3.9.2. Raise

Fig. 3.9.2.1 Photo by Photos by Lanty on Unsplash

Note

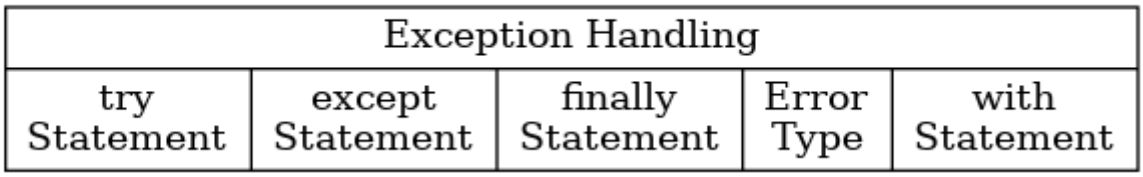
Outline

1. Introduction
2. Examples
    a. Ex1: Basic Usage
    b. Ex2: Built-in Exceptions
    c. Ex3: Handle Exceptions
    d. Ex4: Re-raise Exceptions
    e. Ex5: Custom Exceptions

Note

Roadmap

1. This topic: TryExcept

> myBlock
>
> 
>

2. Course: Python 1
3. Subject: Programming
4. Field
    a. Software Engineering (SE)
    b. Computer Science and Information Engineering (CSIE)
    c. Electrical/Electronics Engineering (EE)

3.9.2.1. Introduction

1. The raise statement in Python is used to **manually** raise exceptions.
2. This allows encapsulating error handling logic cleanly within try/except blocks.

3.9.2.2. Examples

1. There are at least 5 type of usages such as
    a. Basic usage
    b. Raise built-in exceptions
    c. Handle raised exceptions
    d. Re-raise raised exceptions
    e. Raise custom exceptions

3.9.2.2.1. Ex1: Basic Usage

1. This raises a ValueError with the given error message.

```
1  raise ValueError('Invalid value...')
```

2. Code+Output

> Code Output

Listing 3.9.2.2.1.1 /src/ExceptionHandling/Raise/__init__.py

```
'''
author: CPH
since:  20230826
'''

def isNum(i):
    try:
        complex(i)
        return True
    except:
        return False

if __name__ == '__main__':
    while (1):
        i = input('Please input number: ')
        if isNum(i):
            print(f'It is number')
        else:
            print(f'Inside Error: This line will be shown...')
            raise ValueError('Found: Not number...')
            print(f'Inside Error: This line will NOT be shown...')
```

3.9.2.2.2. Ex2: Built-in Exceptions

1. Raise Built-in Exceptions

    a. Common built-in exceptions like TypeError, ValueError, etc. can be raised.

```
raise TypeError('Argument must be a string...')
```

2. Code+Output

__init__.py | value.py | index.py | file.py | Output

Listing 3.9.2.2.2.2 /src/ExceptionHandling/Raise2/value.py

```
'''
author: cph
since:  20230827
'''
# Raising a ValueError
def tryValue():
    try:
        iVal = int(input('Enter a negative value: '))
        if iVal >= 0:
            raise ValueError('Value cannot be positive...')
    except ValueError as ve:
        print(f'ValueError: {ve}')

if __name__ == '__main__':
    tryValue()
```

3.9.2.2.3. Ex3: Handle Exceptions

1. raise manually raises an exception which can be handled with try/except.

```
try:
    raise RuntimeError('Error...')
except RuntimeError as err:
    print(err)
```

2. Code+Output

Code | Output

Listing 3.9.2.2.3.1 /src/ExceptionHandling/Raise3/__init__.py

```
'''
author: CPH
since:  20230826
'''

if __name__ == '__main__':
    i = 10
    for j in range(3, -4, -1):
        try:
            if (j == 0):
                raise ZeroDivisionError('Found: Division Zero...')
            print("%d / %d = %0.3f" %(i, j, i / j))
        except ZeroDivisionError as err:
            #print("除數為0，無法進行除法。")
            print(err)
```

3.9.2.2.4. Ex4: Re-raise Exceptions

1. Exceptions can be re-raised after handling.

```
try:
    # code
except:
    # log error

    # re-raise last exception
    raise ValueError('Invalid value...')
```

