Python Programming



RGM College of Engineering & Technology (Autonomous)

Department of Computer Science & Engineering

EXCEPTION HANDLING - 5



Guido Van Rossum

Agenda:

- 1. else block with try-except-finally
- 2. else block with try-except-finally demo programs
- 3. Various possible combinations of try-except-else-finally

16. else block with try-except-finally

- ☐ This concept is Python specific concept.
- □ In Java, you can't use else block with try-catch-finally.

Where 'else' is used so far,

i. if - else:

□ If condition is false, then only else will be executed.

ii. for - else:

□ If loop executed without break then only else will be executed.

iii. while - else:

□ If loop executed without break then only else will be executed.

try - except - else - finally:

- We can use else block with try-except-finally blocks.
- else block will be executed if and only if there are no exceptions inside try block.

Syntax:

```
try:
    Risky Code
except:
    will be executed if exception inside try
else:
    will be executed if there is no exception inside try
finally:
    will be executed whether exception raised or not raised and handled or not handled
```

17. else block with try-except-finally demo programs

Demo Program 1:

```
try:
    print("try")
except:
    print("except")
else:
    print("else")
finally:
    print("finally")
```

try else finally

Demo Program 2:

```
try:
    print("try")
    print(10/0)
except:
    print("except")
else:
    print("else")
finally:
    print("finally")
```

try except finally

Demo Program 3:

```
try:
    print("try")
else:
    print("else")
finally:
    print("finally")

SyntaxError: invalid syntax
```

Key Observations:

- 1. There is no chance of executing both except and else blocks simultaneously.
- 2. If we want to take else block, compulsory except block should be there (i.e., else block without except block is invalid.)

SyntaxError: invalid syntax

Where exactly 'else' block is required?

Demo Program:

```
f = None
try:
    f = open('abc.txt','r')
except:
    print("Some Problem while locating and opening the specified file")
else:
    print('File opened successfully')
    print('The data present in the specified file is :')
    print(f.read())
    print('#'* 50)
finally:
    if f is not None:
        f.close()
```

Note: Now, I am creating abc.txt file in the present working directory.

```
f = None
try:
   f = open('abc.txt','r')
except:
   print("Some Problem while locating and opening the specified file")
else:
   print('File opened successfully')
   print('The data present in the specified file is :')
   print(f.read())
   print('#'* 70)
finally:
   if f is not None:
       f.close()
File opened successfully
The data present in the specified file is:
Ηi,
We are demonstrating 'else' block in try-except-finally block
Dept. of CSE, RGMCET(Autonomous), Nandyal
```

Test your Skills:

Q1. Which of the following is true about 'else' block?

- a) We can use else with try-except-finally blocks.
- b) else block will be executed if there is no exception inside the try block.
- c) There is no chance of executing both except and else blocks simultaneously.
- d) else block without except block is invalid because of SyntaxError.
- e) All of the above.

Ans: e

18. Various possible combinations of try-except-else-finally

Key Conclusions:

- 1. Whenever we are writing try block, compulsory we should write except or finally block (i.e., without except or finally block we cannot write try block).
- 2. Whenever we are writing except block, compulsory we should write try block (i.e., except without try is always invalid).
- 3. Whenever we are writing finally block, compulsory we should write try block (i.e., finally without try is always invalid).

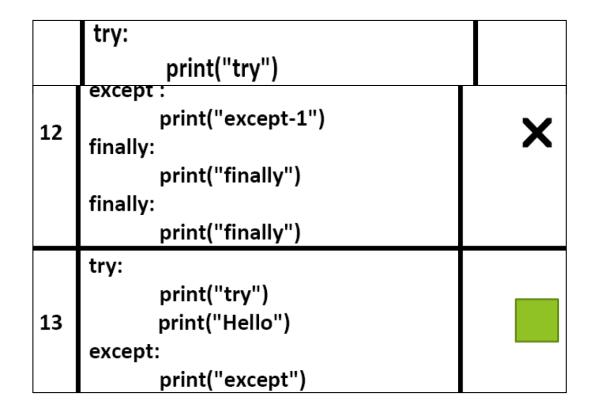
- 4. We can write multiple except blocks for the same try, but we cannot write multiple finally blocks for the same try.
- 5. Whenever we are writing else block compulsory except block should be there (i.e., without except we cannot write else block).
- 6. In try-except-else-finally order is important.
- 7. We can define try-except-else-finally inside try, except, else and finally blocks (i.e., nesting of try-except-else-finally is always possible).

