**Hata Yakalama Türleri**

TRY, EXCEPET komutları: bir hata almaktan ziyade

Hataya şekil vermeye yarıyor. Ve devam edebiliyor

https://python-istihza.yazbel.com/hata\_yakalama.html

Try: denemek

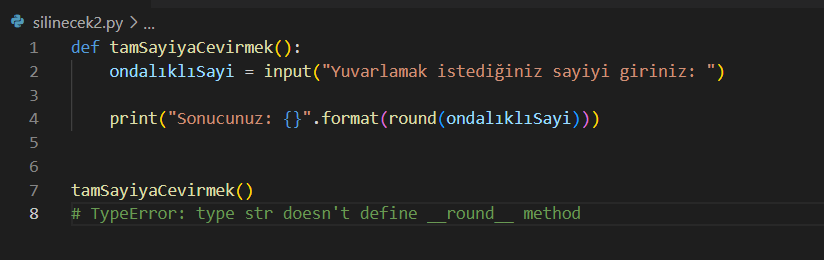
Except : harici durumda

1. **Try (denemek) ve Except(harici durumda)**

Örnk.

Round(sayi) ifadedsi bir ondalıklı sayıyı tam sayıya yuvarlamaya yarıyor. Bunu kullanarak bir örnek

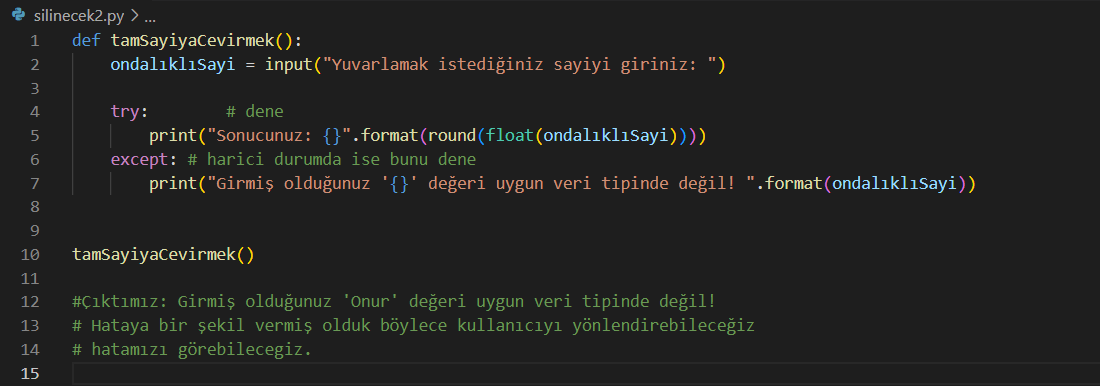
Round : yuvarlak / yuvarlamak



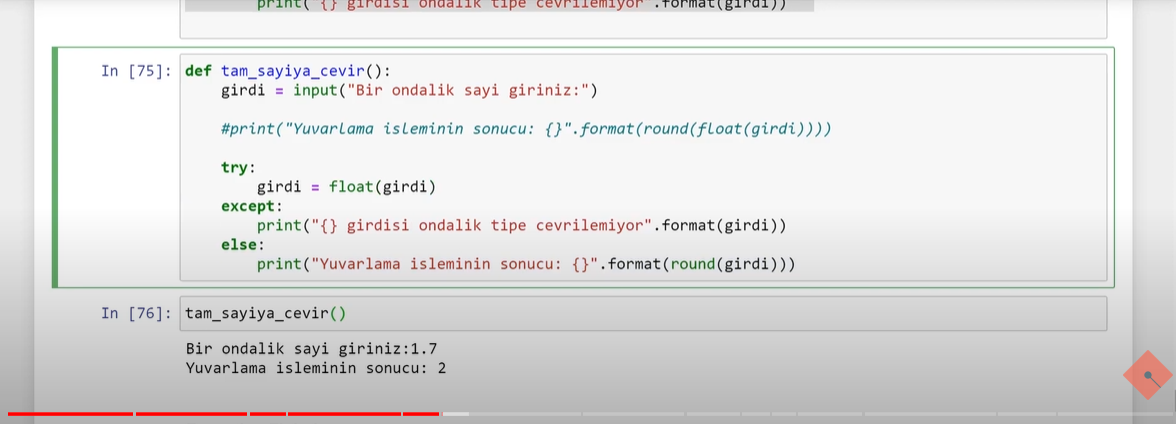
Blinçli olarak float girilmedi ve type error vermsi sağlandı

