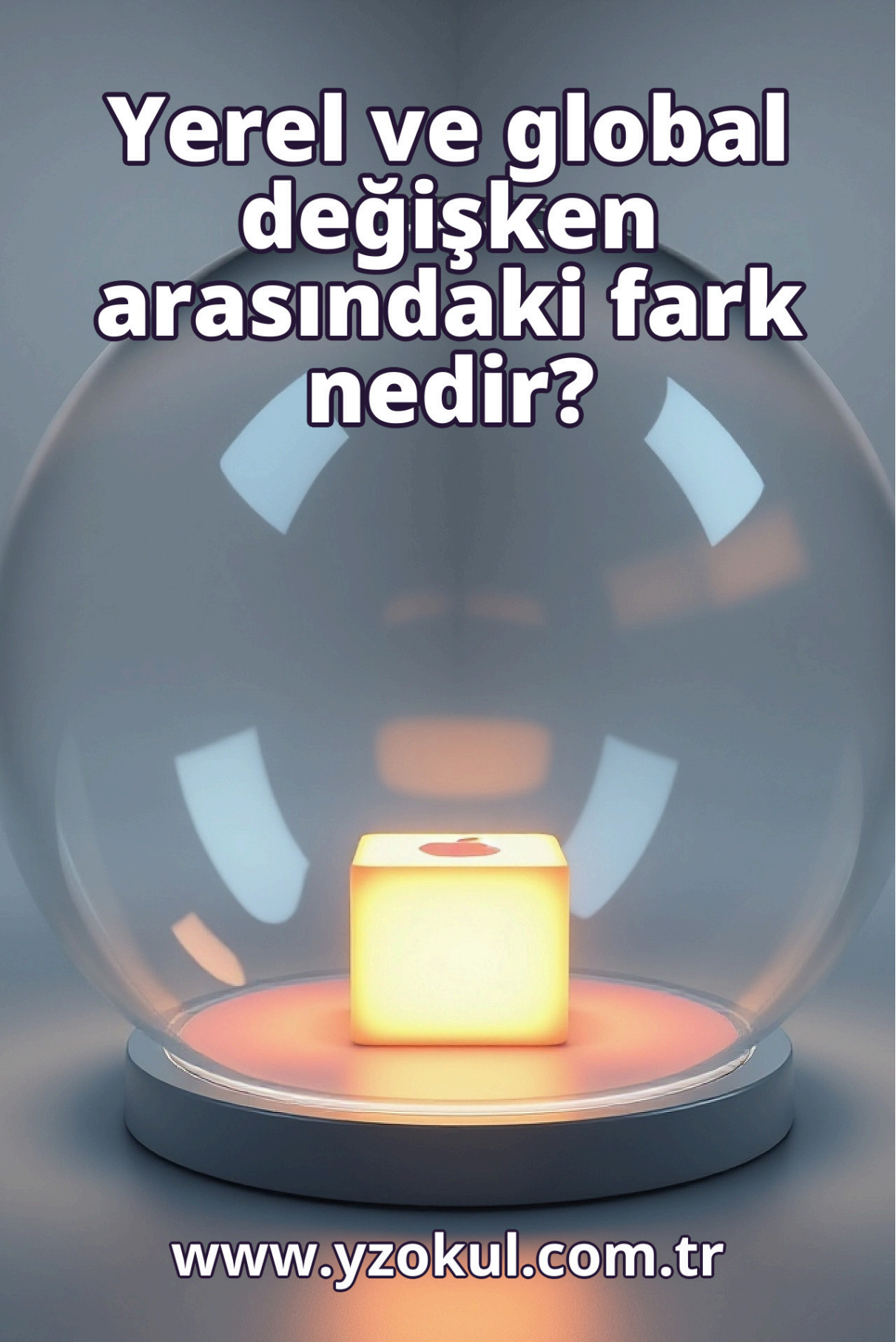
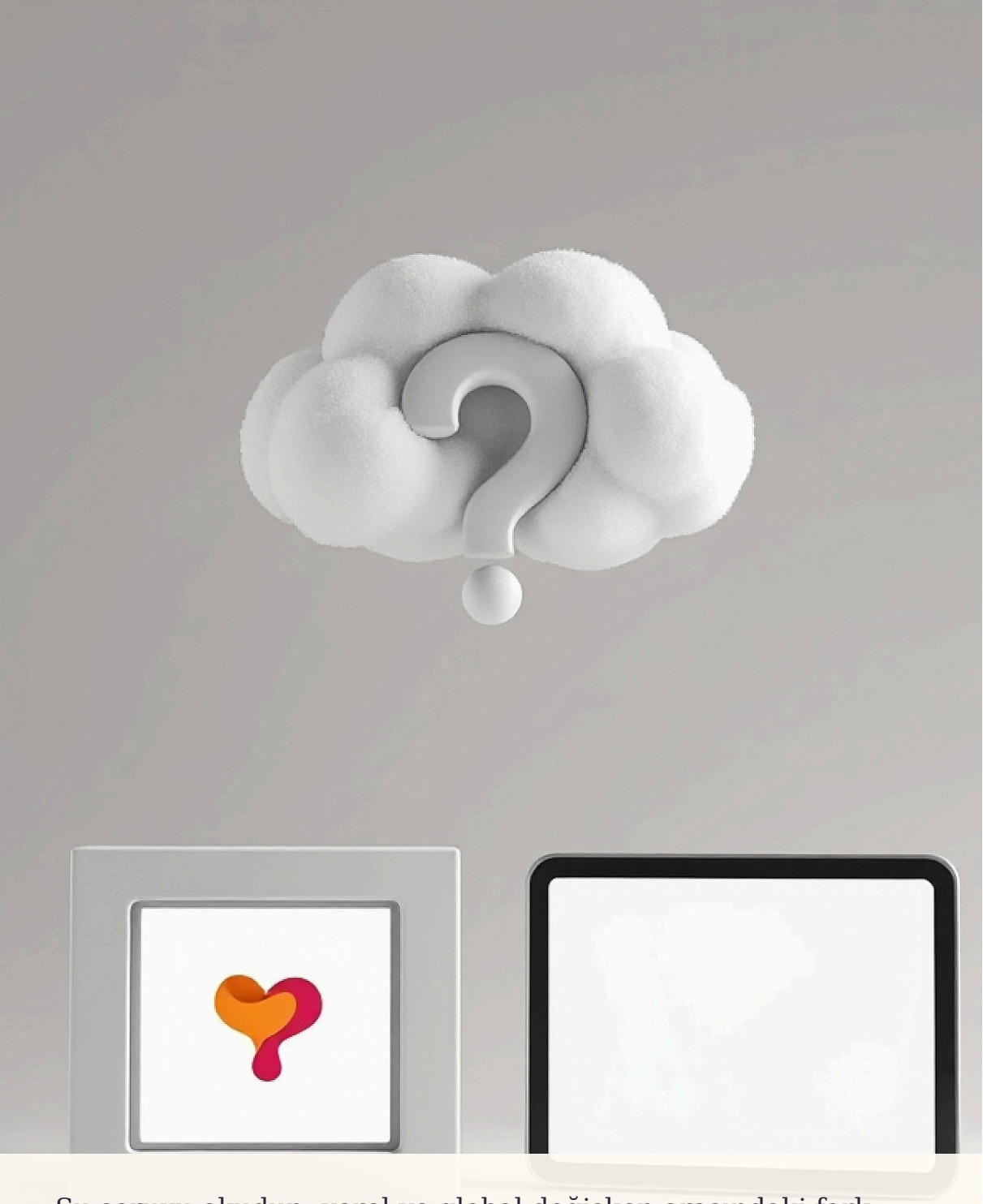
Yerel ve global değişken arasındaki fark nedir?



