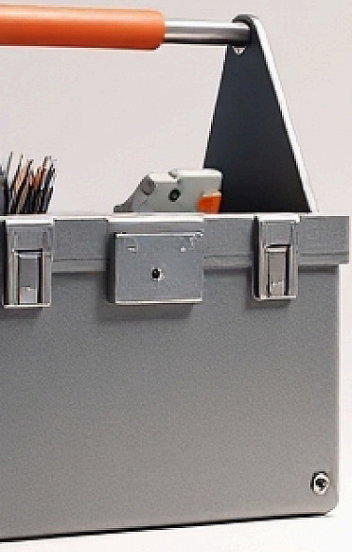
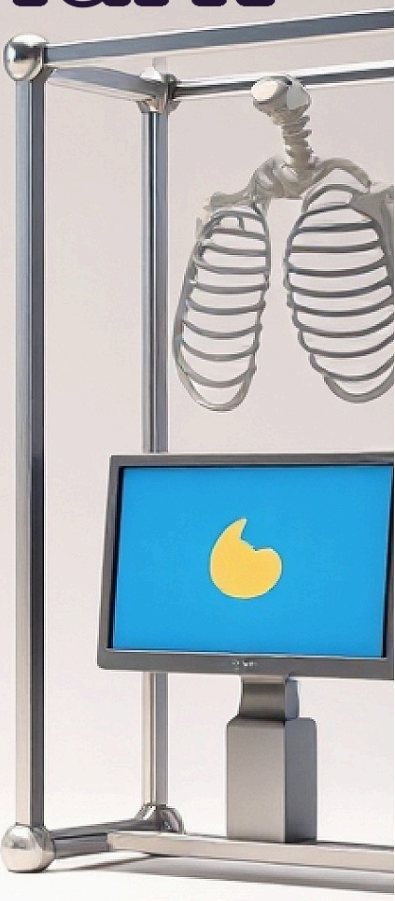
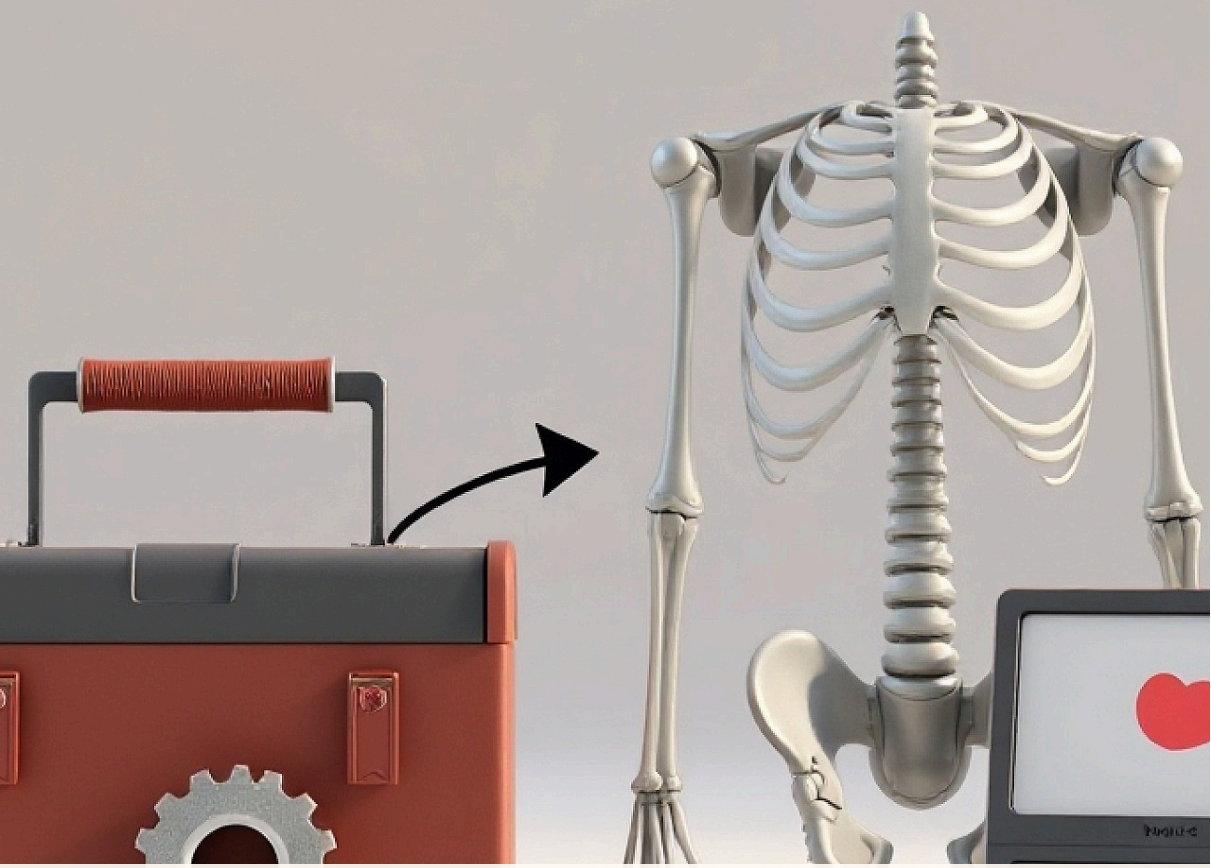
Bir framework ile kütüphane arasındaki fark nedir?





