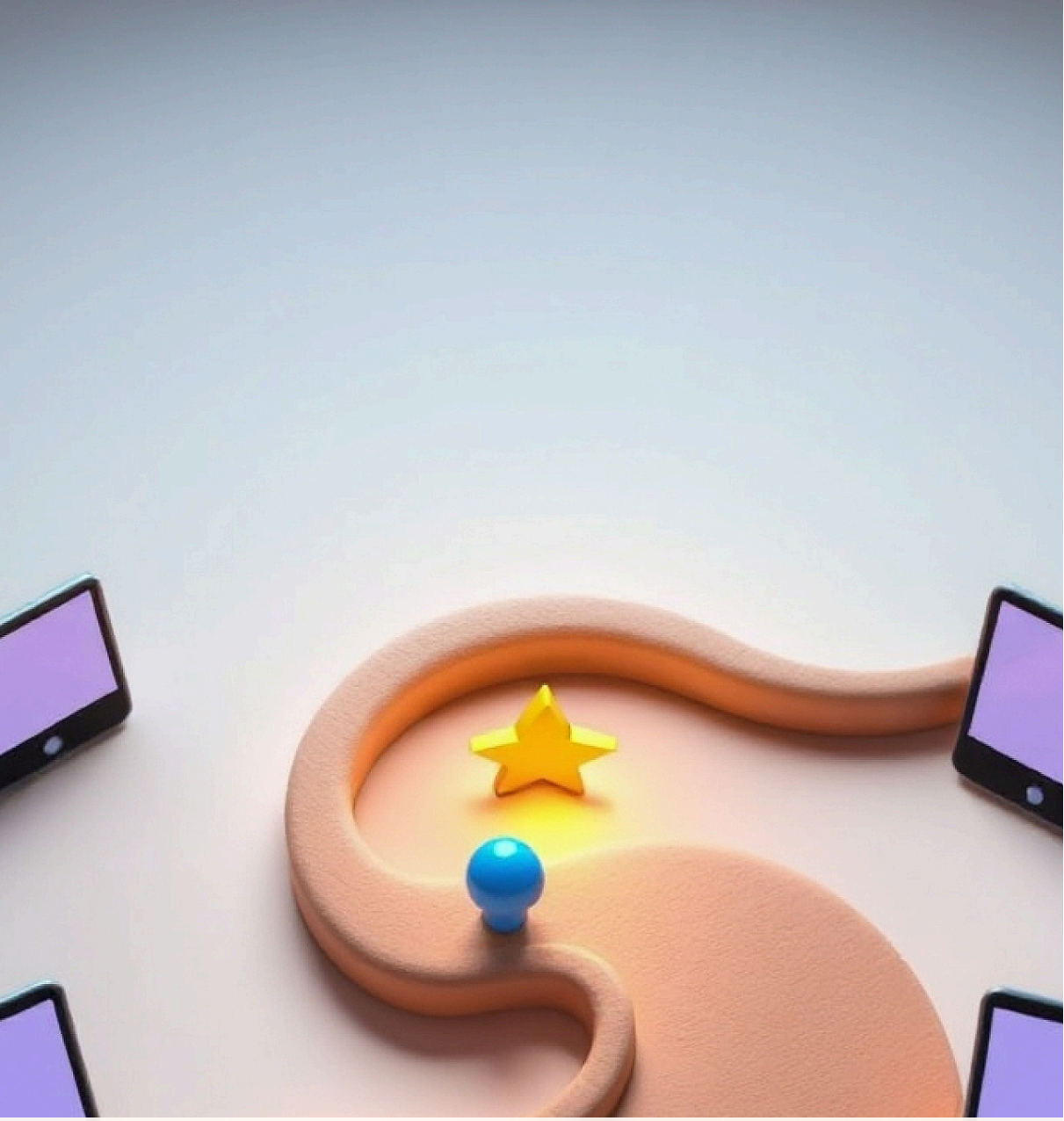
Bir sayı tahmin oyunu nasıl kodlanır?



www.yzokul.com.tr

Bir sayı tahmin oyunu nasıl kodlanır?



