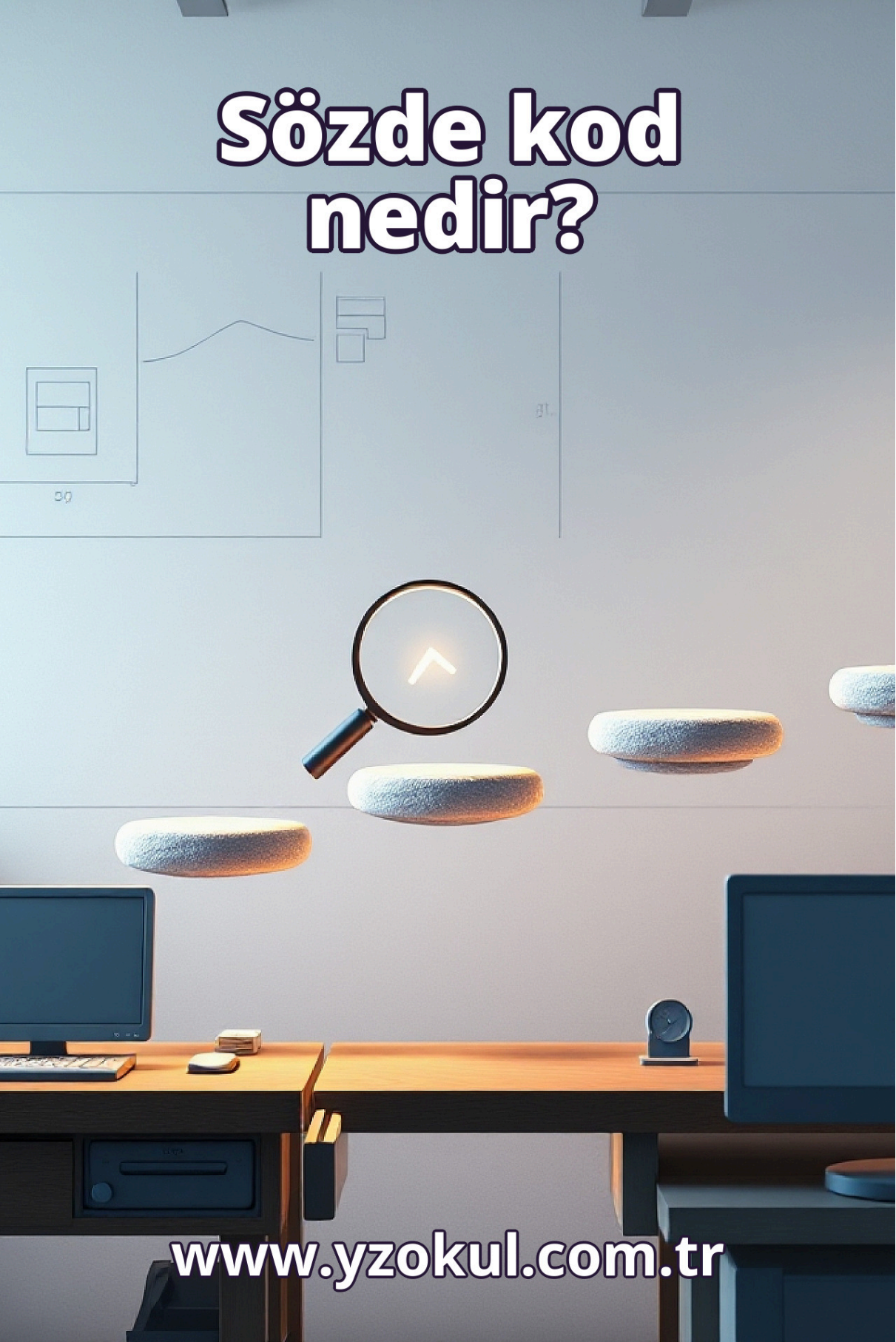
