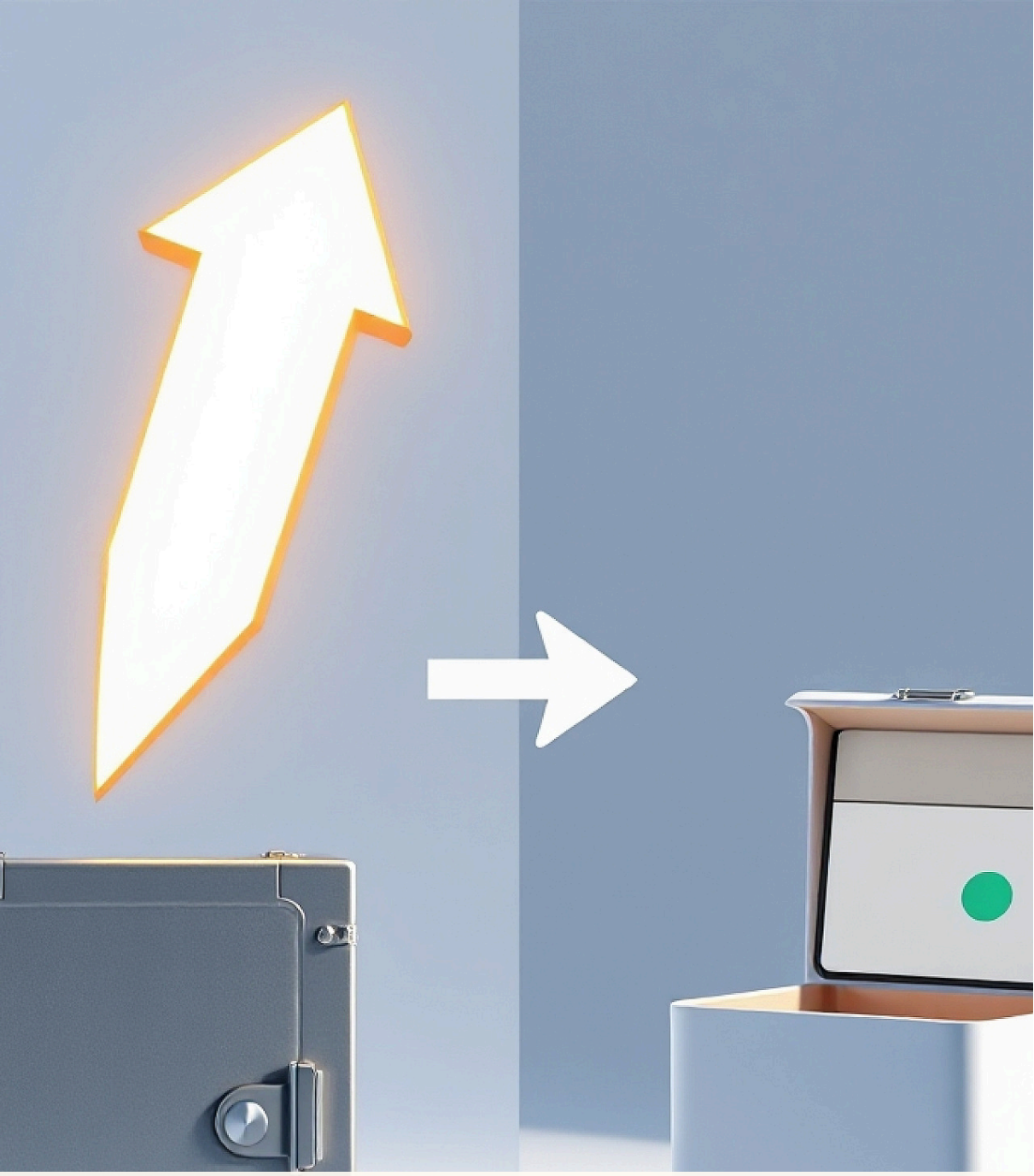
Sabit ile değişken arasındaki fark nedir?



www.yzokul.com.tr

Sabit ile değişken arasındaki fark nedir?



