Kütüphane nedir?



www.yzokul.com.tr

Kütüphane nedir?

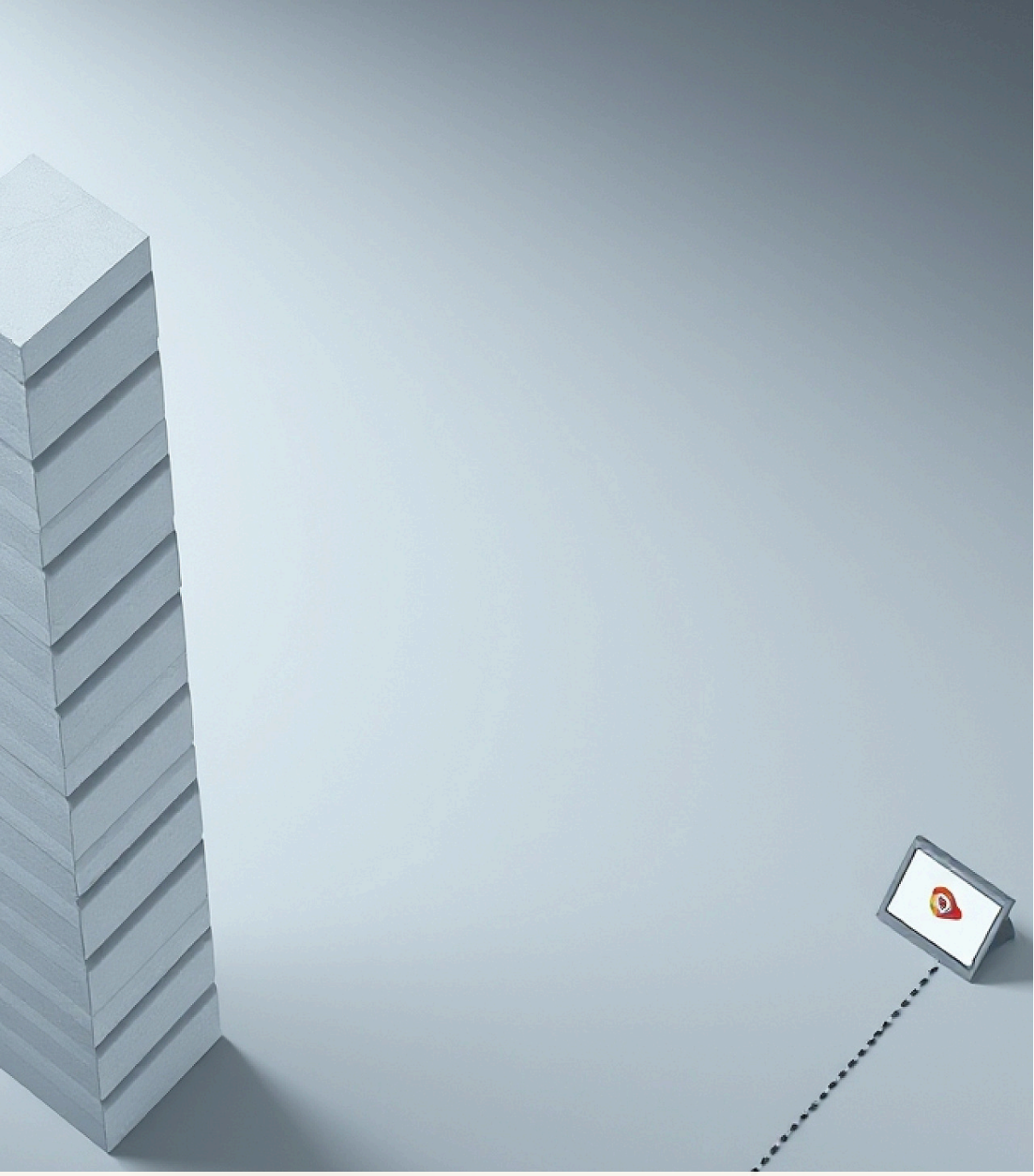
www.yzokul.com.tr



Kütüphane sözcüğünü duymuş ve programlamada ne anlama geldiğini merak etmiş olabilirsin. Kısa yanıt basittir: kütüphane, belirli görevleri yapmak için hazırlanmış yeniden kullanılabilir kod koleksiyonudur. Her talimatı kendin yazmak yerine, başka birinin zaten yazıp test ettiği kodu kullanabilirsin. Bu kitap bu yanıtı en baştan adım adım açıklar.