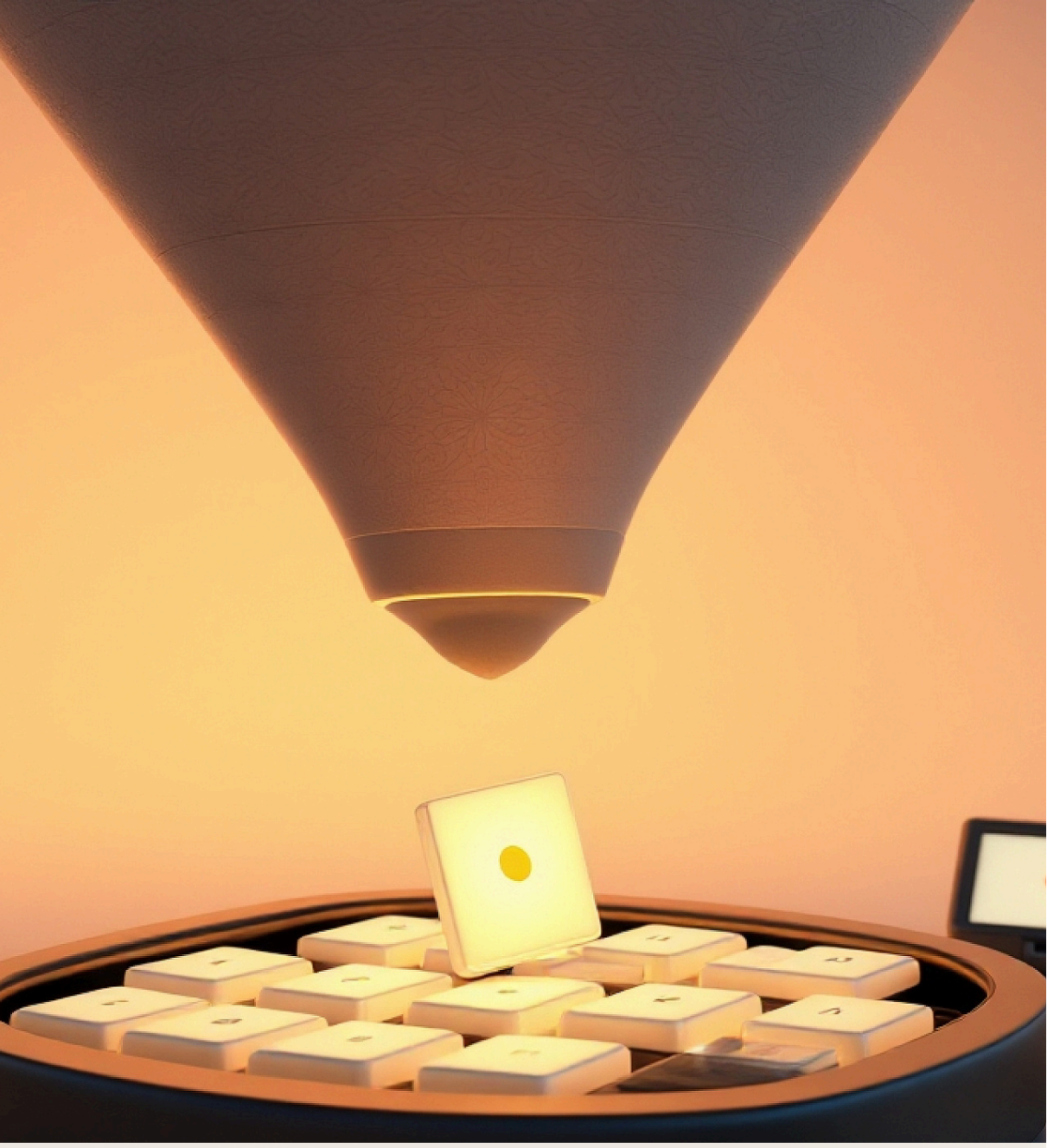
Bir sayı tahmin oyunu, seninle tahmin yarışı oynayan küçük bir programdır. Bilgisayar gizli bir sayı seçer ve sen tahminler yazarak onu bulmaya çalışırsın. Her tahminden sonra program sana daha yükseğe mi yoksa daha düşüğe mi gitmen gerektiğini söyler. Oyun, gizli sayıyı tam olarak tahmin edene kadar devam eder. Nasıl kodlayacağının kısa yanıtı şudur: gizli bir sayı seç, her tahmini karşılaştır, ipuçları ver ve doğru olana kadar döngü kur.