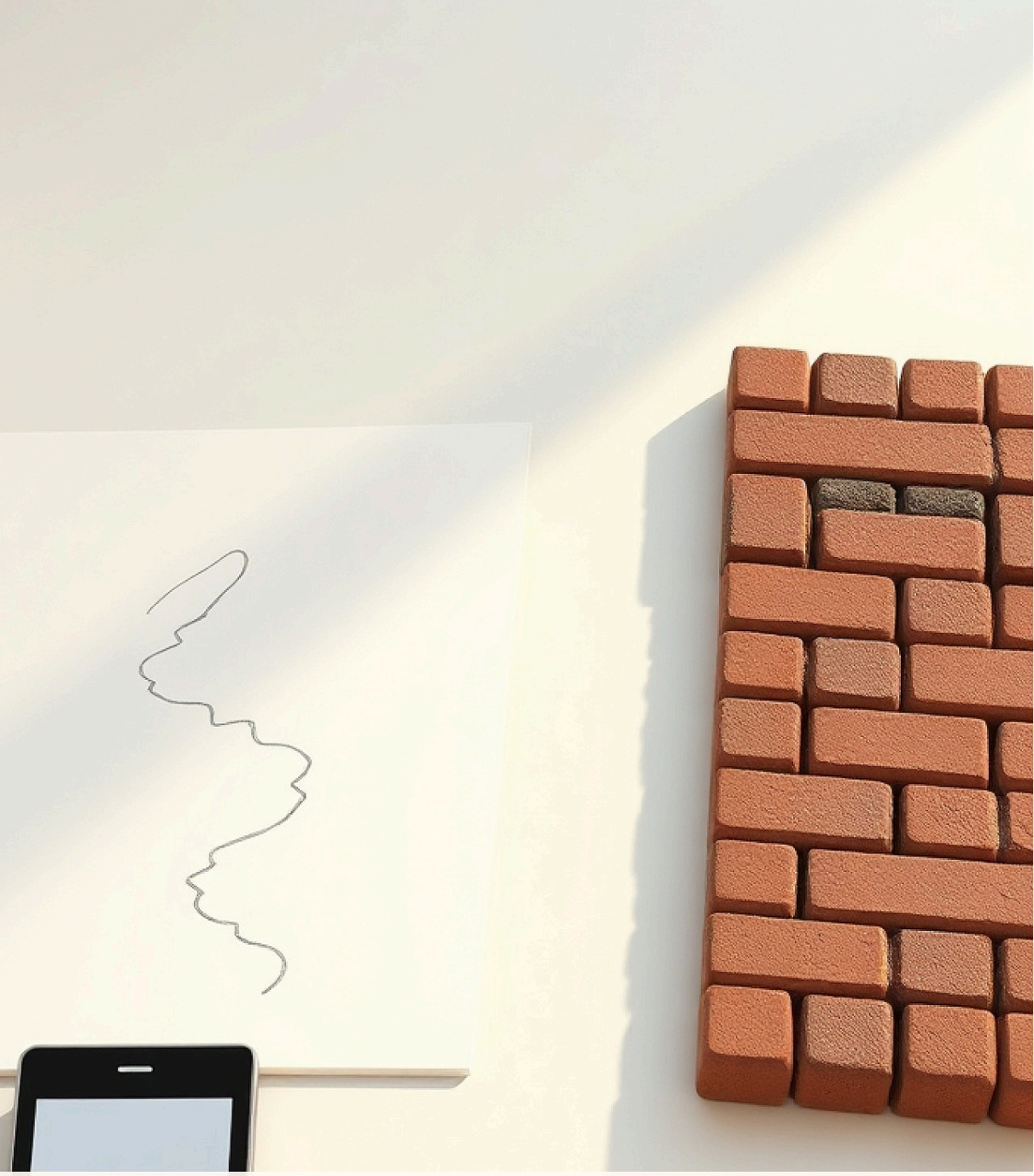
Sözde kod nedir?