www.yzokul.com.tr

Yerel ve global değişken arasındaki fark nedir?

www.yzokul.com.tr



Şu soruyu okudun: yerel ve global değişken arasındaki fark nedir? Kısa yanıt, yerel değişkenin yalnızca tanımlandığı küçük bölümde çalıştığı, global değişkene ise programın daha geniş bir alanından erişilebildiğidir. Bu cevap kitabı her fikri basit örneklerle adım adım açıklar.



Değişken, bilgisayarın belleğinde bir değeri, örneğin bir sayıyı tutan adlandırılmış bir kutudur. Kutuya bir değer koyabilir, sonra okuyabilir ve ihtiyaç duyduğunda değiştirebilirsin. Örneğin score adlı bir kutu 10 sayısını tutabilir. Ad, o değerin nerede saklandığını hatırlamadan ona başvurmanı sağlar.



Bir program düz bir talimat listesi değildir. Her biri tek bir iş yapan fonksiyonlar gibi bölümleri vardır. Fonksiyon, tekrar tekrar kullanılabilen küçük, adlandırılmış bir talimat bloğudur. Örneğin add adlı bir fonksiyon iki sayı alıp toplamlarını döndürebilir. Bölümler kodunu düzenlemene yardımcı olur.



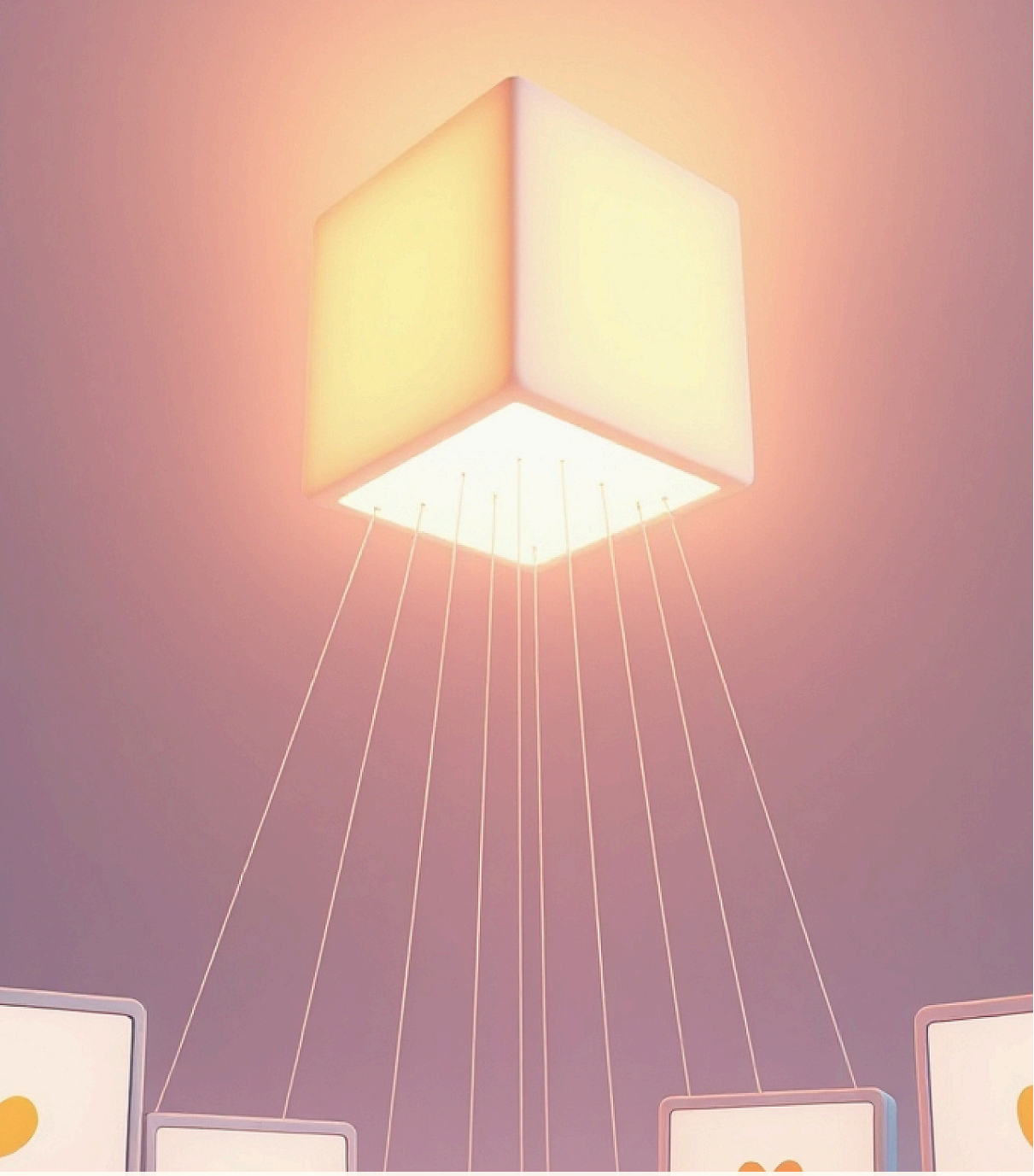
Her değişken program metninde bir yerde, bir bölümün içinde oluşturulur. Tanımlandığı yer, nerede kullanılabileceğini belirler. Bir değişkeni bir fonksiyonun içinde tanımlarsan o fonksiyona aittir. Tüm fonksiyonların dışında tanımlarsan tüm programa aittir. Tanım yeri kuralları belirler.