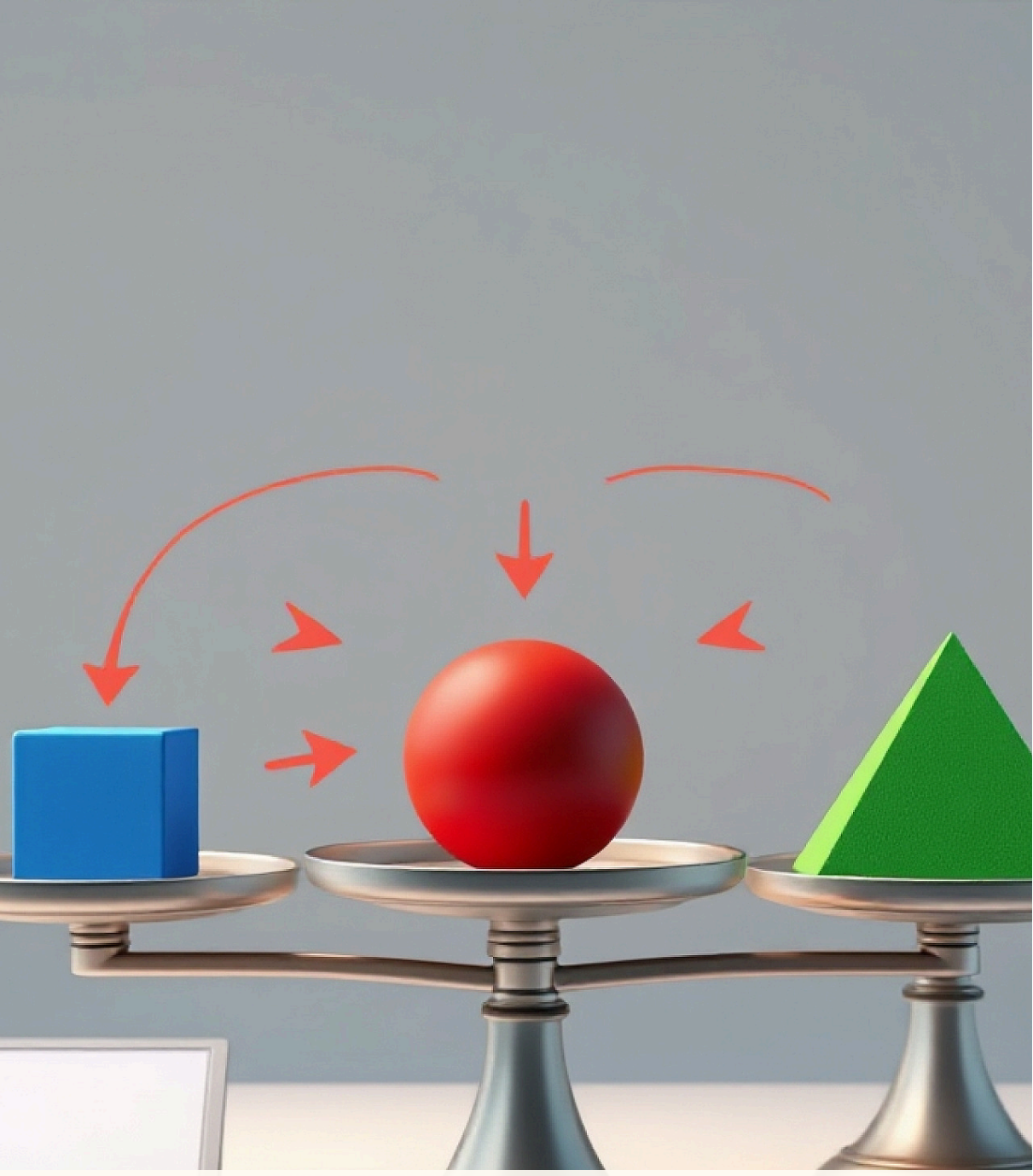
Programın 1 ile 100 gibi sabit bir aralıkta gizli bir sayı seçmesini sağlayarak başla. Program, öngörülemez bir değer üreten yerleşik bir araç olan rastgele sayı üreteci kullanır, böylece oyun her seferinde farklı olur. Bu gizli sayıyı bir değişkende sakla ki programın geri kalanı tahminleri onunla karşılaştırabilsin. İlk başta aralığı 1 ile 10 gibi küçük tut, böylece oyunu hızlıca test edebilirsin.



Değişken, bilgisayarın belleğinde bir değer tutan adlandırılmış bir kutudur. Ona secret gibi bir ad verirsin ve içine bir sayı koyarsın. Daha sonra değeri okuyabilir veya yeni bir değerle değiştirebilirsin. Bu oyunda gizli sayı için bir, deneme sayısı için başka bir değişken kullanırsın. Ad, değerin nerede saklandığını hatırlamana gerek kalmadan ona başvurmanı sağlar.



Program şimdi oyuncudan bir tahmin yazmasını ister ve bu girdiyi okur. Klavyeden gelen girdi genellikle metin olarak gelir, bu yüzden kullanmadan önce onu bir sayıya dönüştür. Birçok dilde bunun için bir fonksiyon vardır, örneğin Python'da int(input()). Dönüştürülen tahmini de bir değişkende sakla ki gizli sayıyla karşılaştırabilesin. Dönüştürme olmadan program sayı karşılaştırmalarını doğru yapamaz.



Şimdi program tahmini gizli sayıyla karşılaştırır. Üç olası sonuç vardır: tahmin hedefe eşittir, hedeften küçüktür veya hedeften büyüktür. Her sonuç farklı bir mesaja yol açar. Bu karşılaştırma oyunun kalbidir, çünkü her ipucu ve her kazanma ondan gelir. Tek bir karşılaştırma satırı, koşullarla birleştiğinde üç sonucun hepsini üretebilir.



