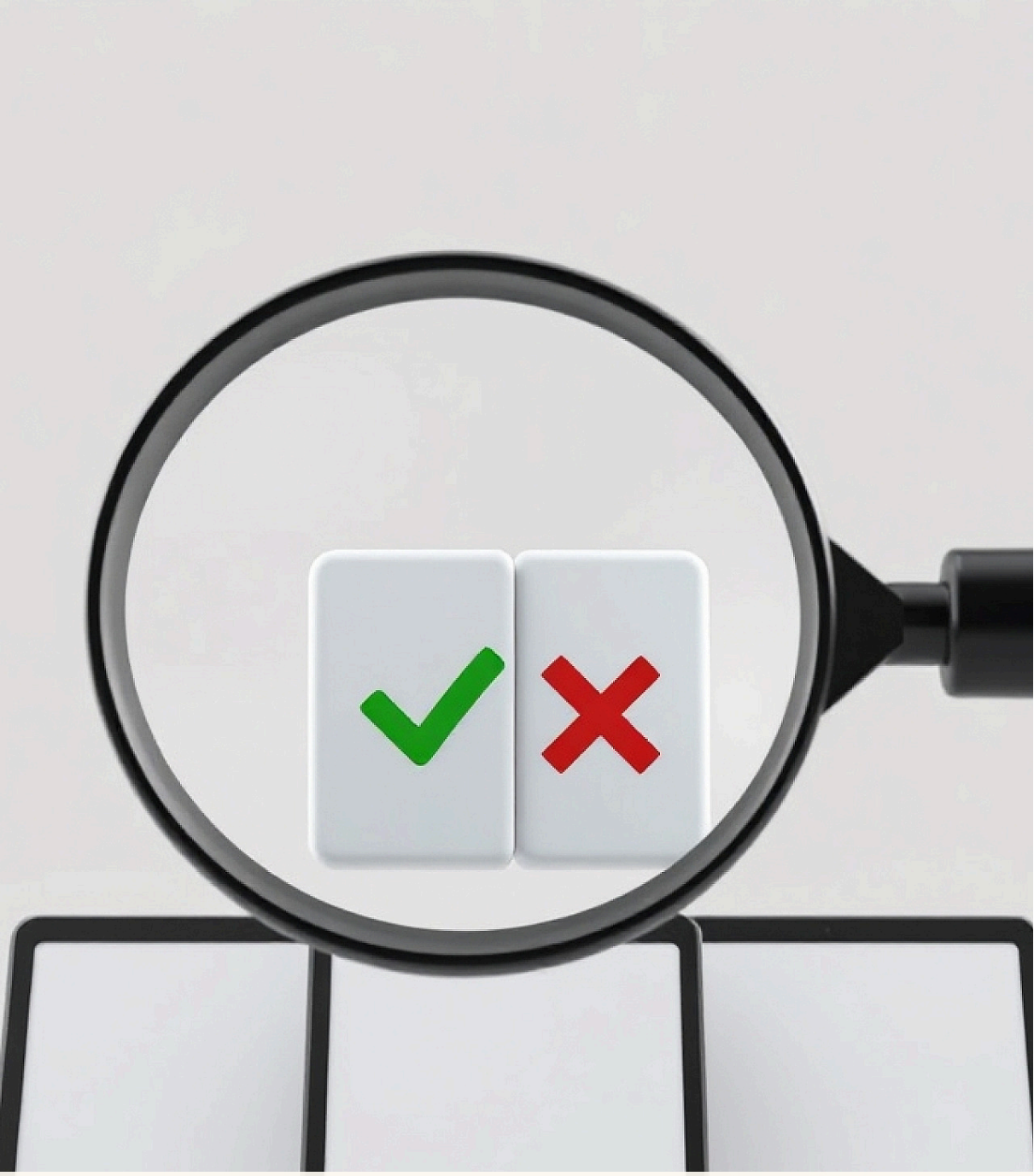
Koşul nedir?

