

کلیات استثناگردانی

سیستم مدیریت استثناها و خطاها در پایتون شبیه به زبان‌های دیگر است.

- نکته: استثناها با خطاهای دستوری، متفاوت هستند و در زمان اجرا رخ می‌دهند.
- همه‌ی استثناها در پایتون از کلاس `Exception` ارث‌بری می‌کنند. (نه خطاها)

برخی از خطاهای رایج در پایتون

نام خطا	توضیح
<code>ImportError</code>	مفسر نمی‌تواند ماژولی را پیدا کند.
<code>IOError</code>	اشکالی در باز شدن فایل وجود دارد.
<code>ValueError</code>	آرگومان تابع یا عملگر نوع درست ولی مقدار غیرقابل قبولی دارد.
<code>EOFError</code>	فایل به اتمام رسیده است.

پرتاب استثنا

هنگامی که در بخشی از کد می‌خواهید استثنائی پرتاب کنید، می‌توانید از کلمه‌ی `raise` استفاده کنید.

```
1 | raise Exception('x should not exceed 5. The value of x was: {}'.format(x))
```

کلاس `Exception` و اکثر زیرکلاس‌های معروف آن، در تابع سازنده خود آرگومانی دارند که توضیحات خطای پرتاب شده را می‌توانید در آن قرار دهید و زمانی که استثنا رخ دهد، می‌توانید آن خطا را ببینید.

بلوک try-except

برای این که جلوی استثناهای پرتاب شده را بگیرید تا برنامه را قطع نکنند، می‌توانید از بلوک `try-except` استفاده کنید. اگر کدی که داخل `try` هست، استثنایی پرتاب کند برنامه قطع نمی‌شود؛ بلکه مفسر به قسمت‌های `except` می‌رود تا اگر آن نوع استثنا پیش‌بینی شده، وارد آن قسمت شود.

- اگر هیچ خطایی رخ ندهد وارد قسمت `else` می‌شود.
- همواره پس از اجرای کامل قسمت‌های دیگر بلوک `finally` اجرا می‌شود.
- می‌توان چند نوع خطا یا استثنا را در یک بلوک `except` قرار داد.
- اگر بخواهیم بلوکی را در پایتون خالی بگذاریم و در آن دستوری نباشد، باید کلمه‌ی `pass` را در آن بلوک قرار دهیم.

```
1 | try:  
    # Use "raise" to raise an error
```

```
2     raise IndexError("Index Error Description")
3 except IndexError as e:
4     print(e)                #Print e Description
5 except (TypeError, NameError):
6     pass                    # Multiple exceptions can be handled together, if required.
7 else:                       # Optional clause to the try/except block. Must follow all except blocks
8     print("All good!")      # Runs only if the code in try raises no exceptions
9 finally:                    # Execute under all circumstances
10    print("We can clean up resources here")
11
```