Yerel değişken bir fonksiyonun içinde tanımlanır, örneğin geçici bir hesaplama sayısı. Yalnızca o fonksiyon çalışırken ve yalnızca o fonksiyonun içinde var olur. Örneğin iki sayıyı toplayan bir fonksiyonun içinde sonucu tutmak için total adlı yerel bir değişken oluşturabilirsin. Fonksiyon bitince total kaybolur.



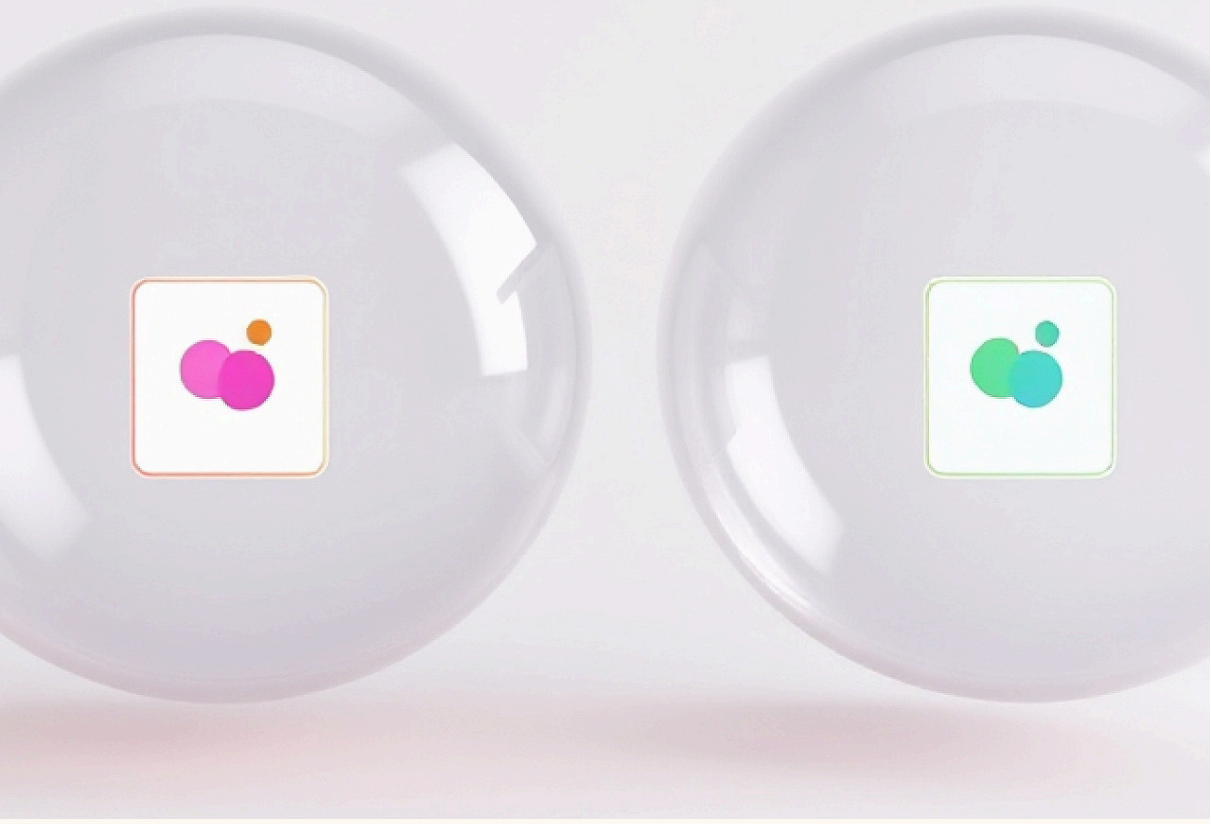
Fonksiyonun dışındaki kod yerel bir değişkeni göremez veya kullanamaz. Onu başka bir bölümden kullanmaya çalışırsan program onu bilinmeyen olarak ele alır, çünkü yalnızca o küçük bölüme aittir. Örneğin total'i bir fonksiyonun içinde tanımlarsan total'i o fonksiyonun dışından yazdıramazsın. Ad orada basitçe var olmaz.



