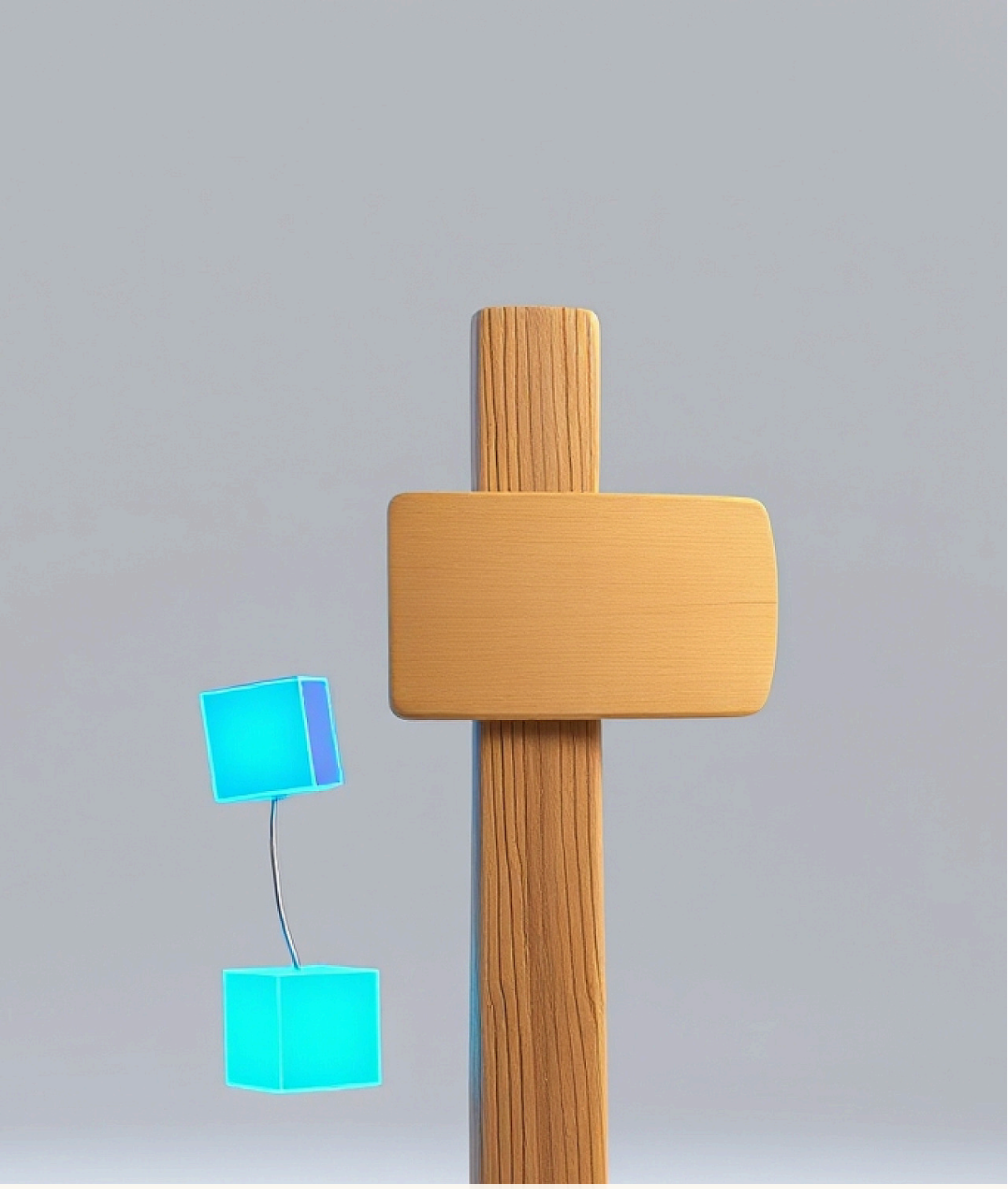
Bir değişkene değer atamak ne anlama gelir?



www.yzokul.com.tr

Bir değişkene değer atamak ne anlama gelir?