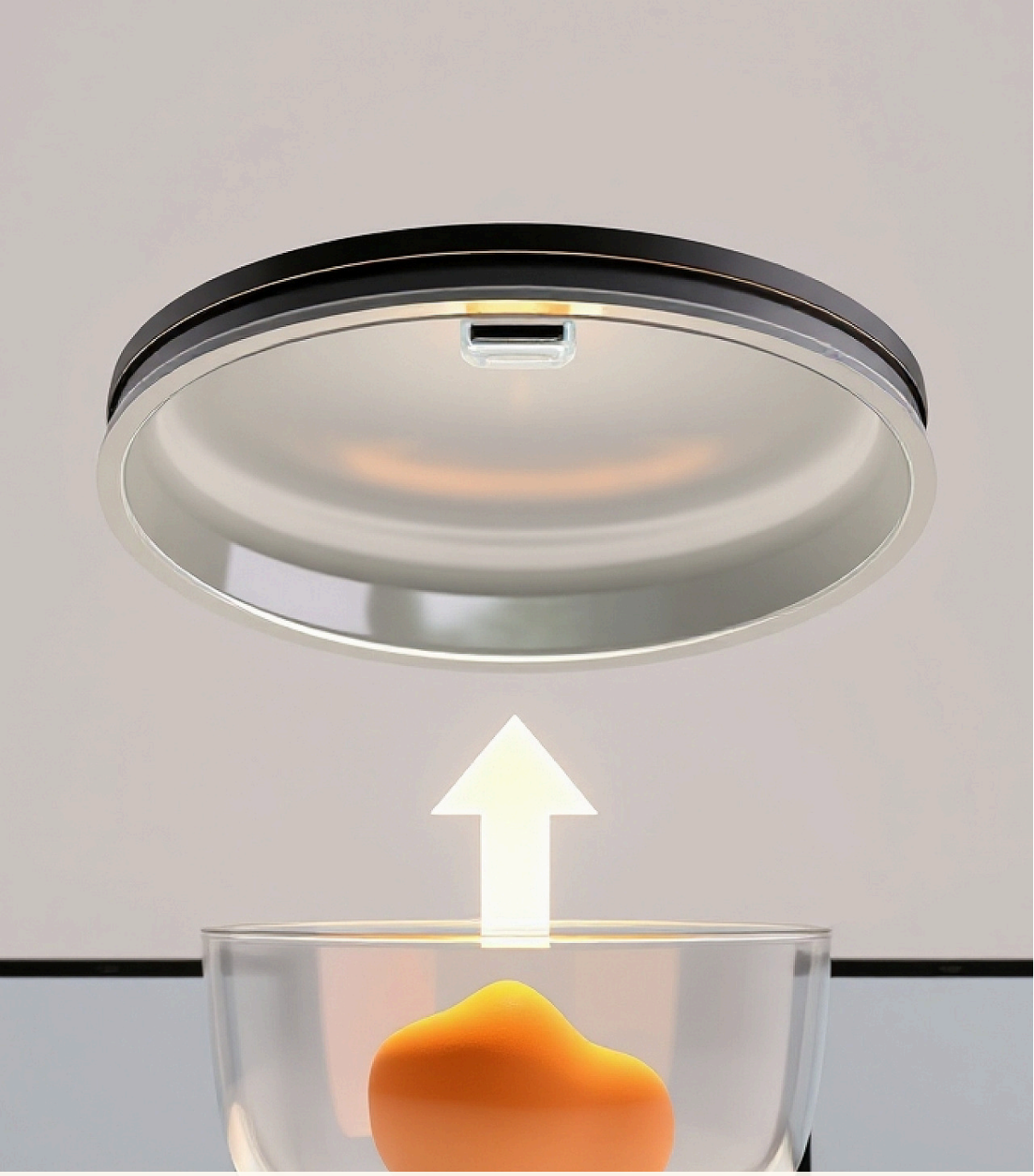
Sabit ile değişken arasındaki fark nedir? Kısa yanıt şudur: program kurallarına göre bir sabit yeniden atanamaz, oysa bir değişken değer değiştirebilir. Yeniden atama, bir ada yeni bir değer vermek demektir. Bu kitap her fikri adım adım açıklar ve değerler ile adların temelleriyle başlar. Sonunda temel farkı anlayacak ve yaygın hatalardan kaçınacaksın.



Bir değer, basitçe bir programın sakladığı bir bilgi parçasıdır; örneğin bir sayı, bir sözcük ya da doğru/yanlış bayrağı. Her sabit ve her değişken bir değer tutar. Örneğin 42 sayısı bir değerdir ve hello sözcüğü bir değerdir. Programlar değerleri daha sonra kullanabilmek için saklar. Değerleri anlamak, sabitleri ve değişkenleri anlamanın ilk adımıdır.