Type error aldık / bilinçli bir type error

Bunları try ve except fonksiyonları ile birlikte deneyelim



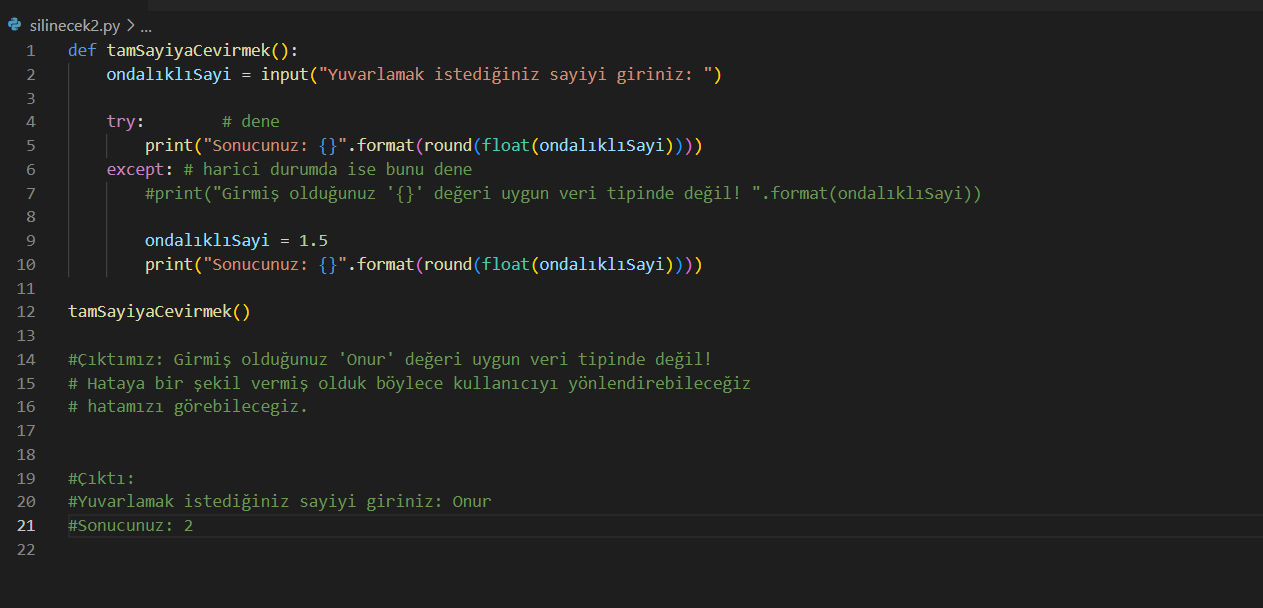
Bunun bir diğer kullanım şekli ise



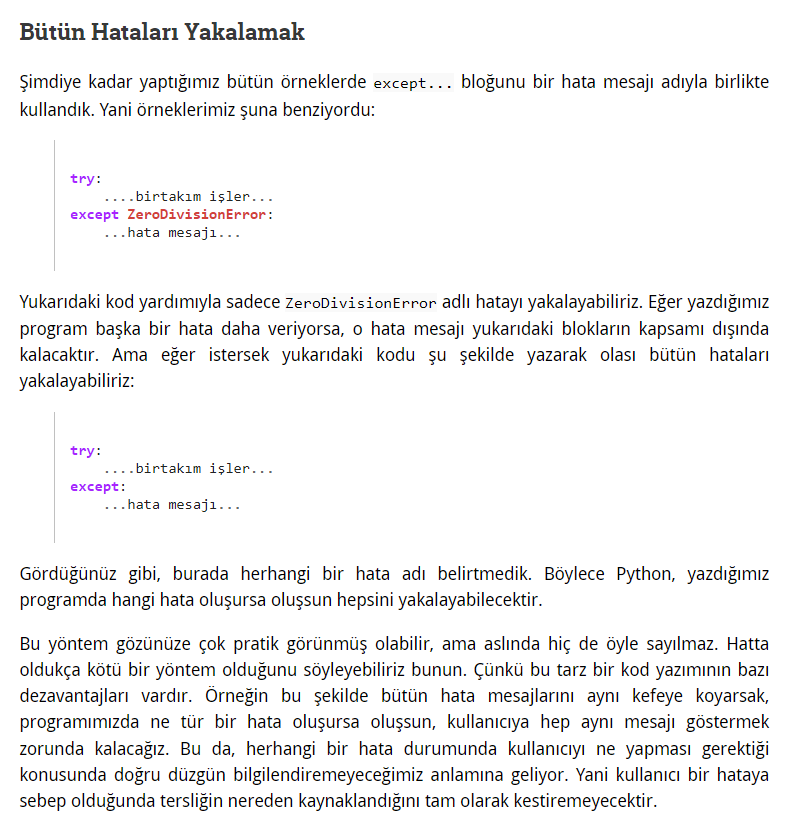
Bilgi: Örnegin bir test aşamasında hatayi görmezden gelerek kodun devamınını çalıştırılması için