www.yzokul.com.tr

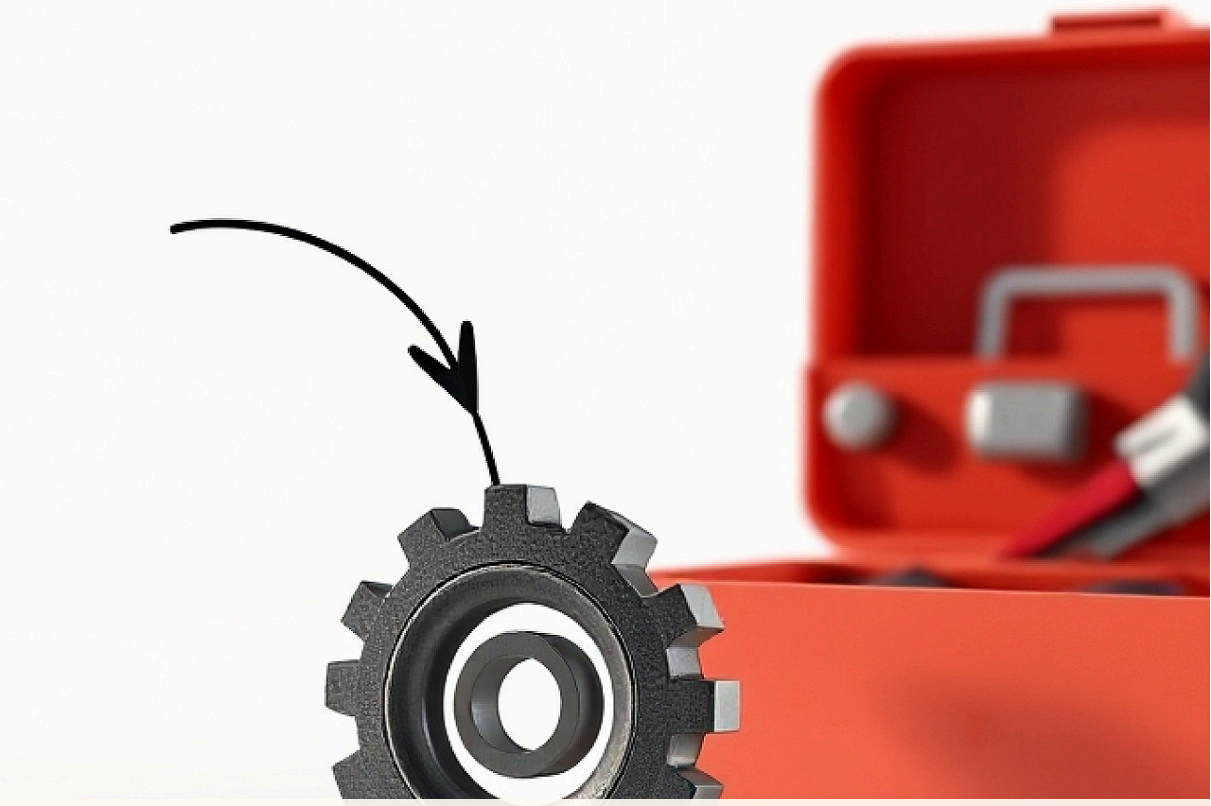
Bir framework ile kütüphane arasındaki fark nedir?