www.yzokul.com.tr

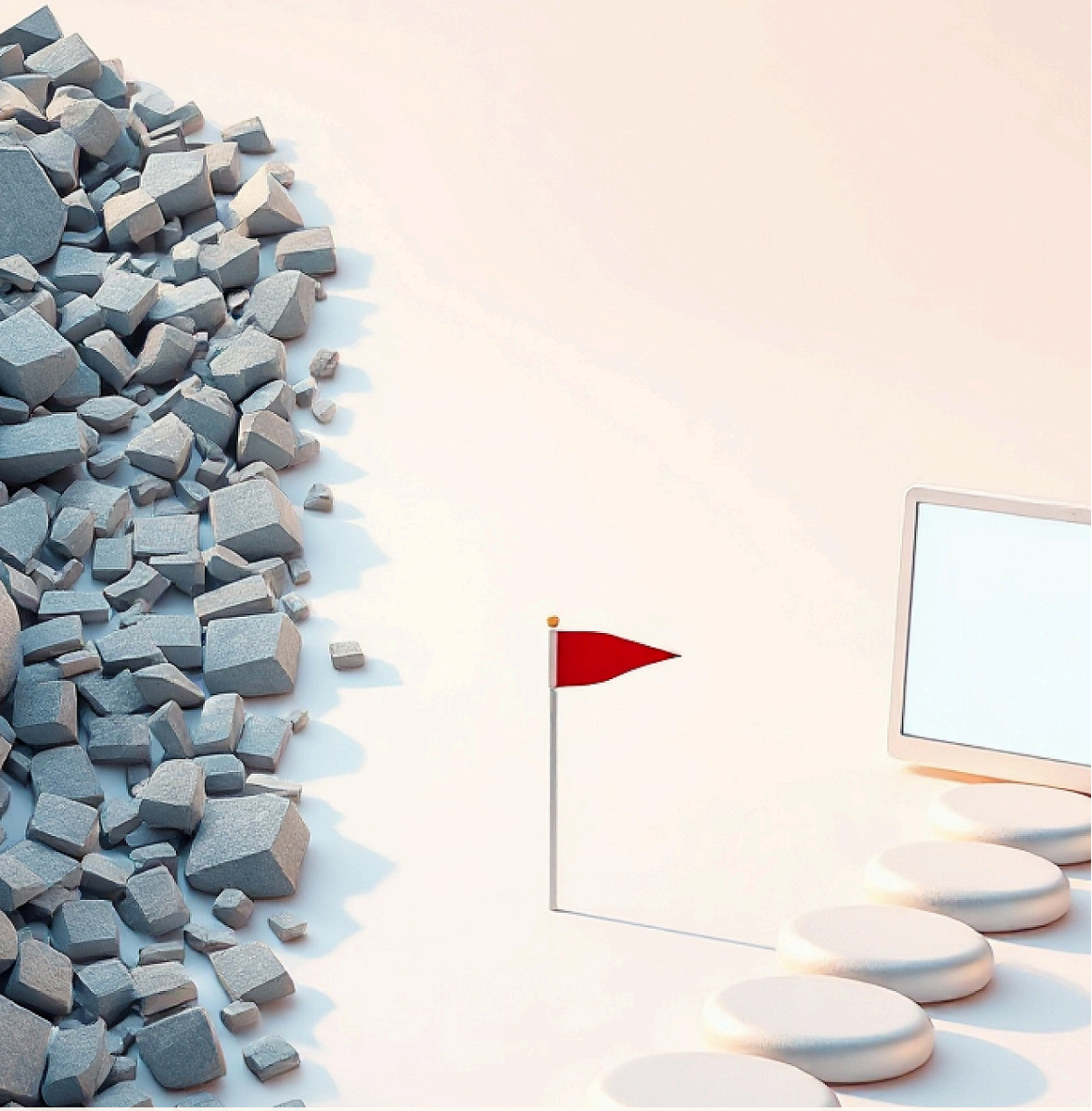
Sözde kod nedir?



Sözde kod, bir çözümün adımlarını sade, insan dostu sözcüklerle yazmanın bir yoludur. Gerçek bir programlama dili değildir; yalnızca bir algoritmanın açık bir tanımıdır. Gerçek kod yazmaya başlamadan önce bir problemin nasıl çözüleceğini planlamak için kullanılır. Bunu, insanların kolayca okuyabilmesi için yazılmış mantığının bir taslağı olarak düşün.



Algoritma, bir problemi çözmek için gereken adımların basit bir listesidir; tıpkı bir tarifin sana kekin nasıl pişirileceğini söylemesi gibi. Her adım bir eylemdir ve bunları sırayla izlemek sonucu verir. Sözde kod, böyle bir listeyi yazmanın bir yoludur. Bir algoritmayı anlamak için bilgisayara ihtiyacın yoktur; sadece adımları okursun.



Gerçek programlama dillerinde birçok katı yazım kuralı vardır: tam anahtar sözcükler, noktalama ve biçimlendirme. Bu kurallar, sen hâlâ çözümü düşünürken işini zorlaştırır. Sözde kod, tam sözdizimi konusunda endişelenmeden önce fikre odaklanmanı sağlar. Mantığı serbestçe yazabilir ve bir bilgisayarın bunu nasıl çalıştıracağını ancak daha sonra düşünebilirsin.



