Kodla basit bir hesap makinesi nasıl yapılır?

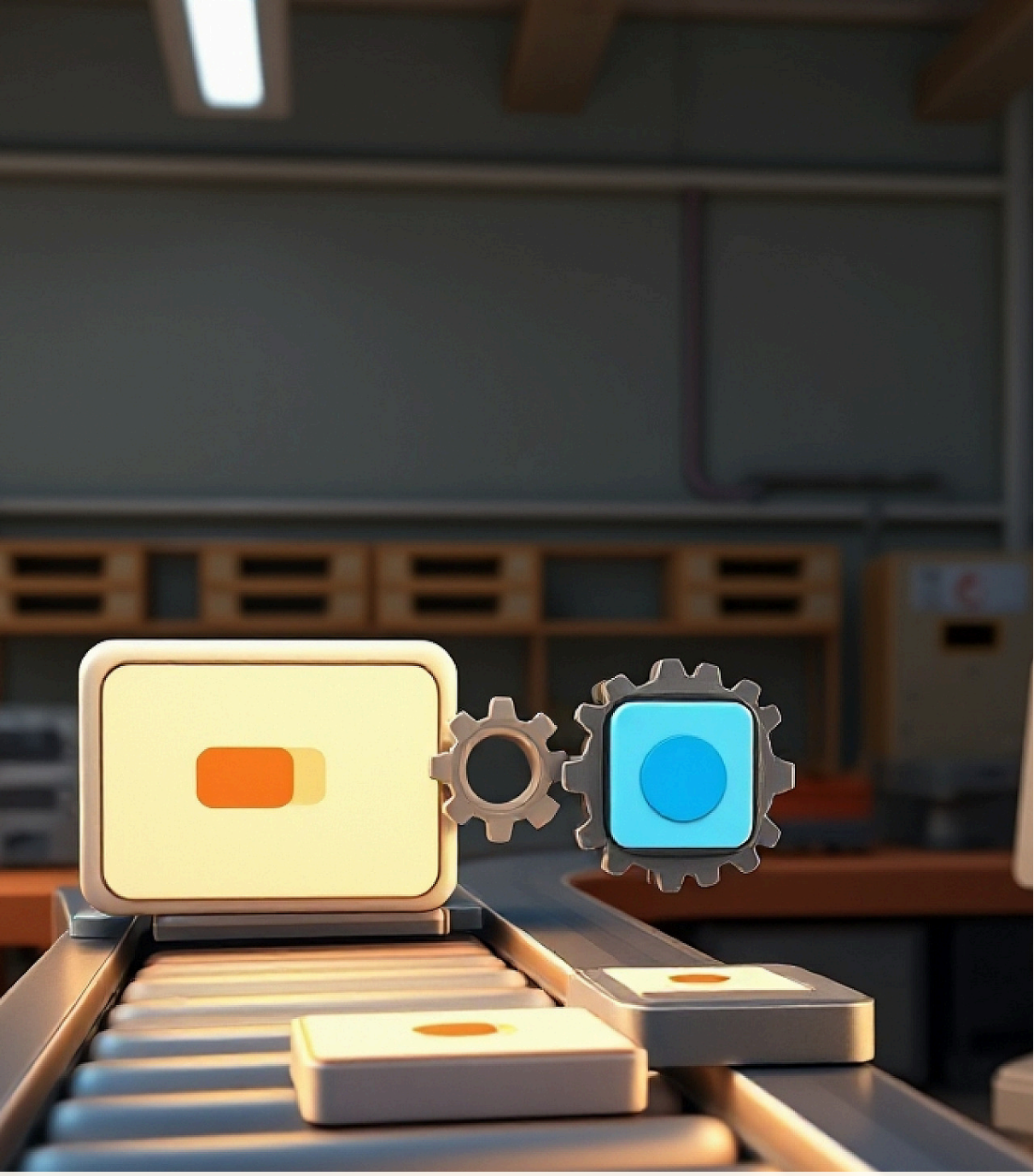


www.yzokul.com.tr

Kodla basit bir hesap makinesi nasıl yapılır?



