

코딩입문(Python)(0301) 13주차
강의노트 실습 보고서
2022125028 안기겸

A screenshot of the IDLE Shell 3.10.1 window. The window has a title bar with three colored buttons (red, yellow, green) on the left and the text "IDLE Shell 3.10.1" in the center. The main area contains a Python REPL session. The session starts with a header line: "Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin". This is followed by a prompt "Type 'help', 'copyright', 'credits' or 'license()' for more information." The user enters several commands: ">>> x = 10", ">>> def foo():", ">>> ...", ">>> ...", ">>> foo()", ">>> 10", ">>> print(x)", ">>> 10", ">>> global()", which results in a "SyntaxError: invalid syntax" error. The user then enters ">>> globals()", which displays a dictionary of the current global namespace. This is followed by ">>> y='Hello'", ">>> globals()", which displays an updated dictionary including the new variable 'y'. The session ends with ">>> |". At the bottom right of the window, the status bar shows "Ln: 19 Col: 0".

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> x = 10
>>> def foo():
>>> ...
>>> ...
>>> foo()
10
>>> print(x)
10
>>> global()
SyntaxError: invalid syntax
>>> globals()
{'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__': <class '_frozen_importlib.BuiltinImporter'>, '__spec__': None, '__annotations__': {}, '__builtins__': <module 'builtins' (built-in)>, 'x': 10, 'foo': <function foo at 0x108d616c0>}
>>> y="Hello"
>>> globals()
{'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__': <class '_frozen_importlib.BuiltinImporter'>, '__spec__': None, '__annotations__': {}, '__builtins__': <module 'builtins' (built-in)>, 'x': 10, 'foo': <function foo at 0x108d616c0>, 'y': 'Hello'}
>>> |
```

Ln: 19 Col: 0

A screenshot of the IDLE Shell 3.10.1 window. The window has a title bar with three colored buttons (red, yellow, green) on the left and the text "IDLE Shell 3.10.1" in the center. The main area is a text editor with a light gray background. It contains Python code and its output. The code defines a function 'foo()' that prints the value of 'var' and then increments it by 3. It then sets 'var' to 3, prints it, and calls 'foo()'. The output shows the value 3 printed twice. The status bar at the bottom right shows "Ln: 17 Col: 0".

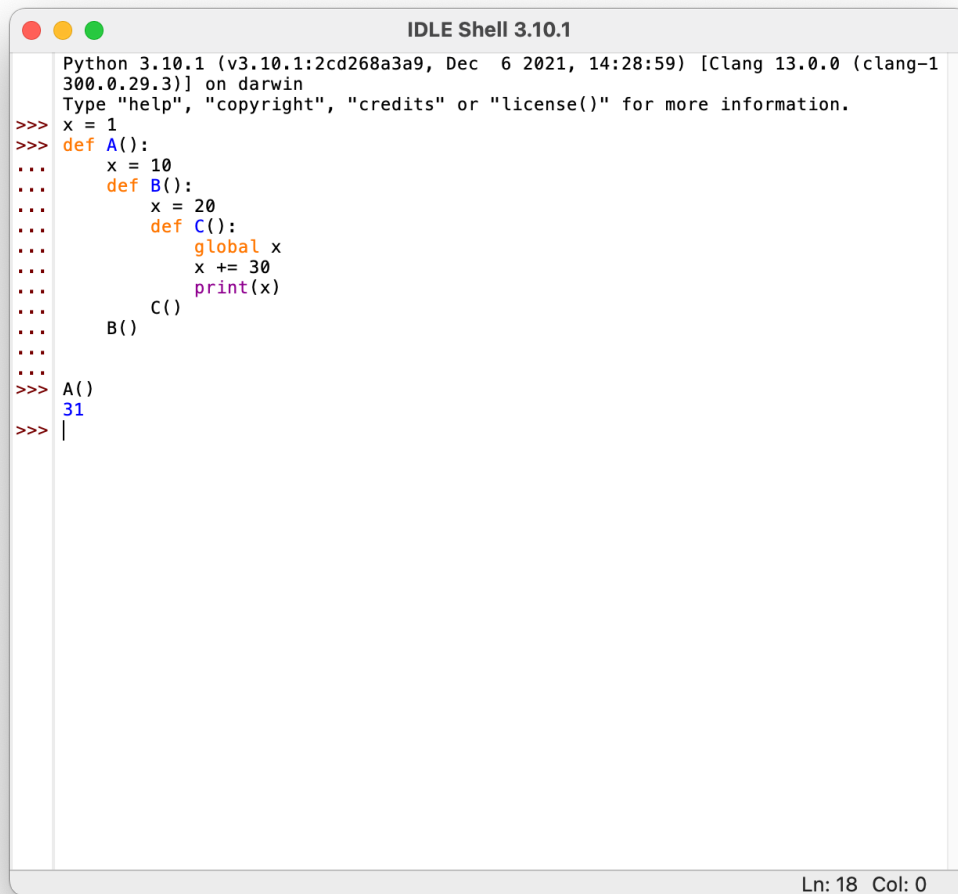
```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> def foo():
...     print(var)
...     var1 = 33
...     def fooo():
...         print(var1)
...         fooo()
...
...
>>> var = 3
>>> print(var)
3
>>> foo()
3
33
>>>
```

Ln: 17 Col: 0

A screenshot of the IDLE Shell 3.10.1 window. The window has a title bar with three colored buttons (red, yellow, green) on the left and the text "IDLE Shell 3.10.1" in the center. The main area contains a Python shell session. The session starts with the Python version and platform information. Then, a function 'foo' is defined with a local variable 'var'. The user sets 'var' to 3 and prints it, getting 3. Then they call 'foo()', which results in a 'UnboundLocalError: local variable 'var' referenced before assignment' because the function tries to use 'var' before it has been assigned a new value. The user then redefines 'foo' to use 'global var', sets 'var' to 3, prints it (3), and calls 'foo()' again, which prints 4. The status bar at the bottom right shows "Ln: 30 Col: 0".

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> def foo():
...     var += 1
...     print(var)
...
...
>>> var = 3
>>> print(var)
3
>>> foo()
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    foo()
  File "<pyshell#3>", line 2, in foo
    var += 1
UnboundLocalError: local variable 'var' referenced before assignment
>>>
>>> def foo():
...     global var
...     var += 1
...     print(var)
...
...
>>> var = 3
>>> print(var)
3
>>> foo()
4
>>> |
```