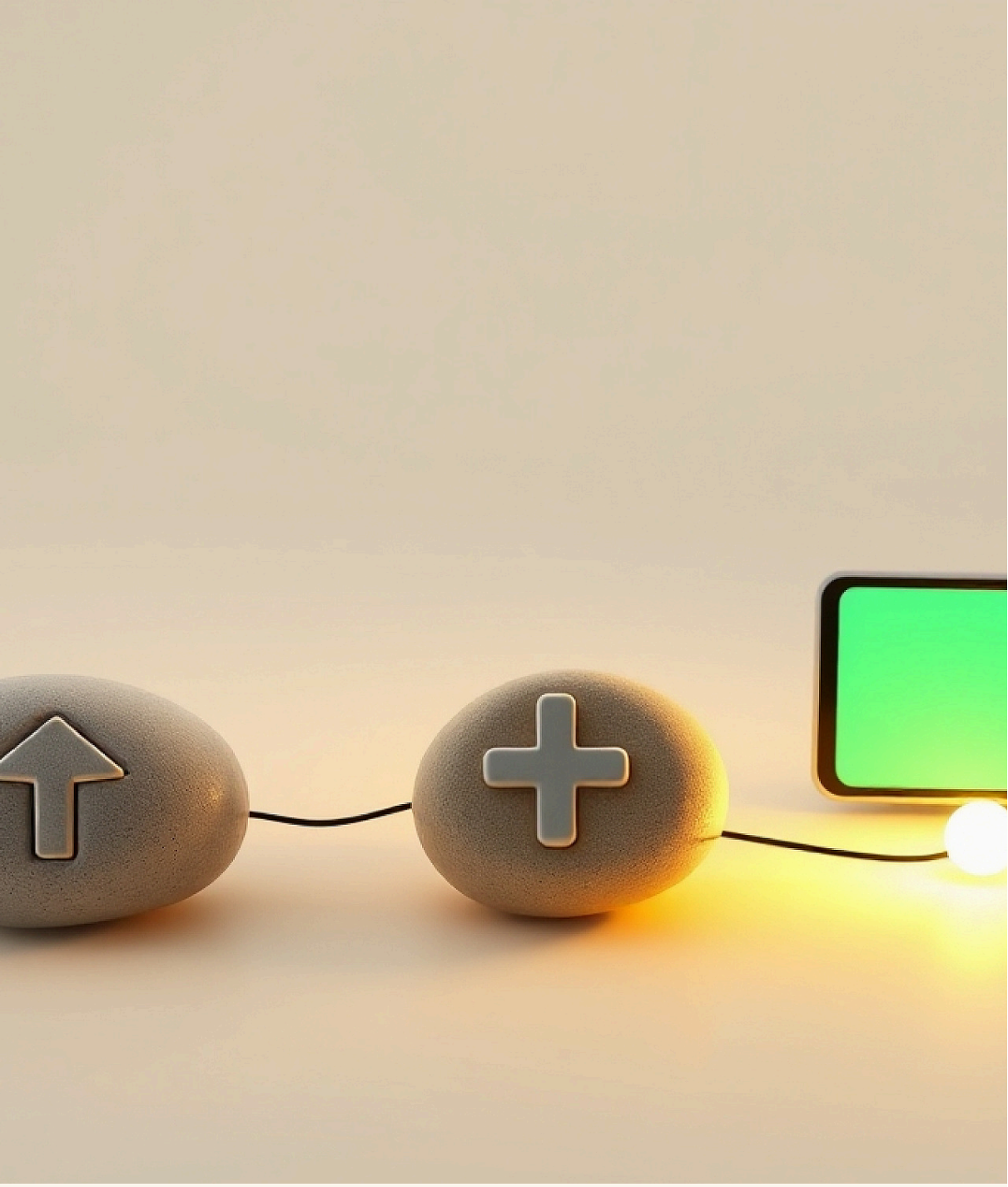
Sözde kod sıradan dille yazılabilir. Örneğin şöyle yazabilirsin: Sayıyı al, sıfırdan büyükse ekrana yazdır. Bu cümle açıktır ve tam olarak ne olması gerektiğini anlatır. Özel simgelere veya komutlara ihtiyacın yoktur. Anlamı sade sözcükler taşır ve herkes bunları okuyabilir.