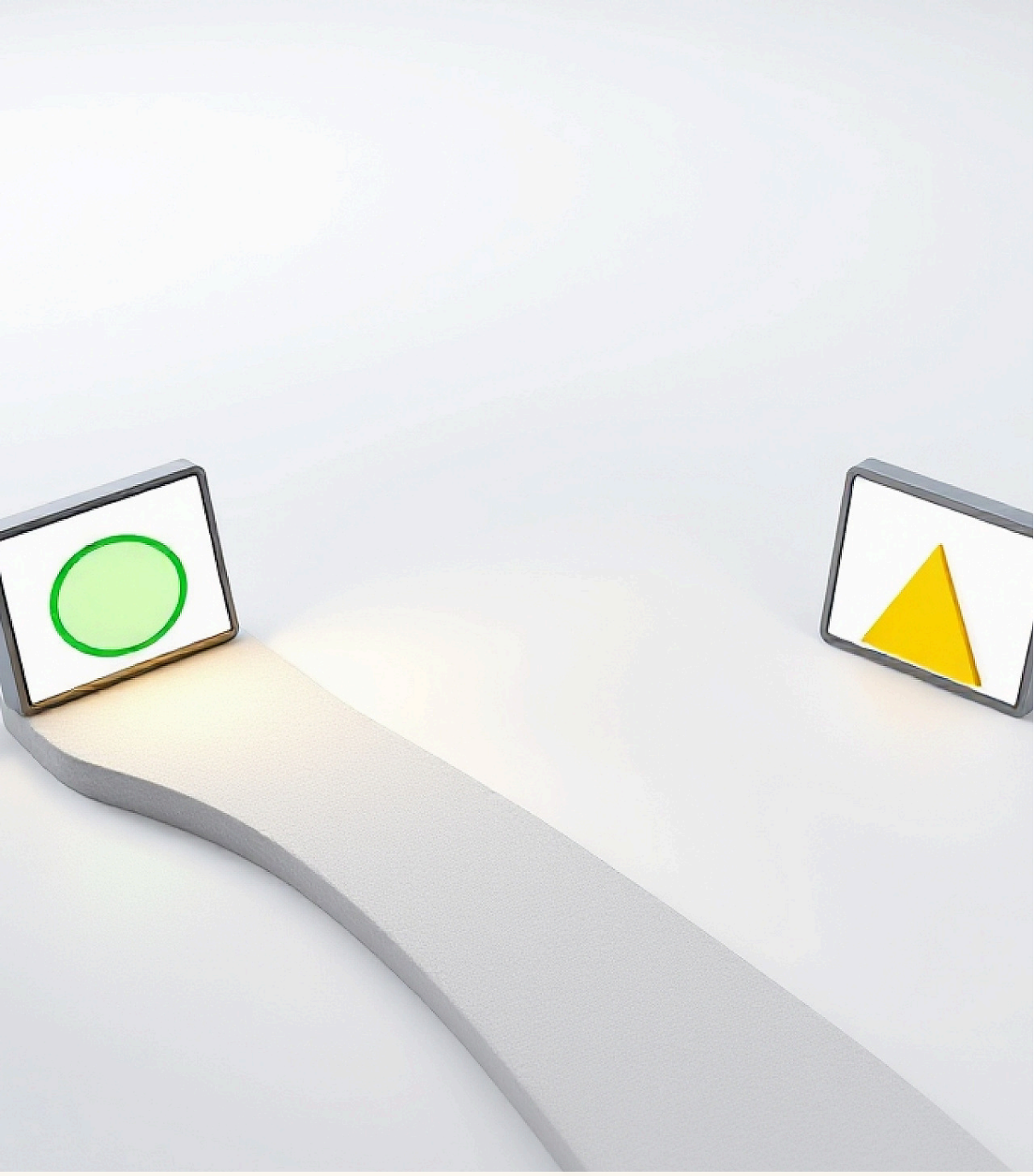


www.yzokul.com.tr

Koşul nedir?



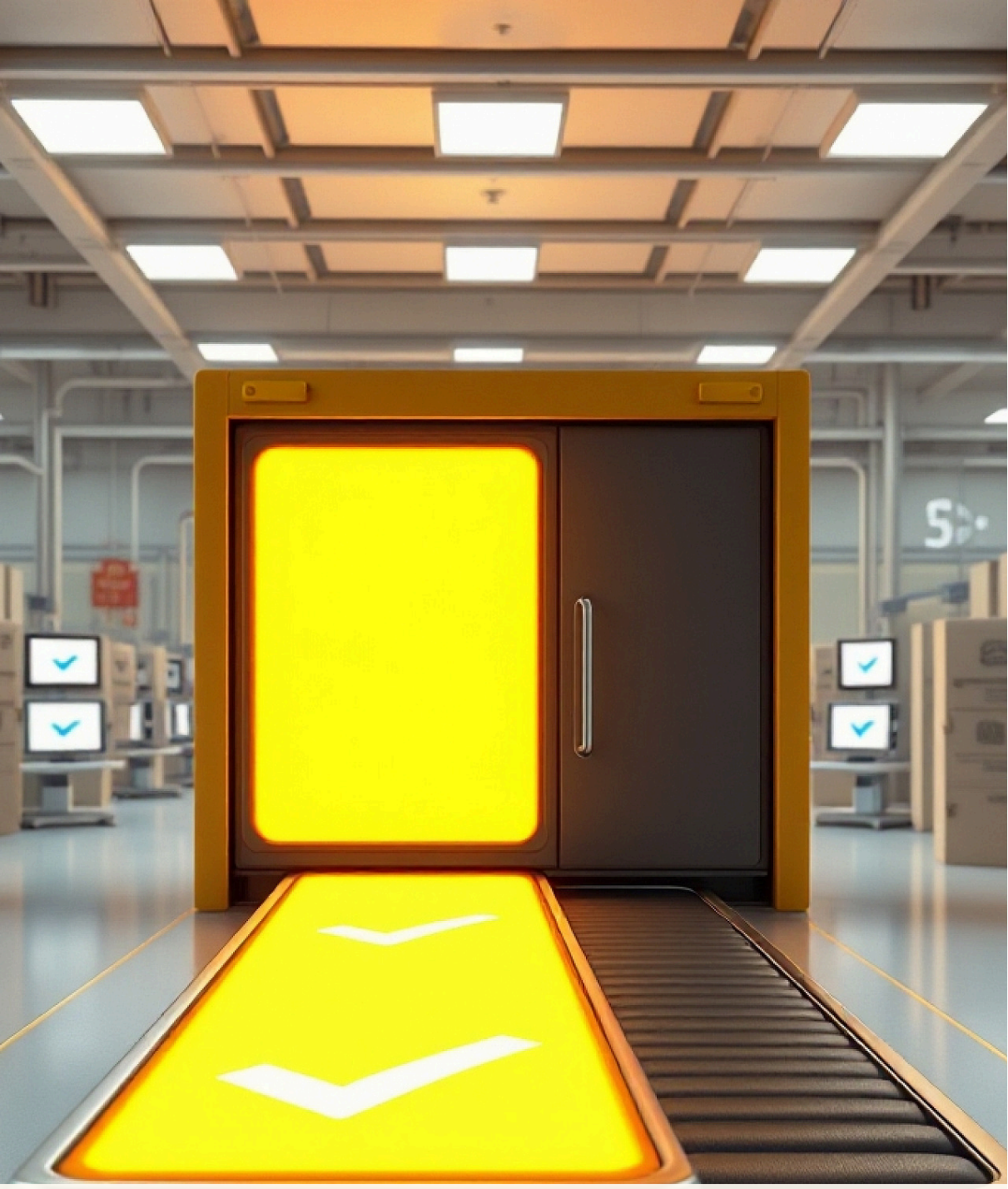
Koşul, bir programın bundan sonra ne yapacağına karar vermek için denetlediği şeydir. Bunu, programın mevcut durum hakkında sorduğu bir soru olarak düşünebilirsin. Bu sorunun yanıtı her zaman ya doğru ya da yanlıştır. Programa yanıta göre bir eylemi ya da diğerini seçer. Koşullar, programların karar vermesini sağlayan temel araçtır.