Program, bilgisayarın birbiri ardına izlediği talimatlar listesidir. Kod, bilgisayara ne yapacağını söyleyen yazılı metindir; örneğin iki sayıyı toplaması veya ekranda bir sözcük göstermesi. Bilgisayar, verilen sırayla kodun söylediğini tam olarak yapar. Kod yazmak, bu talimatları bilgisayarın çalıştırabileceği bir dilde yazmak demektir.



Basit bir program için bile her talimatı satır satır kendin yazdığını düşün. Her ayrıntıya sen karar verirdin: verinin nereden geldiğine, nasıl saklandığına ve ekranda ne görüneceğine. Bu işe yarar ve program doğru olabilir. Ama çok zaman ve emek ister; üstelik bu işin çoğu daha önce başkaları tarafından yapılmıştır.



Bazı kodlar zaten yazılmıştır ve yeniden kullanılabilir, böylece her seferinde sıfırdan başlamazsın. Kütüphane tam olarak budur: bir arada tutulan ve diğer programcıların kullanımına hazır hale getirilen yeniden kullanılabilir kod koleksiyonu. Onu programına eklersin ve gerektiğinde çağırırsın. Her kütüphanenin arkasındaki temel fikir budur.



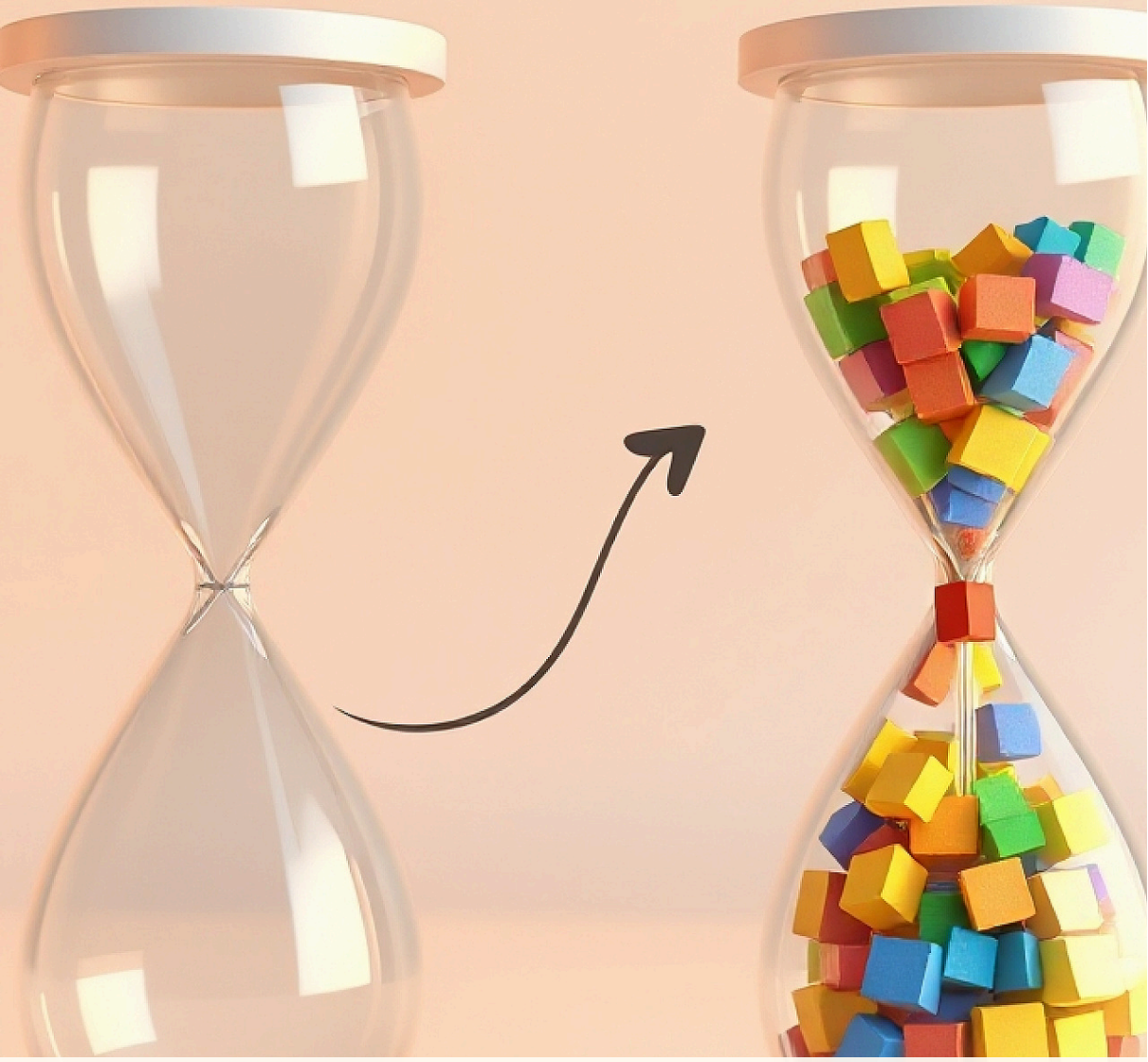
Bir kütüphane belirli görevleri yapmak için hazırlanır ve her kütüphane genellikle tek bir iş türüne odaklanır. Bir kütüphane görüntülerle, bir diğeri metinle, bir başkası sayılarla ilgilenebilir. Bu odak, kütüphaneyi öğrenmeyi ve kullanmayı kolaylaştırır. İhtiyacın olan işe uyan kütüphaneyi seçersin.