Değişken, bir programda bir değeri temsil eden bir addır. Bir değişkene değer atamak, o ada belirli bir bilgi parçasını vermek anlamına gelir; böylece program onu daha sonra kullanabilir. Örneğin, score = 10 yazmak, score adlı değişkene 10 değerini verir. Kısa yanıt: atama, bir adı bir değere bağlar.



Değişken, bir programda bir değeri temsil eden, etiket gibi bir addır. Henüz herhangi bir kod bilmen gerekmez; bunu bir isim etiketi olarak düşün. Ad, veriye başvurmak için programında yazdığın şeydir. Örneğin, score adı bir sayıyı temsil edebilir. Değişken değerin kendisi değildir; değere işaret eden addır.



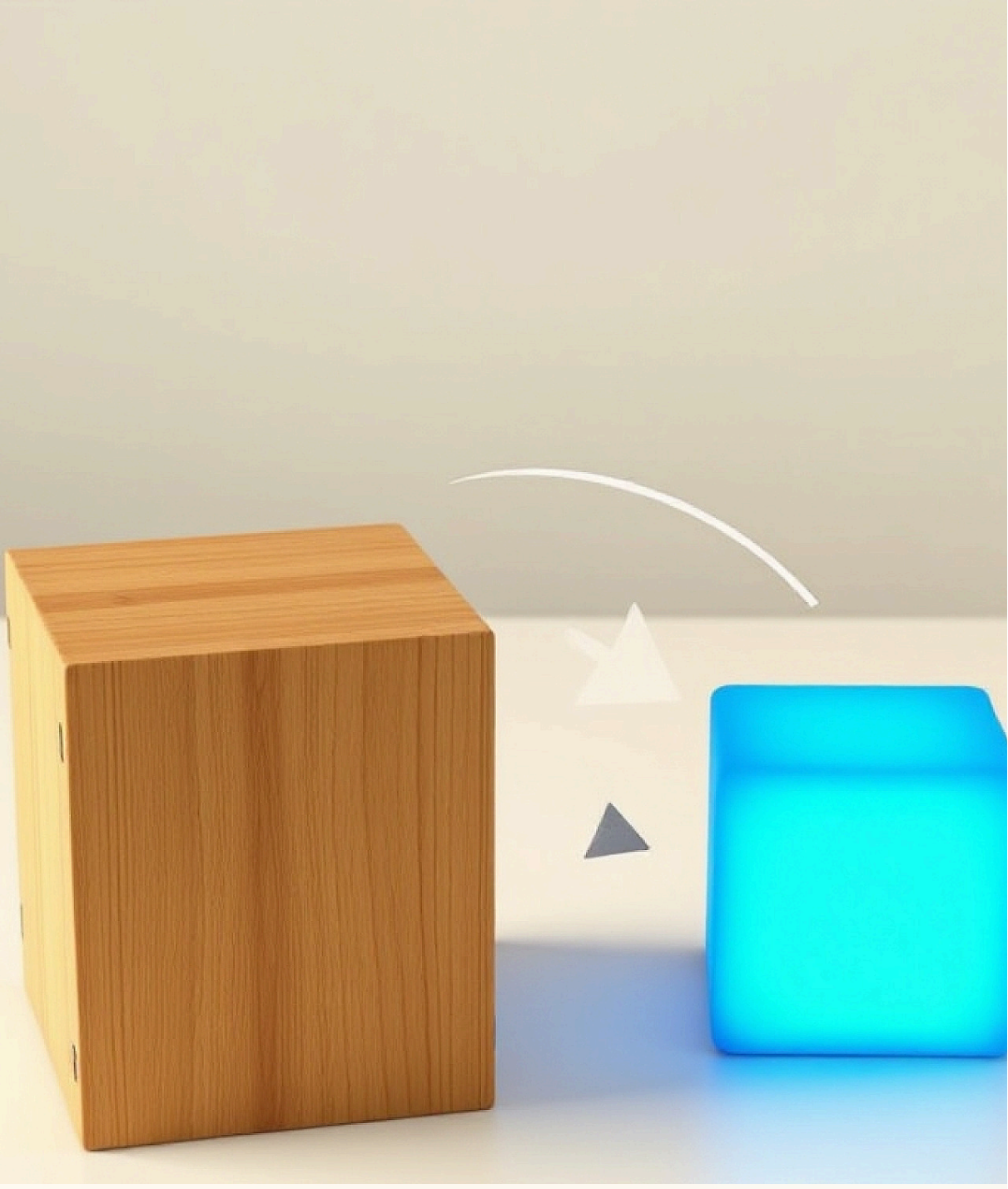
Değer, 10 sayısı veya hello gibi bir sözcük gibi bir bilgi parçasıdır. Ad bu bilgi parçasına işaret eder. Değerler, programının üzerinde çalıştığı gerçek verilerdir. Örneğin, 10 bir değerdir ve score ona işaret edebilen bir addır. Atama yaptığında, bir adı bir değere bağlarsın.