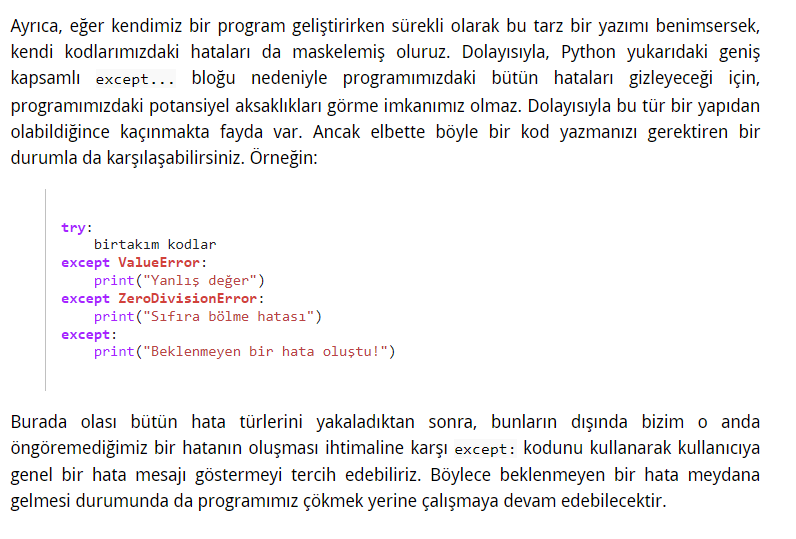
#test aşamasında şöyle bir şeyde yapabiliriz