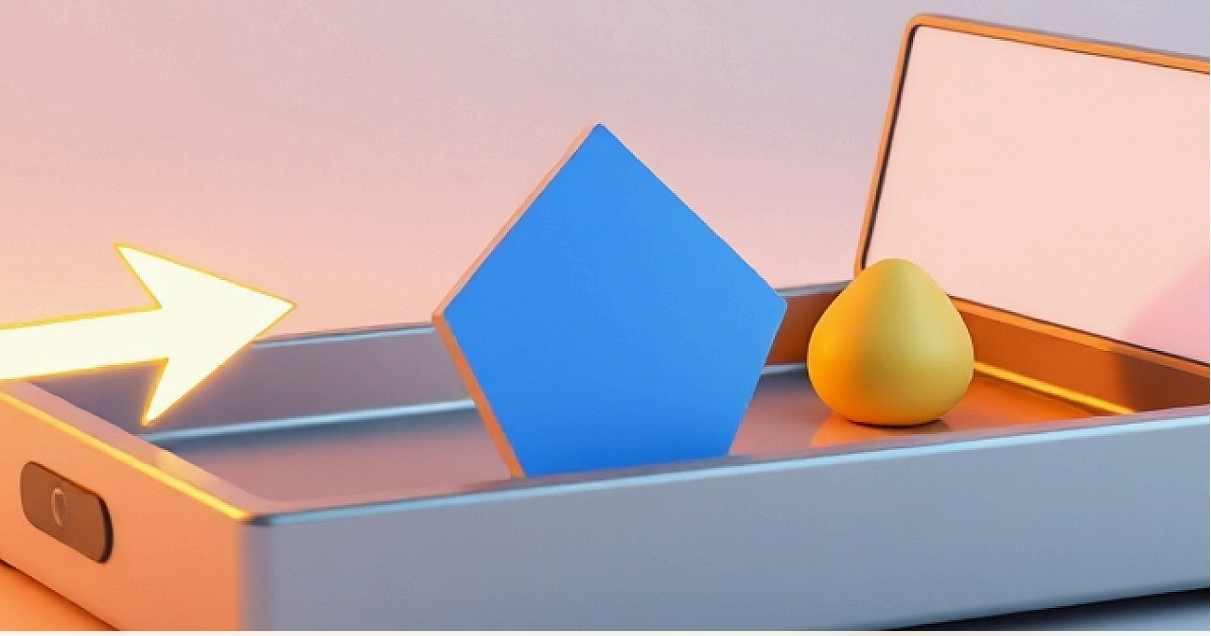
Programlar değerlere ad verir, böylece daha sonra onlara başvurabilir. Bir ad, yalnızca bir değeri işaret eden bir etikettir. Örneğin bir değere score ya da maximum adını verebilirsin. Adı programında kullandığında, işaret ettiği değeri temsil eder. Adlar programları okumayı ve yazmayı kolaylaştırır. Adlar değerlerin kendisi değildir; değerlere yapılan başvurulardır.



Bir değişken, değişmesine izin verilen adlandırılmış bir değerdir. Bir değişkene istediğin zaman yeni bir değer verebilirsin ve ad bundan sonra yeni değeri işaret eder. Örneğin score adlı bir değişken önce 0'ı, sonra 10'u işaret edebilir. Ad aynı kalır, ancak işaret ettiği değer değiştirilebilir. Bir değişkeni yararlı kılan şey bu değişebilme yeteneğidir.



