جلسه ۱۰ – مدیریت خطاها و استثناها

در این جلسه با **مدیریت خطاها و استثناها (Exceptions)** در پایتون آشنا می‌شویم.
مدیریت صحیح خطاها باعث می‌شود برنامه ما در شرایط غیرمنتظره متوقف نشود.

مفهوم Exception

در پایتون، وقتی خطایی رخ می‌دهد (مانند تقسیم بر صفر یا دسترسی به اندیسی که وجود ندارد)، یک **Exception** ایجاد می‌شود. اگر این استثنا مدیریت نشود، اجرای برنامه متوقف خواهد شد.

In [ ]:
```
# نمونه خطا
x = 10
y = 0
print(x / y)  # ZeroDivisionError ایجاد می‌شود
```

استفاده از `try`, `except`, `finally`

برای جلوگیری از توقف برنامه می‌توانیم از بلوک‌های `try`, `except` و در صورت نیاز `finally` استفاده کنیم.

In [ ]:
```
try:
    x = 10
    y = 0
    result = x / y
except ZeroDivisionError:
    print("خطا: تقسیم بر صفر مجاز نیست.")
finally:
    print("این بخش همیشه اجرا می‌شود.")
```

ایجاد خطای سفارشی با `raise`

می‌توانیم در شرایط خاص، خطاهای سفارشی ایجاد کنیم.

In [ ]:
```
def check_age(age):
    if age < 0:
        raise ValueError("سن نمی‌تواند منفی باشد.")
    return f"سن شما {age} سال است."

print(check_age(25))
print(check_age(-5))  # ایجاد خطای سفارشی
```

تمرین: ماشین حساب مقاوم در برابر ورودی اشتباه

برنامه‌ای بنویسید که دو عدد و یک عملگر (`+` , `-` , `*` , `/`) از کاربر بگیرد و نتیجه را نمایش دهد. برنامه باید در برابر ورودی‌های اشتباه (مانند متن به جای عدد، تقسیم بر صفر یا عملگر نامعتبر) مقاوم باشد.

In [ ]:
```
def safe_calculator():
    try:
        a = float(input("عدد اول را وارد کنید: "))
        b = float(input("عدد دوم را وارد کنید: "))
        op = input("عملگر را وارد کنید (+, -, *, /): ")

        if op == "+":
            result = a + b
        elif op == "-":
            result = a - b
        elif op == "*":
            result = a * b
        elif op == "/":
            if b == 0:
                raise ZeroDivisionError("تقسیم بر صفر مجاز نیست.")
            result = a / b
        else:
            raise ValueError("عملگر نامعتبر است.")

        print("نتیجه:", result)

    except ValueError as ve:
        print("خطای ورودی:", ve)
    except ZeroDivisionError as zde:
        print("خطا:", zde)
    except Exception as e:
        print("یک خطای ناشناخته رخ داد:", e)
    finally:
        print("پایان اجرای ماشین حساب.")

# اجرای ماشین حساب
# safe_calculator()
```