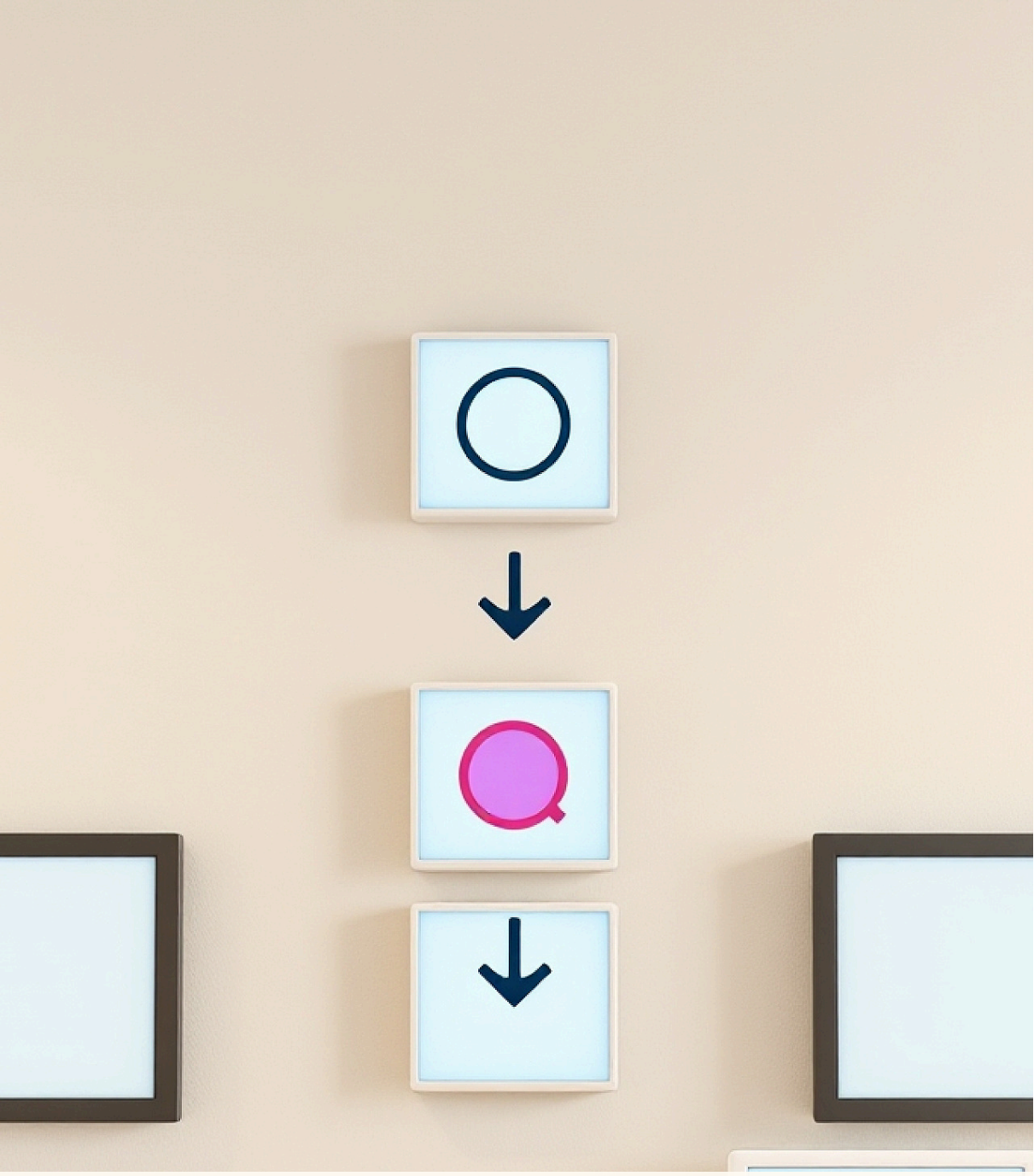
Sözde kodun temel amacı, çözümün mantığını kolayca görünür kılmaktır. Okuyan herkes, bir programlama dili bilmeden adımları izleyebilmelidir. Mantık açık olduğunda hataları kontrol edebilir ve planı geliştirebilirsin. İyi sözde kod, belirli bir dilin ayrıntılarını değil, çözümün arkasındaki düşünmeyi gösterir.