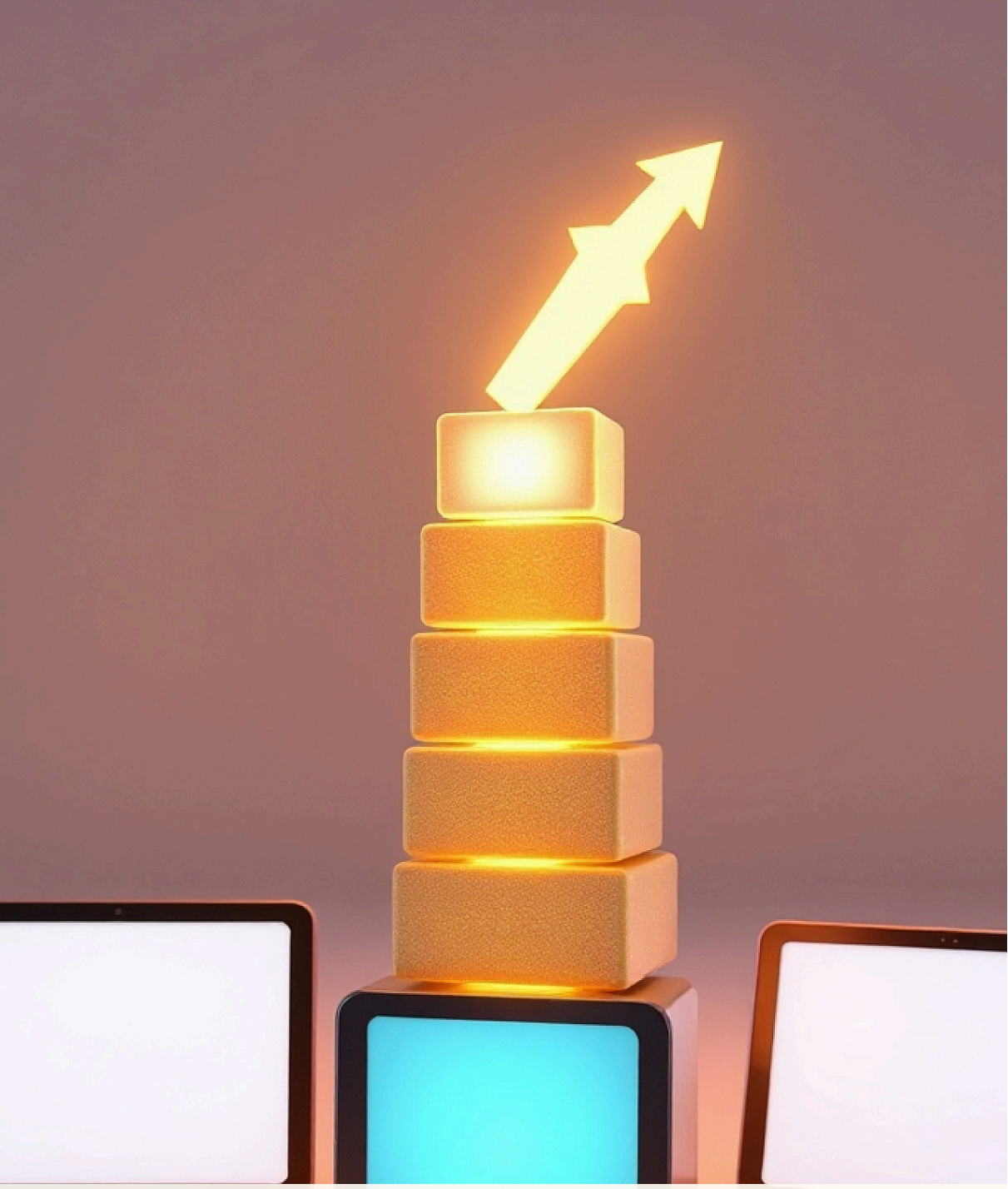
Bir sabit, yeniden atanmasına izin verilmeyen adlandırılmış bir değerdir. Bir sabiti belirledikten sonra, program kurallarına göre ona yeni bir değer veremezsin. Örneğin MAX_SCORE adlı bir sabit 100 olarak ayarlanabilir ve 100 olarak kalmalıdır. Ad, değerine kilitlidir. Bir sabiti yeniden atamaya çalışmak birçok programlama dilinde hataya yol açar.