Atama, bir adı bir değere bağlama eylemidir. Atamadan sonra program adı gördüğünde hangi değerin kastedildiğini bilir. Örneğin, score = 10 ifadesinden sonra program score'un 10 anlamına geldiğini bilir. Atama bir eylemdir, bir olgu bildirimi değildir. Adın neyi temsil ettiğini değiştirir.



Birçok dilde score = 10 yazmak, score adlı değişkene 10 değerini verir. score adı artık 10'u temsil eder. Buradaki eşittir işareti atama operatörüdür; sağdaki değeri soldaki ada sakla anlamına gelir. Örneğin, bu ifadeden sonra programda score kullanmak sana 10 verir. Bu, atamanın temel biçimidir.



score = 10 ifadesini score 10 alır veya score artık 10 tutar diye oku. Ad soldadır, değer sağdadır. Sol taraf bir değer değil, bir değişken adı olmalıdır. Sağ taraf saklamak istediğin değerdir. Örneğin, score = 10 içinde score addır ve 10 değerdir. Bu okuma kafa karışıklığından kaçınmana yardımcı olur.



Matematikte score = 10 iki şeyin eşit olduğunu iddia ederdi. Programlamada bu bir eylemdir: 10 değerini score değişkenine koyar. Eşittir işareti burada bir karşılaştırma değildir; saklama komutudur. Örneğin, score = 10 yazmak score 10'a eşit mi diye sormaz; score'un 10 tutmasını sağlar. Bu fark yeni başlayanlar için önemlidir.



İfadeden önce score'un değeri olmayabilir veya farklı bir değeri olabilirdi. İfadeden sonra programın durumu değişmiştir: score 10 tutar. Program durumu, tüm değişkenlerin mevcut tüm değerlerinin toplamıdır. Örneğin, score önceden 3 idiyse, score = 10 ifadesinden sonra 10 olur. Atama, çalışan bir programın durumunu değiştirme biçimindir.