Global değişken herhangi bir fonksiyonun dışında, programın üst düzeyinde tanımlanır. Bir kez oluşturulur ve program çalıştığı sürece kullanılabilir kalır. Örneğin dosyanın en üstünde app_name adlı global bir değişken tanımlayabilirsin. Programdaki her fonksiyon o değeri okuyabilir. Tüm program boyunca yaşar.



Programın birçok farklı bölümü global bir değişkeni okuyabilir ve değiştirebilir. Onu global yapan tam da bu daha geniş erişimdir. Örneğin bir fonksiyon app_name'i yeni bir değere ayarlayabilir, başka bir fonksiyon da o yeni değeri sonra okuyabilir. Değişiklik her yerde görünür, çünkü değişken paylaşılır.



İki fonksiyonun her biri aynı ada sahip kendi yerel değişkenine sahip olabilir ve bunlar çakışmaz. Her yerel kutu yalnızca kendi fonksiyonuna aittir. Örneğin birinci fonksiyonda count adlı yerel bir değişken, ikinci fonksiyonda da count adlı yerel bir değişken olabilir. Bunlar hiçbir zaman birbirine karışmayan ayrı kutulardır.



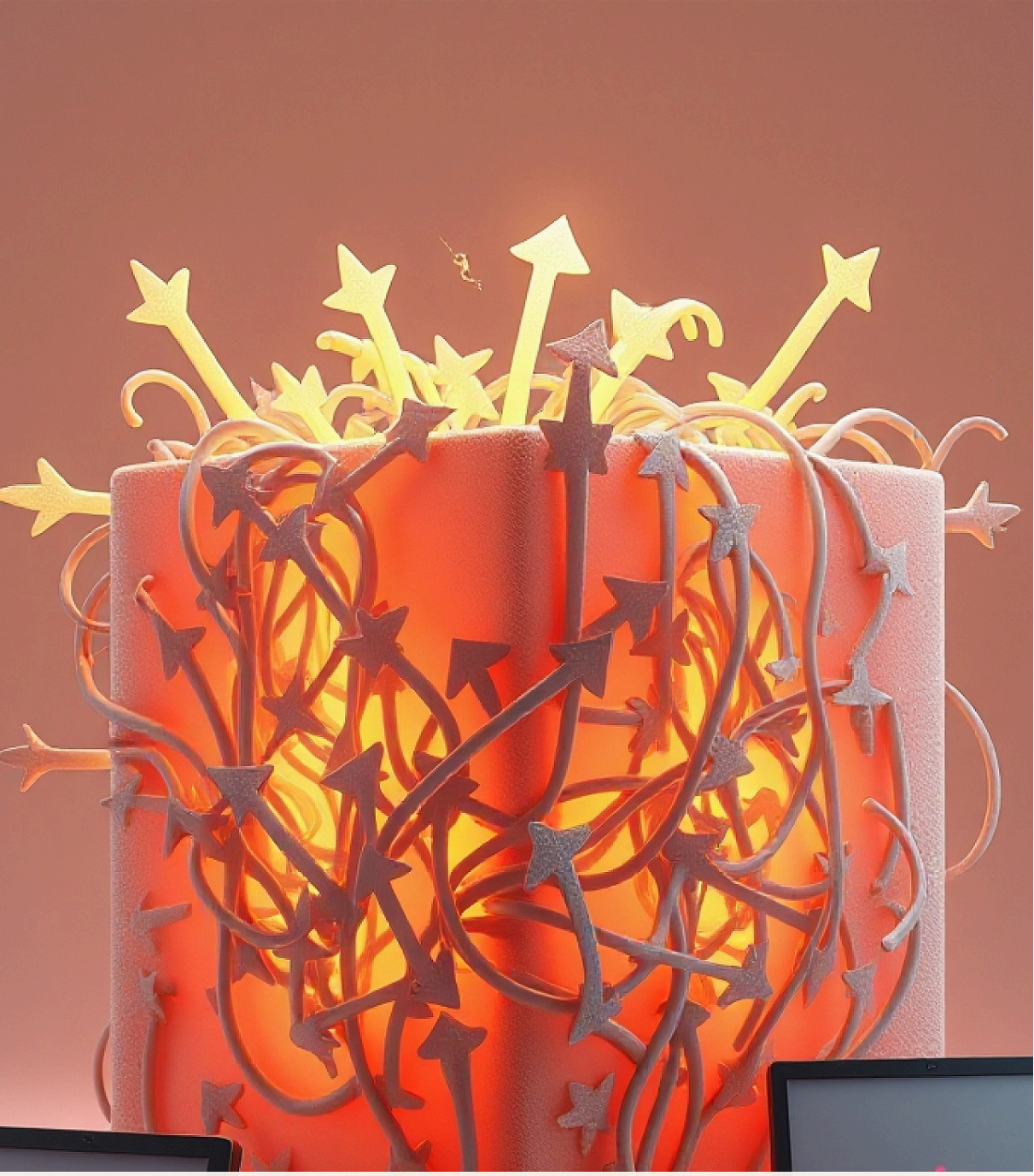
Bir değişkenin erişim alanına kapsam denir. Yerel değişkenin kapsamı küçüktür; global değişkenin kapsamı geniştir. Örneğin bir fonksiyonun içinde tanımlanan değişkenin kapsamı o fonksiyonla sınırlıdır. Üst düzeyde tanımlanan değişkenin kapsamı tüm programı kapsar. Kapsam, bir adın nerede kullanılabileceğini söyler.