Programlar her çalıştıklarında her zaman aynı şeyi yapmaz. Bazen bir programın bir duruma bakması ve olası eylemlerden birini seçmesi gerekir. Örneğin bir oyunun, oyuncunun kazanıp kazanmadığını ya da kaybedip kaybetmediğini denetlemesi gerekebilir. Koşul, programın ne yapacağını seçmeden önce bu durumu inceleme biçimidir.



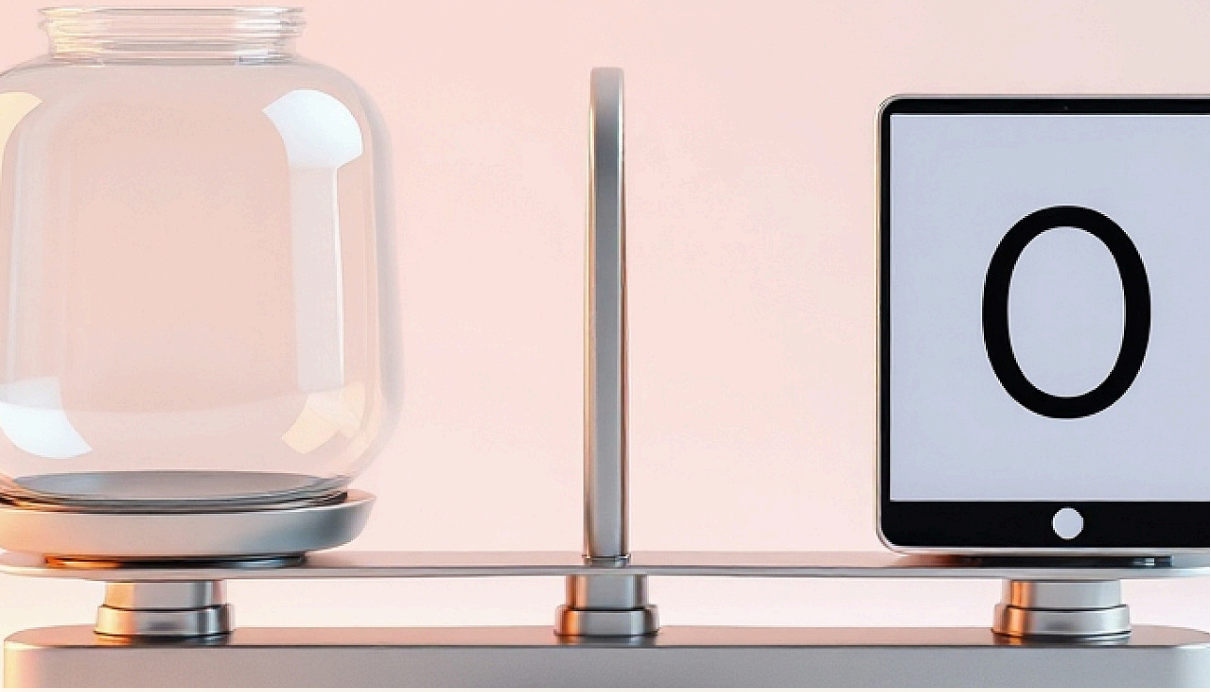
Oyuncunun can puanı olduğu bir oyun düşün. Can sıfıra düştüğünde oyun bitmelidir. Can hâlâ sıfırın üzerindeyse oyun devam etmelidir. Programın can değerini denetlemesi ve doğru tepki vermesi için bir yola ihtiyacı vardır. Bu denetim bir koşuldur ve oyunun durup durmayacağına karar verir.



Bir eylemi seçmeden önce program durumu denetler. Bu denetim koşuldur ve sonucu bir sonraki adıma karar verir. Program tahmin etmez ya da rastgele davranmaz; önce koşulu değerlendirir. Ancak sonuç bilindikten sonra uygun eylemi seçer. Denetim her zaman seçimden önce gelir.



Koşul, tam olarak iki değerden birine çıkan bir ifadedir: doğru ya da yanlış. Arada bir sonuç ve belki diye bir şey yoktur. Program doğru ve yanlışı olası tek yanıtlar olarak ele alır. Bu basitlik, koşulları karar vermek için güvenilir kılan şeydir. Yazdığın her koşul bu iki değerden birini üretecektir.