Kütüphane ve framework sözcüklerini duymuş ve bunların nasıl farklı olduğunu merak etmiş olabilirsin. Kısa yanıt şudur: bir kütüphane, senin çağırdığın araçlar sunar; bir framework ise programının yapısını kurabilir ve akışını kontrol edebilir. Bu kitap bu farkı basit örnekler ve sade bir dille adım adım açıklar. Sonunda temel fikri anlayacak ve birçok durumda hangisinin hangisi olduğunu ayırt edebileceksin.



Kütüphane, programında kullanabileceğin hazır araçlar topluluğudur. Bunu, başka birinin yazıp test ettiği yararlı fonksiyonlarla dolu bir alet çantası olarak düşün. Kütüphane orada bekler ve senin kodun araçlarından birini çağırana kadar hiçbir şey yapmaz. Her aracı ne zaman ve nasıl kullanacağına sen karar verirsin ve programın kendi adımlarını çalıştırmaya devam eder. Kütüphaneler, yaygın kodu sıfırdan yazmaktan kaçınmana yardımcı olur.



Bir kütüphanede programın yetkilidir ve bir aracı ne zaman kullanacağına karar verir. Kodunu, yalnızca bir şeye ihtiyaç duyduğunda kütüphaneye uzanan patron olarak düşün. Kütüphane programını çalıştırmaz veya ona ne yapacağını söylemez; yalnızca