



Ngoại lệ trong Python

Nội dung

1. Exception là gì?
2. Xử lý ngoại lệ bằng Try-except
3. Try-except-finally
4. Bài tập thực hành

Exception

- ❑ **Exception** - Ngoại lệ có thể là bất kỳ điều kiện **bất thường** nào trong chương trình mà phá vỡ luồng thực thi chương trình đó. Bất cứ khi nào một ngoại lệ xuất hiện, mà không được xử lý, thì chương trình ngừng thực thi và vì thế code không được thực thi.
- ❑ Python đã định nghĩa sẵn rất nhiều ngoại lệ. Trong chương này chúng ta sẽ tìm hiểu cách xử lý ngoại lệ cũng như cách tạo các **Custom Exception** như thế nào.

❖ Có thể hiểu:

- Ngoại lệ là những điều '**bất thường**' nằm ngoài luồng đúng của các khối lệnh.
- Những điều '**bất thường**' này có thể do Python định nghĩa sẵn. Nhưng cũng có thể do lập trình viên tự định nghĩa ra: thế nào là '**bất thường**' đối với các khối code của mình.

Xử lý ngoại lệ

- ❑ Khi lập trình viên cảm thấy bất cứ đoạn code nào là khả nghi (có thể gây ra ngoại lệ) thì bạn có thể phòng thủ chương trình của mình bằng cách đặt các khối code khả nghi này trong một khối try.
- ❑ Khối **try** này được theo sau bởi lệnh **except**. Sau đó, nó được theo sau bởi các lệnh mà xử lý vấn đề đó.

Cú pháp đơn giản

try:

Ban thực hiện các hoạt động của mình tại đây;

Va đây là phần code có thể tạo exception;

.....

except:

Neu có bất kỳ exception nào, thì thực thi khối code này

.....

else:

Neu không có exception nào, thì thực thi khối code này

Cú pháp đã xác định loại except

try:

Ban thực hiện các hoạt động của mình tại đây;

Va đây là phần code có thể tạo exception;

.....

except *ExceptionI*:

Neu có **ExceptionI**, thì thực thi khối code này

except *ExceptionII*:

Neu có **ExceptionII**, thì thực thi khối code này

.....

else:

Neu không có exception nào thì thực thi khối code này

Lưu ý

- ❑ Phần code khả nghi mà có khả năng tạo exception cần được bao quanh trong khối **try**.
- ❑ Khối try được theo sau bởi lệnh **except**. Có thể có một hoặc nhiều lệnh except với một khối try đơn.
- ❑ Lệnh except xác định exception mà xảy ra. Trong trường hợp mà exception đó xảy ra, thì lệnh tương ứng được thực thi.
- ❑ Ở cuối khối try, bạn có thể cung cấp lệnh else. Nó được thực thi khi không có exception nào xảy ra. Khối else là địa điểm tốt cho code mà không cần sự bảo vệ của khối try.


```
print ("Three")
```

```
# Phép chia này không vấn đề.  
value = 10 / 2
```

```
print ("Two")
```

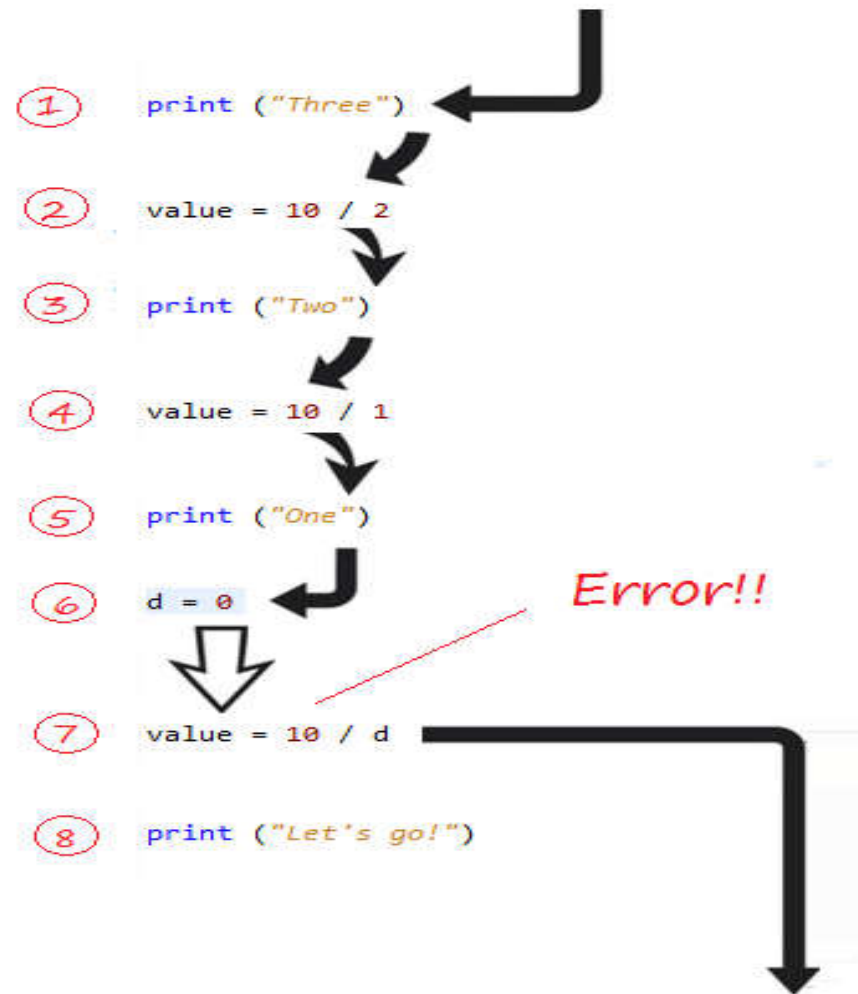
```
# Phép chia này không vấn đề.  
value = 10 / 1
```

```
print ("One")
```

```
d = 0
```

```
# Phép chia này có vấn đề, chia cho 0  
# Một lỗi được phát ra tại đây.  
value = 10 / d
```

```
# Dòng mã dưới đây sẽ không được thực thi.  
print ("Let's go!")
```



```
print ("Three")
value = 10 / 2
print ("Two")
value = 10 / 1
print ("One")
```

```
d = 0
```

```
try :
```

```
# Phép chia này có vấn đề, chia cho 0.
```

```
# Một lỗi được phát ra tại đây (ZeroDivisionError).
```

```
value = 10 / d
```

