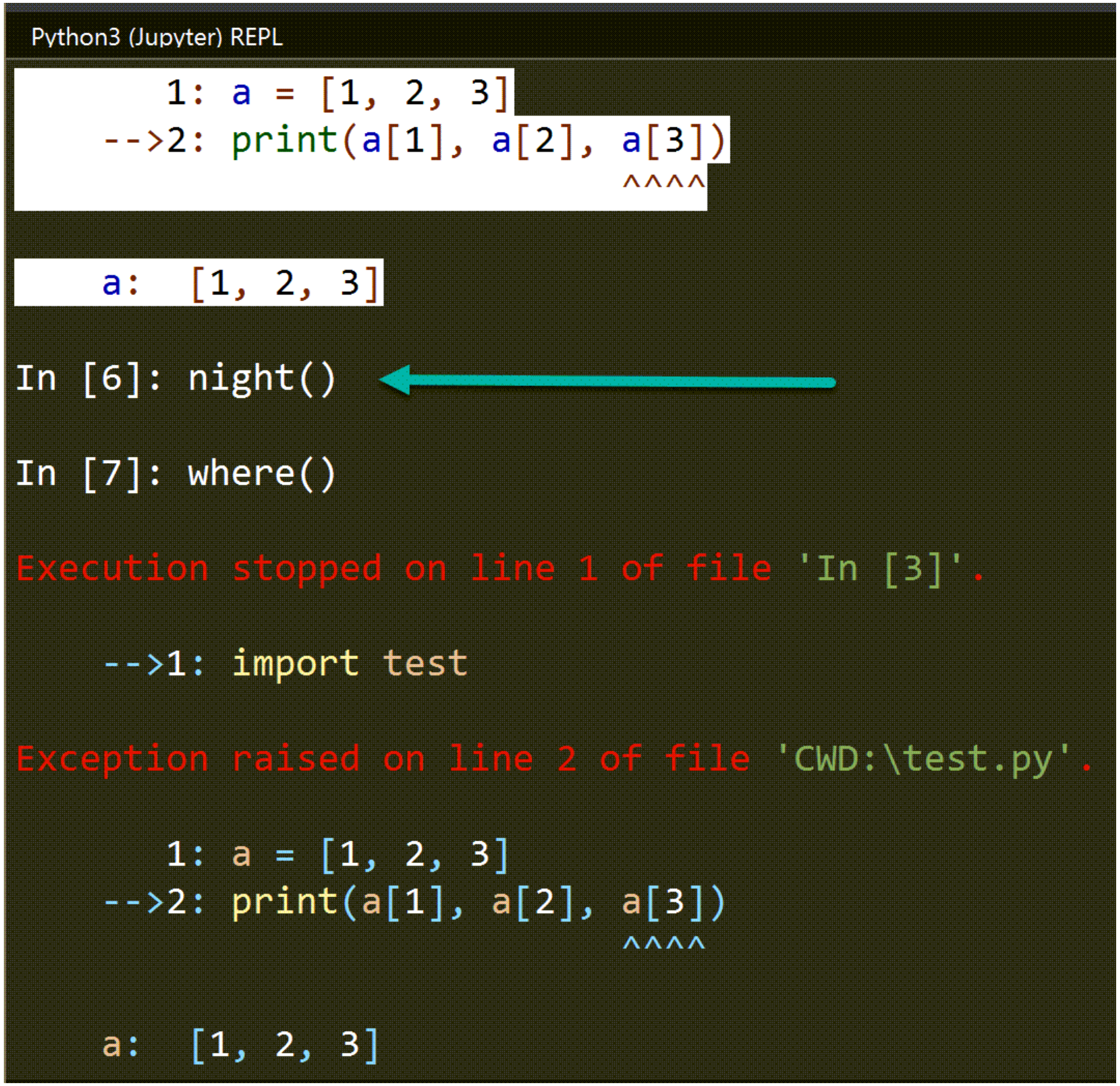
```
# Ecrit ton programme ici ;-)

```

```
>>>
>>>
>>> import pen from turtle
  File "<stdin>", line 1
    import pen from turtle
                  ^
SyntaxError: invalid syntax
>>>
```

```
In [1]: from friendly.mu import *

In [2]: import pen from turtle

Traceback (most recent call last):
  File "In [2]", line 1
    import pen from turtle
               ^
SyntaxError: invalid syntax

Did you mean from turtle import pen?

In [3]:
```

