

Python'da **fonksiyon kapsamı (scope)**, bir değişkenin tanımlı olduğu alanı ifade eder. Python'da kapsam iki ana kategoriye ayrılır: **yerel (local)** ve **genel (global)** kapsam. Bir fonksiyon içinde tanımlanan değişkenler yerel kapsamda olurken, fonksiyon dışında tanımlananlar genel kapsamda olur.

1. Global ve Yerel Değişkenler

Fonksiyon dışında tanımlanan bir değişken globaldir, içeride tanımlanan ise lokaldir.

```
x = 10 # Global değişken

def fonksiyon():
    x = 5 # Yerel değişken
    print("Fonksiyon içindeki x:", x)

fonksiyon() # Çıktı: Fonksiyon içindeki x: 5
print("Fonksiyon dışındaki x:", x) # Çıktı: Fonksiyon dışındaki x: 10
```

Açıklama: `x` fonksiyon içinde ve dışında farklı değerler alır çünkü fonksiyon içindeki `x` yerel, dışındaki ise global kapsamda tanımlıdır.

2. Global Anahtar Kelimesi

Fonksiyon içinden global bir değişkene erişmek ya da onu değiştirmek için `global` anahtar kelimesi kullanılır.

```
x = 10 # Global değişken

def fonksiyon():
    global x
    x = 5 # Global değişkeni değiştiriyoruz
    print("Fonksiyon içindeki x:", x)

fonksiyon() # Çıktı: Fonksiyon içindeki x: 5
print("Fonksiyon dışındaki x:", x) # Çıktı: Fonksiyon dışındaki x: 5
```

Açıklama: `global x` ifadesiyle, fonksiyon içinden global değişken olan `x`'e eriştik ve onu değiştirdik.

3. Nonlocal Anahtar Kelimesi

İç içe fonksiyonlar kullanıldığında, dış (ancak global olmayan) bir fonksiyondaki değişkene erişmek için `nonlocal` kullanılır.

```
def dis_fonksiyon():
    x = 10 # Dış fonksiyon değişkeni

    def ic_fonksiyon():
        nonlocal x
        x = 5 # Dış fonksiyonun değişkenini değiştiriyoruz
        print("İç fonksiyon içindeki x:", x)

    ic_fonksiyon()
    print("Dış fonksiyon içindeki x:", x)

dis_fonksiyon()
# Çıktı:
# İç fonksiyon içindeki x: 5
# Dış fonksiyon içindeki x: 5
```

Açıklama: `nonlocal x` ile, iç fonksiyon dış fonksiyonun değişkenini değiştirebilir. Ancak, bu değişken global değildir.