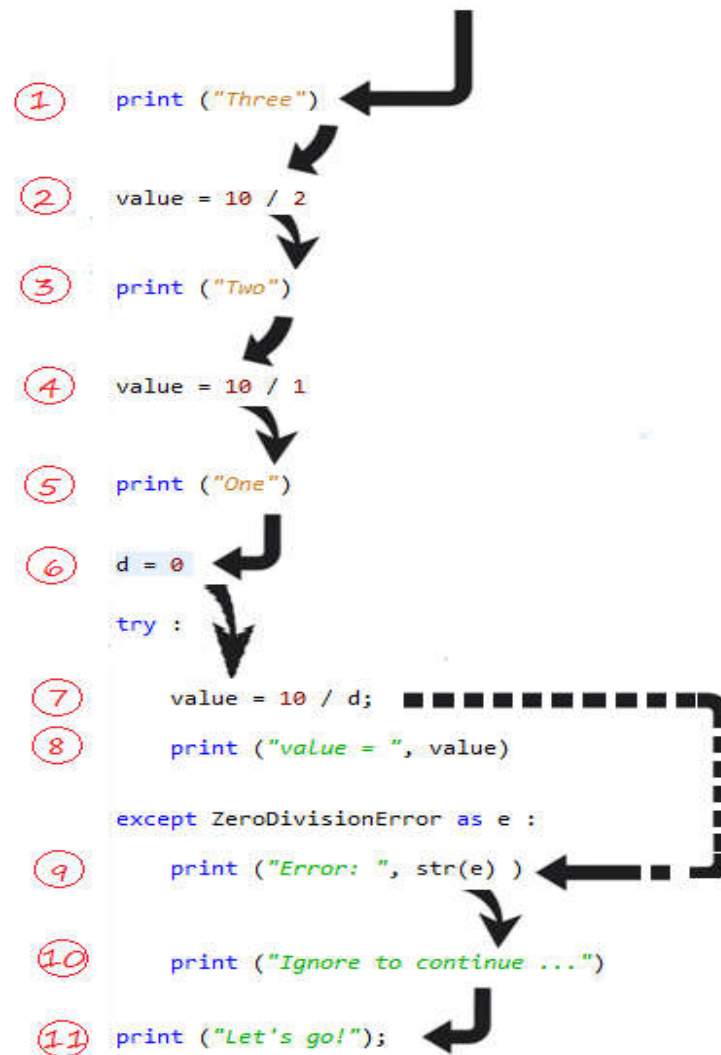
```
print ("value = ", value)
```

```
except ZeroDivisionError as e :
```

```
print ("Error: ", str(e) )
```

```
print ("Ignore to continue ...")
```

```
print ("Let's go!")
```



Tham số trong ngoại lệ

❑ Một Exception có thể có một tham số, mà là một giá trị mà cung cấp thông tin bổ sung về vấn đề. Nội dung của tham số là đa dạng tùy vào các exception.

❑ Cú pháp:

```
try:  
    Ban thực hiện các hoạt động của mình tại đây;  
    Va đây là phần code có thể tạo exception;  
.....  
except Kieu_exception, Tham_so:  
    Ban có thể in giá trị của Tham_so tại đây...
```

```
# Định nghĩa hàm convert một đầu vào thành 1 số
def temp_convert(var):
    try:
        return int(var)
    except ValueError:
        print("Tham số truyền vào không phải là số.")

# Gọi hàm
temp_convert("xyz")
```

Try-except-finally

❑ Việc xử lý ngoại lệ đầy đủ là **try-except-finally**. Khối **finally** luôn được thực thi bất kể ngoại lệ có xảy ra tại khối **try** hay không.

❑ Cú pháp:

```
try :  
    # Làm gì đó tại đây.  
except Exception1 as e :  
    # Làm gì đó tại đây.  
except Exception2 as e :  
    # Làm gì đó tại đây.  
finally :  
    # Khối finally luôn luôn được thực thi.  
    # Làm gì đó tại đây.
```

```
def temp_convert(var):  
    try:  
        return int(var)  
    except ValueError:  
        print("Tham số truyền vào không phải là số: ", var)  
    finally:  
        print("Khôi finally vẫn được thực hiện: ", var)  
  
# Gọi hàm  
temp_convert("xyz")
```

THANKS FOR ATTENDING!