çağrıldığında yanıt verir. Bu, programındaki eylemlerin sırası üzerinde tam kontrole sahip olduğun anlamına gelir. Hangi araçları kullanacağını ve bunları ne zaman kullanacağını sen seçersin.



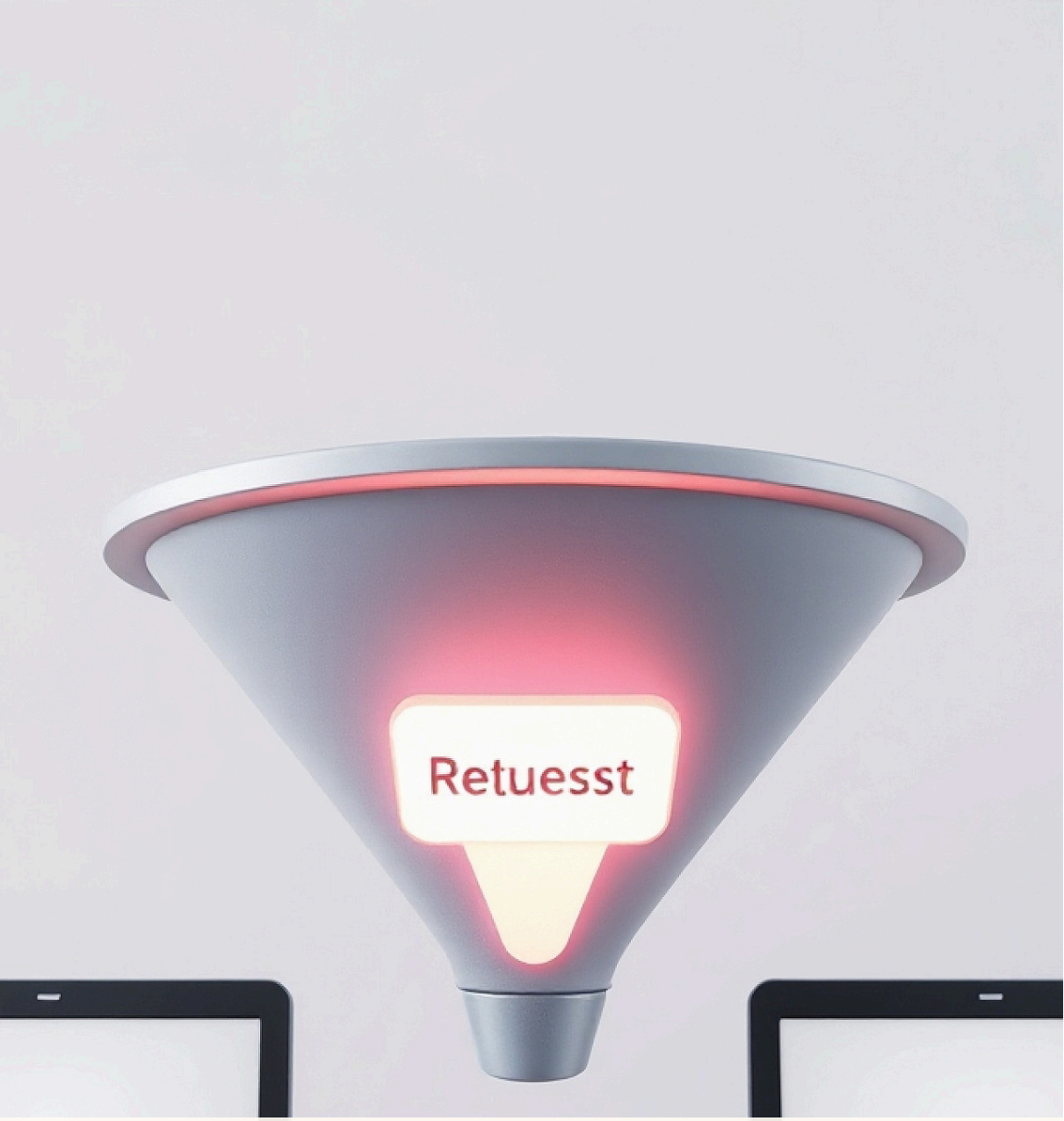
Programının biraz matematik yapması gerektiğini ve bu yüzden bir kütüphaneden bir matematik aracı çağırdığını düşün. Programın kendi adımlarını çalıştırmaya devam eder ve yalnızca bir sonuç almak için aracı bir an ödünç alır. Araç yanıtı döndürdükten sonra programın kendi mantığıyla devam eder. Kütüphane, programının bundan sonra ne yapacağına asla karar vermez; yalnızca hesaplamayı sağlar. Kütüphaneler böyle çalışır: edilgen yardımcılardır.



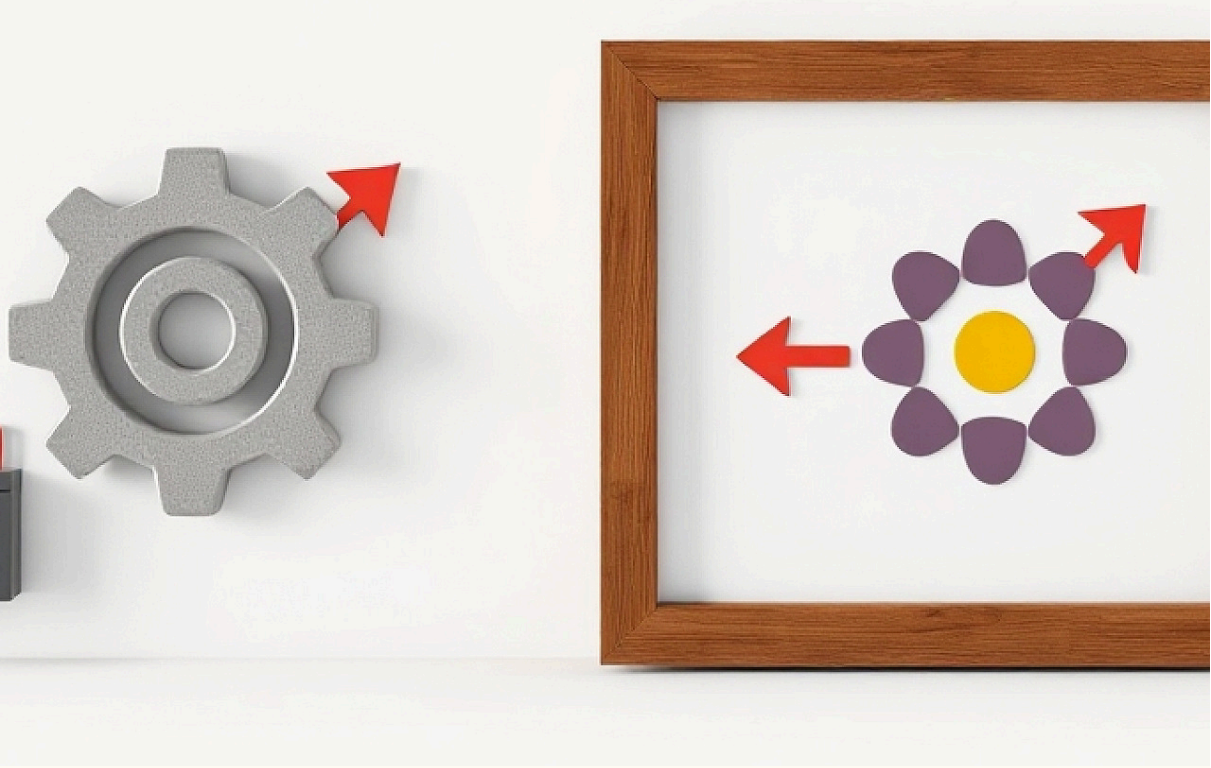
Framework bir araçlar kümesinden fazlasıdır: tüm uygulamanın genel yapısını sağlayabilir. Kodunun içine oturduğu hazır bir iskelet sunar. Framework, kodunun nereye gideceğini ve farklı parçaların nasıl bağlanacağını tanımlar. Ayrıntıları sen doldurursun ama genel biçim zaten belirlenmiştir. Bu sana çok fazla yapı sağlar ve özellikle yaygın uygulama türleri için geliştirmeyi hızlandırabilir.



