

انواع خطا

۱. خطاهای دستوری یا نوشتاری (syntax errors): مانند املای اشتباه یک دستور، فراموش کردن: در انتهای if و for و استفاده نادرست از ساختار یک دستور و غیره. در صورتی که برنامه شامل چنین خطاهایی باشد با پیام خطا error مواجه می شویم و با دنبال کردن شماره خط حاوی خطا نسبت به رفع آن اقدام می کنیم.
مثال: در برنامه زیر چند خطای دستوری وجود دارد، آنها را مشخص کنید:

```
n=int(input())
s=0
for i in range(1,n+1)
    if n%i = 0 :
        s=s+i
if s=2n:
    print("ok")
else
    print("no")
```

پراتنز

:

==

2*n

:

۲. خطاهای زمان اجرا (runtime errors): این خطاها در زمان اجرای برنامه رخ می دهند. در قطعه برنامه زیر چنانچه کاربر عدد دوم را صفر وارد کند خطای تقسیم بر صفر اتفاق می افتد. برنامه نویس باید اینگونه خطاها را پیش بینی کند و طوری برنامه را بنویسد که اینگونه خطاها اتفاق نیفتند.

```
n=int(input())
m=int(input())
print(n/m)
```

برنامه بهتر است به اینصورت نوشته شود

```
n=int(input())
m=int(input())
if m!=0:
    print(n/m)
else:
    print("عدد دوم نباید صفر باشد")
```

۳. خطاهای منطقی یا مفهومی (Symantec errors): در مورد این خطاها، هشدار یا پیامی از طرف مترجم پایتون صادر نمی شود اما خروجی آن، چیزی که کاربر انتظار دارد نیست. برای رفع این خطاها می توان از **جدول تعقیب** یا **print اضافی** در برنامه کمک گرفت.

مثال: در برنامه زیر که برای محاسبه میانگین سه عدد ورودی نوشته شده است، خطای منطقی وجود دارد آنرا بیابید

```
a=float(input())
b=float(input())
c=float(input())
average=a+b+c/3
print("Average=",average)
```

تمرین: در برنامه زیر که برای محاسبه مقلوب عدد ورودی نوشته شده است، چه نوع خطایی وجود دارد؟ آنرا برطرف کنید.

```
n=int(input())
m=0
while n!=0:
    r=n%10
    m=m+r*10
    n=n//10
print(m)
```

درج توضیحات در برنامه

برای تشریح عملکرد یک دستور یا یک متغیر می توان توضیحاتی به برنامه اضافه کرد. توضیحات توسط مترجم پایتون نادیده گرفته می شود و آنرا اجرا نمی کند. در ابتدای سطر توضیحات علامت # درج می شود.

مثال: برنامه زیر مقلوب عدد ورودی را با استفاده از تابع محاسبه می کند.(همراه با توضیحات)

```
# تعریف تابع برای محاسبه مقلوب عدد ارسالی به آن
```

```
def Reverse(n):
```

```
    m=0
```

```
    while n!=0:
```

```
        r=n%10
```

```
        m=m*10+r
```

```
        n=n//10
```

```
    return m
```

```
# بدنه اصلی برنامه
```

```
n=int(input("عدد مورد نظر را وارد کنید"))
```

```
print("عدد:",n,"مقلوب:",Reverse(n))
```