Python Raise, Try and Catch Exceptions

Fan Wang

2020-11-04

Contents

1 Exception Handling

Go to the **RMD**, **PDF**, or **HTML** version of this file. Go back to Python Code Examples Repository (bookdown site) or the pyfan Package (API).

1.1 A function That Raises an Error with Try Statement Catching it

Below, we have a function that will raise a *TypeError* unless we provide an integer input. The function is called with integer input and then called with a string input. The string input is wrapped in a try and except call, where the exception catches the TypeError and prints it.

```
# define the function
def ffi_error_test(gn_speckey=None):
    if isinstance(gn_speckey, int):
        print(f'{gn_speckey=} is an integer')
    else:
        raise TypeError(f'{gn_speckey=} is not an integer')
# Call function with integer
error_test(1)
# Call function with string
```

```
## gn_speckey=1 is an integer
```

```
try:
    ffi_error_test('abc')
except TypeError as error:
    print('Caught this error: ' + repr(error))
```

```
## Caught this error: TypeError("gn_speckey='abc' is not an integer")
```

1.2 Catch an Exception Noisily

Rather than only catching the last element of the exception, show the full trace. Notice in the example below, the second element of the loop does not work as function input. With exception catching, the loop continued despite the second element not working. The traceback shows more details at the end with full trace of the exception from the second element of the list as input for ffi_error_test().

```
import traceback
import numpy as np

ls_ob_inputs = [2, ['abc','efg'], 1, 123]

for ob_input in ls_ob_inputs:
    print(f'try input {ob_input=} with the error_test function:')
    try:
        ffi_error_test(ob_input)
    except TypeError as error:
        traceback.print_exc()
        print('Caught this error: ' + repr(error))
```

```
## try input ob_input=2 with the error_test function:
## gn_speckey=2 is an integer
## try input ob_input=['abc', 'efg'] with the error_test function:
## Caught this error: TypeError("gn_speckey=['abc', 'efg'] is not an integer")
## try input ob_input=1 with the error_test function:
## gn_speckey=1 is an integer
## try input ob_input=123 with the error_test function:
## gn_speckey=123 is an integer
##
## Traceback (most recent call last):
##   File "<string>", line 4, in <module>
##   File "<string>", line 5, in ffi_error_test
## TypeError: gn_speckey=['abc', 'efg'] is not an integer
```

see How to print the stack trace of an exception object in Python?

1.3 Handle Parameters When Conditions Not Satisfied

There is a function, that can estimate or simulate, under both functionalities, there is a common string parameter, that requires specifying estimation or simulation conditions. The common string parameter should be a simple string without special separators in the case of simulation, and should be four strings concatenated together with equal sign for estimation. Generate an exception if the function is called for estimation but the string parameter does not have the required structure.

- Python 3 TypeError
- Manually raising (throwing) an exception in Python

```
# ls_st_spec_key_dict = ['NG_S_D', 'NG_S_D=KAP_MO_NLD_M_SIMU=2=3']
# st_connector = '='
# ls_st_esti_simu = ['esti', 'simu']
# for st_spec_key_dict in ls_st_spec_key_dict:
#   for st_esti_simu in ls_st_esti_simu:
#     if st_esti_simu == 'simu':
#       if len(st_spec_key_dict.split(st_connector)) and
#         print('simulate with ' + st_spec_key_dict)

if estimate and not isinstance(spec_key_dict, str):
    pass
elif (estimate is False and isinstance(spec_key_dict, str)) or (estimate is False and isinstance(spec_k
    pass
else:
    st_error = 'speckey=' + speckey + ' and estimate=' + str(estimate)
```

```
    raise ValueError(st_error)
```

1.4 Proceed Despite Error

Sometimes, code should proceed despite error, to finish a loop for example:

```
# estimate at each initial random points
for it_esti_ctr in range(esti_param_vec_count):
    # Update the 3rd element of combo_type, which determines which draw index to use
    combo_type[3] = it_esti_ctr
    try:
        invoke_run_main.invoke_main(combo_type, **dc_invoke_main_args)
    except Exception:
        logging.critical(f'Finished this {it_esti_ctr=} of {range(esti_param_vec_count)=}')
```