Ln: 30 Col: 0

A screenshot of the Python IDLE Shell window. The title bar at the top reads "IDLE Shell 3.10.1". The main text area contains the following code:

```
>>> Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> x = 1
>>> def A():
...     x = 10
...     def B():
...         x = 20
...         def C():
...             global x
...             x += 30
...             print(x)
...         C()
...     B()
>>> A()
31
>>> |
```

The code defines a function A() which contains a function B(). B() contains a function C() which uses a global variable x. The code calls A(), which prints 31. The cursor is on the line following the last prompt. The status bar at the bottom right shows "Ln: 18 Col: 0".

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> def A():
...     x = 10
...     y = 100
...     def B():
...         x = 20
...         def C():
...             nonlocal x
...             nonlocal y
...             x += 30
...             y += 300
...             print(x)
...             print(y)
...         C()
...     B()
...
>>> A
SyntaxError: unmatched ')
>>> A()
50
400
>>> |
```

Ln: 24 Col: 0



The image shows a screenshot of the Python IDLE Shell window. The title bar reads "IDLE Shell 3.10.1". The shell displays the Python version and compiler information: "Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin". It also shows a prompt to type "help", "copyright", "credits" or "license()" for more information. The user has entered a recursive function `A()` that calls itself and a nested function `B()` that uses `nonlocal x` to modify the variable `x` in the scope of `A()`. The function `A()` prints the value of `x` before calling `B()`. The user has executed `A()`, and the output shows the value `20` being printed. The status bar at the bottom right indicates "Ln: 14 Col: 0".

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> def A():
...     x = 10
...     def B():
...         nonlocal x
...         x = 20
...     B()
...     print(x)
...
>>> A()
20
>>> |
```