Sözde kod tam anahtar sözcükler, noktalama veya biçimlendirme talep etmez. Onu, adımları sana ve okuyucularına en açık hâle getiren şekilde yazarsın. Eksik bir noktalı virgül için şikâyet edecek bir derleyici yoktur. Adımları gruplamak için girinti veya satır sonları kullanabilirsin, ama bunlar tercihtir, katı kurallar değildir. Bu özgürlük, anlama odaklanmana yardımcı olur.



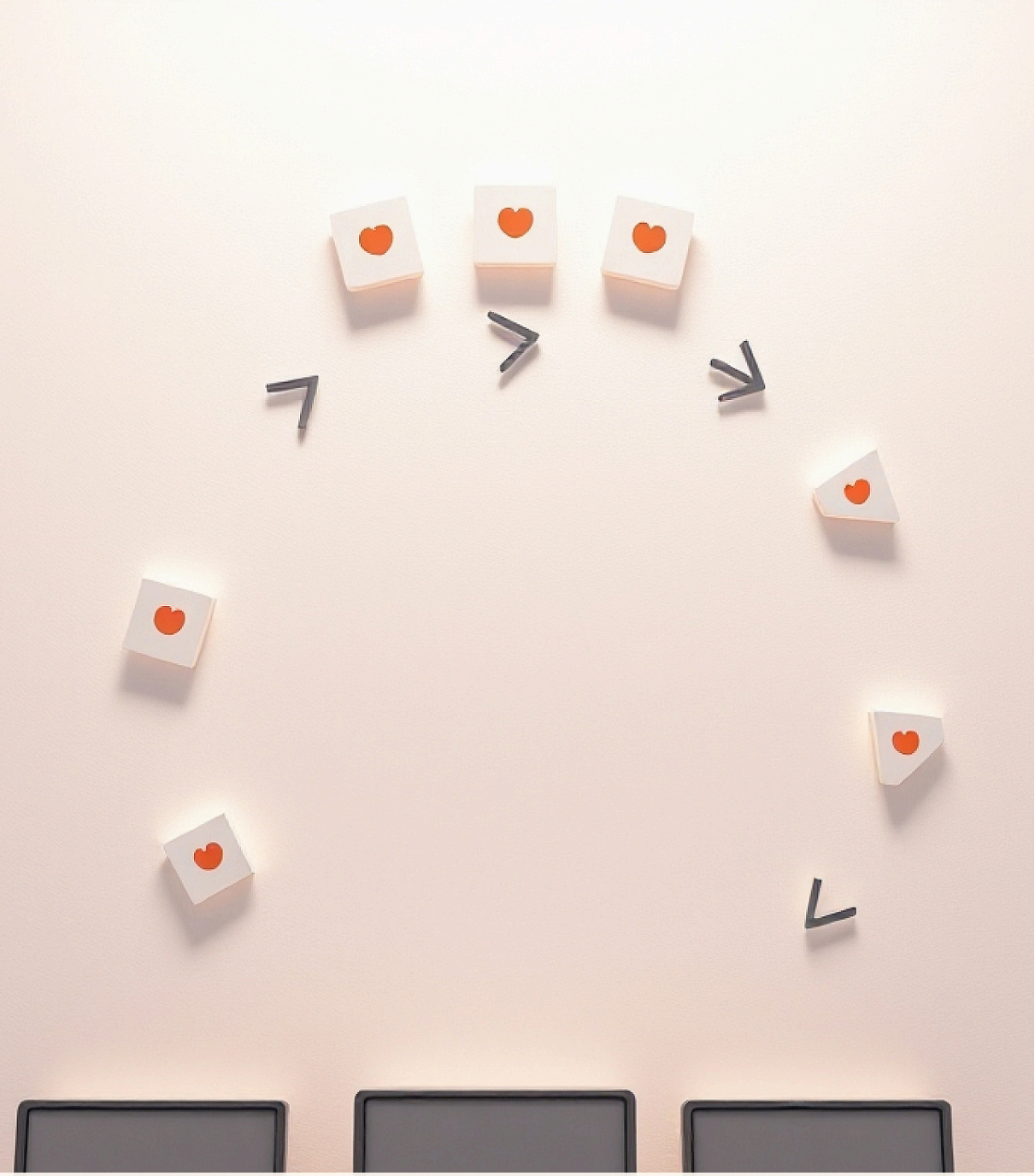
Küçük bir örnek üzerinden geç: bir sayının pozitif olup olmadığını kontrol et ve yazdır. Sözde kodda şöyle yazabilirsin: Sayıyı oku. Sayı sıfırdan büyükse sayıyı yazdır. Aksi hâlde hiçbir şey yapma. Her satır sırayla bir eylemi anlatır. Kararı ve eylemi herhangi bir kod sözdizimi olmadan açıkça görebilirsin.