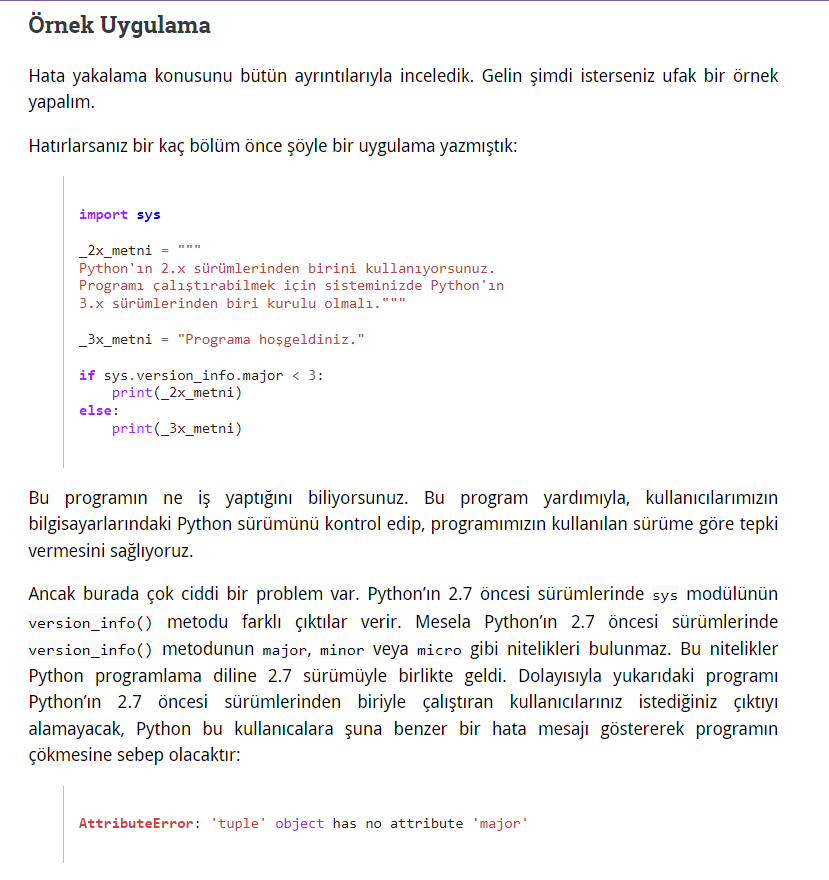
Bir nevi default değeri olarak

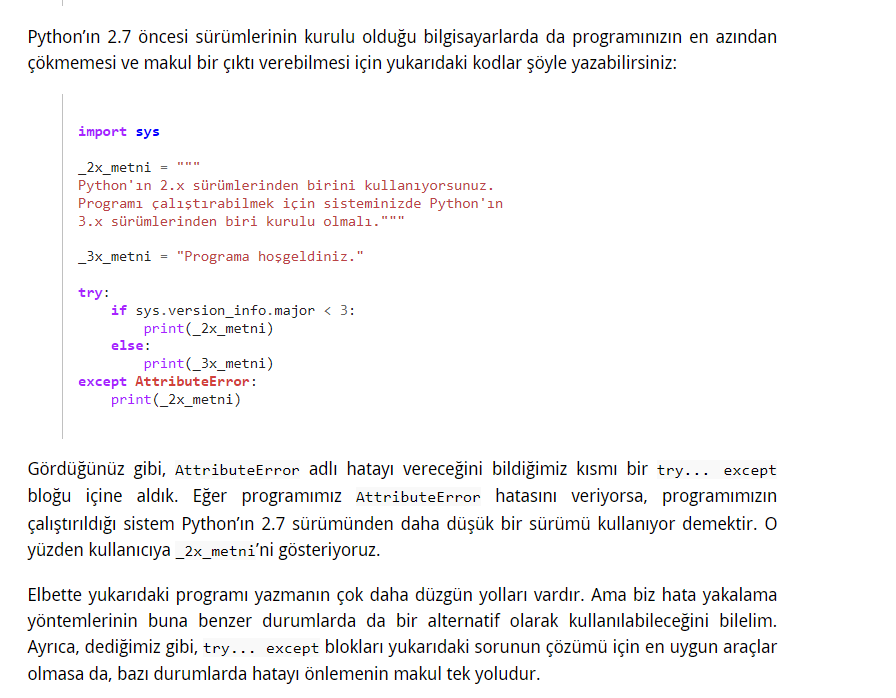


**Bilgi:**



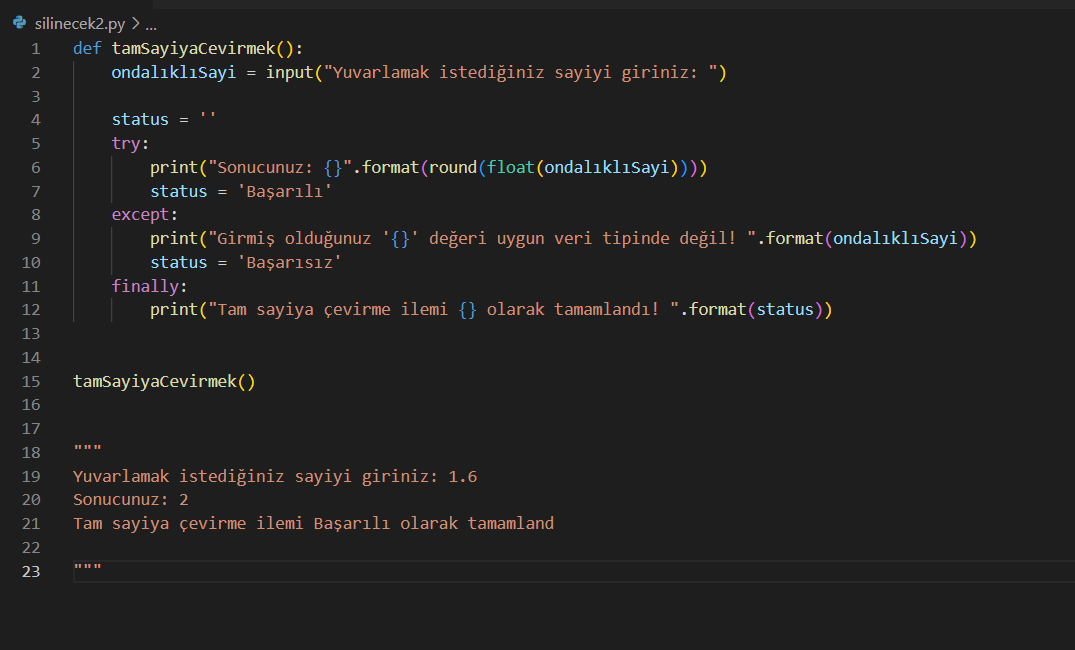