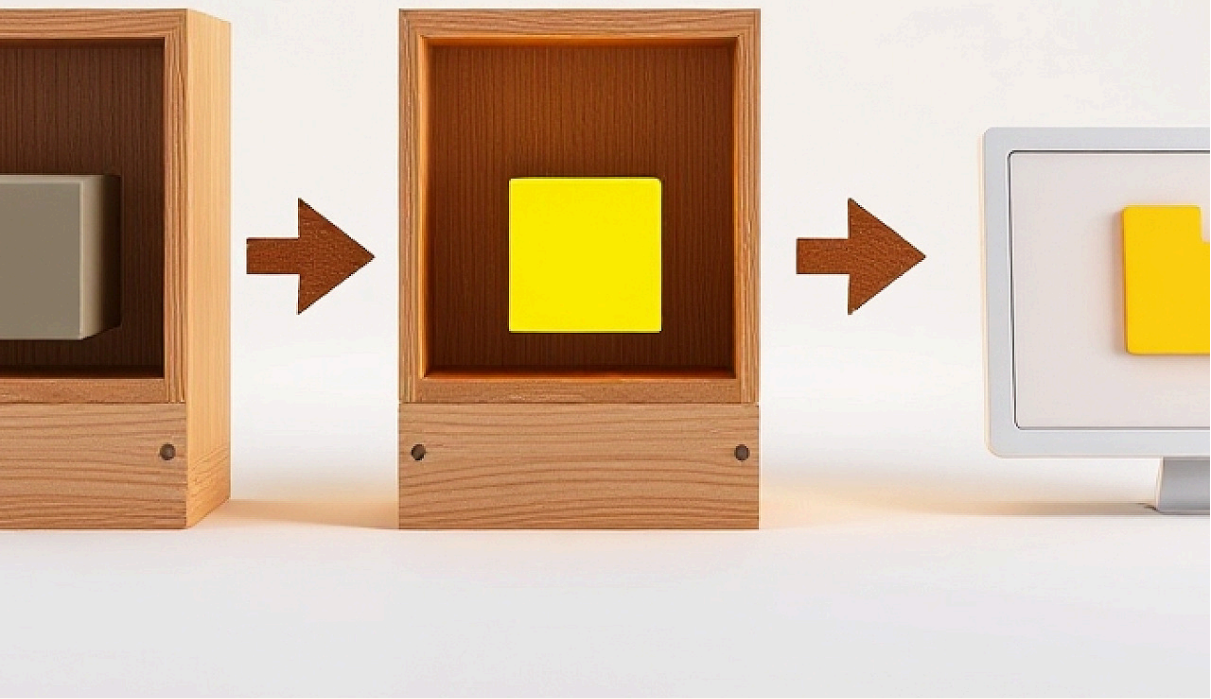
score 10 tuttuğunda, diğer ifadeler score adını kullanabilir ve 10 elde edebilir. Ad, değerin yerine geçen bir temsilci olarak çalışır. Örneğin, print(score) gibi sonraki bir ifade 10 gösterir. 10 değerini tekrarlaman gerekmez; sadece adı kullanırsın. Değişkenlerin yararlı olmasının nedeni budur: değerleri saklamana ve yeniden kullanmana olanak tanırlar.



Sonraki bir ifade, yeni bir değer hesaplamak için eski değeri kullanabilir. Örneğin, score = score + 5 eski score değerini (10) okur, 5 ekler ve 15'i tekrar score'a atar. Sağ taraf önce mevcut değer kullanılarak hesaplanır. Sonra sonuç soldaki değişkende saklanır. Bu kalıp programlarda yaygındır.