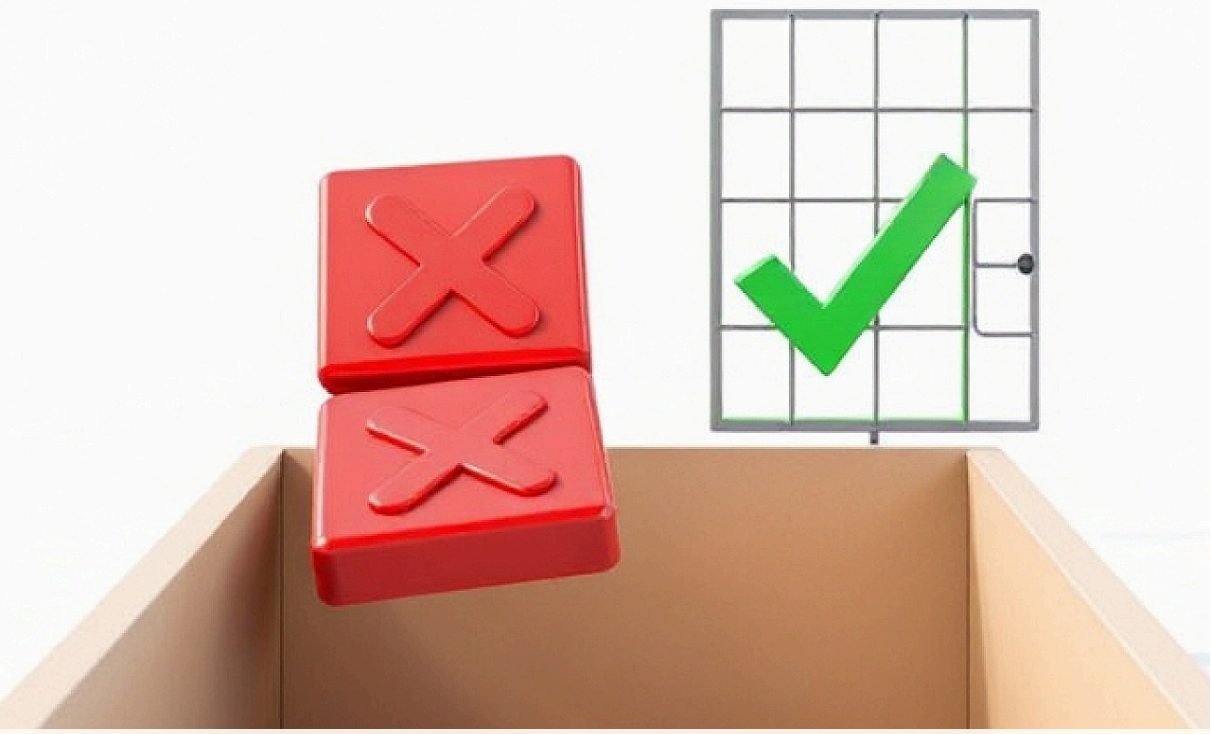
Can sıfıra eşittir ifadesi bir koşuldur. Can sıfırsa koşul doğrudur. Can sıfırın üzerindeyse koşul yanlıştır. Program bu sonucu elde etmek için mevcut can değerini sıfırla karşılaştırır. Bu karşılaştırma, bir durumu denetlemenin basit ama güçlü bir yoludur.