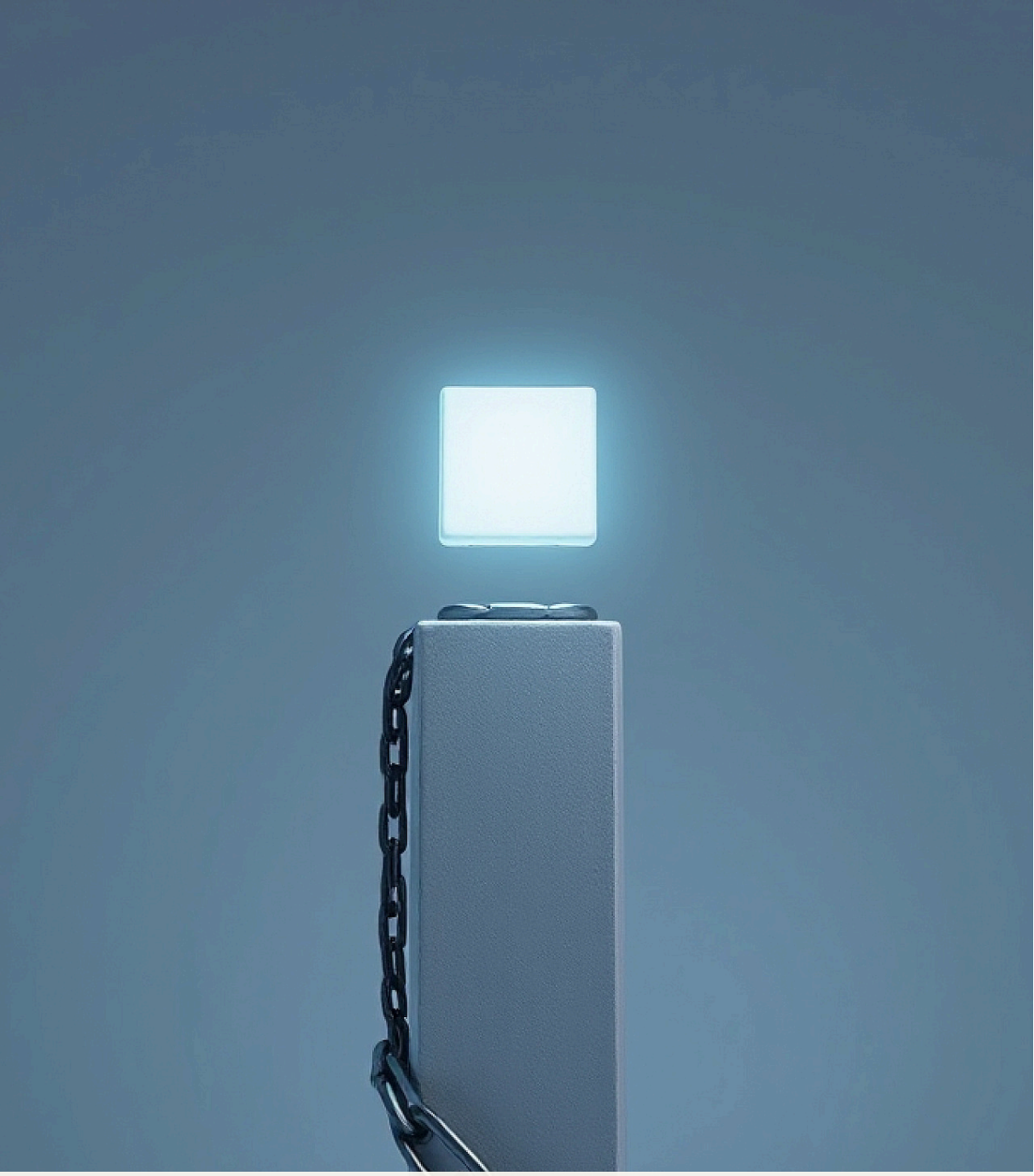
Bir değişkeni değiştirmek için ona yeni bir değer atarsın. Ad aynı kalır, ancak işaret ettiği değer değiştirilir. Örneğin score 0 ise, onu 10 yapmak için score = 10 yazabilirsin. Bundan sonra score adı 0 yerine 10'u işaret eder. Buna yeniden atama denir. Değişkenler bu tür bir değişime izin verecek şekilde tasarlanmıştır.



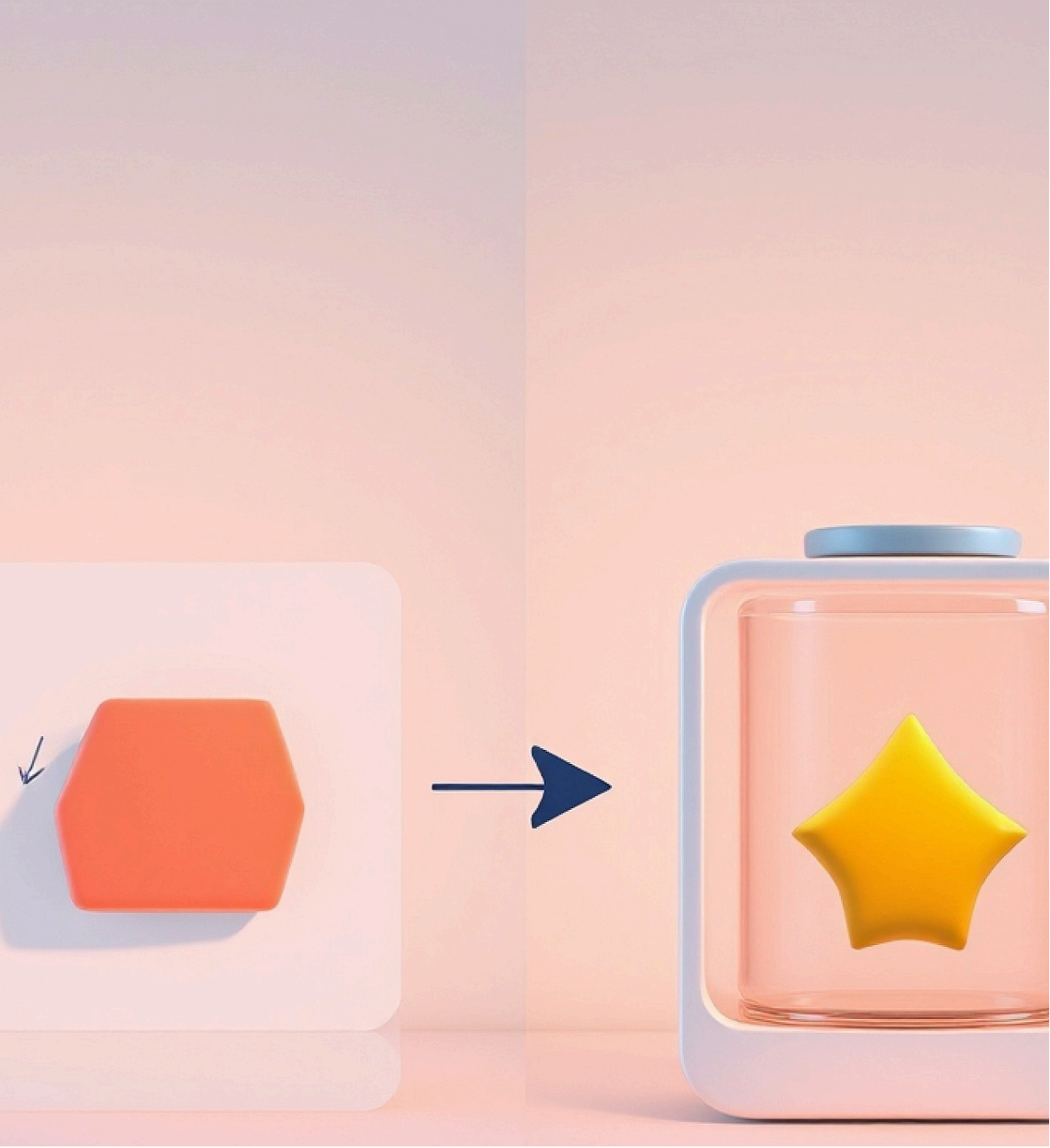
Bir sabiti yeniden atamaya çalışmak program kurallarını çiğner. MAX adlı bir sabit tanımlayıp onu 100 olarak ayarlarsan, daha sonra MAX = 200 yazamazsın. Program bu girişimi reddeder, çoğu zaman bir hata mesajıyla. Zihnindeki resim kilitli bir kutu olmalıdır: ad, değerine kilitlidir. Bu koruma, önemli değerlerin yanlışlıkla değişmesini önler.