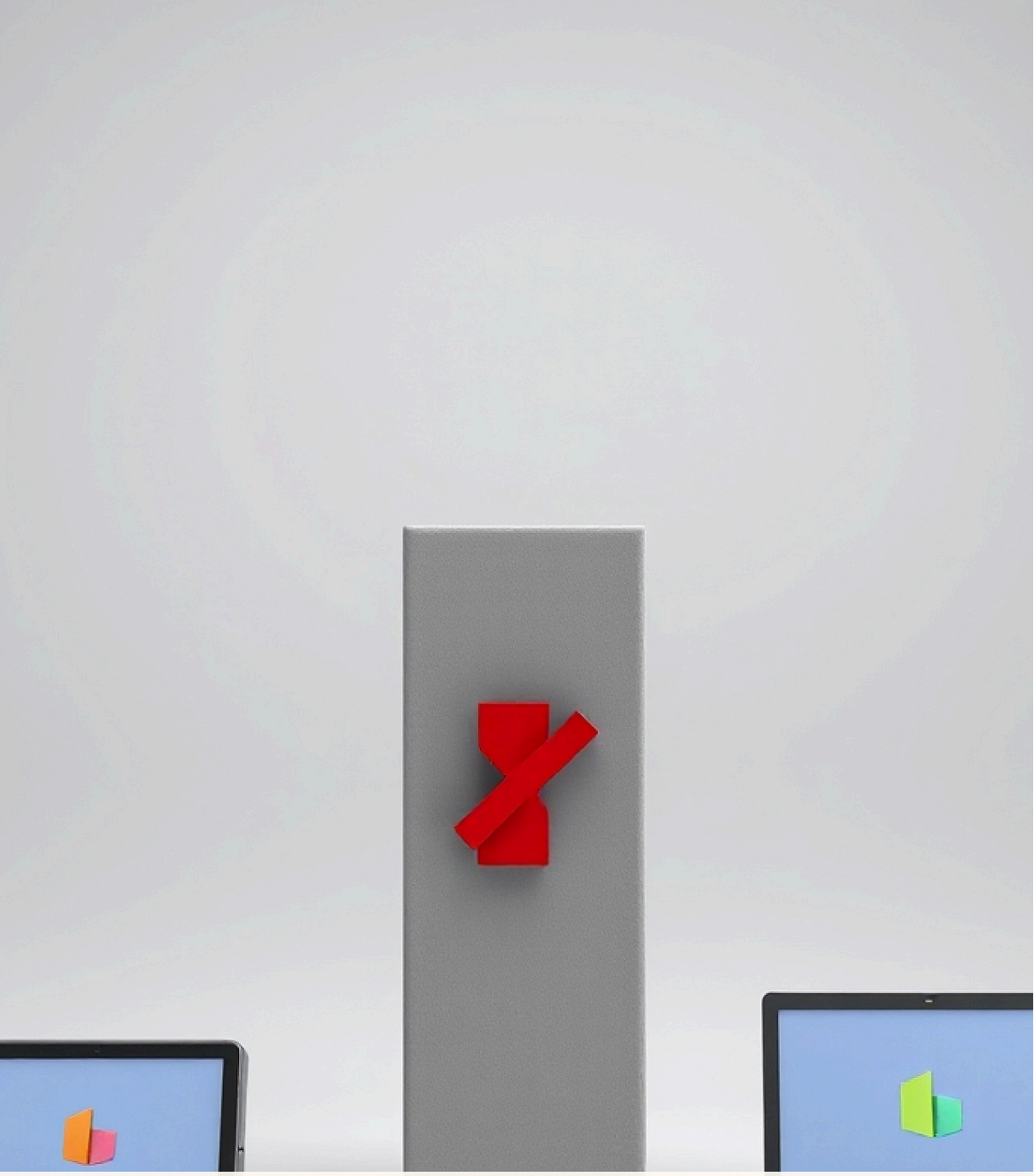
Sağ taraf önce mevcut değer kullanılarak hesaplanır. Sonra sonuç, eski değerin yerine geçerek değişkende saklanır. Örneğin, score = score + 5 içinde program önce score'un mevcut değerini (10) bulur, 5 ekleyerek 15 elde eder, sonra 15'i score'da saklar. Atamadan sonra eski değer gitmiştir. Bu sıra her zaman aynıdır.