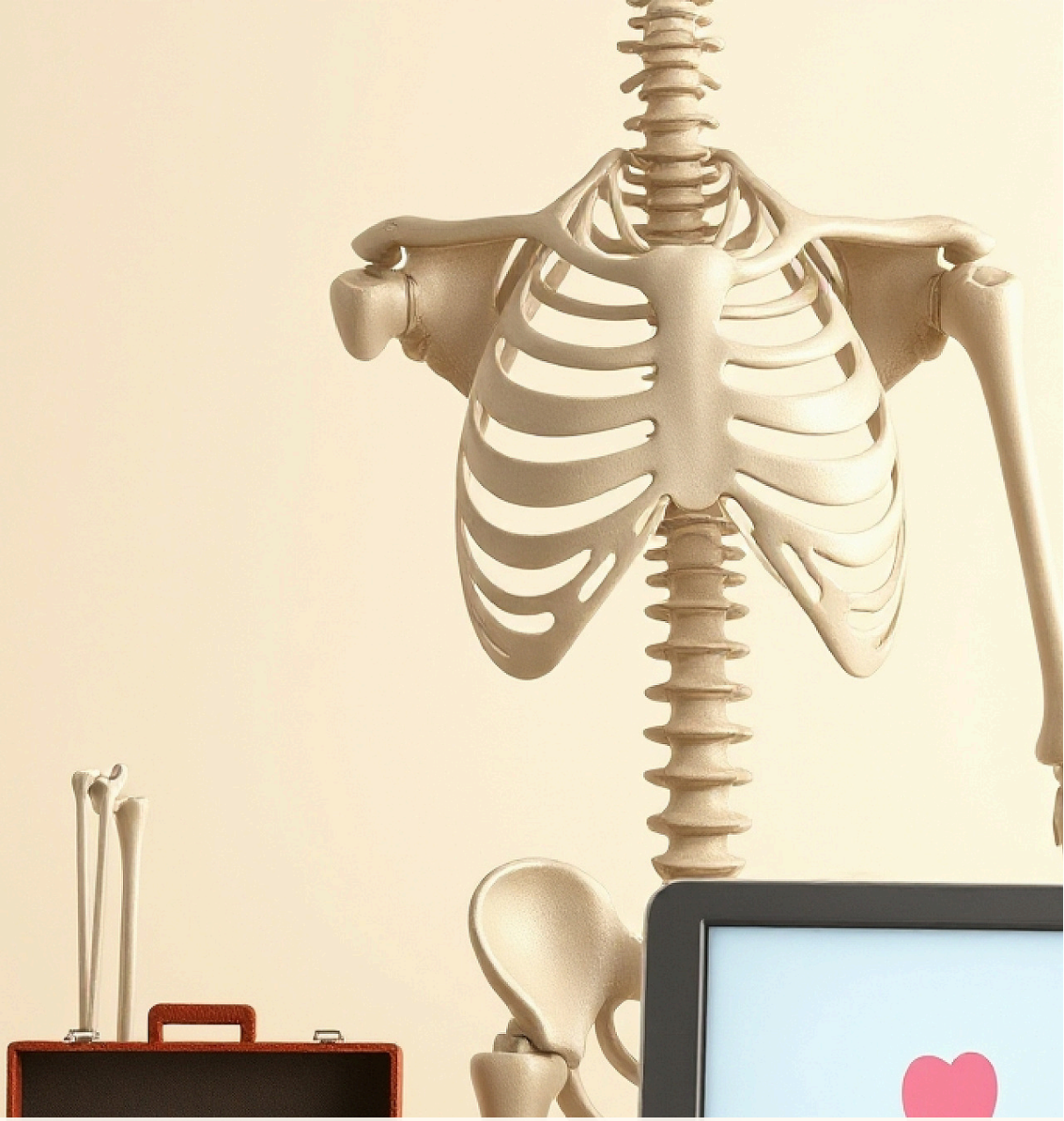
Bir framework'te temel dönüş noktası, akışın sorumluluğunun framework'te olması ve gerektiğinde senin kodunu çağırmasıdır. Kodun, framework'ün istediği parçaları, örneğin belirli olayları işleyen belirli fonksiyonları doldurur. Bu fonksiyonların ne zaman ve hangi sırayla çalıştırılacağına framework karar verir. Buna genellikle kontrolün tersine çevrilmesi denir: framework'ü sen çağırmazsın; framework seni çağırır. Kodun, framework'ün akışına takılan parçalar kümesi hâline gelir.



