

ایجاد استثنای سفارشی با raise در پایتون

گاهی اوقات در برنامه‌های خود نیاز داریم که خطاهای خاصی را مدیریت کنیم که در خطاهای پیش فرض پایتون وجود ندارند. در این موارد، می‌توانیم از `raise` برای ایجاد استثنای سفارشی (Custom Exceptions) استفاده کنیم.

1. استفاده از raise برای ایجاد استثنا

پایتون به ما اجازه می‌دهد که با استفاده از `raise` یک استثنا را به صورت دستی ایجاد کنیم.

مثال 1: ایجاد یک خطای سفارشی با raise

```
age = int(input("لطفاً سن خود را وارد کنید: "))

if age < 18:
    raise ValueError("خطا: سن شما باید حداقل 18 سال باشد")
else:
    print("ثبت نام شما با موفقیت انجام شد")
```

◆ توضیح:

- اگر مقدار وارد شده کمتر از ۱۸ باشد، یک خطای `ValueError` با پیام سفارشی نمایش داده می‌شود.
- در غیر این صورت، پیام موفقیت نشان داده می‌شود.

مدیریت خطای ایجاد شده

اگر از `raise` استفاده کنیم، برنامه در صورت بروز خطا متوقف می‌شود. برای جلوگیری از این اتفاق، می‌توان از `try-except` استفاده کرد:

```
try:
    age = int(input("لطفاً سن خود را وارد کنید: "))

    if age < 18:
        raise ValueError("خطا: سن شما باید حداقل 18 سال باشد")
    else:
        print("ثبت نام شما با موفقیت انجام شد")

except ValueError as e:
    print(f"یک خطا رخ داد: {e}")
```

◆ توضیح:

- اگر ورودی نامعتبر باشد (مانند یک مقدار غیرعددی)، `ValueError` اجرا شده و پیام مناسب نمایش داده می‌شود.
- اگر مقدار کمتر از ۱۸ باشد، خطای `raise` اجرا می‌شود.

2. تعریف استثنای سفارشی با کلاس‌ها

می‌توانیم استثنای سفارشی خود را با ایجاد کلاس‌هایی که از `Exception` ارث‌بری می‌کنند، تعریف کنیم.

مثال 2: تعریف یک کلاس خطای سفارشی

```
class AgeTooLowError(Exception):
    """استثنای سفارشی برای سن کمتر از حد مجاز"""
    def __init__(self, age, message="!سن شما باید حداقل 18 سال باشد"):
        self.age = age
        self.message = message
        super().__init__(self.message)

# استفاده از استثنای سفارشی
try:
    age = int(input("لطفاً سن خود را وارد کنید: "))

    if age < 18:
        raise AgeTooLowError(age) # استفاده از استثنای سفارشی
    else:
        print(".ثبت نام شما موفقیت آمیز بود")

except AgeTooLowError as e:
    print(f"خطای ثبت نام: {e}")
except ValueError:
    print("لطفاً یک عدد صحیح وارد کنید")
```

توضیح:

- یک کلاس است که از `Exception` ارث‌بری کرده است.
- اگر سن کمتر از ۱۸ باشد، یک نمونه از این استثنا با پیام مناسب `raise` می‌شود.
- این خطا را گرفته و پیام مربوطه را نمایش می‌دهد.

3. ارسال پیام‌های خطای مفید برای رفع اشکال

بهتر است هنگام ایجاد استثنای سفارشی، اطلاعات بیشتری به پیام خطا اضافه کنیم تا اشکال‌زدایی آسان‌تر شود.

مثال 3: ارائه جزئیات بیشتر در پیام خطا

```
class NegativeNumberError(Exception):
    """استثنای سفارشی برای جلوگیری از ورود اعداد منفی"""
    def __init__(self, number):
        self.number = number
        self.message = f"مجاز نیست! لطفاً عدد مثبت وارد کنید {number} عدد"
        super().__init__(self.message)

# استفاده از استثنا
try:
    num = int(input("یک عدد مثبت وارد کنید: "))

    if num < 0:
        raise NegativeNumberError(num)
    else:
        print(f".مورد قبول است {num} عدد")

except NegativeNumberError as e:
```

```
print(e)
except ValueError:
    print("لطفأً یک عدد صحیح وارد کنید")
```

◆ توضیح:

- `NegativeNumberError` پیام مشخصی درباره مقدار نامعتبر ارائه می‌دهد.
- اگر مقدار وارد شده منفی باشد، پیام مشخصی درباره مقدار ورودی نشان داده می‌شود.

4. مدیریت چندین خطای سفارشی همزمان

اگر بخواهیم چندین نوع خطای سفارشی را مدیریت کنیم، می‌توانیم چند `except` استفاده کنیم.

مثال 4: چندین استثنای سفارشی

```
class AgeTooLowError(Exception):
    """استثنا برای سن کمتر از 18"""
    pass

class AgeTooHighError(Exception):
    """استثنا برای سن بیشتر از 100"""
    pass

try:
    age = int(input("سن خود را وارد کنید: "))

    if age < 18:
        raise AgeTooLowError("سن شما کمتر از 18 سال است، ثبت نام ممکن نیست")
    elif age > 100:
        raise AgeTooHighError("سن شما بیش از 100 سال است، مقدار وارد شده غیرمنطقی است")
    else:
        print("ثبت نام موفقیت آمیز بود.")

except AgeTooLowError as e:
    print(f"خطای سنی: {e}")
except AgeTooHighError as e:
    print(f"خطای سنی: {e}")
except ValueError:
    print("لطفأً یک عدد صحیح وارد کنید")
```

◆ توضیح:

- `AgeTooLowError` برای سن کمتر از ۱۸
- `AgeTooHighError` برای سن بالاتر از ۱۰۰
- برنامه بسته به مقدار ورودی، پیام مناسب را نمایش می‌دهد.

جمع‌بندی

- ✅ با `raise` می‌توانیم خطاهای خاصی را در برنامه خود ایجاد کنیم.
- ✅ با تعریف کلاس‌های سفارشی از `Exception` می‌توانیم خطاهای مخصوص به خود را مدیریت کنیم.
- ✅ پیام‌های خطا را می‌توان به **اطلاعات مفید** مجهز کرد تا اشکال‌زدایی راحت‌تر شود.
- ✅ ترکیب `try-except` با استثنای سفارشی، برنامه را در برابر ورودی‌های نامعتبر و شرایط پیش‌بینی‌نشده مقاوم‌تر می‌کند.

🎯 تمرین:

یک کلاس استثنای سفارشی برای بررسی اعتبار **رمز عبور** ایجاد کنید که اگر رمز عبور کمتر از ۸ کاراکتر بود، خطا بدهد.