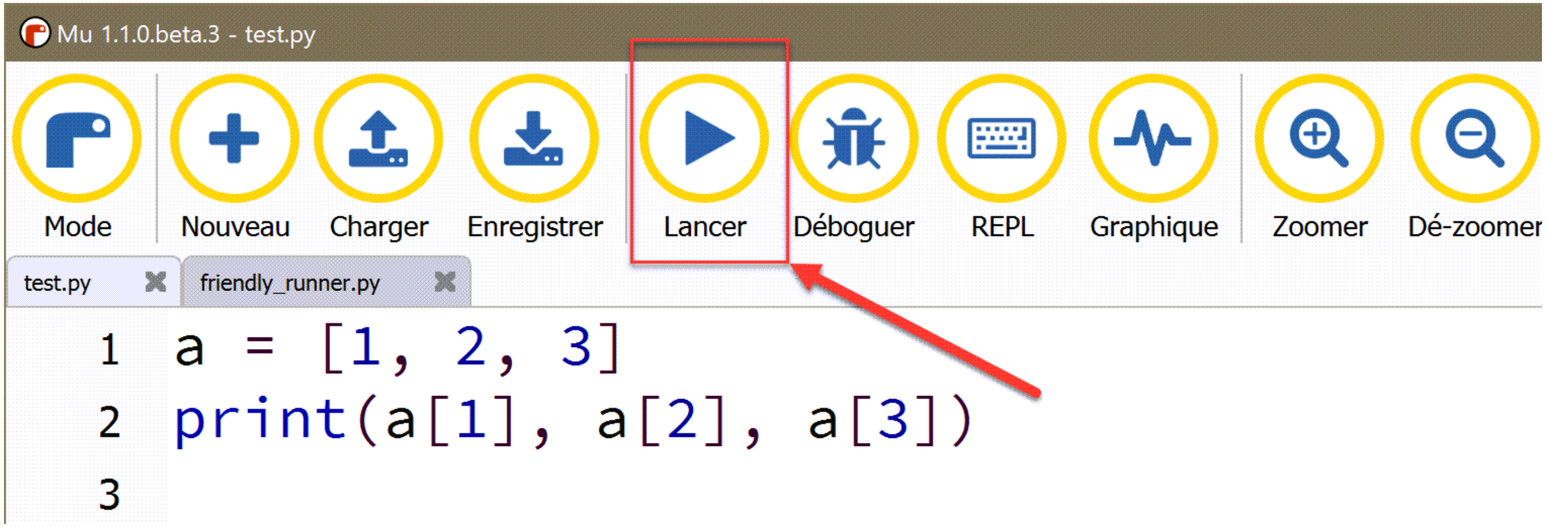


```
In [4]: why()

You have tried to get the item with index 3 of a, a list of length 3. The
largest valid index of a is 2.

In [5]: where()

Execution stopped on line 1 of file 'In [3]'.

    -->1: import test

Exception raised on line 2 of file 'CWD:\test.py'.

       1: a = [1, 2, 3]
    -->2: print(a[1], a[2], a[3])
                            ^^^^

    a:  [1, 2, 3]
```

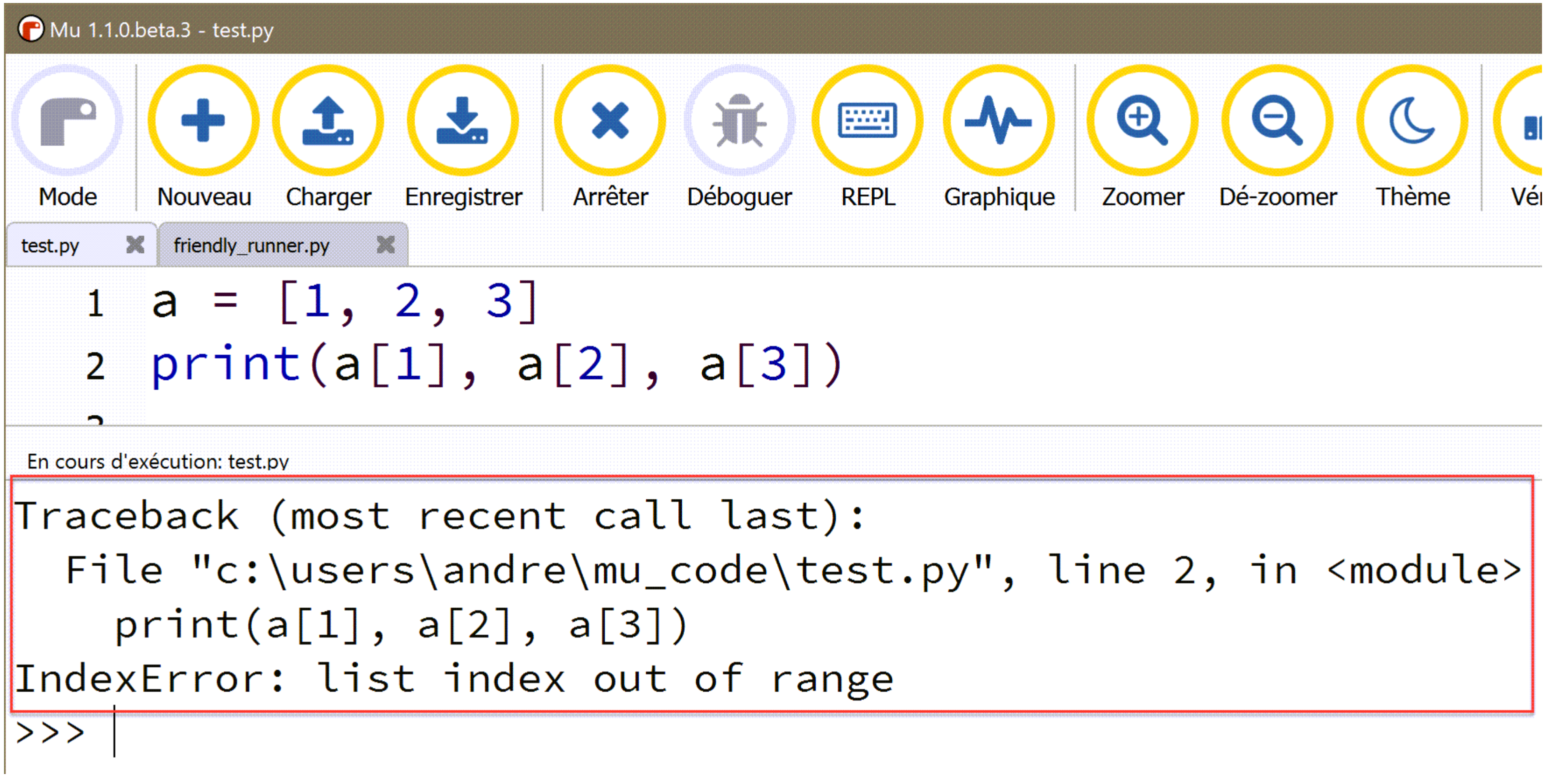


Python3 (Jupyter) REPL

```
      1: a = [1, 2, 3]
    -->2: print(a[1], a[2], a[3])
                              ^^^^

    a:  [1, 2, 3]

In [6]: night()

In [7]: where()

Execution stopped on line 1 of file 'In [3]'.

    -->1: import test

Exception raised on line 2 of file 'CWD:\test.py'.

      1: a = [1, 2, 3]
    -->2: print(a[1], a[2], a[3])
                              ^^^^

    a:  [1, 2, 3]
```