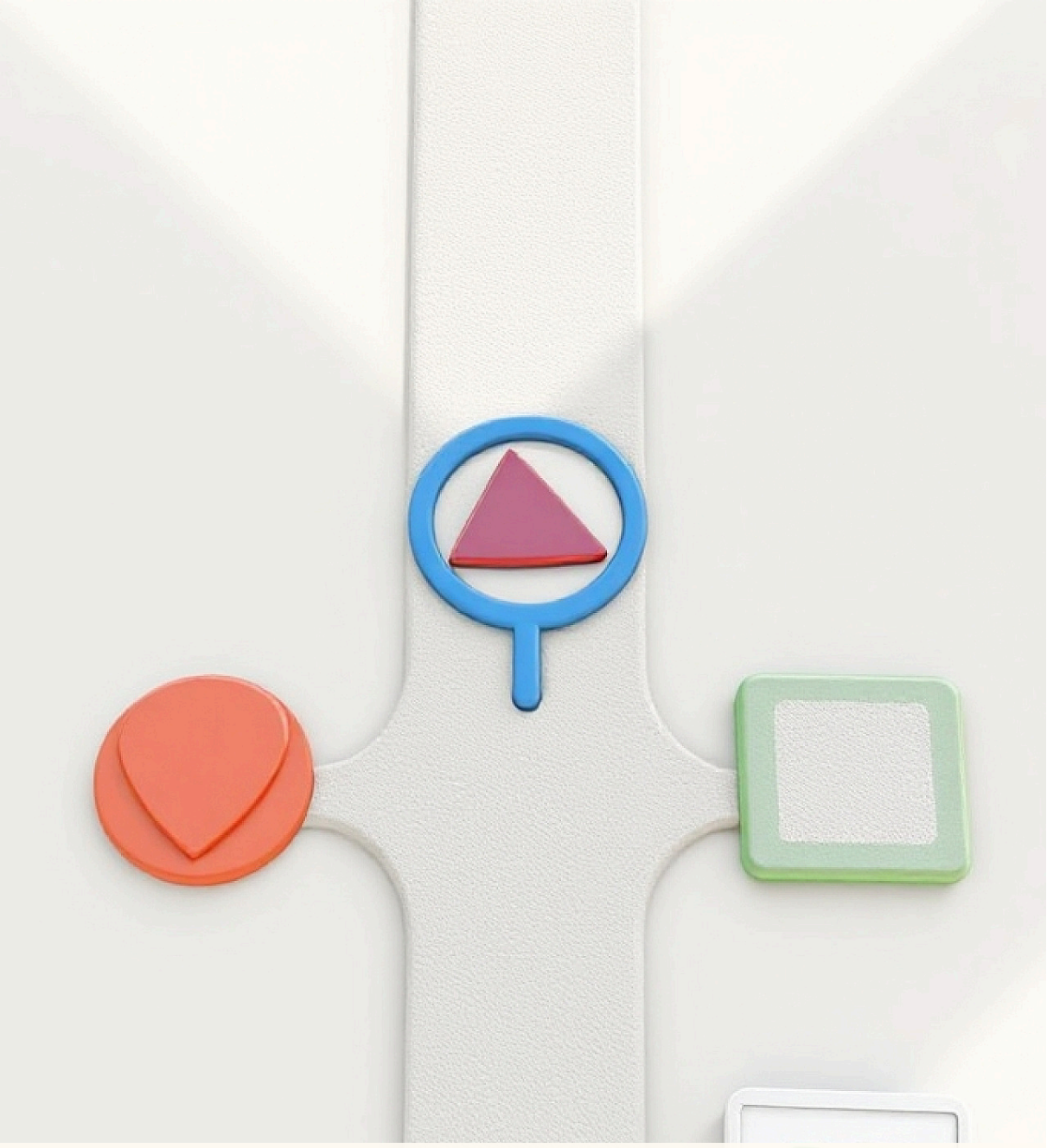
Tahmin gizli sayıdan küçük olduğunda program daha yüksek gibi bir ipucu yazdırır. Bu, oyuncuya gizli sayının daha büyük olduğunu, yani bir sonraki tahminin daha büyük olması gerektiğini söyler. İpucu yalnızca yazdırılan bir mesajdır; gizli sayıyı değiştirmez. Her yanlış tahminden sonra net bir ipucu vermek, oyunu oynanabilir ve adil kılan şeydir.



Tahmin gizli sayıdan büyük olduğunda program daha düşük gibi bir ipucu yazdırır. Bu, oyuncuya gizli sayının daha küçük olduğunu, yani bir sonraki tahminin daha küçük olması gerektiğini söyler. Çok düşük ipucuyla birlikte bu, olası sayıların aralığını adım adım daraltır. Oyuncu bu ipuçlarını, tıpkı bir arama gibi, yanıta yaklaşmak için kullanır.



