

みんなのPython

10. 例外処理

山木翔馬

Pythonの例外

- 例外の発生
 - 例外オブジェクトが作られる
 - 親クラスは Exception
- 例外を捕まえる
 - try ~ except 文
- 例外を発生させる
 - raise ValueError("Error message")

例外処理の書式

書式	説明
except:	全ての例外を受け取り、例外発生時の処理を行う。
except 例外クラス名:	クラスを指定して、特定の例外だけを受け取る。
except 例外クラス名 as 変数名:	例外クラスと、例外オブジェクトを受け取る変数名を指定する。
else:	例外が発生しなかった場合の処理を記述したい場合に利用する。
finally:	例外が発生してもしなくても、実行するブロックを記述するときを利用する。

例外処理のコード例

```
import sys
```

```
for fn in sys.argv[ 1: ] :
```

```
    try :
```

```
        f = open( fn )
```

```
    except FileNotFoundError :
```

```
        print( "{ }というファイルは存在しません".format( fn ) )
```

```
    else :
```

```
        try :
```

```
            print( fn, len( f.read() ) )
```

```
        finally :
```

```
            f.close()
```

with文

```
with open( fn ) as f :  
    for line in f :  
        print( line )
```

- ファイルを開くことができればブロック内の処理を行う
- ブロック内でエラーが発生した場合でも `f.close()` が必ず行われる
- with文に添えるオブジェクトはコンテキストマネージャに対応したもの(`__enter__`, `__exit__` メソッドを持っている必要がある)

トレースバックの表示

- tracebackモジュールを使う

```
import traceback
```

```
try :
```

```
    #処理を実行するコード
```

```
except :
```

```
    traceback.print_exc()          #例外を表示
```

```
    ex = traceback.format_exc()    #文字列として取り出す
```

よく起こるエラー

エラー名	
AttributeError	オブジェクトに定義されていないアトリビュートを参照しようとしたとき、またはアトリビュートの追加が許されていないオブジェクトに対してアトリビュートを追加しようとしたときに発生。
IndexError	リストなどのシーケンスをインデックスで参照しようとしたとき、シーケンスの要素数を超える数を指定した場合などに発生。
KeyError	ディクショナリオブジェクトの要素をキーで参照するとき、存在しないキーを指定すると発生する。
UnicodeDecodeError UnicodeEncodeError	文字列やバイト型のデコード、エンコード中にエラーが起こった場合に発生する。

練習問題

a / b を返す関数 `division(a, b)` を定義し、引数の型によるエラーを処理せよ。

```
>>> division(10, 2)
```

```
5
```

```
>>> division(10, 0)
```

```
0では割れません。
```

```
>>> division(10, "2")
```

```
引数の型が正しくありません。
```