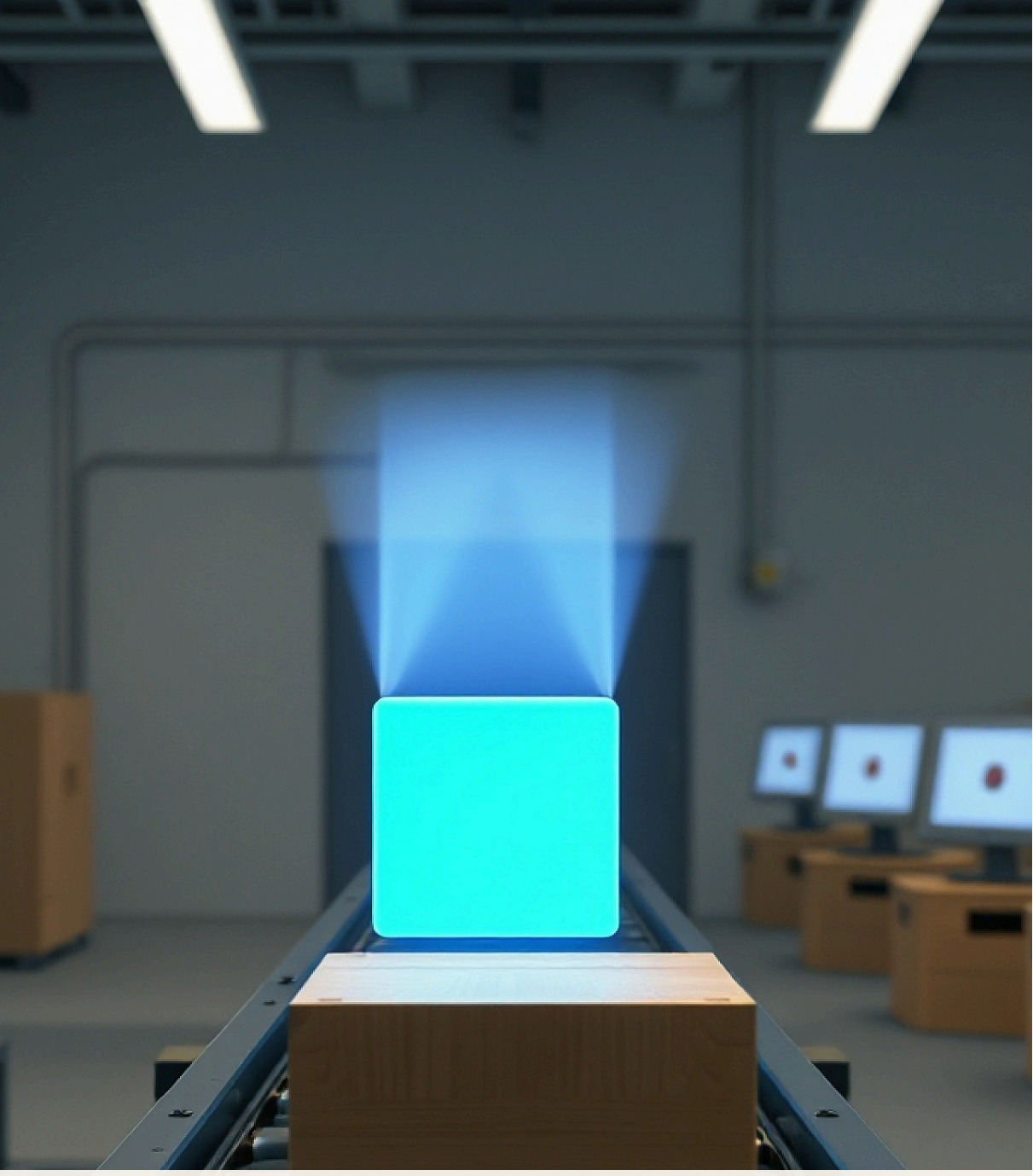
İfadeler birbiri ardına çalışır, bu yüzden atamaların sırası önemlidir. Sırayı değiştirmek değişkenin son değerini değiştirebilir. Örneğin, önce score = 10 sonra score = score + 5 yazarsan score 15 olur. Sırayı tersine çevirirsen sonuç farklı olur veya hata çıkar. Atamalarının sırasını her zaman kontrol et.