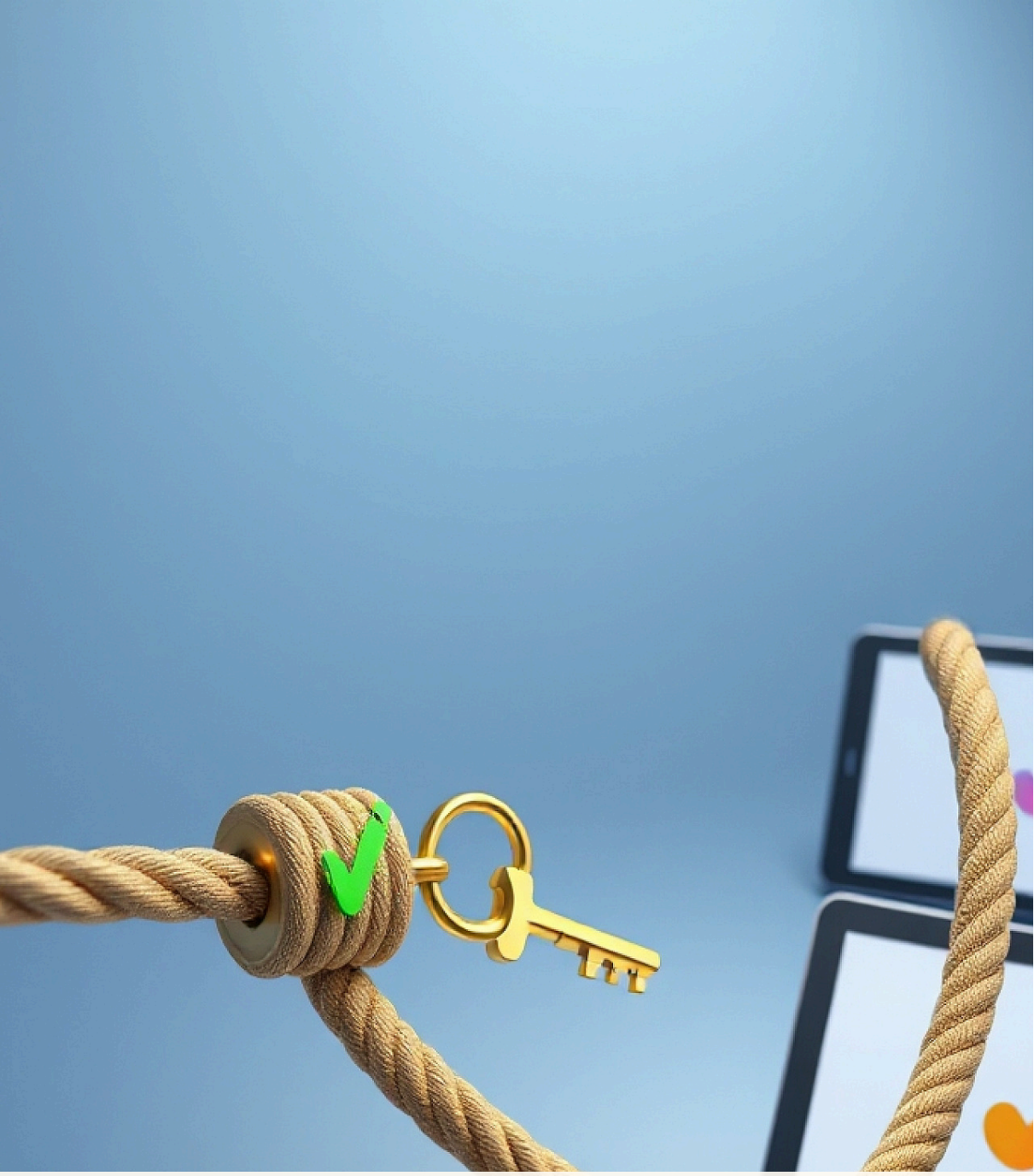
Tahmin gizli sayıya eşit olduğunda program oyuncuya kazandığını söyler. Bu, kazanma koşuludur, yani oyunu bitiren testtir. Kısa bir mesaj yazdır ve istersen gizli sayıyı göster ki oyuncu onu doğrulayabilsin. Bu noktadan sonra döngüyü bitirmek veya programı durdurmak dışında hiçbir şey olmamalıdır.