Puan kazandığın bir oyun düşün. Mevcut puanın oynadıkça artar, bu yüzden değişmelidir. Puanı bir değişkende saklarsın; örneğin başlangıçta score = 0, sonra puan kazandığında score = score + 10. Puan değiştiği için doğru seçim bir değişkendir. Oyun, oyun sırasında puanı birçok kez günceller.



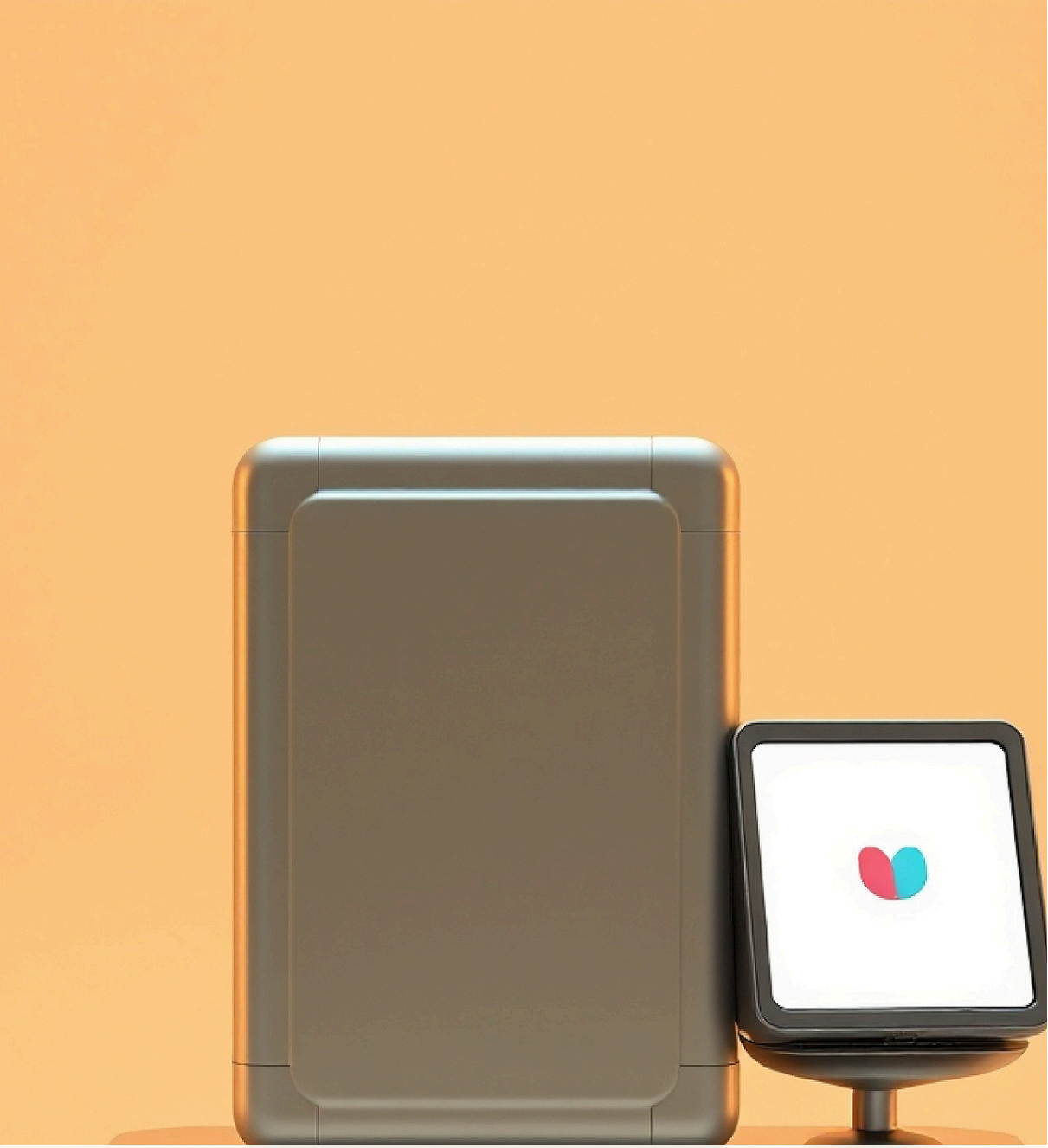
Olası en yüksek puan gibi sabit bir üst sınır, oyun sırasında değişmemelidir. Bunu ayrı bir tanımda tutarsın, örneğin MAX_SCORE = 100. Bu bir sabittir çünkü bir kez ayarlanır ve asla yeniden atanmaz. Bir sabit kullanmak, bu değerin sabit kalmasının amaçlandığını açıkça gösterir. Ayrıca oyunu bozabilecek yanlışlıkla değişiklikleri de önler.