Sözde kod, eylemleri tıpkı gerçek bir programın çalışması gibi yukarıdan aşağıya, birbiri ardına listeler. Adımların sırası sonuç için önemlidir. Sayıyı okumadan önce yazdırırsan, henüz değere sahip olmazsın. Adımları sırayla yazmak, eksik eylemleri veya yanlış sıralamaları fark etmene yardımcı olur. Adımları her zaman olmaları gereken sıraya göre düzenle.



Bir seçim sade sözcüklerle yazılır. Örneğin: sayı sıfırdan büyükse, o zaman sayıyı yazdır. Bu, özel simgeler olmadan bir kararı gösterir. Ayrıca bir else kısmı ekleyebilirsin: aksi hâlde bir mesaj yazdır. if, then ve otherwise sözcükleri dalı açık hâle getirir. Okuyan herkes hangi koşulda hangi eylemin gerçekleştiğini görebilir.



Sözde kod, bir döngüyü listedeki her öğe için bunu tekrarla gibi sade sözcüklerle gösterebilir. Özel döngü anahtar sözcüklerine ihtiyacın yoktur. Örneğin: listedeki her sayı için, sayı sıfırdan büyükse onu yazdır. Bu, okuyucuya aynı eylemin her öğe için gerçekleştiğini söyler. Anlam, herhangi bir programlama sözdizimi olmadan açıktır.



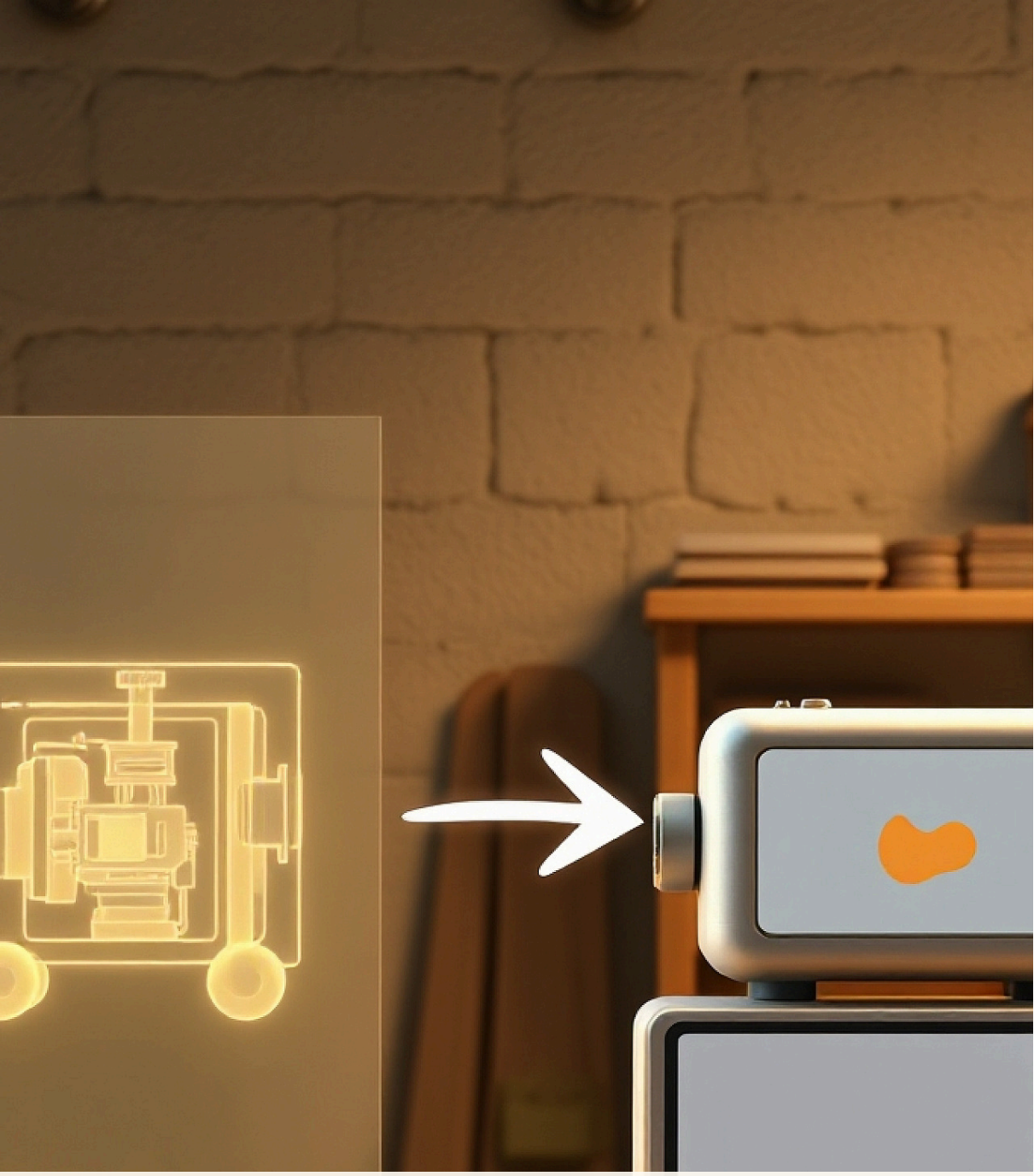
Herkesin kullanması gereken tek bir resmî sözde kod yoktur. Farklı kişiler onu biraz farklı yazar ve bu sorun değildir. Bazıları read ve print gibi sözcükler kullanır; bazıları get ve show kullanır. Bazıları girinti koyar, bazıları koymaz. Önemli olan adımların anlaşılır olmasıdır. Sana ve kitlene uygun bir üslup seçebilirsin.