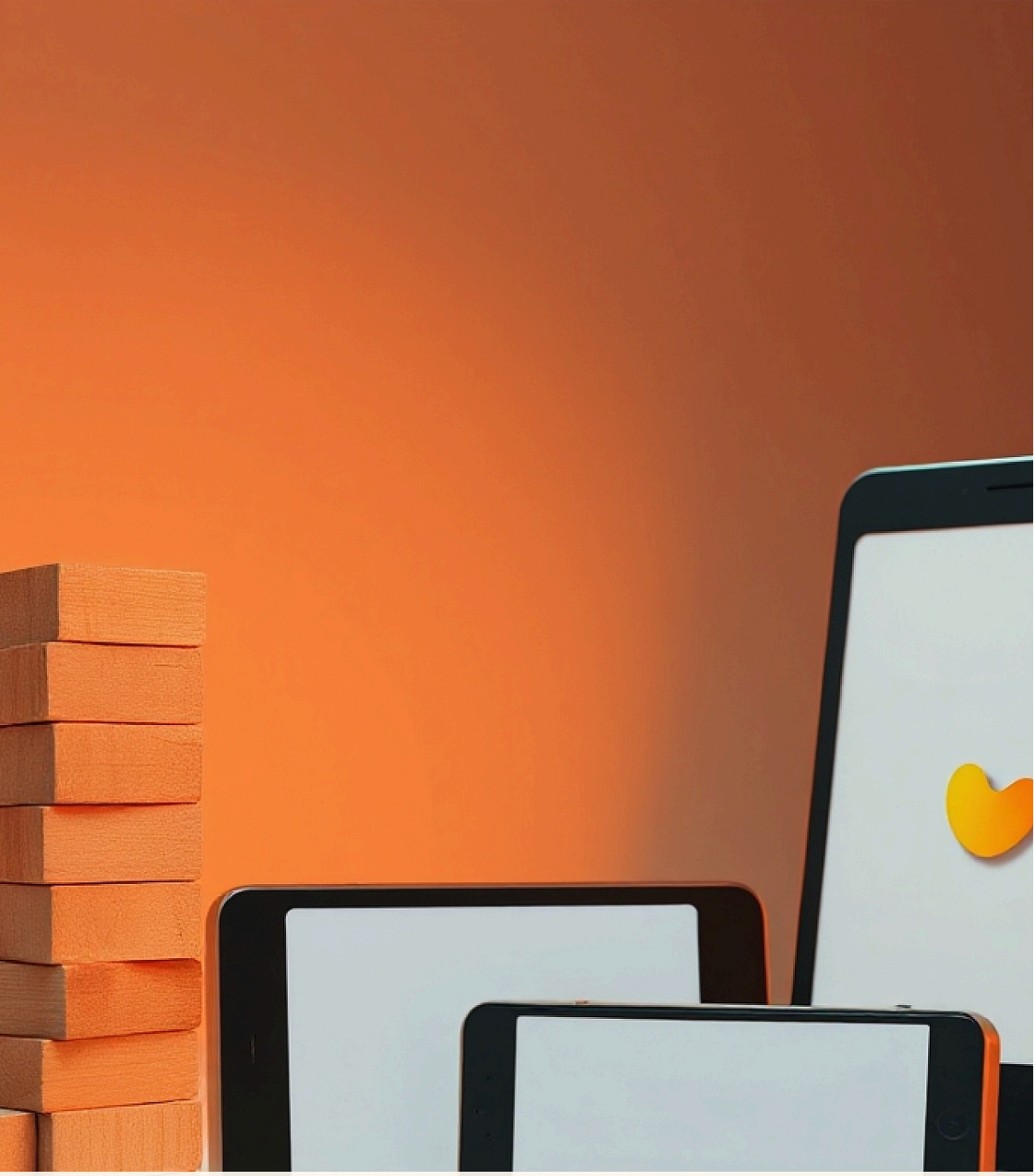
Koşul, doğru ya da yanlış olan bir testtir; örneğin tahmin hedeften küçüktür. Program hangi ipucunu göstereceğine ve oyuncunun kazanıp kazanmadığına karar vermek için koşulları kullanır. Koşullar karşılaştırma operatörleriyle yazılır: eşit, küçüktür ve büyüktür. Onları zincirleyerek her tahmin için tam olarak bir dalın çalışmasını sağlarsın: çok düşük, çok yüksek veya doğru.



Döngü, bir talimat bloğunu tekrar tekrar yürütür. Sor-karşılaştır-ipucu ver adımlarını bir döngünün içine sar ki oyuncu programı yeniden başlatmadan tahmin etmeye devam edebilsin. Döngüdeki her geçiş oyunun bir turudur. Tek bir soruyu, biri kazanana kadar süren gerçek bir oyuna dönüştüren şey döngüdür.