Ln: 14 Col: 0

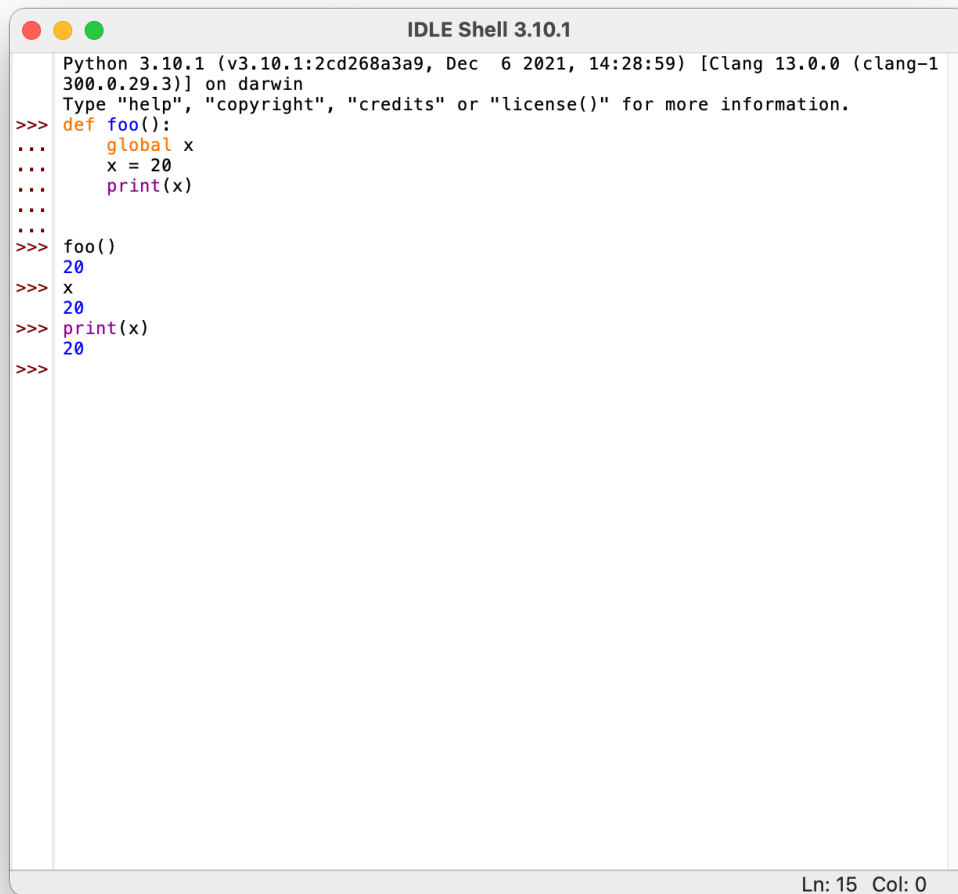

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> def A():
...     x = 10
...     def B():
...         x = 20
...
>>> B()
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    B()
NameError: name 'B' is not defined
>>> B
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    B
NameError: name 'B' is not defined
>>> A
<function A at 0x10a2996c0>
>>> def A():
...     x = 10
...     def B():
...         x = 20
...         B()
...         print(x)
...
>>> A()
10
>>>
```

Ln: 31 Col: 0

A screenshot of the IDLE Shell 3.10.1 window. The window has a title bar with three colored buttons (red, yellow, green) on the left and the text "IDLE Shell 3.10.1" in the center. The main area is a text editor with a light gray background. It contains Python code with syntax highlighting: red for prompt characters, blue for function names, green for strings, and purple for print statements. The code defines a function print_hello() which sets a variable hello to "Hello, World!" and then calls a nested function print_message() which prints the value of hello. The function is then called, resulting in the output "Hello, World!". The bottom right corner of the window shows the status "Ln: 12 Col: 0".

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> def print_hello():
...     hello = "Hello, World!"
...     def print_message():
...         print(hello)
...     print_message()
...
...
>>> print_hello()
Hello, World!
>>> |
```

Ln: 12 Col: 0

A screenshot of the IDLE Shell 3.10.1 window. The window has a title bar with three colored buttons (red, yellow, green) on the left and the text "IDLE Shell 3.10.1" in the center. The main area is a text editor with a light gray background. It contains Python code with syntax highlighting: red for prompt characters, orange for keywords, blue for identifiers, and purple for function calls. The code defines a function 'foo' that uses a global variable 'x' and prints its value. The function is then called, and the variable 'x' is printed. The status bar at the bottom right shows "Ln: 15 Col: 0".

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> def foo():
...     global x
...     x = 20
...     print(x)
...
...
>>> foo()
20
>>> x
20
>>> print(x)
20
>>>
```

Ln: 15 Col: 0

A screenshot of the IDLE Shell 3.10.1 window. The window has a title bar with three colored buttons (red, yellow, green) on the left and the text "IDLE Shell 3.10.1" in the center. The main area contains a Python shell session. The first line is the Python version and build information: "Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin". The second line is a prompt followed by a message: "Type 'help', 'copyright', 'credits' or 'license()' for more information." The third line is a prompt followed by "x = 10". The fourth line is a prompt followed by "def foo():". The fifth line is a prompt followed by "x = 20". The sixth line is a prompt followed by "print(x)". The seventh line is a prompt followed by "foo()". The eighth line is a prompt followed by "20". The ninth line is a prompt followed by "print(x)". The tenth line is a prompt followed by "10". The eleventh line is a prompt followed by an empty line. The status bar at the bottom right shows "Ln: 13 Col: 0".

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> x = 10
>>> def foo():
...     x = 20
...     print(x)
...
>>> foo()
20
>>> print(x)
10
>>>
```

Ln: 13 Col: 0

A screenshot of the IDLE Shell 3.10.1 window. The window has a title bar with three colored buttons (red, yellow, green) on the left and the text "IDLE Shell 3.10.1" in the center. The main area contains a Python shell session. The session starts with the Python version and platform information: "Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin". It then shows a prompt "Type 'help', 'copyright', 'credits' or 'license()' for more information." followed by a function definition: ">>> def foo():", ">>> x = 10", ">>> print(x)", ">>>". Then the function is called: ">>> foo()", ">>> 10", ">>> print(x)". This is followed by a red traceback message: "Traceback (most recent call last):", " File "<pyshell#5>", line 1, in <module>", " print(x)", "NameError: name 'x' is not defined". The session ends with ">>>". At the bottom right of the window, the status bar shows "Ln: 15 Col: 0".

```
Python 3.10.1 (v3.10.1:2cd268a3a9, Dec 6 2021, 14:28:59) [Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> def foo():
...     x = 10
...     print(x)
...
>>> foo()
10
>>> print(x)
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    print(x)
NameError: name 'x' is not defined
>>>
```

Ln: 15 Col: 0