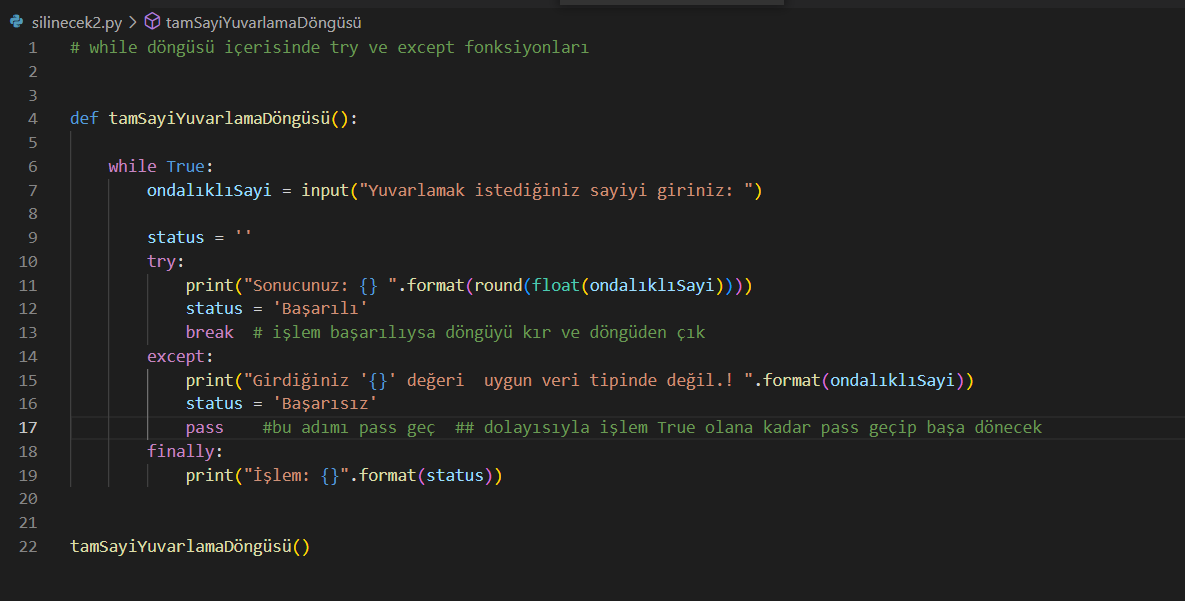


**Finally**

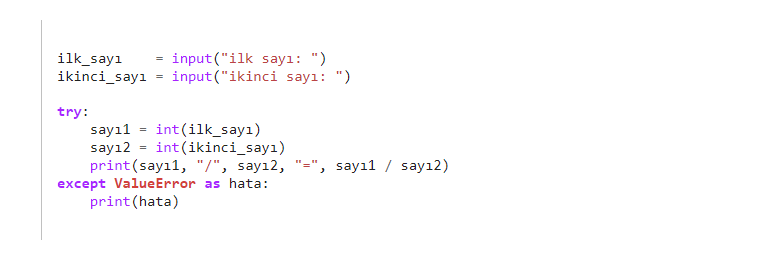
Finally fonksiyonu, hangi durum gerçekleşirse gerçekleşsin (try veya except) çalışan bir fonksiyondur



