

تلاش برای کرش نکردن

استثنا (Exception)

کد ما از هیچ خطایی مستثنا نیست پس ما باید حواسمان باشد که این خطاها منجر به کار نکردن کد نشود پایتون روشی به ما پیشنهاد داده تا در صورت رخ داد خطا کد بسته نشود و ادامه پیدا کند در زیر نمونه هایی از این خطاها که منجر به پایان برنامه میشود را مشاهده میکنید :

```

1  >>> x = 1/0
2  Traceback (most recent call last):
3    File "<stdin>", line 1, in <module>
4  ZeroDivisionError: division by zero
5  >>> print(y)
6  Traceback (most recent call last):
7    File "<stdin>", line 1, in <module>
8  NameError: name 'y' is not defined
9  >>> z = int('salam')
10 Traceback (most recent call last):
11   File "<stdin>", line 1, in <module>
12 ValueError: invalid literal for int() with base 10: 'salam'

```

به طور کلی ساختار خروجی استثناها توسط Interpreter به این ترتیب است که ابتدا مشخص میکند که در کدام بخش برنامه این خطا رخ داده و stack traceback برنامه در آن زمان به چه شکل بوده است. در خط آخر نیز نوع استثنا به همراه یک توضیح کوتاه بعنوان دلیل رخ دادن خطا نمایش داده خواهد شد.

مثال جلوگیری از کرش برنامه

گاهی اوقات ممکن است برنامه ما هنگام اجرا با خطا مواجه شود. برای جلوگیری از توقف ناگهانی، از try-except استفاده میکنیم. ورودی اگر عدد نباشد وارد بخش except میشود. میتوانیم جلوی except یک نوع خطا را مشخص کنیم و اگر از این خطا باشد وارد بخش except میشود در غیر این صورت برنامه کرش میکند.

```

1  try:
2      x = int(input("Enter a number: "))

```

```
3 |     print("Your number is:", x)
4 | except ValueError:
5 |     print("Invalid input! Please enter a valid number.")
```

میتوانید جلوی `except` هیچ دلیل برای خطا ننویسید آنگاه هر خطای که رخ دهد وارد اکسپت میشود. همچنین اگر در پایین `except` از `else` استفاده کنید در صورت رخ ندادن خطا دستورات داخل `else` اجرا میشود و همچنین کلید واژه `finally` هم هست که همیشه چه خطا رخ بدهد چه ندهد اجرا میشود. به عنوان مثال:

```
1 | try:
2 |     f = open("example.txt", "r")
3 |     content = f.read()
4 |     print(content)
5 | except FileNotFoundError:
6 |     print("File not found!")
7 | else:
8 |     print("File read successfully.")
9 | finally:
10 |     if 'f' in locals():
11 |         f.close()
12 |     print("Execution finished.")
```