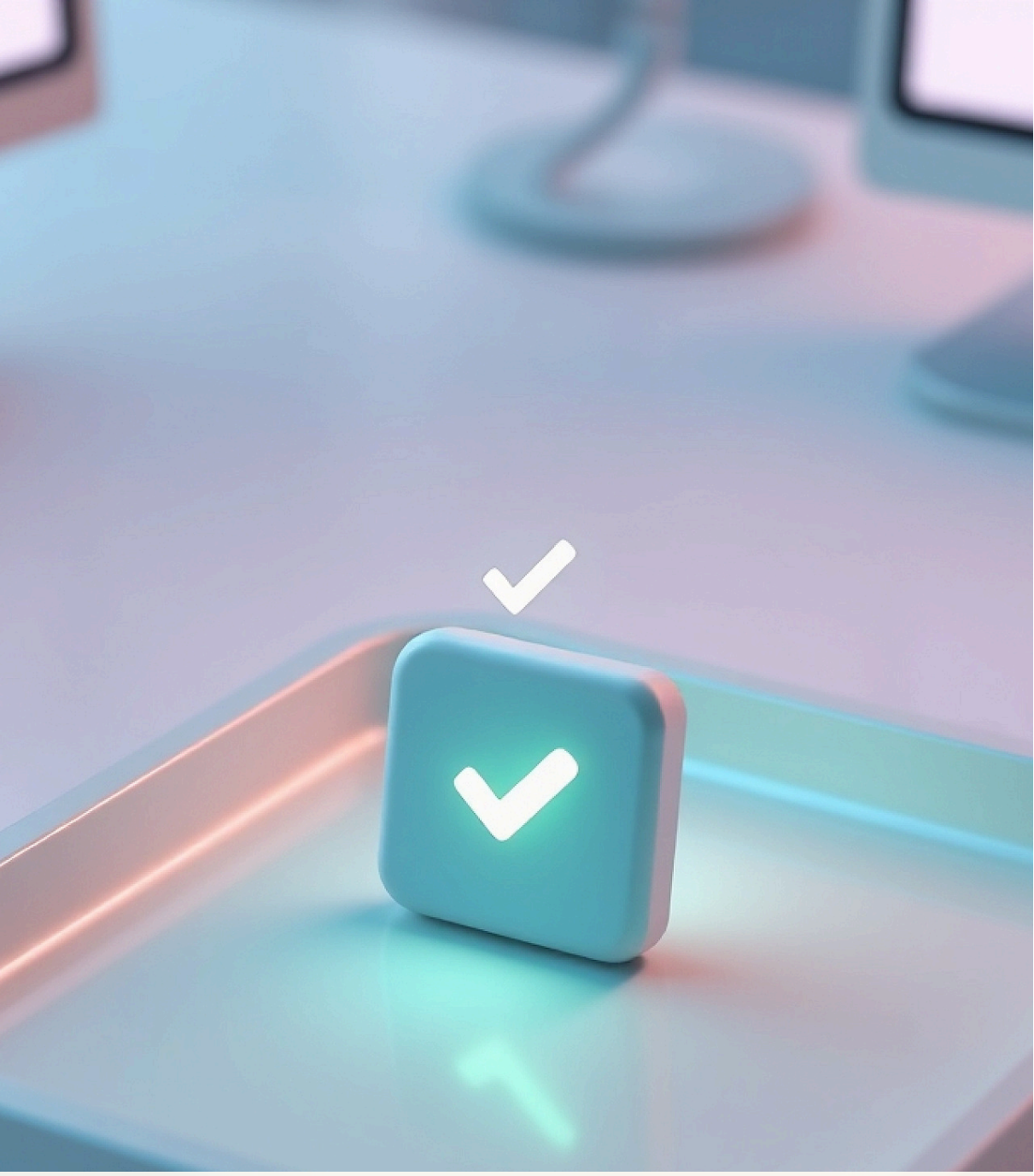
Basit bir hesap makinesi, iki sayı ve bir işlem alıp sonucu gösteren bir programdır. Kısa yanıt şudur: kod girdileri saklar, doğru matematik işlemini seçer ve yanıtı yazdırır. Örneğin, 5 ve 3 girip toplamayı seçersen program 8 gösterir. Bu kitap her parçayı açıklar, böylece daha önce kodlama deneyimin olmasa bile kendin bir tane yapabilirsin.



Her hesap makinesi aynı temel işi yapar: iki sayı ister, hangi işlemin yapılacağını sorar, matematiği yapar ve sonucu gösterir. Bu plan, sonraki tüm adımların temelidir. Örneğin, program önce birinci sayıyı, sonra ikinciyi, sonra toplama, çıkarma, çarpma veya bölme yapılıp yapılmayacağını sorabilir. Her parçayı oluştururken bu basit akışı aklında tut.