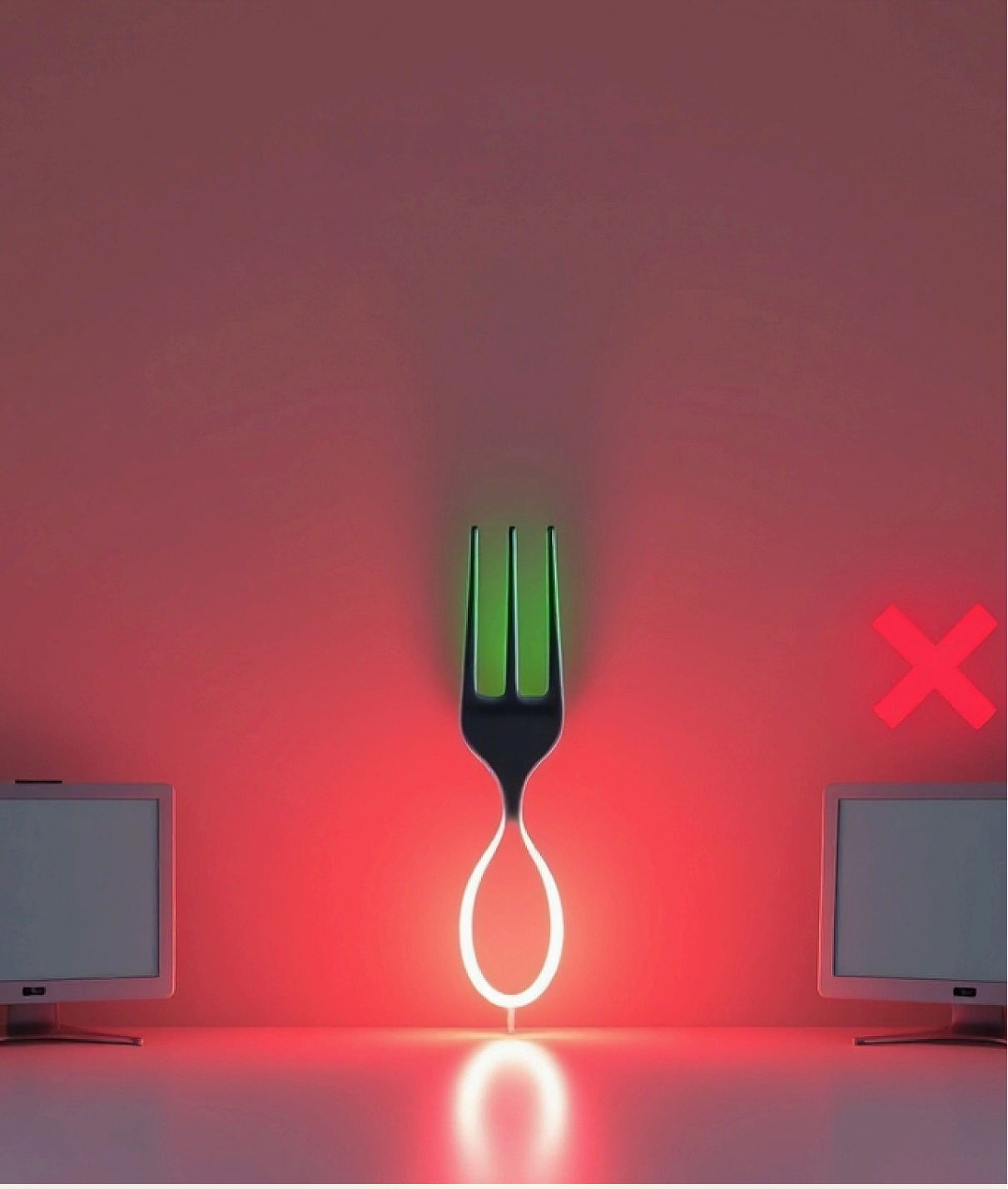
Programlar bir koşula göre eylemde bulunmak için eğer anlamına gelen bir yapı kullanır. Bu yapı şunu söyler: koşul doğruysa şunu yap; değilse başka bir şey yap. if yapısı, bir koşulu karara dönüştürmenin ana aracıdır. Koşulu bir kez yazarsın ve program bir yol seçmek için sonucunu kullanır. Bu, kararı açık ve düzenli tutar.



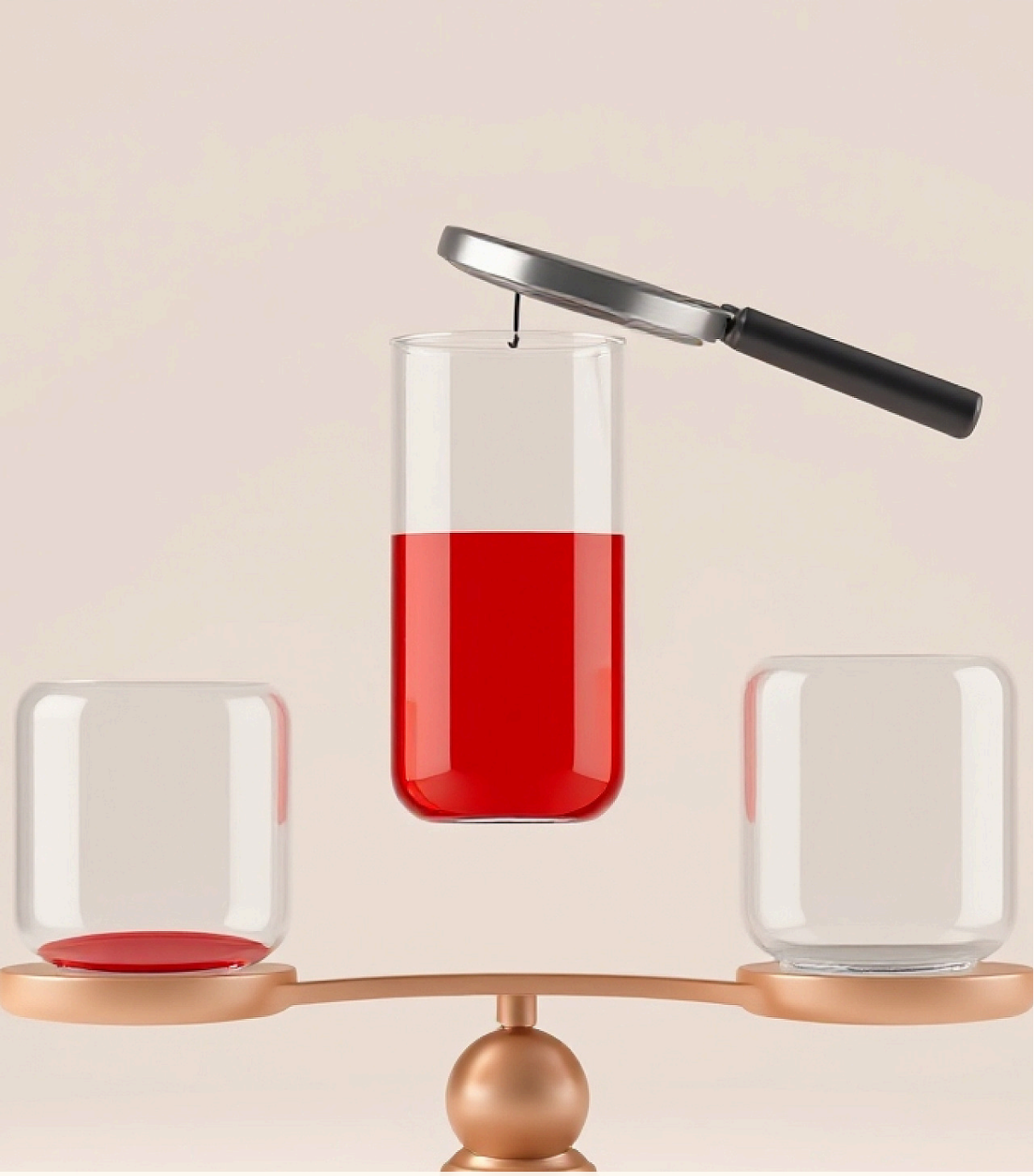
Koşul doğru olduğunda program eşleşen eylemi çalıştırır. Oyun örneğinde doğru, canın sıfıra ulaştığı anlamına gelir, bu yüzden oyun biter. Program doğru durumu için if yapısının söylediğini tam olarak yapar. Yanlış eylemi aynı anda çalıştırmaz. Yalnızca bir yol izlenir.