Programlama dilinin kuralları bir değişkenin tam olarak nerede kullanılabileceğini belirler. Kapsamı serbestçe seçmezsin; tanım yeri onu belirler. Örneğin çoğu dilde bir değişkeni fonksiyonun içinde tanımlamak onu otomatik olarak yerel yapar. Onu yerel olarak bildirmen gerekmez. Dil bu kuralları senin için uygular.



Bir fonksiyonun geçici hesaplama sayısı yerel değişken için iyi bir seçimdir. Kısa süre kullanılır ve sonra unutulur, bu yüzden başka hiçbir şey ona ihtiyaç duymaz. Örneğin indirim hesaplayan bir fonksiyon ara bir sonucu tutmak için temp adlı yerel bir değişken kullanabilir. Fonksiyon dönünce temp'e artık ihtiyaç kalmaz. Onu yerel tutmak programı temiz tutar.



Tüm bilgileri global yapmak, farklı bölümlerin birbirini beklenmedik şekilde etkilemesine yol açar. Bir kısım paylaşılan değeri değiştirir, başka bir kısım belirgin bir neden olmadan bozulur. Örneğin iki fonksiyon da counter adlı global bir değişkeni değiştirirse son değer çalışma sıralarına bağlı olur. Bu, bulunması zor hatalara yol açabilir.



Yeni başlayan biri bir fonksiyonun yerel değişkenini fonksiyonun dışından kullanmaya çalışıp hata alabilir. Çözüm, değişkeni gerçekten ihtiyaç duyulan yerde tanımlamak ya da değeri içeri ve dışarı aktarmaktır. Örneğin bir fonksiyonun dışında bir sonuca ihtiyacın varsa yerel değişkeni doğrudan okumaya çalışmak yerine onu fonksiyondan döndür.



Yeni başlayan biri hatalardan kaçınmak için her değişkeni global yapabilir ve sonra kısımların birbirine karışmasını izler. Geliştirilmesi gereken alışkanlık şudur: bir şey gerçekten paylaşılması gerekmedikçe şeyleri yerel tut. Örneğin bir değeri yalnızca bir fonksiyon kullanıyorsa onu o fonksiyonun içinde tanımla. Yalnızca birkaç bölüm gerçekten aynı değere ihtiyaç duyuyorsa onu global yap.



Yerel değişken yalnızca tanımlandığı bölümde çalışır, global değişkene daha geniş bir alandan erişilebilir ve kapsam bu erişim alanının adıdır. Değişkenleri varsayılan olarak yerel tutmayı ve yalnızca paylaşım gerçekten gerektiğinde onları global yapmayı unutma. Bu basit alışkanlık birçok kafa karıştırıcı hatayı önler.