Başlamak için Python gibi yeni başlayanlara uygun bir dil seç. Aynı fikirler—değişkenler, koşullar, fonksiyonlar ve hata yönetimi—her dilde işe yarar, yani burada öğrendiklerin aktarılabilir. Bu kitaptaki tüm örnekler Python kullanır çünkü kodu kısa ve okunması kolaydır. Örneğin, Python'da bir mesaj yazdırmak yalnızca bir satır sürer: print("Hello").



Araçlarının çalıştığını doğrulamak için mümkün olan en küçük programı yaz ve çalıştır. Python'da tek satırla bir selamlama yazdırabilirsin. Bu, gerçek mantığı eklemeden önce korkuyu ortadan kaldırır. Örneğin, print("Calculator ready") yaz ve çalıştır; mesajın göründüğünü görmelisin. Bu çalıştığında, hesap makinesini yapmaya hazırsın.



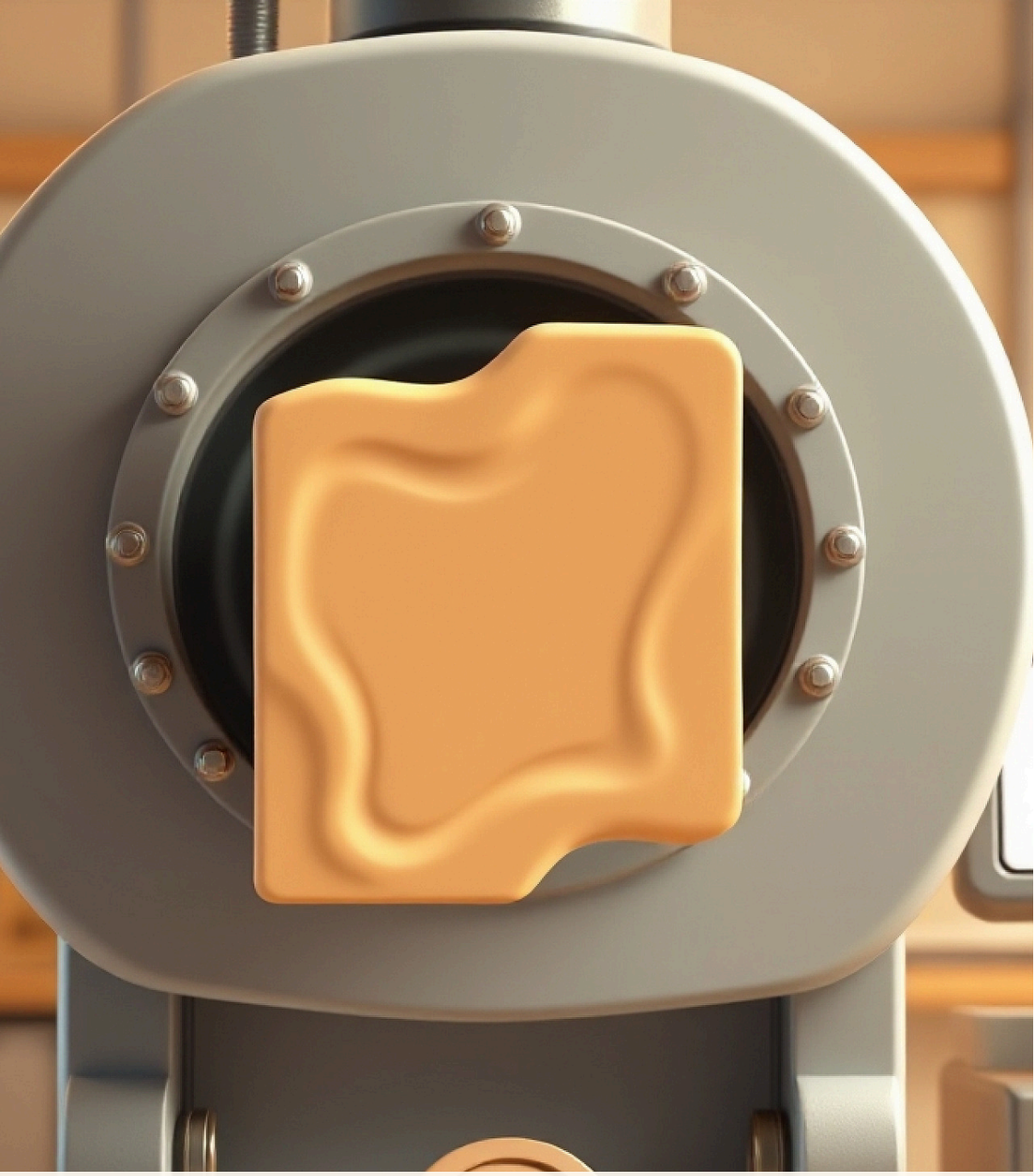
Değişken, bir sayı veya metin gibi bir değeri saklayan adlandırılmış bir kutudur. Kullanıcının yazdığı iki sayıyı tutmak için iki değişken kullanacaksın. Örneğin, first_number adında bir değişken oluşturabilir ve sonra içine bir değer koyabilirsin. Değişkenler, aynı değere yeniden yazmadan birden çok kez başvurmanı sağlar.