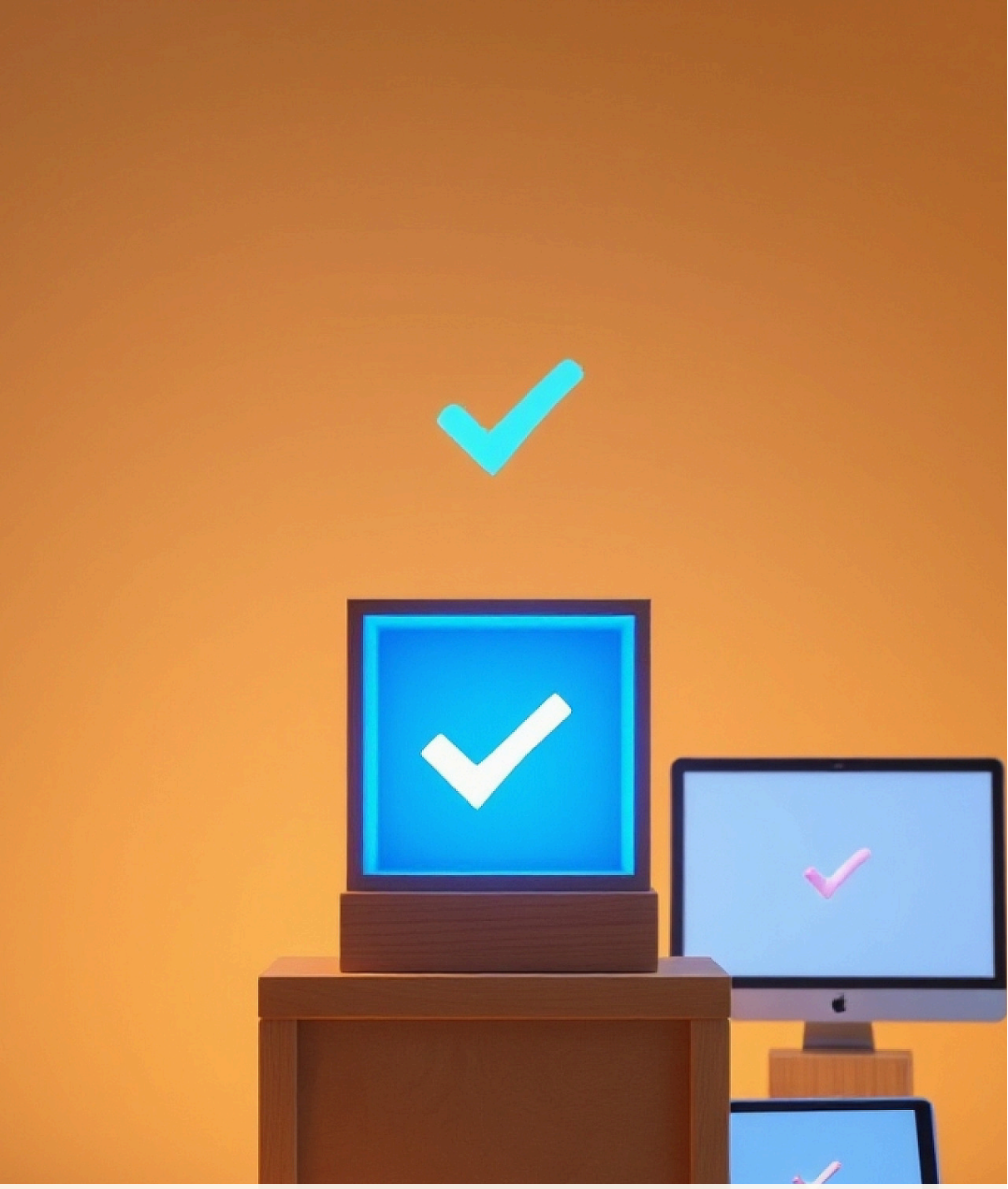
Farklı diller atamayı biraz farklı biçimlerde yazar. Simgeler veya anahtar sözcükler farklı olsa da fikir aynıdır. Örneğin, bazı diller = kullanır, bazıları := veya let kullanır. Python'da score = 10 çalışır; başka dillerde int score = 10; yazabilirsin. Temel anlam—bir adda bir değer saklamak—değişmez.



Yeni başlayanlar = işaretini matematik dersindeki eşittir diye okur. Atamada bu, bu değeri bu adda sakla anlamına gelir, bir karşılaştırma değildir. Örneğin, score = 10 bir soru sormaz; bir emir verir. Değerleri karşılaştırmak için programlama dilleri genellikle == olan farklı bir simge kullanır. Unutma: tek eşittir işareti genellikle atama anlamına gelir.



score = score + 5 içinde yeni başlayanlar score'un aynı anda iki kez kullanıldığını düşünebilir. Aslında önce eski değer okunur, sonra yeni değer saklanır. Örneğin, score 10 ise sağ taraf 10 + 5 = 15 olur ve sonra score 15 olur. Soldaki ad, saklama gerçekleşene kadar okunmaz. Bu iki adımlı süreç anahtardır.



Atama, score = 10 gibi bir adı bir değere bağlamak anlamına gelir ve programın durumunu değiştirir. Unutma: önce sağ tarafı oku, sonra sonucu soldaki adda sakla. Örneğin, score = score + 5 yeni bir değer yapmak için eski değeri kullanır. Bu basit fikir, programların verileri nasıl takip ettiğinin temelidir.