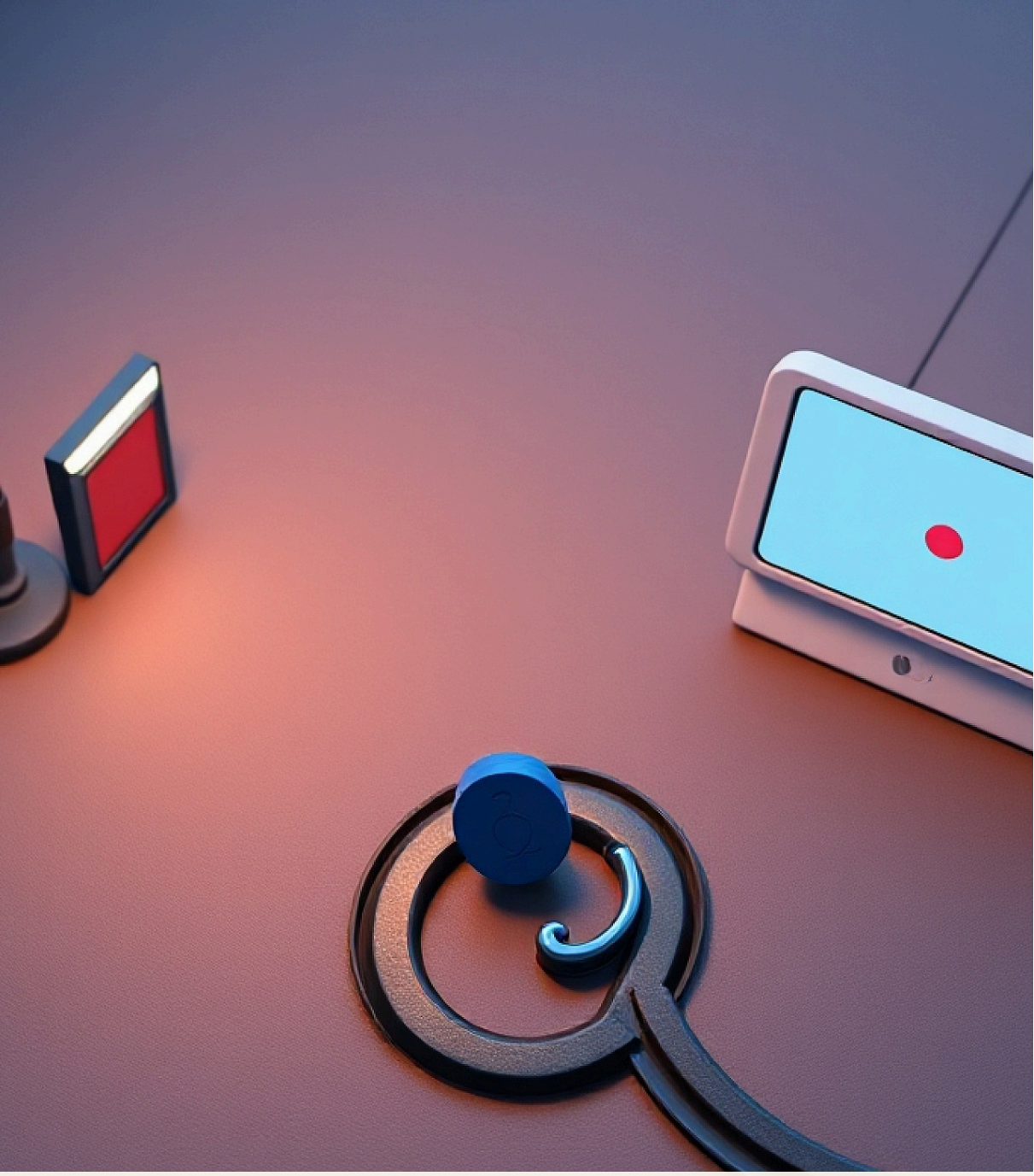
Döngü, tahmin gizli sayıyla eşleştiği anda durmalıdır. Kazanma koşulunu döngüye bağla: tahmin doğru olduğunda döngüden çık veya onu bitiren bir bayrak ayarla. Bu bağlantı olmadan program, oyuncu kazandıktan sonra bile sonsuza kadar sormaya devam ederdi. Doğru anda bitirmek, oyunu tamamlanmış hissettiren şeydir.