Örnek



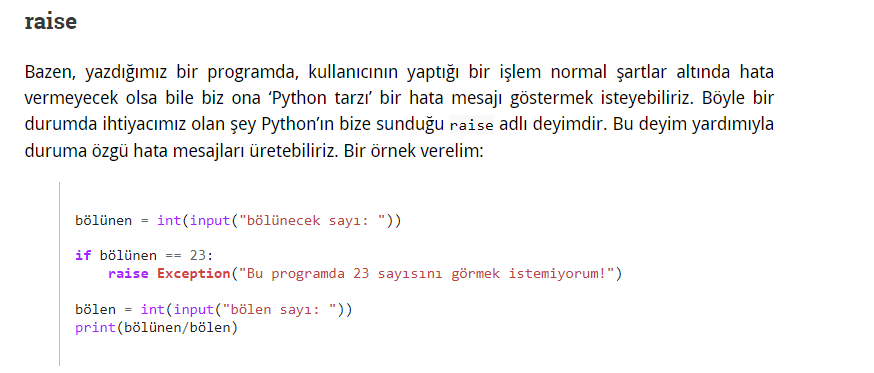
**As**



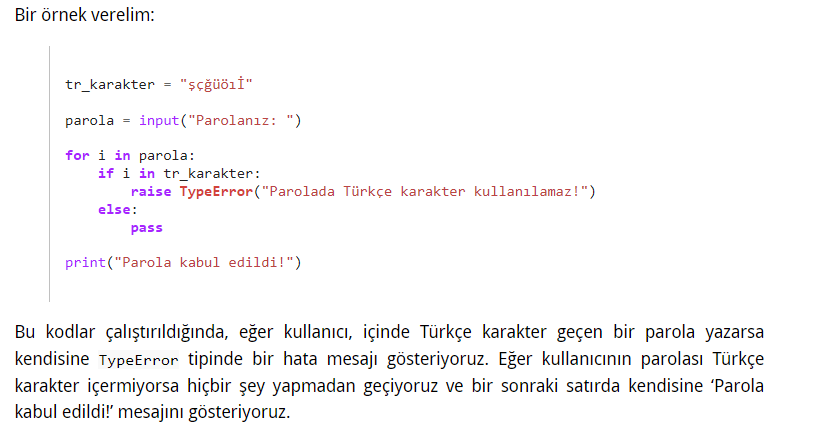
**Else**



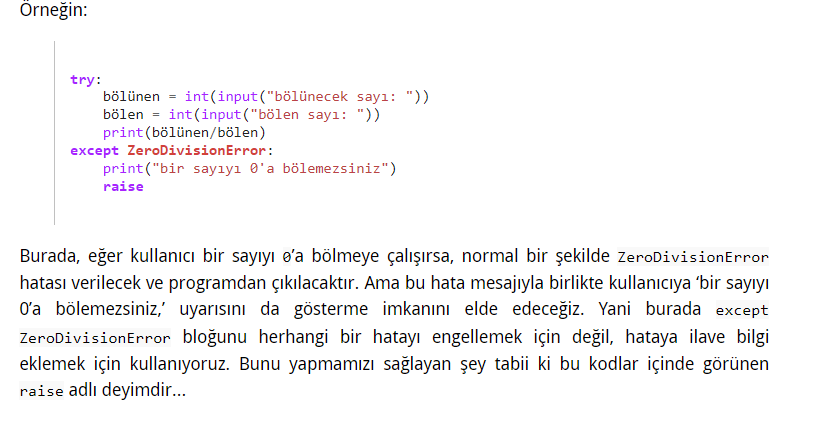
**Raise()**



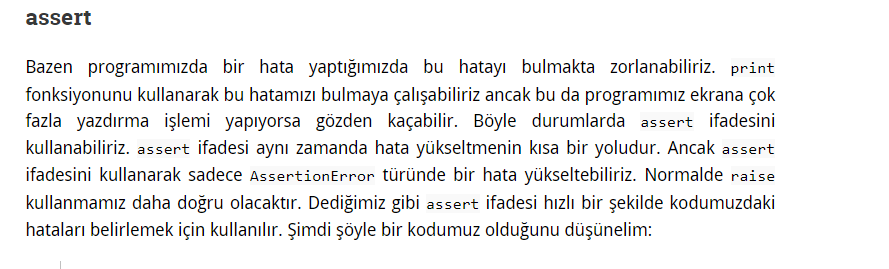
**Örnek**



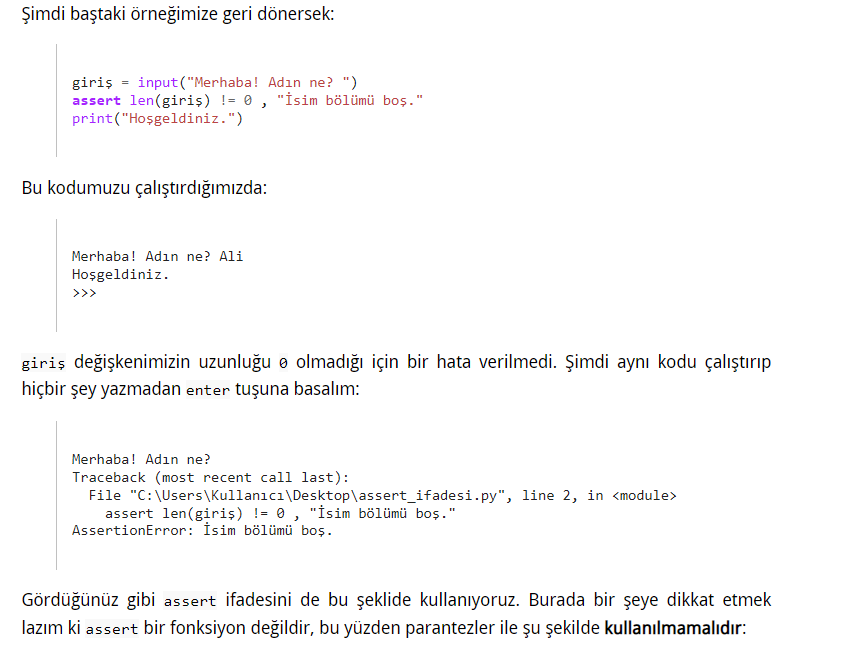
**Örnek**

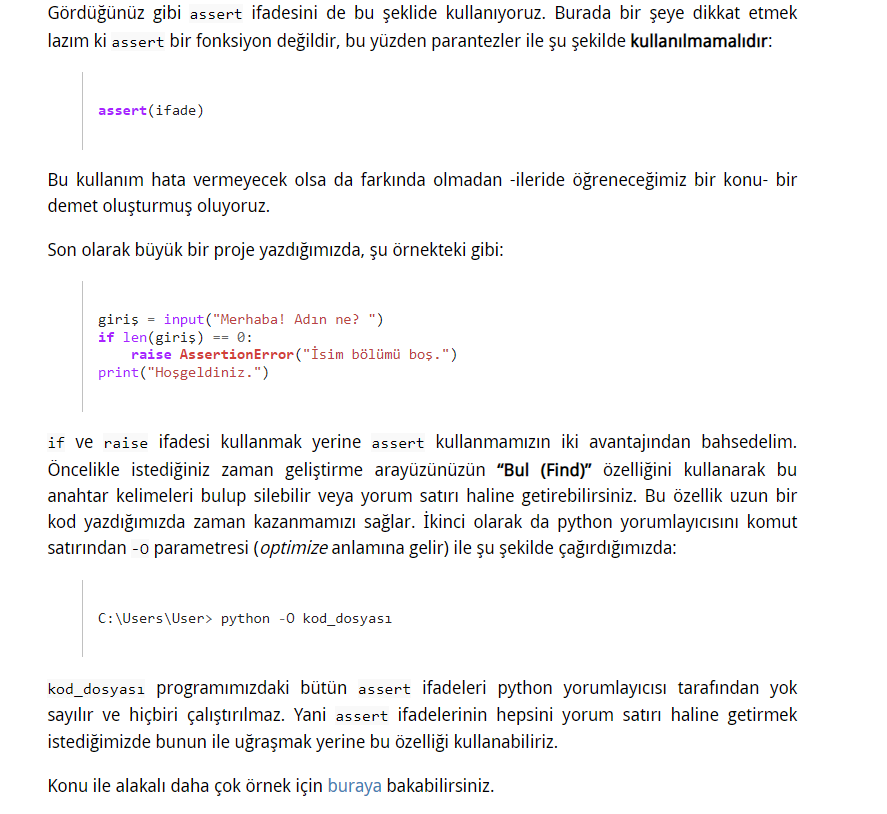


**Assert**





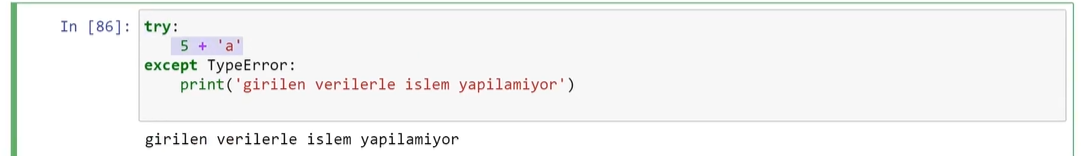