Yeniden atama, bir adı yeni bir değere yöneltmek demektir. İçeriği değiştirmek ise var olan bir nesnenin tuttuğu şeyi değiştirmek demektir. Bunlar farklı eylemlerdir. Örneğin score değişkenini 0'dan 10'a yeniden atamak, adı yeni bir sayıya yöneltir. İçeriği değiştirmek, adı yeni bir listeye yöneltmeden bir listeye öğe ekleyerek onu değiştirmek gibidir. Bu iki fikri ayrı tut.



Bazı dillerde bir sabit adı, içeriği yine de değişebilen bir nesneye bağlı olabilir. Ad farklı bir nesneye yeniden atanamaz, ancak nesnenin kendisi değiştirilebilir. Örneğin bir sabit listeye öğe eklenebilir veya çıkarılabilir, ancak sabit adını yeni bir listeyi işaret edecek şekilde değiştiremezsin. Bu, yaygın bir kafa karışıklığı kaynağıdır, bu yüzden unutma: sabit ad sabittir, ancak nesnenin içeriği sabit olmayabilir.



Sabitler sabit kalması gereken değerleri korur, değişkenler ise değişmesi gereken değerlere izin verir. Bu, programların açık ve doğru kalmasına yardımcı olur. Örneğin bir vergi oranı için sabit kullanmak, onun yanlışlıkla değişmemesini sağlar. Bir kullanıcının yaşı için değişken kullanmak, onun güncellenebilmesini sağlar. Doğru olanı seçmek kodunu anlamayı kolaylaştırır ve hata olasılığını azaltır.