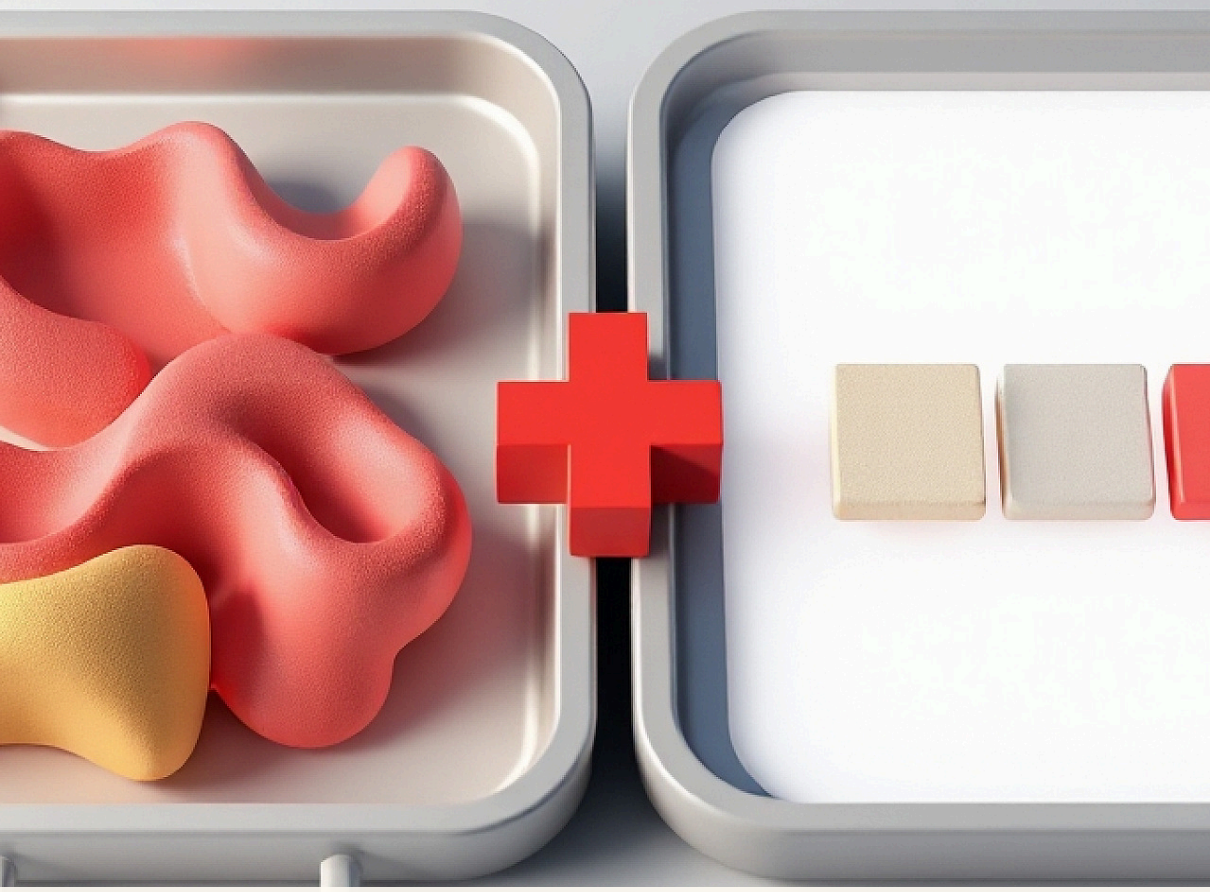
Sıfırdan başlayan ve her tahminden sonra bir artan bir değişken ekle. Bu, oyuncunun ihtiyaç duyduğu denemeleri sayar. Sonunda sayıyı kazanma mesajıyla birlikte yazdır, örneğin: 7 denemede buldun. Sayaç, döngü içinde değişen ikinci bir değişkendir ve değişkenlerle döngülerin nasıl birlikte çalıştığını gösterir.



Şimdi parçaları küçük bir programda birleştir: gizli bir sayı seç, deneme sayacını sıfırdan başlat, sonra döngü kur. Döngünün içinde bir tahmin oku, onu bir sayıya dönüştür, karşılaştır, daha yüksek veya daha düşük yazdır ve doğru olduğunda dur. Her parça tek bir iş yapar ve birlikte eksiksiz oyunu oluştururlar. Önce küçük bir aralıkla test et, sonra aralığı genişlet.



Yaygın bir hata, yalnızca bir tahmin isteyip sonra durmaktır. Döngü olmadan oyuncu yalnızca bir deneme yapar, bu yüzden oyun bozuk ve adil değilmiş gibi hissettirir. Çözüm basittir: girdiyi, karşılaştırmayı ve ipucunu, tahmin doğru olana kadar tekrarlanan bir döngünün içine koy. Döngünün gerçekten bu üç eylemin hepsini sardığını, yalnızca girdiyi değil, kontrol et.



Bir başka yaygın hata, yazılan metni doğrudan bir sayıyla karşılaştırmaktır. Girdi metin olarak gelir, bu yüzden karşılaştırmadan önce onu bir sayıya dönüştür, yoksa sonuçlar yanlış olur. Python'da girdiyi int() içine al, örneğin guess = int(input()). Oyuncu harf yazarsa dönüştürme başarısız olur, bu yüzden yeni başlayan bir sürüm sayı yazıldığını varsayabilir.



Oyun, değişkenlerin, kullanıcı girdisinin, karşılaştırmanın, koşulların ve döngülerin küçük bir projede nasıl birlikte çalıştığını gösterir. Kalıbı hatırla: gizli bir sayı seç, her tahmini karşılaştır, daha yüksek veya daha düşük ipuçları ver, doğru olana kadar döngü kur ve denemeleri say. Bu çalıştığında tahminlere bir sınır koyabilir veya oyunu yeniden oynatabilirsin. Her yeni özellik, zaten bildiğin aynı yapı taşlarını kullanır.