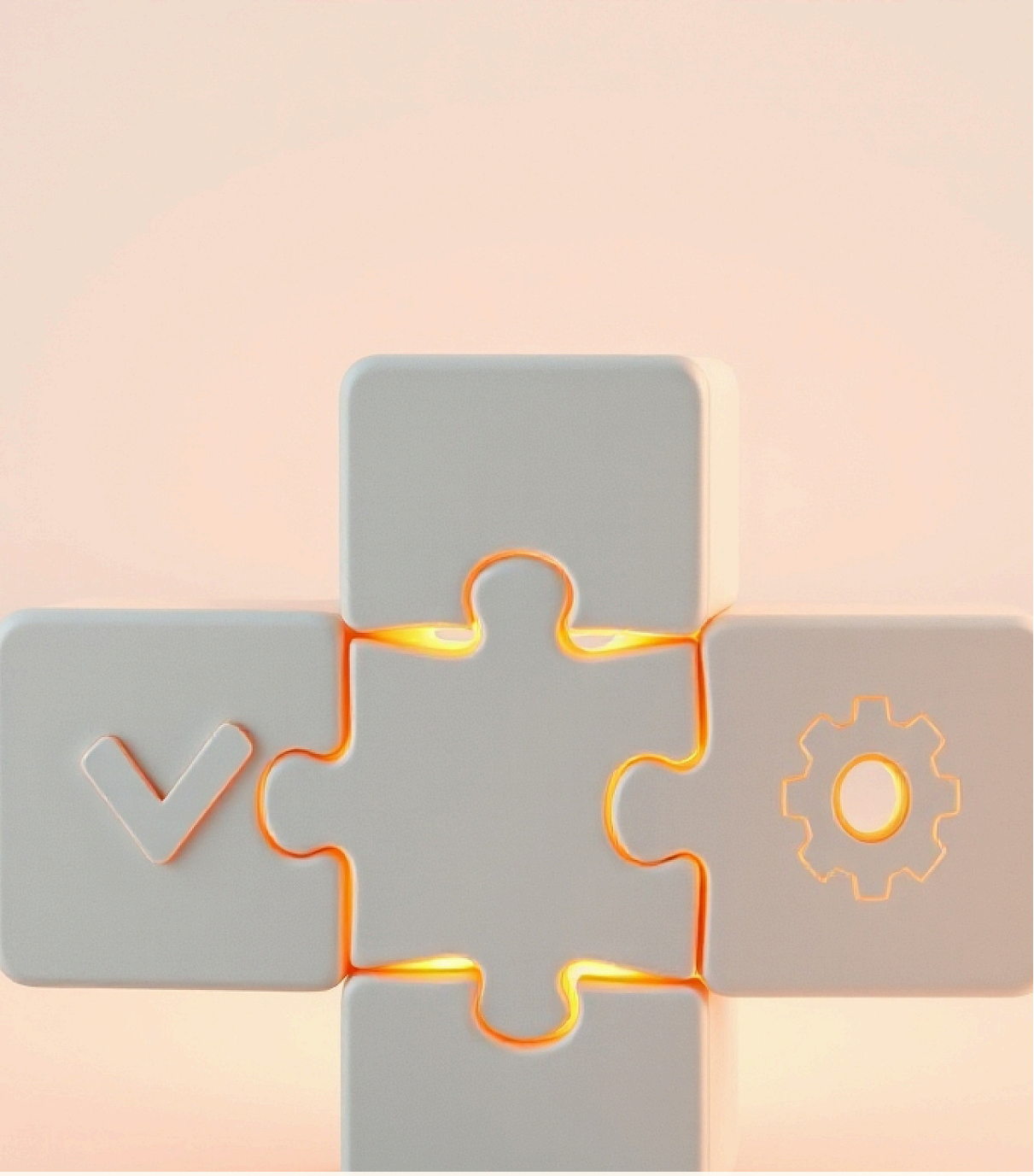
Koşul yanlış olduğunda program doğru eylemini atlar ve başka bir şey yapar. Oyun örneğinde yanlış, canın hâlâ sıfırın üzerinde olduğu anlamına gelir, bu yüzden oyun normal şekilde devam eder. Program bitirme eylemini yok sayar ve diğer yolu izler. Tek bir denetimin farklı sonuçlara nasıl yol açabileceği budur. Yanlış yolu da en az doğru yolu kadar önemlidir.