Below Table illustrates the various possible combinations of try-except-else-finally blocks:

1	try: print("try")	×
2	except: print("Hello")	×
3	else: print("Hello")	×
4	finally: print("Hello")	×
	. ,	

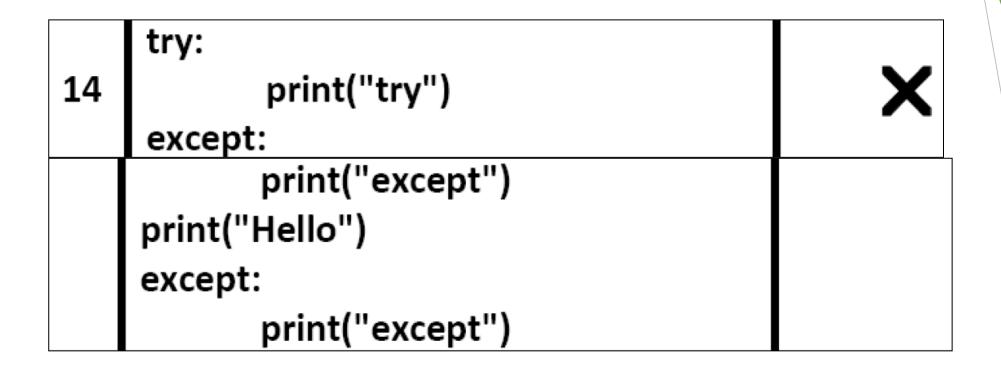
6	try: print("try") finally: print("finally")	1
7	try: print("try")	1
	except:	
	print("except")	
	else:	
	print("else")	
8	try:	
	print("try")	X
	else:	
	print("else")	

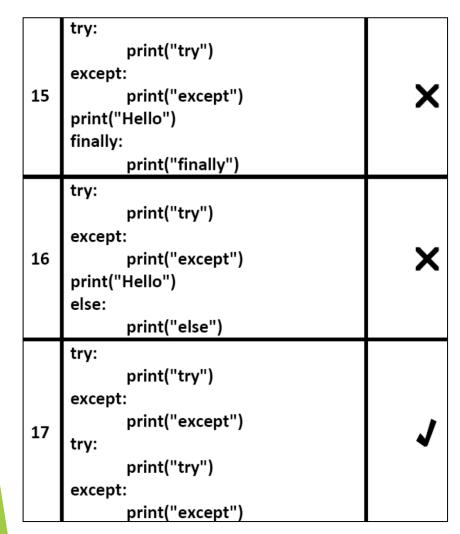
9	try: print("try") else: print("else") finally: print("finally")	×
10	try: print("try") except XXX: print("except-1") except YYY: print("except-2")	J
11	try: print("try") except: print("except-1") else: print("else") else: print("else")	×



```
try:
    print('try')
    print('Hello')
except:
    print('except')

try
Hello
```



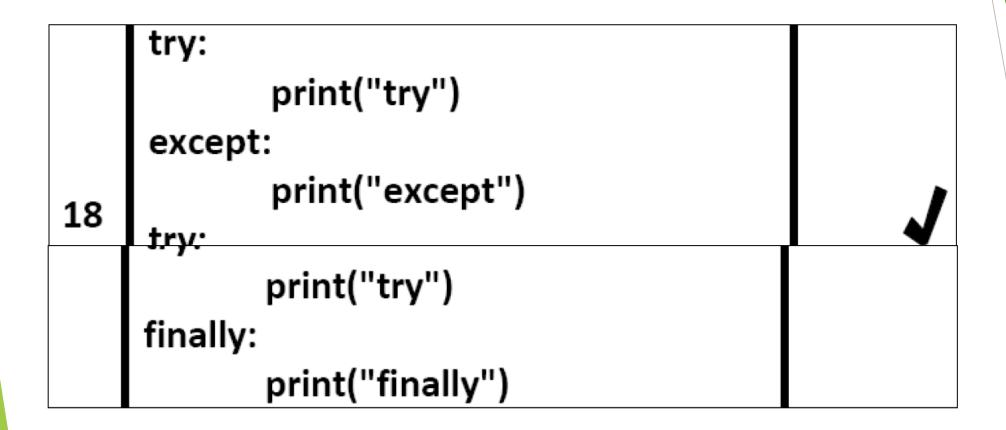


```
try:
    print('try')
print('Hello')
except:
    print('except')

# In between try and except blocks, you can't write any idependent statement.

File "<ipython-input-5-73a478649ae1>", line 3
    print('Hello')
    ^

SyntaxError: invalid syntax
```



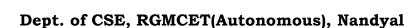
```
try:
    print("try")
    except:
    print("except")
    if 10>20:
        print("if")
    else:
        print("else")
```

Dept. of CSE, RGMCET(Autonomous), Nandyal

print("try") try: print("inner try") except: print("inner except block") finally: print("inner finally block") except: print("except")

try: print("try") except: print("except") 21 try: print("inner try") except: print("inner except block") finally: print("inner finally block")

print("try") except: print("except") finally: 22 try: print("inner try") except: print("inner except block") finally: print("inner finally block")



print("try") except: print("except") 23 print("try") else: print("else")



```
try:
    print("try")
    try:
    print("inner try")
    except:
    print("except")
```



print("try") else: print("else") . 25 except: print("except") finally: print("finally")



Some Programming proofs:

```
I.
            try:
                 print('try')
            finally:
                 print('finally') # Valid code
            try
            finally
II.
            try:
                print('try')
            else:
                print('else')
            except:
                print('except') # Invalid, order is important
              File "<ipython-input-7-3e4df3d694cc>", line 3
                else:
            SyntaxError: invalid syntax
```

III.

```
try:
    print('try')
except:
    print('except')
else:
    try:
        print('inner try')
    finally:
        print('inner finally') # Valid
```

```
try
inner try
inner finally
```

IV.

Any question?



If you try to practice programs yourself, then you will learn many things automatically

Spend few minutes and then enjoy the study

Thamk You