```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit
(AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> def my_test_function():
...     m = 10
...     n = 20
...     result = m + n
...     print(f'result:{result}')
...
...
>>> my_test_function.__code__
<code object my_test_function at 0x000002890700F770, file "<pyshell#5>", line 1>
>>> my_test_function()
result:30
>>> exec(my_test_function.__code__)
result:30
>>> type(my_test_function)
<class 'function'>
>>> id(my_test_function)
2787552483936
>>> my_test_function = 11.13
>>> type(my_test_function)
<class 'float'>
>>> my_test_function()
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in <module>
    my_test_function()
TypeError: 'float' object is not callable
>>> id(my_test_function)
2787551430192
>>> print = 345345
>>> print()
Traceback (most recent call last):
  File "<pyshell#16>", line 1, in <module>
    print()
TypeError: 'int' object is not callable
>>> del print
>>> print("Hello")
Hello
class MyClass:
    def __init__(self, init_n):
        self.__dict__['_n'] = init_n
    def getn(self):
        return self.__dict__['_n']
    def setn(self, val):
        raise AttributeError('Cannot write on attribute n')
    def deln(self):
        raise AttributeError('Cannot delete attribute n')
    n = property(getb, setn, deln)
```

```
Traceback (most recent call last):
  File "<pyshell#29>", line 1, in <module>
    class MyClass:
  File "<pyshell#29>", line 10, in MyClass
    n = property(getb, setn, deln)
NameError: name 'getb' is not defined. Did you mean: 'getn'?
class MyClass:
    def __init__(self, init_n):
        self.__dict__['_n'] = init_n
    def getn(self):
        return self.__dict__['_n']
    def setn(self, val):
        raise AttributeError('Cannot write on attribute n')
    def deln(self):
        raise AttributeError('Cannot delete attribute n')
    n = property(getn, setn, deln)


m = MyClass()
Traceback (most recent call last):
  File "<pyshell#32>", line 1, in <module>
    m = MyClass()
TypeError: MyClass.__init__() missing 1 required positional argument: 'init_n'
m = MyClass(100)
m.n = 400
Traceback (most recent call last):
  File "<pyshell#34>", line 1, in <module>
    m.n = 400
  File "<pyshell#31>", line 7, in setn
    raise AttributeError('Cannot write on attribute n')
AttributeError: Cannot write on attribute n
```