Görüntü açan bir kütüphaneyi düşün. O olmadan, bir görüntü dosyasını okumanın tüm ayrıntılarını kendin yazardın; bu uzun ve karmaşıktır. Kütüphaneyle, hazır kodunu çağırırsın ve görüntü açılır. Programcı bu ayrıntıları sıfırdan yazmaz. Bu, yeniden kullanımın iş başındaki somut bir örneğidir.



İkinci bir örnek düşün: grafik çizen bir kütüphane. Ekranda çizgiler, şekiller ve renkler çizmek için hazır kod içerir. Programın, her çizim ayrıntısını kendisi yazmak yerine onu çağırır. Yine hazır kod, programcıyı tüm alt düzey ayrıntıları yazmaktan kurtarır. Grafik üretmek çok daha kolay hale gelir.



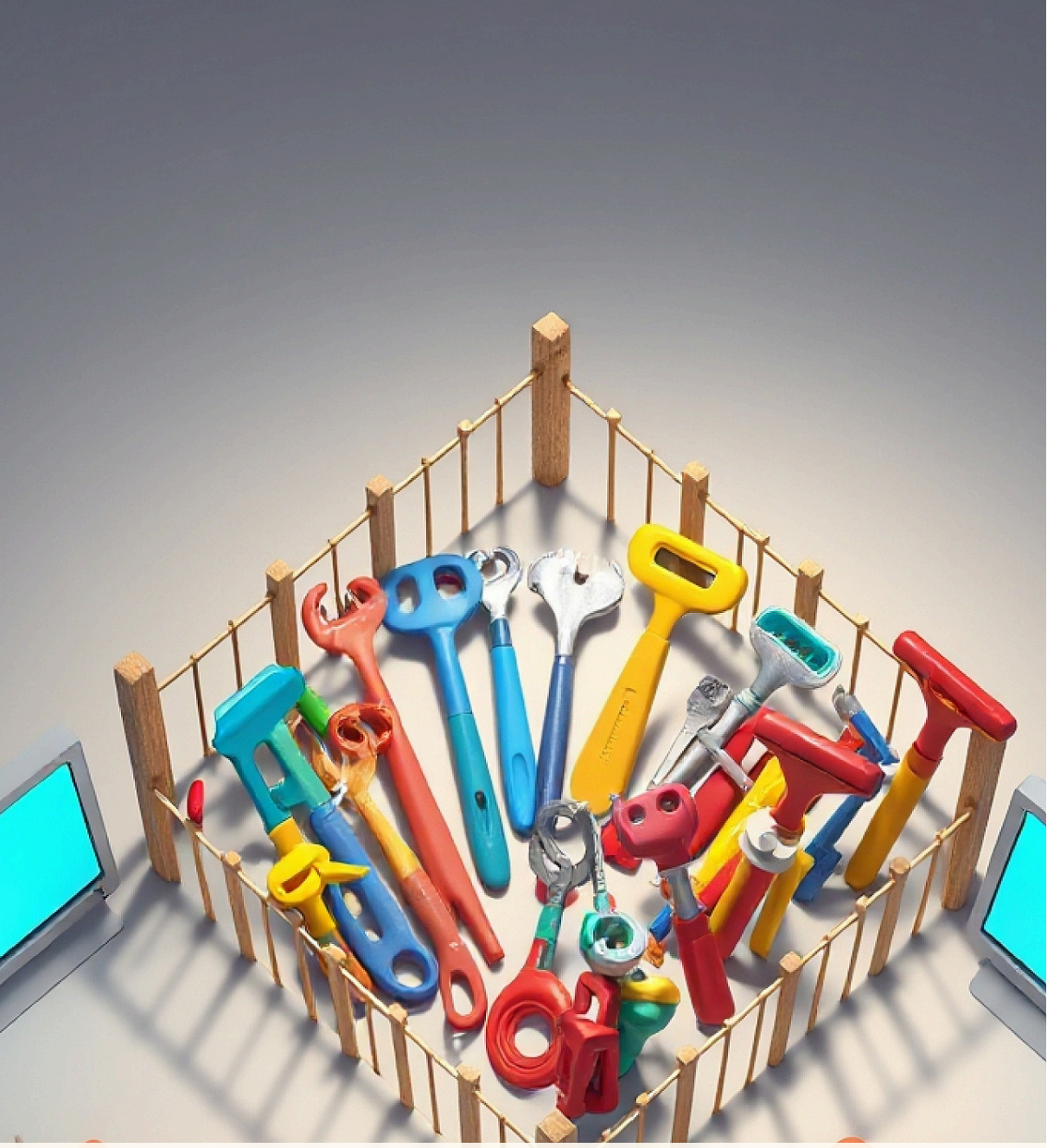
Kütüphaneler vardır çünkü yeniden kullanım zaman kazandırır ve tekrarlanan işi azaltır. Programcı tüm ayrıntıları sıfırdan yazmak zorunda kalmaz, çünkü birileri o kodu zaten yazmış ve paylaşmıştır. Tekrarlamak yerine zaten var olan işin üzerine inşa edersin. Bu, programının gerçekten özgün olan kısımlarına odaklanmanı sağlar.