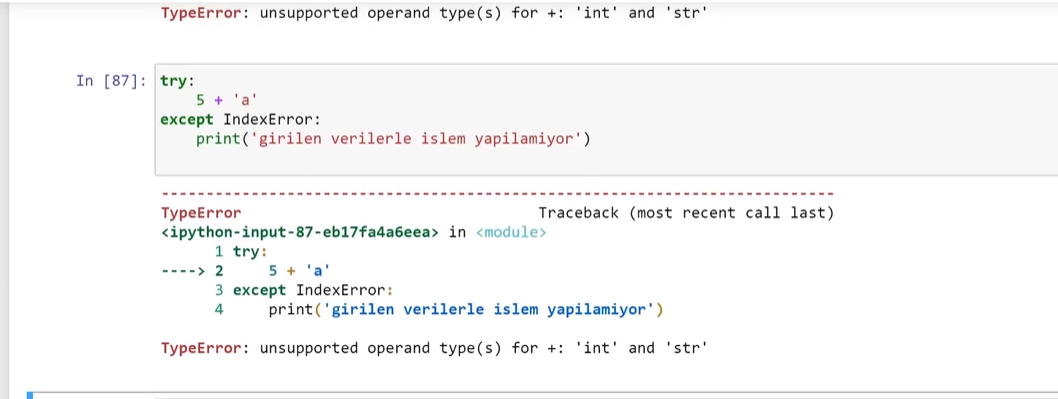
**https://forum.yazbel.com/t/assert-deyimi-nasil-kullanilir/231**

**Exception (Türleri) Türleri**

* Hangi hata türünü alıyorsak ona göre önlem alabiliriz.. (index error, keyerror, typeerror vb)

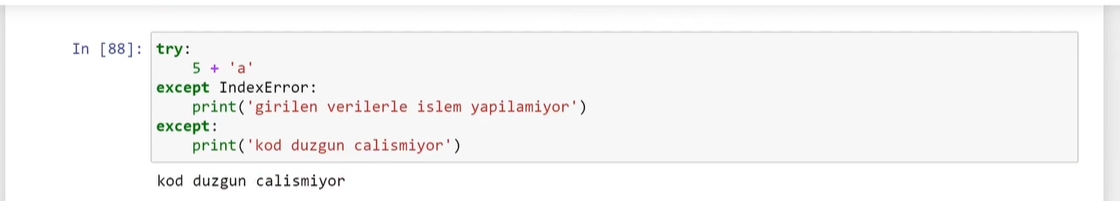


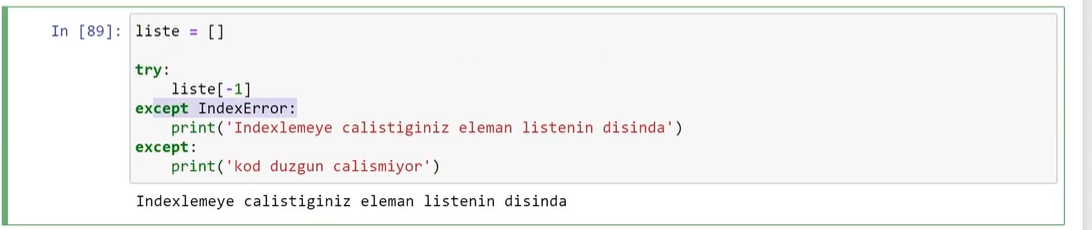
Doğru hatayı girmezsek type degilde indexerror dersek

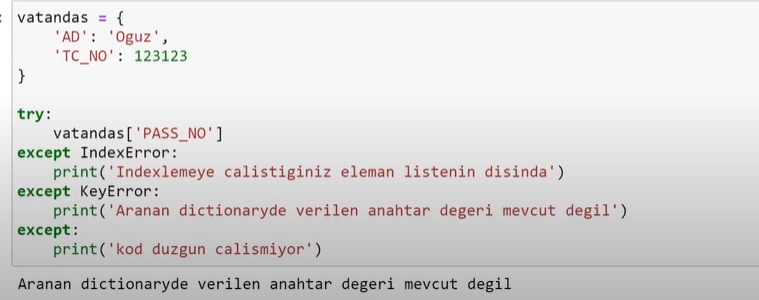


Bu sefer sistem gerçekten hata verir 😊

Birden fazla except de tanımlayabiliyoruz birden fazla hataya karşı önlem almak istersek







Bilgi: Hata türüne karşılık ne kadar çok önlem alırsak o kadar iyidir

Genel bir hata yöntemi son örnekteki en alt satır gibi !

Örnek