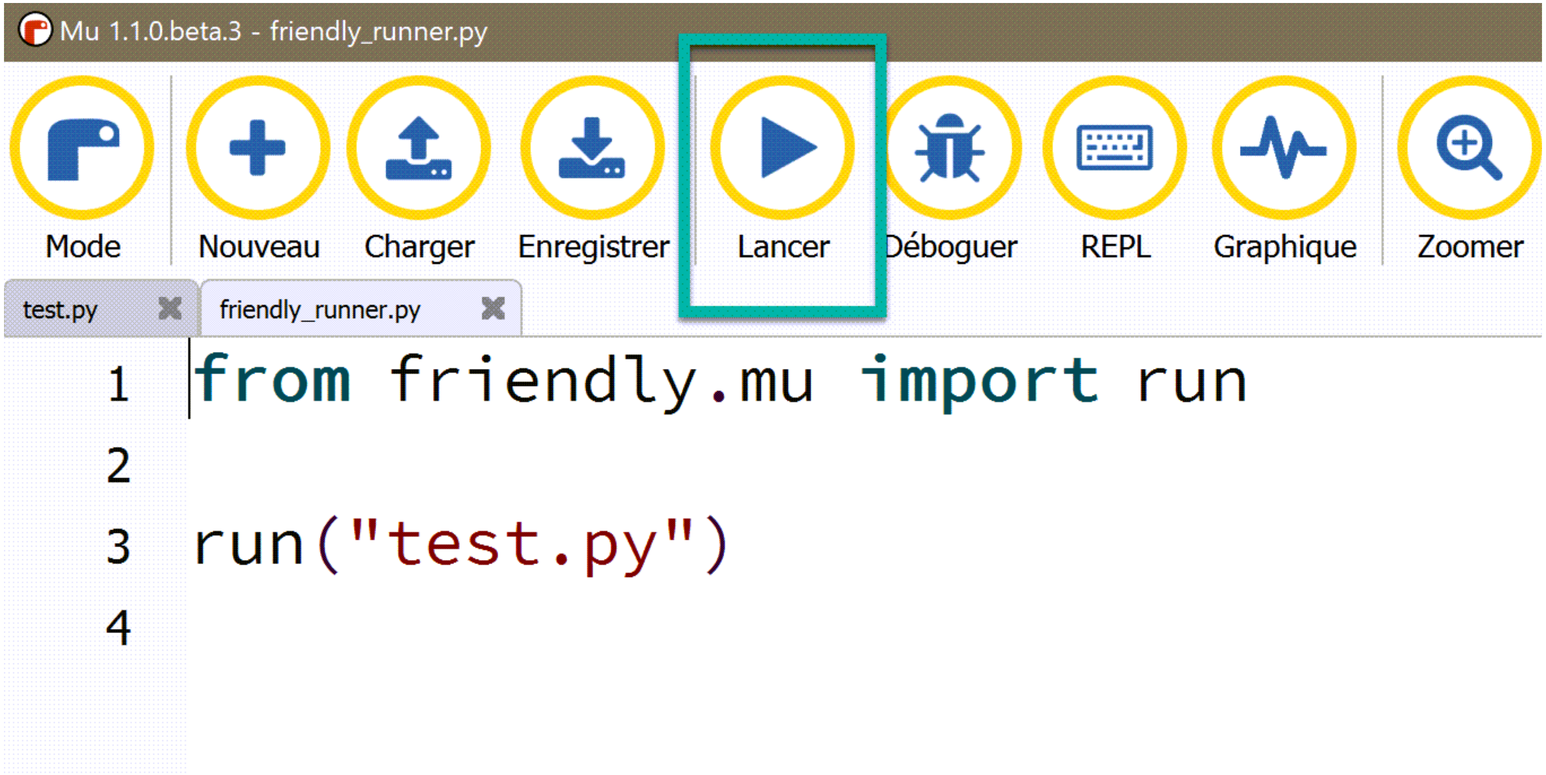
```python
a = [1, 2, 3]
print(a[1], a[2], a[3])