Kullanıcıdan birinci ve ikinci sayıyı istemek için bir input komutu kullan. Program duraklar ve kullanıcı bir şey yazıp Enter'a basana kadar bekler. Örneğin, Python'da first = input("First number: ") yazarsın. Parantez içindeki metin kullanıcıya gösterilen istemdir. Yazılan yanıt first değişkeninde saklanır.



Yazılan girdi sayı olarak değil, metin olarak gelir. Bu yüzden dikkatli olmazsan '2' artı '3', 5 yerine '23' verebilir. Örneğin, girdiyi bir değişkende saklar ve doğrudan başka bir girdiye eklemeye çalışırsan, program metinleri birleştirir. Matematik yapmadan önce metni gerçek bir sayıya dönüştürmelisin.



Metin girdisini bir dönüştürme fonksiyonu kullanarak gerçek bir sayıya dönüştür. Python'da int() fonksiyonu metni tam sayıya, float() fonksiyonu metni ondalık sayıya çevirir. Örneğin, number = int(input("Enter a number: ")) hesaplama yapabileceğin gerçek bir sayı saklar. Artık program değerlerle gerçekten matematik yapabilir.



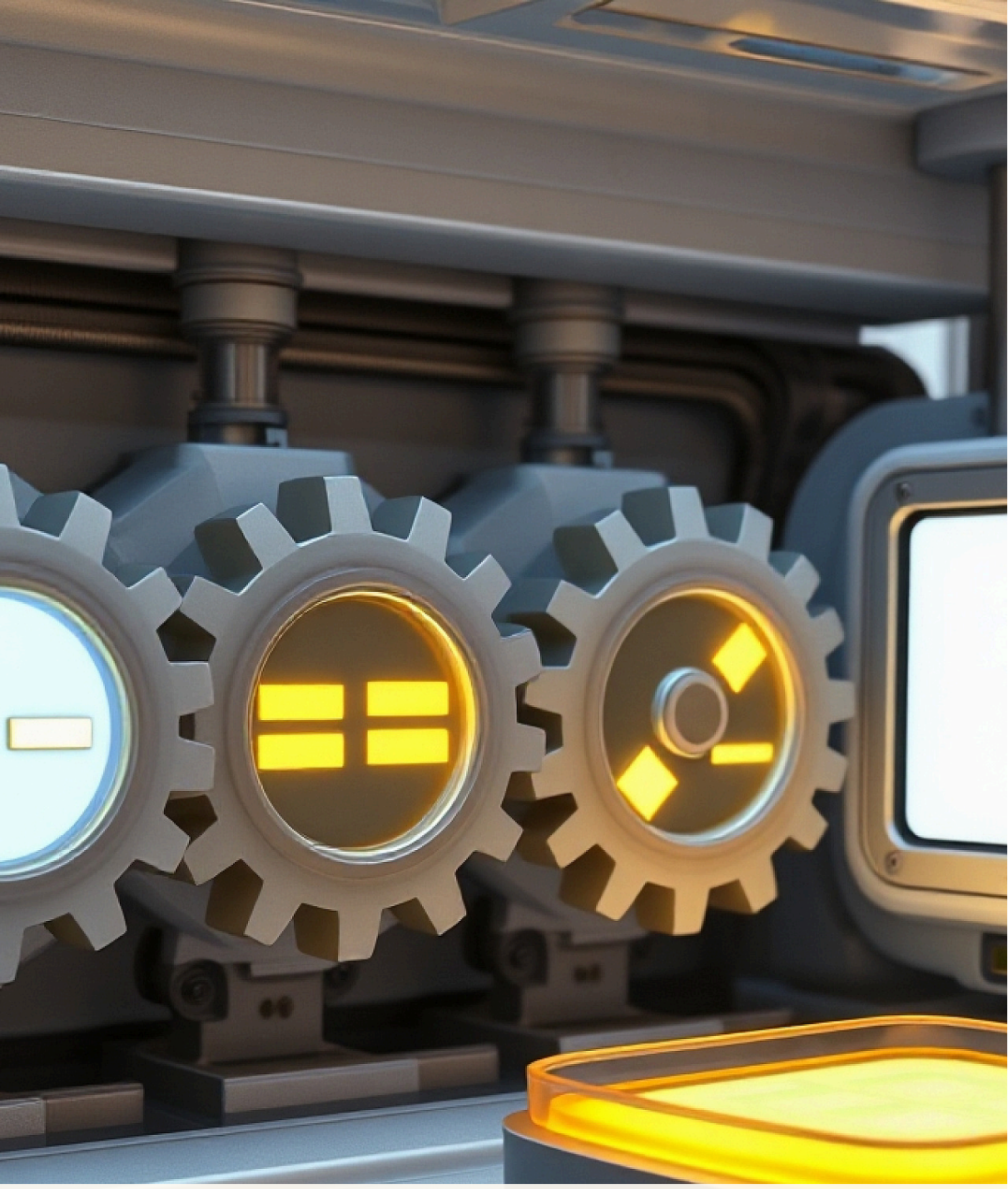
Kullanıcıya toplama, çıkarma, çarpma veya bölme gibi hangi işlemi yapacağını sor. Seçimi bir değişkende sakla, böylece program sonra ne yapacağına karar verebilir. Örneğin, op = input("Choose operation (+, -, *, /): "). Kullanıcı sembollerden birini yazar ve o sembol sonraki kısım için kaydedilir.



Bir if ifadesi bir koşulu kontrol eder ve kodu yalnızca koşul doğruysa çalıştırır. Program, kullanıcının seçiminden doğru işlemi bu şekilde seçer. Örneğin, if op == "+": toplama kodunu çalıştırır. Diğer işlemler için elif ile daha fazla koşul ve beklenmeyen her şey için son bir else ekleyebilirsin.



İki sayıyı toplayan veya çıkaran ve sonucu saklayan dalları yaz. Her dal tam olarak bir işlemi ele alır. Örneğin, if op == "+": result = a + b; elif op == "-": result = a - b. Burada a ve b dönüştürülmüş sayılardır ve result yanıtı tutar.