En önemli kural açıklıktır: adımlar okunması ve anlaşılması kolay olmalıdır. Okuyucu kafası karışırsa sözde kodun yeniden yazılması gerekir. Sade sözcükler ve kısa cümleler kullan. Burada bir şey yap gibi belirsiz ifadelerden kaçın. Bunun yerine tam olarak hangi eylemin yapılacağını söyle. Açık sözde kod, herkesin soru sormadan mantığı izlemesini sağlar.



Tek bir sözde kod parçası içinde baştan sona aynı üslubu ve sözcükleri koru. Başta read sözcüğünü kullanırsan, sonra get'e geçme. Bir if içinde girinti kullanıyorsan bunu tutarlı yap. Üslupları karıştırmak mantığı izlemeyi zorlaştırır. Tutarlılık, okuyucunun senin tercihlerini çözmeye değil anlama odaklanmasına yardımcı olur.



Mantık açık hâle geldikten sonra sözde kod adımları gerçek bir programlama diline çevrilir. Sözde kod plandır; kod ise tamamlanmış yapıdır. Her sade adımı alır ve dilinin tam sözdizimini kullanarak yazarsın. Mantık zaten doğru olduğu için çeviri daha kolay ve daha az hataya açıktır. Yazım kurallarına yalnızca bu aşamada odaklanırsın.



Yeni başlayanların sık yaptığı hatalara dikkat et. Gerçek kod sözdizimini çok erken yazmak, mantığı ayrıntıların arkasında gizler. Üslubu yarı yolda değiştirmek okuyucunun kafasını karıştırır. Girdiyi işle gibi belirsiz adımlar bırakmak, eylemi belirsiz kılar. Her hata sözde kodu daha az kullanışlı hâle getirir. Açık, tutarlı ve sade adımlara odaklanarak bunlardan kaçın.



Sözde kod, bir algoritmayı katı dil kuralları olmadan sade ve açık adımlarla tanımlar. Daha sonra gerçek koda dönüşür. Açık ve tutarlı tutmayı ve önce mantığa odaklanmayı unutma. Sıradan sözcükler kullan, adımları sırayla listele, kararları ve döngüleri sade biçimde göster. Mantığı izlemek kolay olduğunda koda çevirmek basit bir iş hâline gelir.