```



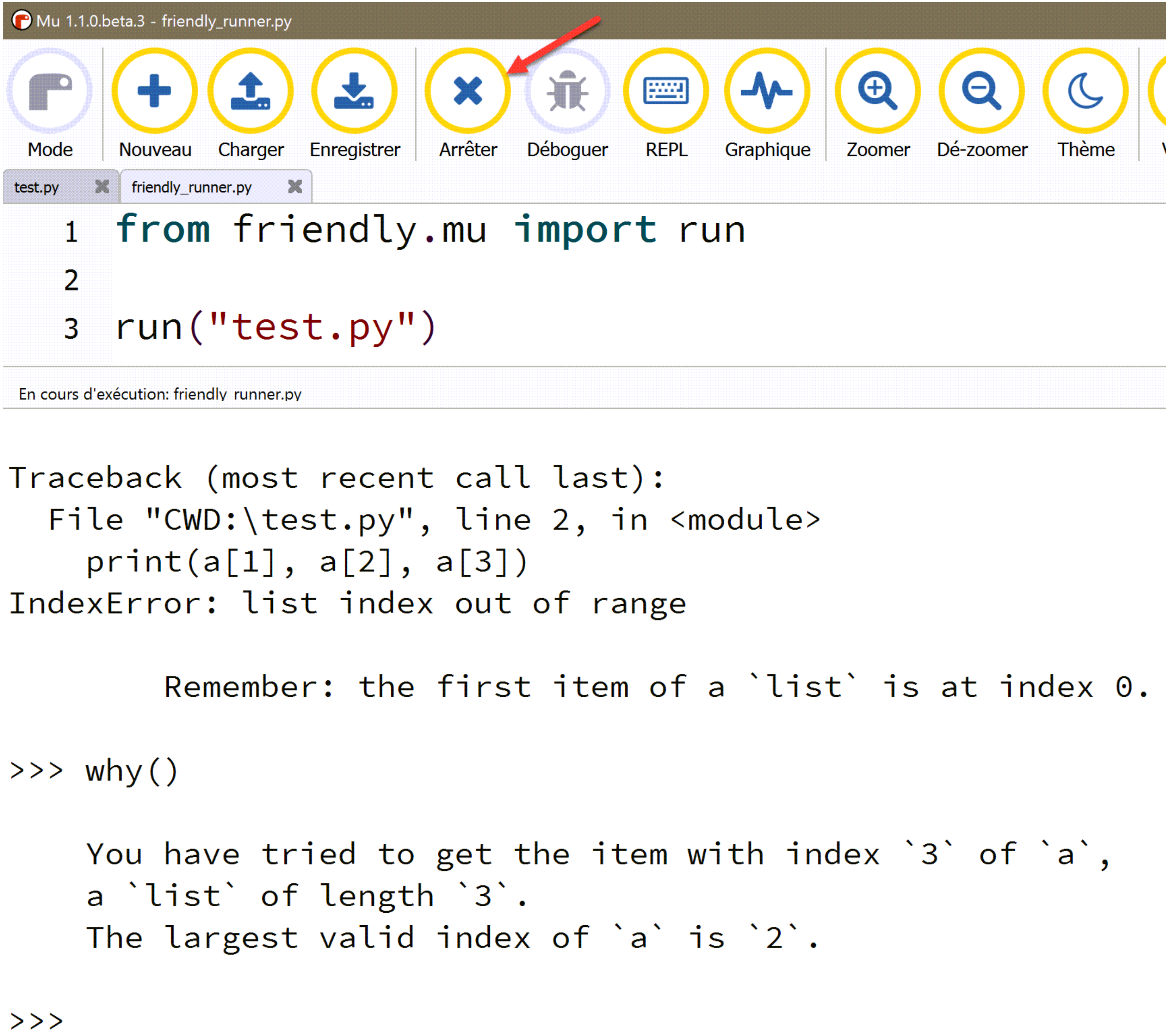
```python
a = [1, 2, 3]
print(a[1], a[2], a[3])
```

En cours d'exécution: test.py

```
Traceback (most recent call last):
  File "c:\users\andre\mu_code\test.py", line 2, in <module>
    print(a[1], a[2], a[3])
IndexError: list index out of range
>>> 
```



```
from friendly.mu import run

run("test.py")

```



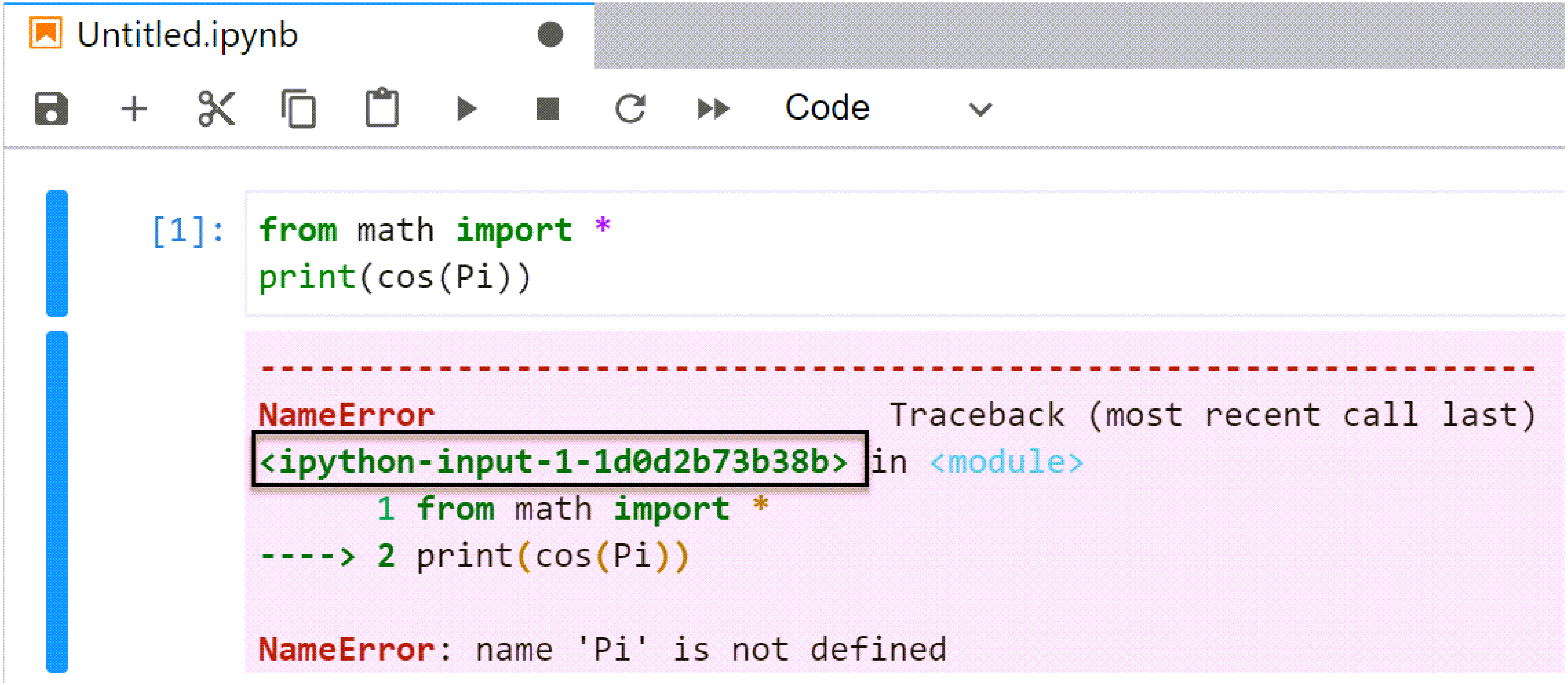
```python
from friendly.mu import run

run("test.py")
```