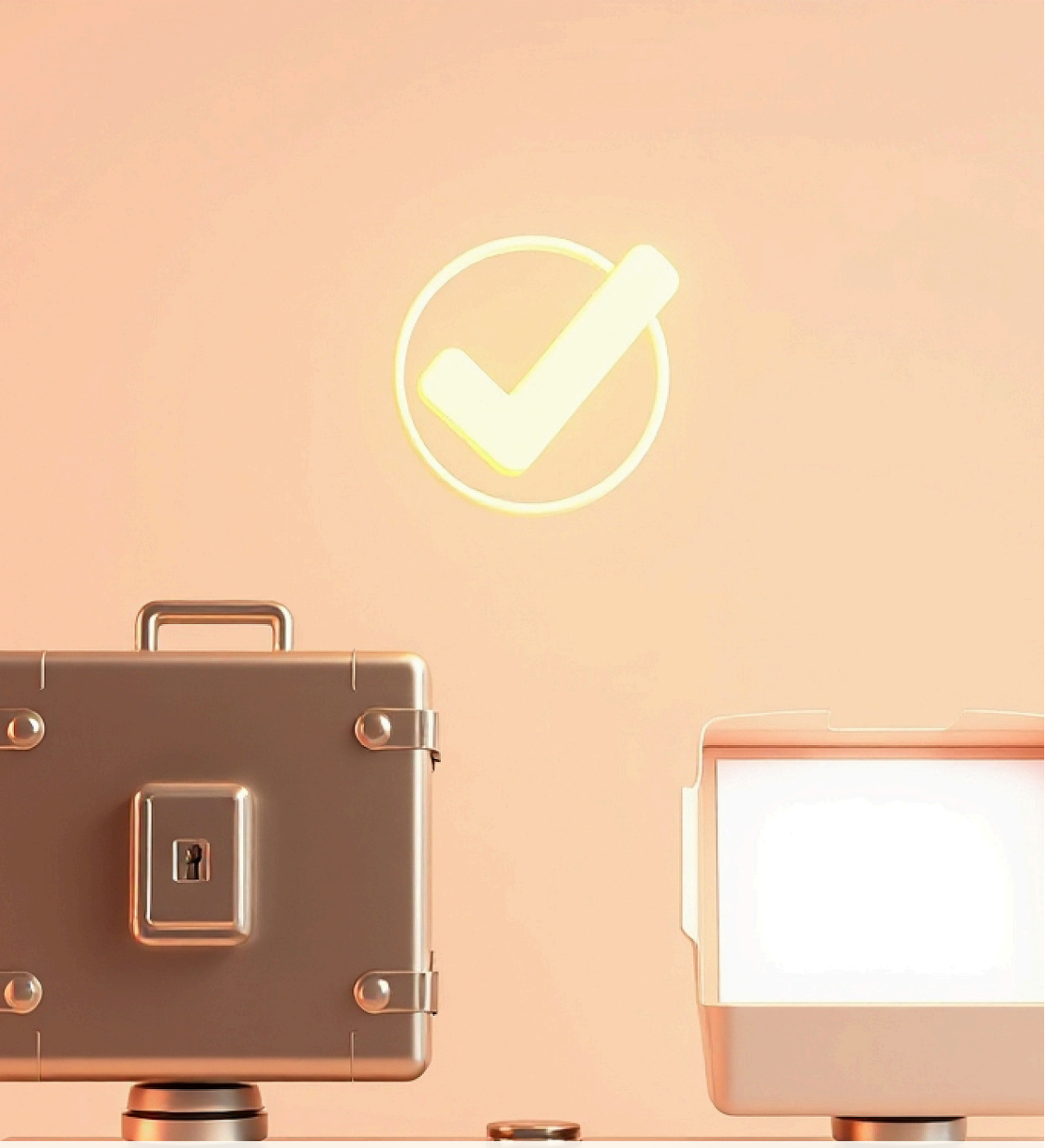
Yaygın bir hata, bir sabitin içeriğinin asla değişemeyeceğini düşünmektir. Gerçekte yeniden atama ile içeriği değiştirmek aynı şey değildir. Bir sabit yeniden atanamaz, ancak değiştirilebilir bir nesneyi işaret ediyorsa o nesnenin içeriği çoğu zaman değiştirilebilir. Örneğin bir sabit listeye öğe eklenebilir. Yani bir sabitin her şeyinin donmuş olduğunu varsayma.



Bir başka hata, bir değişkenin her durumda her zaman değiştirilebileceğini düşünmektir. Bir değişkenin ne zaman ve nasıl yeniden atanabileceğine dair program kuralları yine geçerlidir. Örneğin bir değişken belirli bağlamlarda salt okunur olabilir ya da türü hangi değerleri tutabileceğini kısıtlayabilir. Ayrıca bazı dillerde değişkenler için farklı kurallar vardır. Programlama dilinin kurallarını her zaman kontrol et.



Bir adı, işaret ettiği değerle karıştırmak yaygın bir hatadır. Ad yalnızca bir etikettir; değer ise programın gerçekte sakladığı şeydir. Örneğin score adı 10 sayısı değildir; 10'u işaret eder. Değeri değiştirirsen ad yine yeni değeri işaret eder. Adların değerlerin kendisi değil, başvurular olduğunu unutma.



Özetle: bir sabit yeniden atanamaz, bir değişken değer değiştirebilir ve yeniden atama bir nesnenin içeriğini değiştirmekten farklıdır. Bir sabit adının, içeriği değişebilen bir nesneyi işaret edebileceğini unutma. Sabitleri sabit değerler için, değişkenleri ise değişmesi gereken değerler için kullan. Bu temel noktalar sabitleri ve değişkenleri doğru kullanmana yardımcı olacaktır.