

TRY AND EXCEPT BLOCKS - HATA YÖNETİMİ

NEDEN İHTİYACIMIZ VAR?

- Bazen programlarımızı oluştururken bazı kodlar vardır ki bu kodlar bize program içinde belli hatalara yol açabilir.
- Bu hatalar mantıksal programlama hataları, değişkenlerin tipleri ile ilgili hatalar veya çok daha farklı hata çeşitleri olabilir.

HATA ÇEŞİTLERİ

1. Syntax Hatası
2. Mantık Hatası

HATA TÜRLERİ

- AssertionError,
- AttributeError
 - EOFError,
- Floating PointError
- GeneratorExit,
- ImportError
- IndexError,
 - KeyError
- KeyboardInterrupt,
- Memory Error,
- NameError,
- NotImplementedError,
 - OSError,
- OverflowError,
- ReferenceError,
- RuntimeError,
- StopIteration,
- SyntaxError,
- Indentation Error,
 - TabError
- SystemError,
- SystemExit,
- TypeError,
 - Unbound
- LocalError,
- UnicodeError,
 - UnicodeEncodeError,
 - UnicodeDecodeError,
- UnicodeTranslateError,
 - ValueError,
- ZeroDivisionError

HATA YAKALAMA(TRY-EXCEPT YAPISI)

try:

kodlarımız burada

except Exceptionl:

bir hata olduğunda bu kısmı çalıştır.

except Exceptionll:

hata iki olduğunda bu kodu çalıştır.

HER ZAMAN KULLANMAK GEREKLİ Mİ?

- Her akla geldiğinde kullanılacak bir yapı değildir
- Büyük programlarda sorun olabilir
- Çok nadir durumlarda kullanılmayı gerektirebilir

RAISE

- Kullanıcının yaptığı bir işlem normal şartlar altında hata vermeyecek olsa bile biz ona 'Python tarzı' bir hata mesajı göstermek isteyebiliriz

ASSERT

- Assert (assertion), programınızın belirli bir koşulu kontrol etmesini ve bu koşulun doğru olmadığı durumda bir istisna (exception) fırlatmasını sağlayan bir Python ifadesidir.
- Assert ifadesi, hata ayıklama (debugging) ve doğrulama (validation) amaçları için kullanılır.