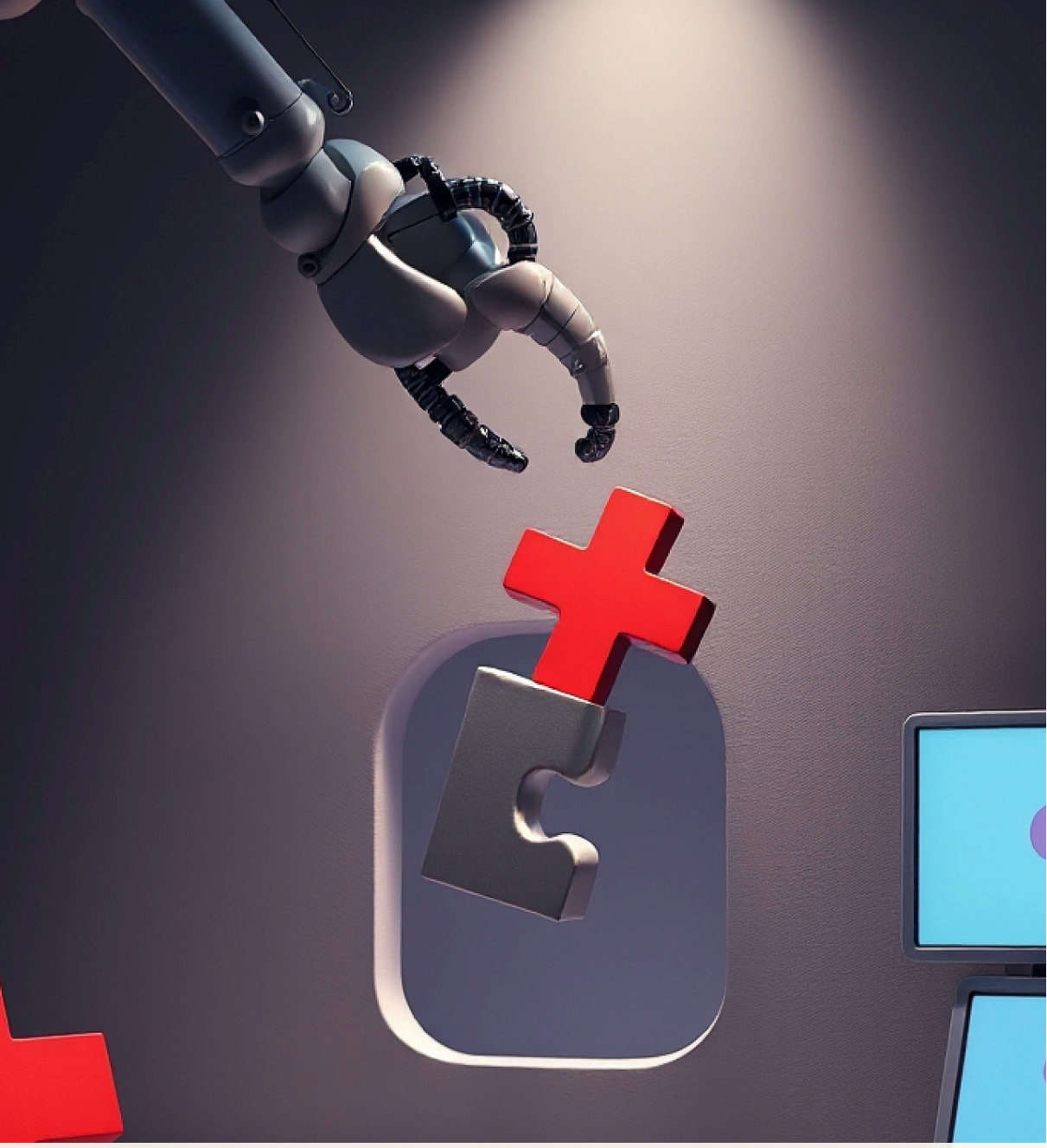
Bir kütüphaneyi kullanmak, onun hazır kodunu kendi programından çağırmak demektir. Kütüphanenin ne zaman kullanılacağına ve hangi bilgiyi alacağına yine programın karar verir. Örneğin programın, görüntü kütüphanesini yalnızca kullanıcı bir fotoğraf açmak istediğinde çağırabilir. Görevi kütüphane yapar, ama akışın kontrolü senin kodunda kalır.