En cours d'exécution: friendly_runner.py

```
Traceback (most recent call last):
  File "CWD:\test.py", line 2, in <module>
    print(a[1], a[2], a[3])
IndexError: list index out of range

        Remember: the first item of a `list` is at index 0.

>>> why()

    You have tried to get the item with index `3` of `a`,
    a `list` of length `3`.
    The largest valid index of `a` is `2`.

>>>
```

DEMO?

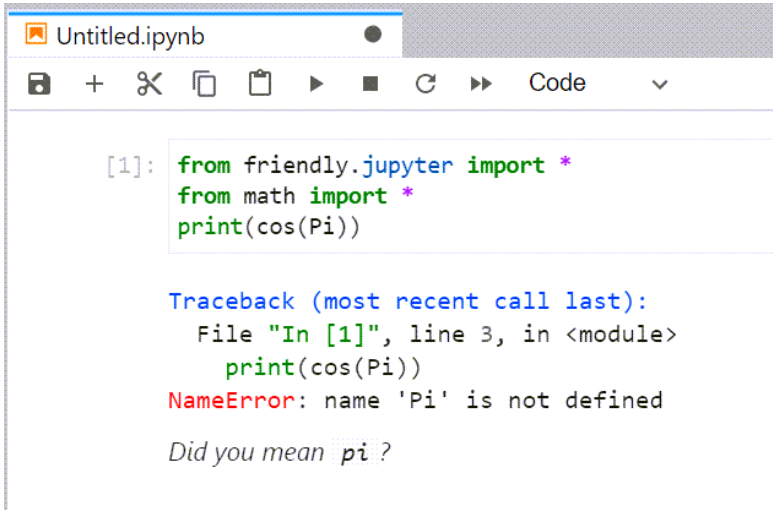
If the demo does not work ...



```
[1]: from math import *
     print(cos(Pi))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-1d0d2b73b38b> in <module>
      1 from math import *
----> 2 print(cos(Pi))

NameError: name 'Pi' is not defined
```

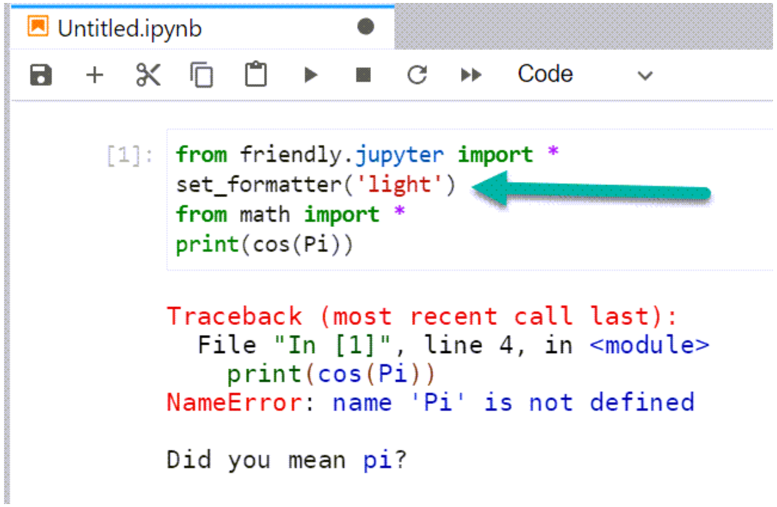
```
from friendly.jupyter import *
```



```
[1]: from friendly.jupyter import *
     from math import *
     print(cos(Pi))
```

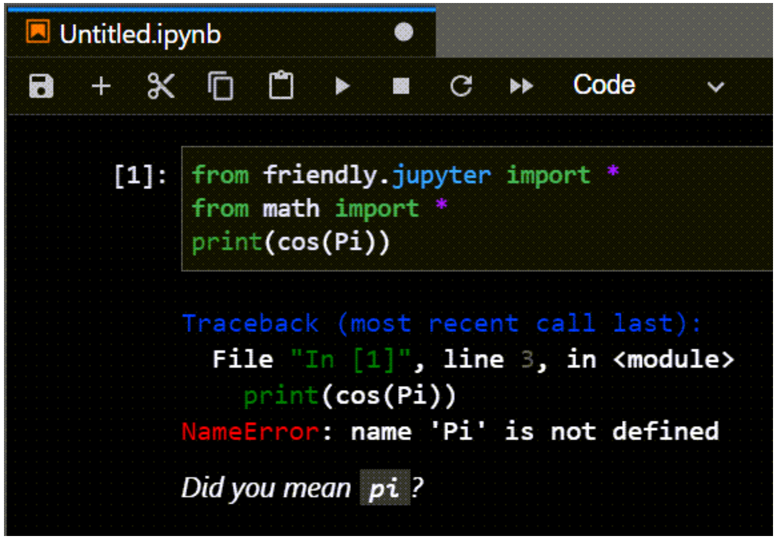
```
Traceback (most recent call last):
  File "In [1]", line 3, in <module>
    print(cos(Pi))
NameError: name 'Pi' is not defined
```