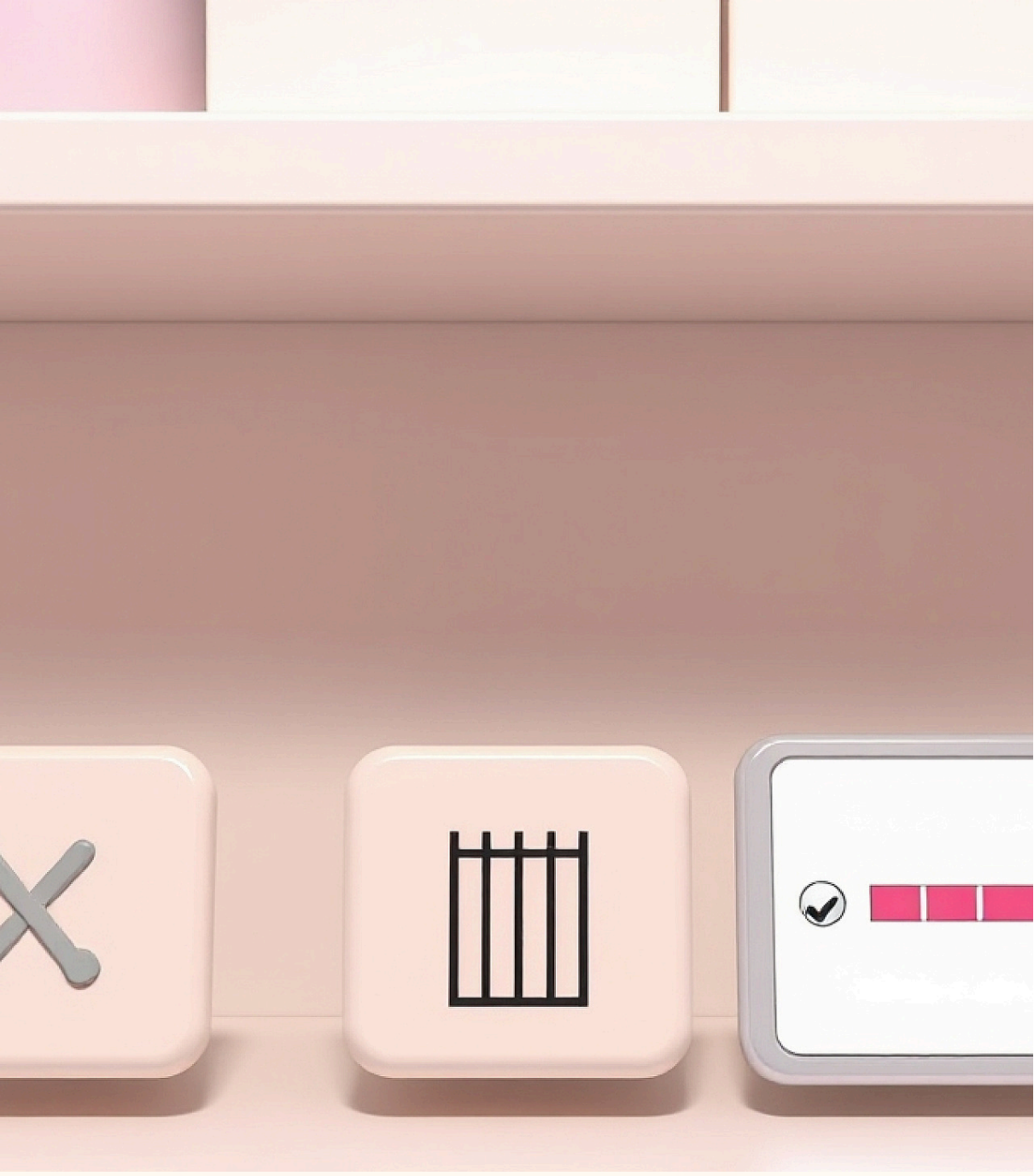
Çarpma ve bölme dallarını ekleyerek dört temel işlemi tamamla. Sonuç aynı result değişkeninde saklanır. Örneğin, elif op == "*": result = a * b; elif op == "/": result = a / b. Artık hesap makinesi dört yaygın matematik işleminin tümünü yapabilir.



Sıfıra bölme bir hatadır, bu yüzden program bölmeden önce böleni kontrol etmelidir. Bölen sıfırsa, çökmek yerine açık bir mesaj göster. Örneğin, bölmeden önce b == 0 olup olmadığını kontrol et: print("Cannot divide by zero"). Yalnızca b sıfır değilse böl. Buna hata yönetimi denir.



Saklanan sonucu 'Result: 8' gibi açık ve okunabilir bir şekilde yazdır. Çıktı alanı, kullanıcının yanıtı sonunda gördüğü yerdir. Örneğin, print("Result:", result) kullan. Mesajın anlaşılması kolay olsun ki kullanıcı sayının ne anlama geldiğini bilsin.



Yaygın yeni başlayan hatalarına dikkat et: girdiyi dönüştürmeyi unutmak, koşulları yanlış sıraya koymak ve sıfır kontrolünü atlamak. Her hatanın basit bir çözümü vardır. Örneğin, matematikten önce girdiyi her zaman int() veya float() ile dönüştür; bölmeden önce sıfıra bölmeyi kontrol et; ve doğru dalın çalışması için if ve elif dallarını sırala.



Tüm yolu hatırla: girdiyi al, dönüştür, koşullarla bir işlem seç, hataları kontrol et ve sonucu göster. Hatırlanması gereken temel fikirler değişkenler, koşullar, fonksiyonlar ve hata yönetimidir. Örneğin, eksiksiz bir hesap makinesi tüm bu adımları güvenilir şekilde çalışan kısa bir programda birleştirir. Bu temelle daha sonra daha fazla özellik ekleyebilirsin.