2. Code+Output

Code | Output

Listing 3.9.2.2.4.1 /src/ExceptionHandling/Raise4/__init__.py

```
'''
author: CPH
since:  20230826
'''
# Raising one exception from another exception
if __name__ == '__main__':
    iVal = 1.0          # Want an integer, but give a float number

    try:
```

```
        if iVal < 10:   # Want >= 10, but lower
            raise ValueError("Initial value: Want >= 10, but lower...")
    except ValueError as ve:
        print(f"Value Error: {ve}")

        try:
            if not isinstance(iVal, int):   # Want integer, but others
                raise TypeError("Initial value: Want integer, but others...")
        except TypeError as te:
            print(f"Type Error: {te}")
```

3.9.2.2.5. Ex5: Custom Exceptions

1. Custom exception classes can also be defined and raised.

```
class MyError(Exception):
    pass

raise MyError('Something wrong...')
```

2. Code+Output: Type 1

Code | Output

Listing 3.9.2.2.5.1 /src/ExceptionHandling/Raise5/__init__.py

```
'''
author: cph
since:  20230827
'''

if __name__ == '__main__':
    class CustomException(Exception):
        print(f'We can do something here...')

    try:
        raise CustomException('This is a custom exception...')
    except CustomException as ce:
        print(f'CustomException: {ce}')
```

3. Code+Output: Type 2

Code | Output

Listing 3.9.2.2.5.2 /src/ExceptionHandling/Raise5b/__init__.py

```
'''
author: cph
since:  20230827
'''
class CustomException(Exception):
    print(f'We can do something here...')

if __name__ == '__main__':
    try:
        raise CustomException('This is a custom exception...')
    except CustomException as ce:
        print(f'CustomException: {ce}')
```

Note

1. Start: 20170719
2. System Environment:

Listing 3.9.2.2.5.3 requirements.txt

```
sphinx==7.1.2                           # Sphinx
graphviz>=0.20.1                        # Graphviz
sphinxbootstrap4theme>=0.6.0            # Theme: Bootstrap
sphinx-material>=0.0.35                 # Theme: Material
sphinxcontrib-plantuml>=0.25            # PlantUML
sphinxcontrib.bibtex>=2.5.0             # Bibliography
sphinx-autorun>=1.1.1                   # ExecCode: pycon
sphinx-execute-code-python3>=0.3        # ExecCode
btd.sphinx.inheritance-diagram>=2.3.1   # Diagram
sphinx-copybutton>=0.5.1                # Copy button
sphinx_code_tabs>=0.5.3                 # Tabs
sphinx-immaterial>=0.11.3               # Tabs

#-----------------------------------------------------
#-- Library Upgrade Error by Library Itself
#   >> It needs to fix by library owner
#   >> After fixed, we need to try it later
#-----------------------------------------------------
pydantic==1.10.10                       # 2.0: sphinx compiler error, 20230701

#-----------------------------------------------------
#-- Minor Extension
#-----------------------------------------------------
sphinxcontrib.httpdomain>=1.8.1         # HTTP API

#sphinxcontrib-blockdiag>=3.0.0         # Diagram: block
#sphinxcontrib-actdiag>=3.0.0           # Diagram: activity
#sphinxcontrib-nwdiag>=2.0.0            # Diagram: network
#sphinxcontrib-seqdiag>=3.0.0           # Diagram: sequence

#-----------------------------------------------------
#-- Still Wait For Upgrading Version
#-----------------------------------------------------

#-----------------------------------------------------
#-- Still Under Testing
#-----------------------------------------------------
#numpy>=1.24.2                          # Figure: numpy

#-----------------------------------------------------
#-- NOT Workable
#-----------------------------------------------------
#sphinxcontrib.jsdemo==0.1.4    # ExecCode: Need replace add_js_file()
#jupyter-sphinx==0.4.0          # ExecCode: Need gcc compiler
#sphinxcontrib.slide==1.0.0     # Slide: Slideshare
#hieroglyph==2.1.0              # Slide: make slides
#matplotlib>=3.7.1              # Plot: Need Python >= v3.8
#manim==0.17.2                  # Diagram: scipy, numpy need gcc
#sphinx_diagrams==0.4.0         # Diagram: Need GKE access
#sphinx-tabs>=3.4.1                     # Tabs: Conflict w/ sphinx-material
```