Bir kütüphane işe yaramadan önce incelenmelidir. Neler sunduğunu, her parçanın ne yaptığını ve onu doğru şekilde nasıl çağıracağını öğrenmen gerekir. Nasıl kullanılacağını açıklayan yazılı rehber olan belgelerini oku. Bu incelemeyi atlamak genellikle hatalara yol açar, çünkü bir kütüphanenin nasıl çalıştığını tahmin etmek nadiren başarılı olur.



Her kütüphanenin sınırları vardır. Belirli görevleri yapar, her şeyi değil; bu yüzden neleri yapamadığını bilmelisin. Örneğin görüntü açan bir kütüphane aynı zamanda grafik çizmeyebilir veya sesle ilgilenmeyebilir. Sınırları bilmek doğru kütüphaneyi seçmene ve hiç verilmek üzere yapılmadığı sonuçları beklemekten kaçınmana yardımcı olur. Neleri kapsadığını belgelerinden kontrol et.



Yaygın bir hata, bir kütüphaneyi anlamadan körü körüne kullanmaktır. Kod hatasız çalışabilir, ama yanlış sonuç üretebilir çünkü onu yanlış çağırmışsındır veya ona uygun olmayan girdi vermişsindir. Örneğin bir görüntü kütüphanesini desteklemediği bir dosyayla çağırmak sessizce başarısız olabilir. Kullanmadan önce kütüphaneyi anlamak bu sorunların çoğunu önler.



Bir başka yaygın hata, hazır kodu kendi durumun için test etmeden ona güvenmektir. Bir kütüphane genel olarak iyi çalışabilir, ama senin verilerinle farklı davranabilir. Bir kütüphane kullanmak, programın doğruluğunu test etme sorumluluğunu ortadan kaldırmaz. Kodun yalnızca çalıştığını değil, sonuçların beklediğinle eşleştiğini her zaman kontrol et.



Programını test etmek, kütüphaneyi kullandığında doğru sonuçlar verdiğini kontrol etmek demektir. Önce basit durumları, sonra daha zorlarını dene ve çıktıyı beklediğinle karşılaştır. Örneğin bilinen bir görüntüyü aç ve doğru göründüğünü doğrula. Çağırdığın kodu başkası yazmış olsa bile test etmek senin işin olarak kalır.



Şimdi fikirleri tek bir resimde birleştir. Kütüphane, belirli görevler için yeniden kullanılabilir koddur; böylece her şeyi sıfırdan yazmazsın. Onu nasıl kullanacağını öğrenirsin, sınırlarını bilirsin ve doğruluğu onaylamak için programını test edersin. Bu parçalar birlikte işler: yeniden kullanım zaman kazandırır, anlamak hataları önler ve test etmek sonucun doğru olduğunu kanıtlar.



Akılda tutulacaklar: kütüphane, belirli görevleri yapmak için hazırlanmış yeniden kullanılabilir kod koleksiyonudur ve tüm ayrıntıları sıfırdan yazmaktan seni kurtarır. Onu nasıl kullanacağını öğren, sınırlarını bil ve programını her zaman doğruluk açısından test et. Hazır kod kullanmak daha hızlı geliştirmene yardımcı olur, ama doğru bir programın sorumluluğu sende kalır.