Did you mean `pi` *?*



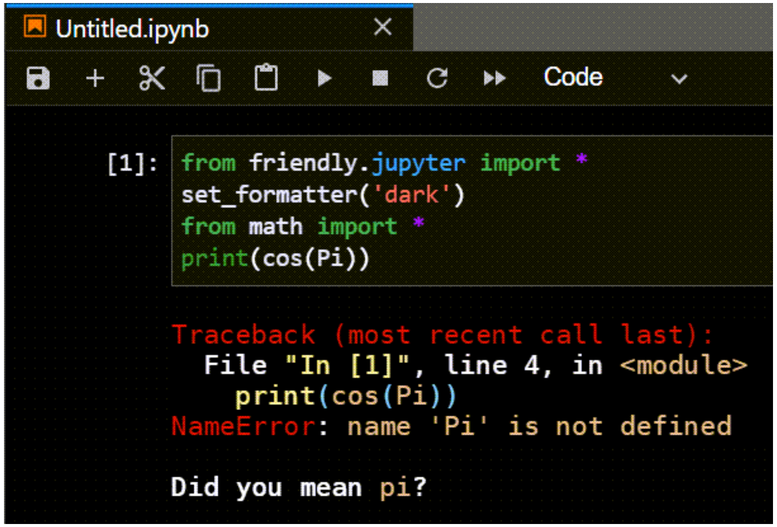
```
[1]: from friendly.jupyter import *
     set_formatter('light')
     from math import *
     print(cos(Pi))
```

```
Traceback (most recent call last):
  File "In [1]", line 4, in <module>
    print(cos(Pi))
NameError: name 'Pi' is not defined

Did you mean pi?
```

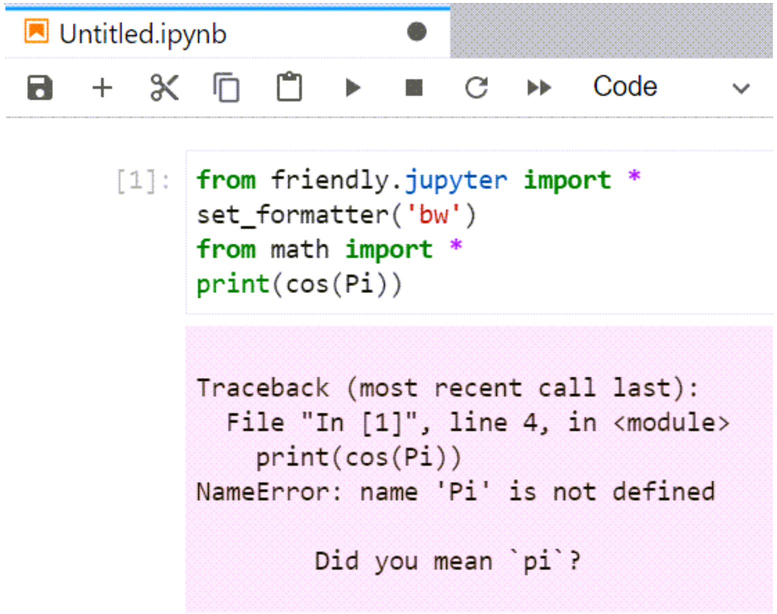


```
[1]: from friendly.jupyter import *
     from math import *
     print(cos(Pi))

Traceback (most recent call last):
  File "In [1]", line 3, in <module>
    print(cos(Pi))
NameError: name 'Pi' is not defined
```

Did you mean `pi` *?*

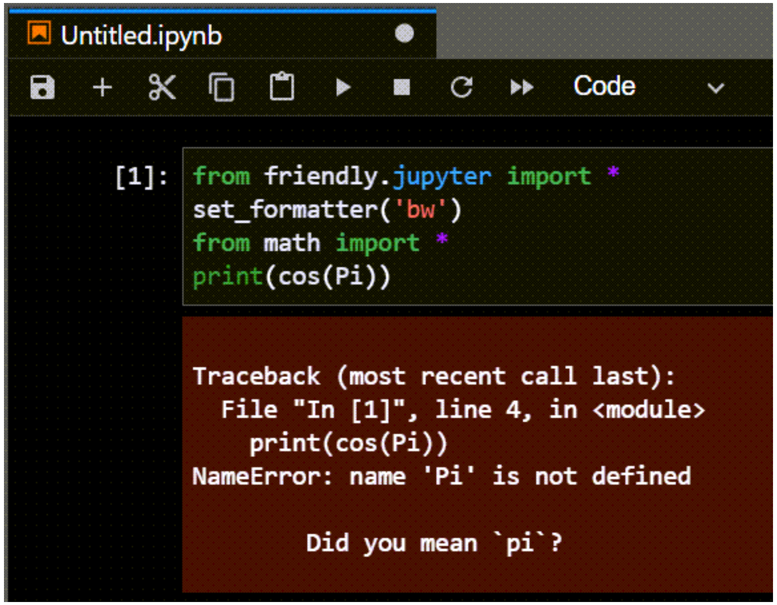


```
[1]: from friendly.jupyter import *
     set_formatter('dark')
     from math import *
     print(cos(Pi))

Traceback (most recent call last):
  File "In [1]", line 4, in <module>
    print(cos(Pi))
NameError: name 'Pi' is not defined

Did you mean pi?
```



```
[1]: from friendly.jupyter import *
     set_formatter('bw')
     from math import *
     print(cos(Pi))
```

```
Traceback (most recent call last):
  File "In [1]", line 4, in <module>
    print(cos(Pi))
NameError: name 'Pi' is not defined

        Did you mean `pi`?
```


Untitled.ipynb
Code

[1]: from friendly.jupyter import *
set_formatter('bw')
from math import *
print(cos(Pi))

Traceback (most recent call last):
  File "In [1]", line 4, in <module>
    print(cos(Pi))
NameError: name 'Pi' is not defined