Bir koşul programı iki olası yola ayırır. Program yalnızca doğru ya da yanlış sonucuyla eşleşen yolu izler. İki yolu aynı anda asla izlemez. Buna karar denir çünkü program koşula göre seçim yapar. İki yol, tek denetim: koşulları kullanmanın temel fikri budur.



Birçok koşul, canın sıfırla karşılaştırılması gibi iki değerin karşılaştırılmasından doğar. Karşılaştırma, if yapısının kullandığı doğru ya da yanlış sonucunu üretir. Programın ihtiyacına göre sayıları, metinleri ya da başka değerleri karşılaştırabilirsin. Karşılaştırmanın kendisi koşuldur. Sonucun kolay tahmin edilebilmesi için karşılaştırmayı basit ve açık tut.



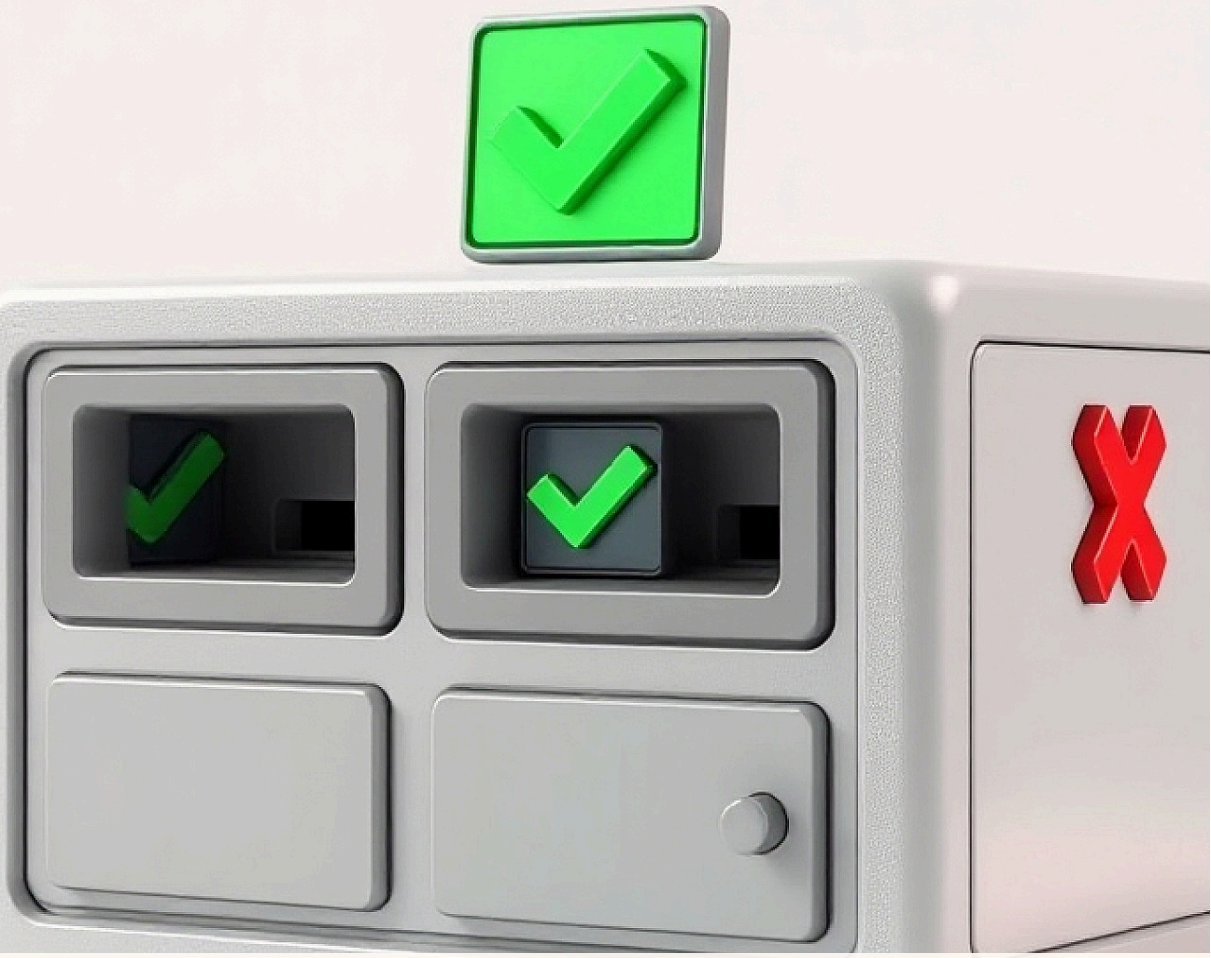
Daha ayrıntılı bir karar vermek için birkaç koşul birleştirilebilir. Örneğin oyunu yalnızca can sıfırsa ve hiç hak kalmadıysa bitir. Koşulları birleştirmek, aynı anda birden fazla şeyi denetlemeni sağlar. Birleşik sonuç yine tek bir doğru ya da yanlış değeridir. Bu, programın ne yaptığı üzerinde daha ince denetim sağlar.