        Did you mean `pi`?

friendly used to be called **friendly-traceback.**

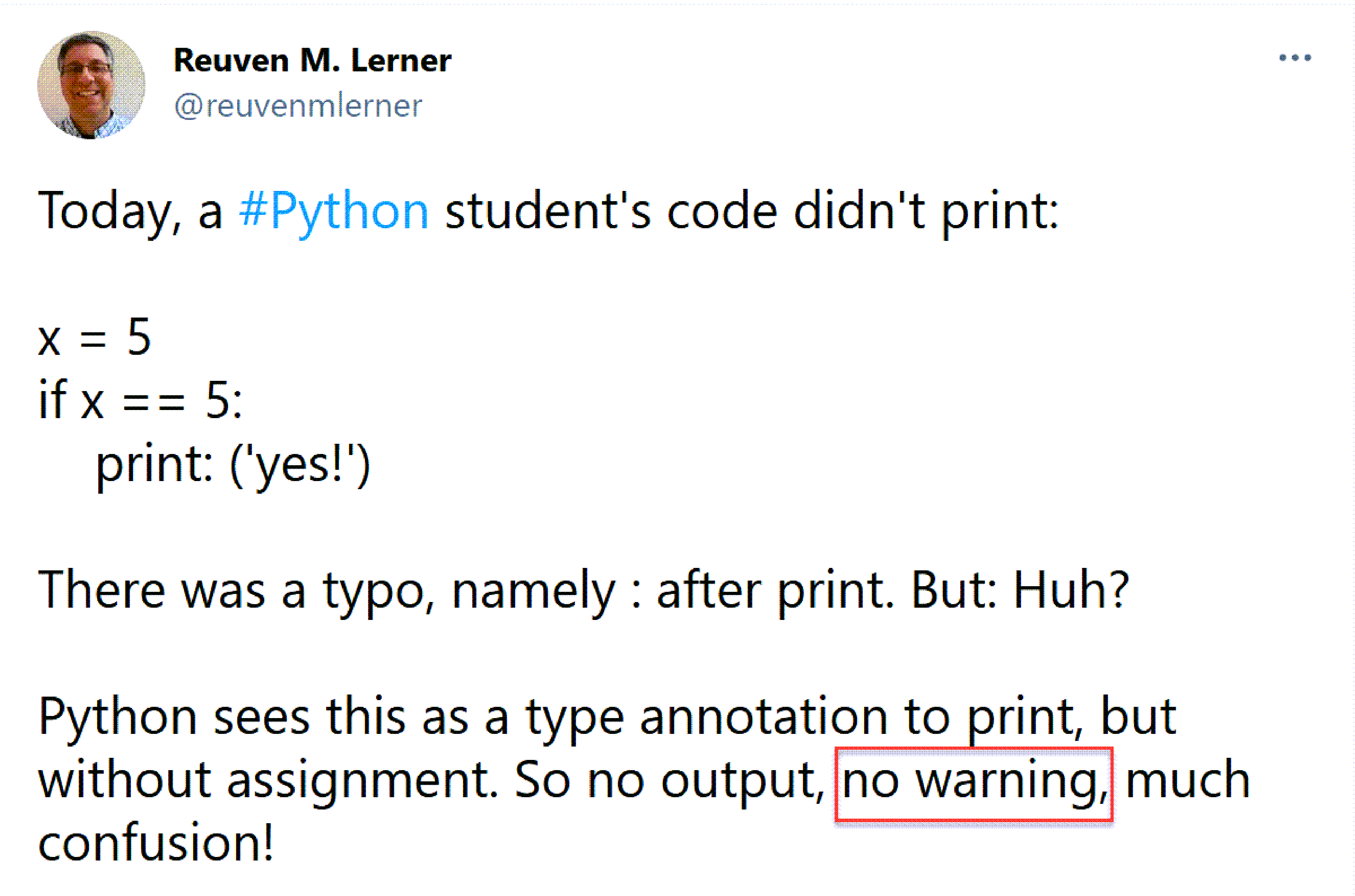
```
>>> import this
...
Explicit is better than implicit.
...
```

friendly-traceback is more explicit than **friendly**.

Why was the name changed?

```python
from friendly          import ...
from friendly.idle     import ...
from friendly.mu       import ...
from friendly.ipython import ...
from friendly.jupyter import ...

# are better than

from friendly_traceback.idle import ...
```

Besides, *traceback* is not exactly a beginner-friendly term and, in the future, **friendly** might do more than simply helping with tracebacks ...