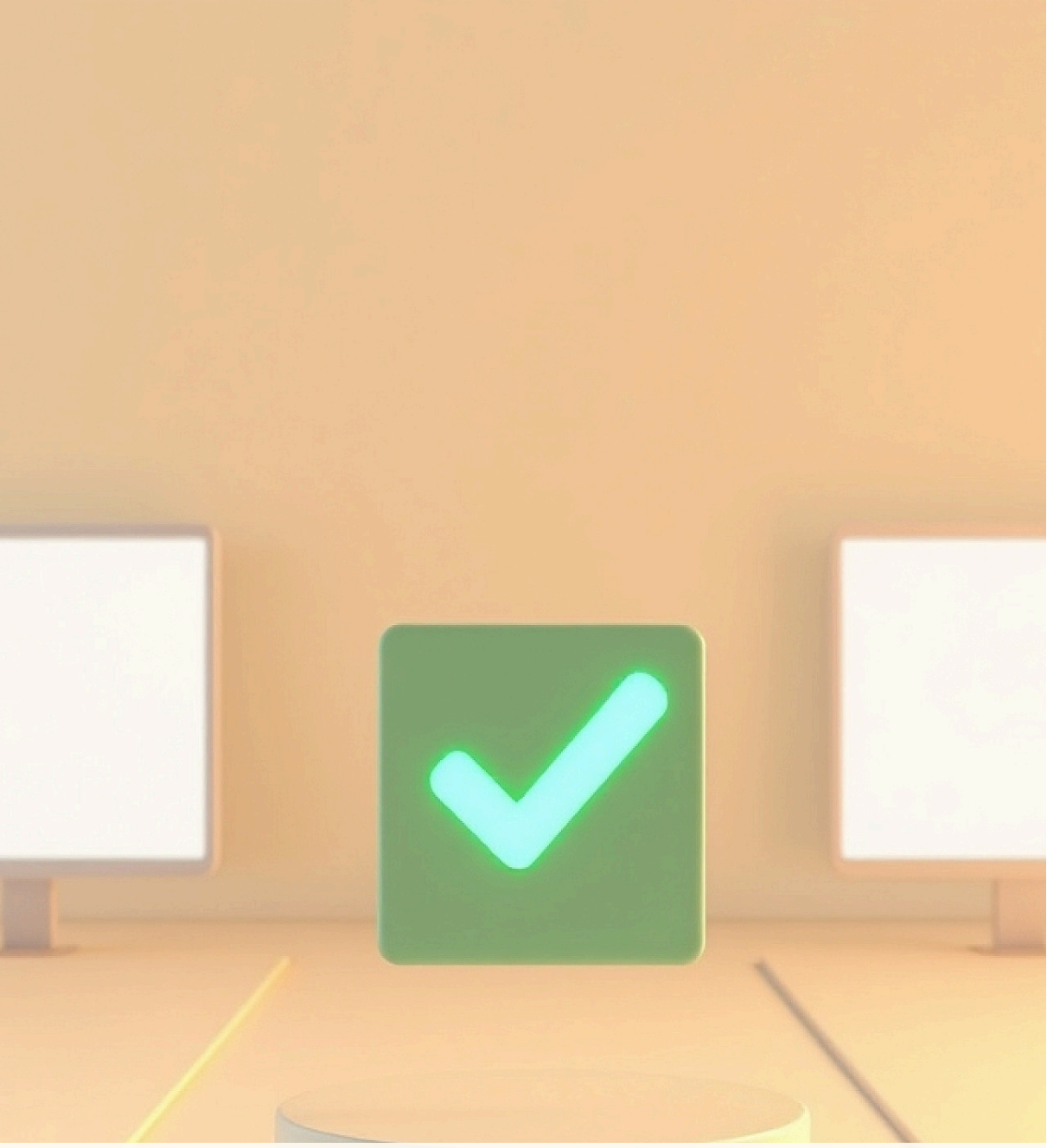
Birleştirilmiş koşullar yine tek bir doğru ya da yanlış sonucu üretir. if yapısı da önceki gibi bu tek sonuca göre eylemde bulunur. Birleştirilmiş koşullar için özel bir yapıya ihtiyacın yoktur; if aynı şekilde çalışır. Program tüm birleşimi değerlendirir ve tek bir yanıt elde eder. Daha fazla ayrıntı temel fikri değiştirmez.



Yeni başlayanların sık yaptığı bir hata, koşulu önce denetlemeden eylemde bulunmaktır. Program bir eylemi seçmeden önce koşulu değerlendirmelidir. Denetimi atlarsan program hangi yolu izleyeceğini bilemez. Koşulu her zaman yaz ve if yapısının sonucunu kullanmasına izin ver. Önce denetlemek, programın davranışını doğru ve tahmin edilebilir tutar.



Bir başka hata, bir koşulun doğru ya da yanlış dışında bir şey vereceğini beklemektir. Bir koşulun her zaman bu iki değerden birine değerlendirildiğini unutma. Sonuç olarak bir sayı ya da bir sözcük döndürmez. Daha fazla ayrıntıya ihtiyacın varsa farklı türde bir yanıt beklemek yerine koşulları birleştir. Sonuç her zaman doğru ya da yanlıştır.



Koşul, bir programın bir durumu denetlemesini ve doğru ya da yanlış sonuca göre ne yapacağını seçmesini sağlar. if yapıları karar vermek için koşulları kullanır ve daha ayrıntılı seçimler için birkaç koşul birleştirilebilir. Denetimin her zaman önce geldiğini ve sonucun her zaman doğru ya da yanlış olduğunu unutma. Bu fikirlerle programlardaki basit kararları okuyup yazabilirsin.