Reuven M. Lerner
@reuvenmlerner

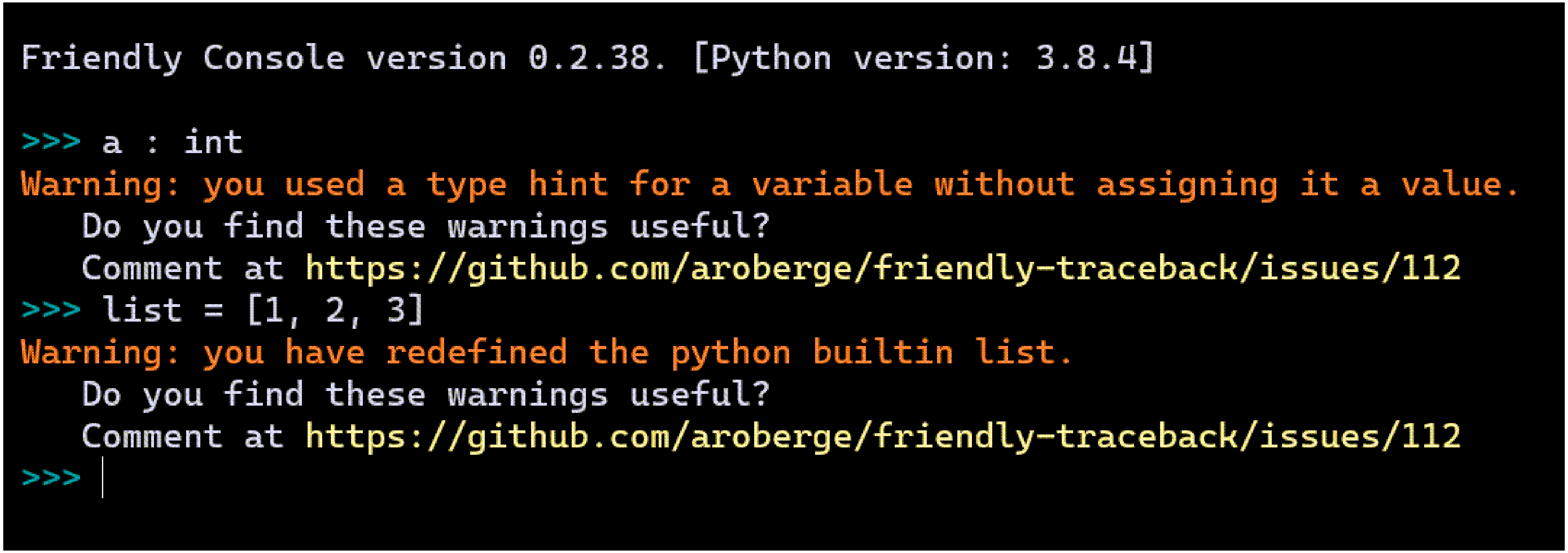
Today, a #Python student's code didn't print:

```
x = 5
if x == 5:
    print: ('yes!')
```

There was a typo, namely : after print. But: Huh?

Python sees this as a type annotation to print, but without assignment. So no output, no warning, much confusion!

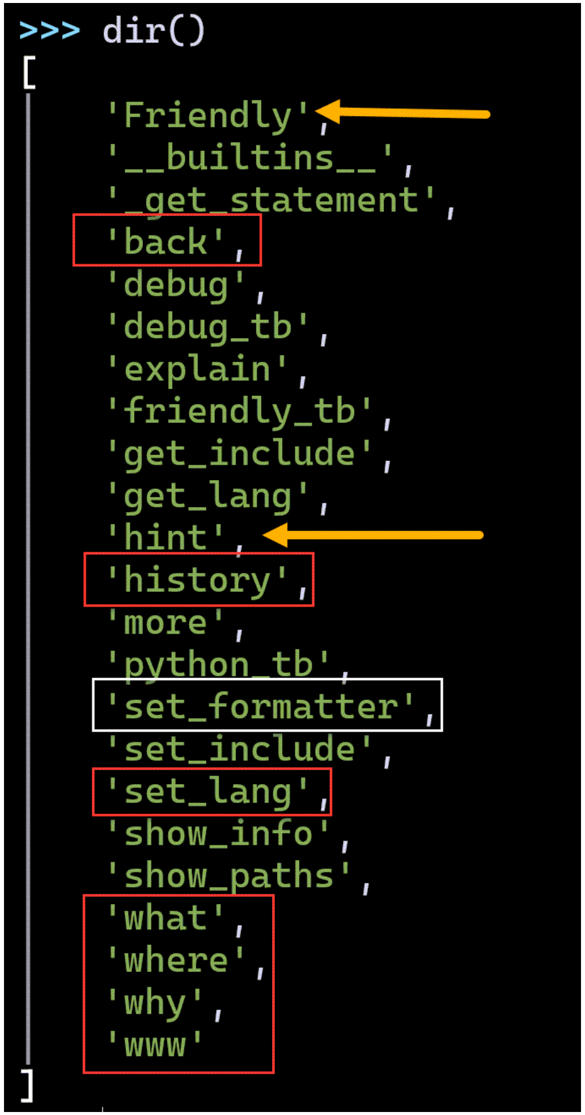
WARNINGS



```
Friendly Console version 0.2.38. [Python version: 3.8.4]

>>> a : int
Warning: you used a type hint for a variable without assigning it a value.
   Do you find these warnings useful?
   Comment at https://github.com/aroberge/friendly-traceback/issues/112
>>> list = [1, 2, 3]
Warning: you have redefined the python builtin list.
   Do you find these warnings useful?
   Comment at https://github.com/aroberge/friendly-traceback/issues/112
>>> 
```

Only works in the friendly console, not with custom modules.

WHY `import *`?



```
>>> dir()
[
    'Friendly',
    '__builtins__',
    '_get_statement',
    'back',
    'debug',
    'debug_tb',
    'explain',
    'friendly_tb',
    'get_include',
    'get_lang',
    'hint',
    'history',
    'more',
    'python_tb',
    'set_formatter',
    'set_include',
    'set_lang',
    'show_info',
    'show_paths',
    'what',
    'where',
    'why',
    'www'
]
```

Friendly.why() == why(), etc

THANKS TO

- Nicholas Tollervey (Mu)
- Aivar Annamaa (Thonny)
- Julien Palard (HackInScience)
- Alex Hall (futurecoder + many packages)
- Sylvain Desodt (DidYouMean-Python inspired by Raymond Hettinger)
- Michael Kennedy, Brian Okken, and Hannah Stepanek in PythonBytes podcast #220
- *plus many others who filed issues, etc.*

WHERE IS THE CODE?

https://github.com/aroberge/friendly

You will find a link to the documentation.

But perhaps there is a more pythonic way ...

where(is the documentation?)😏

```
> python -m pip install friendly
> python -m friendly

Friendly Console version ...

>>> www()
```

friendly is currently at version 0.3.50

Try it!

Let me know how it could be improved
for you and your students.