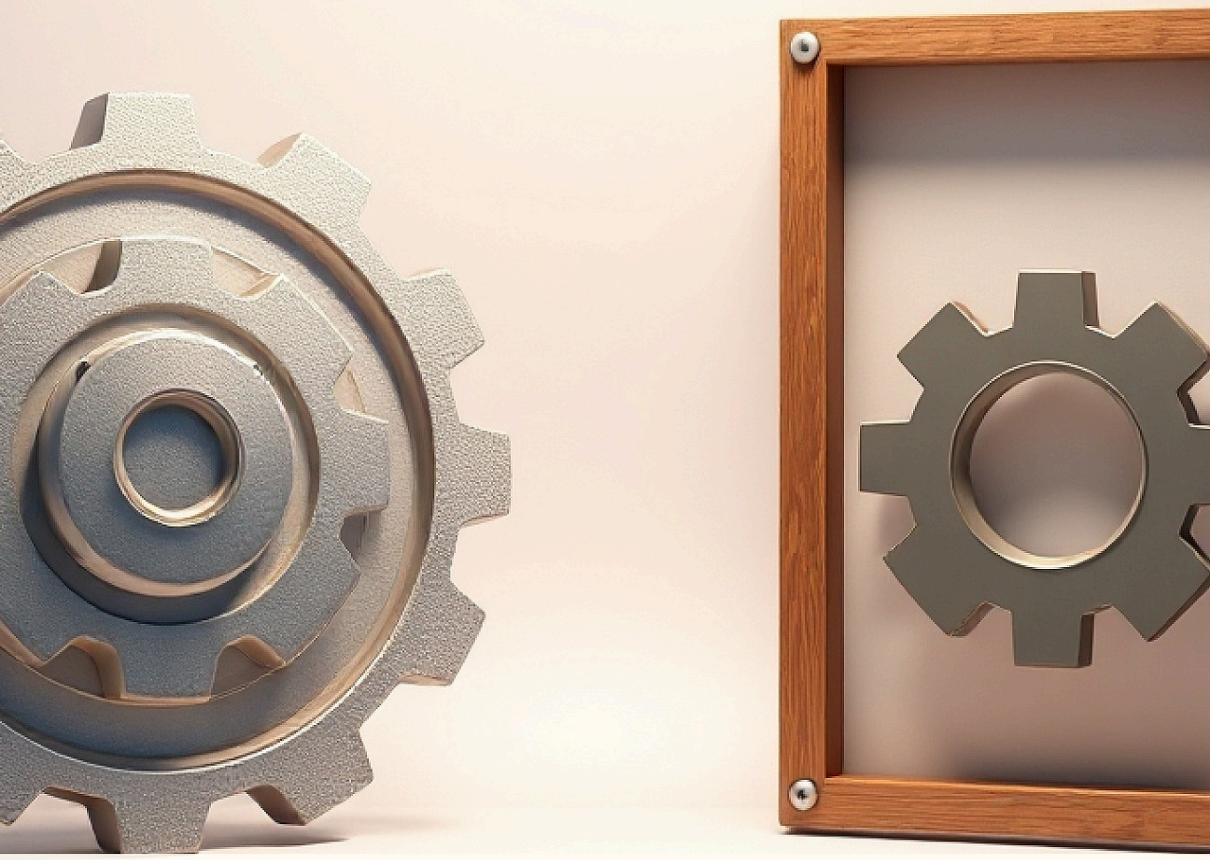
Bir web framework'ünün tarayıcıdan gelen bir isteği aldığını düşün. Framework isteğe bakar ve bunu senin fonksiyonlarından hangisinin işleyeceğine karar verir. Fonksiyonun yalnızca framework isteği ona gönderdiği için çalışır. Fonksiyonun bitince framework bir yanıt göndermek için yeniden kontrolü alabilir. İşleyici fonksiyonları sen yazarsın ama istekleri işlemenin genel akışını framework kontrol eder.



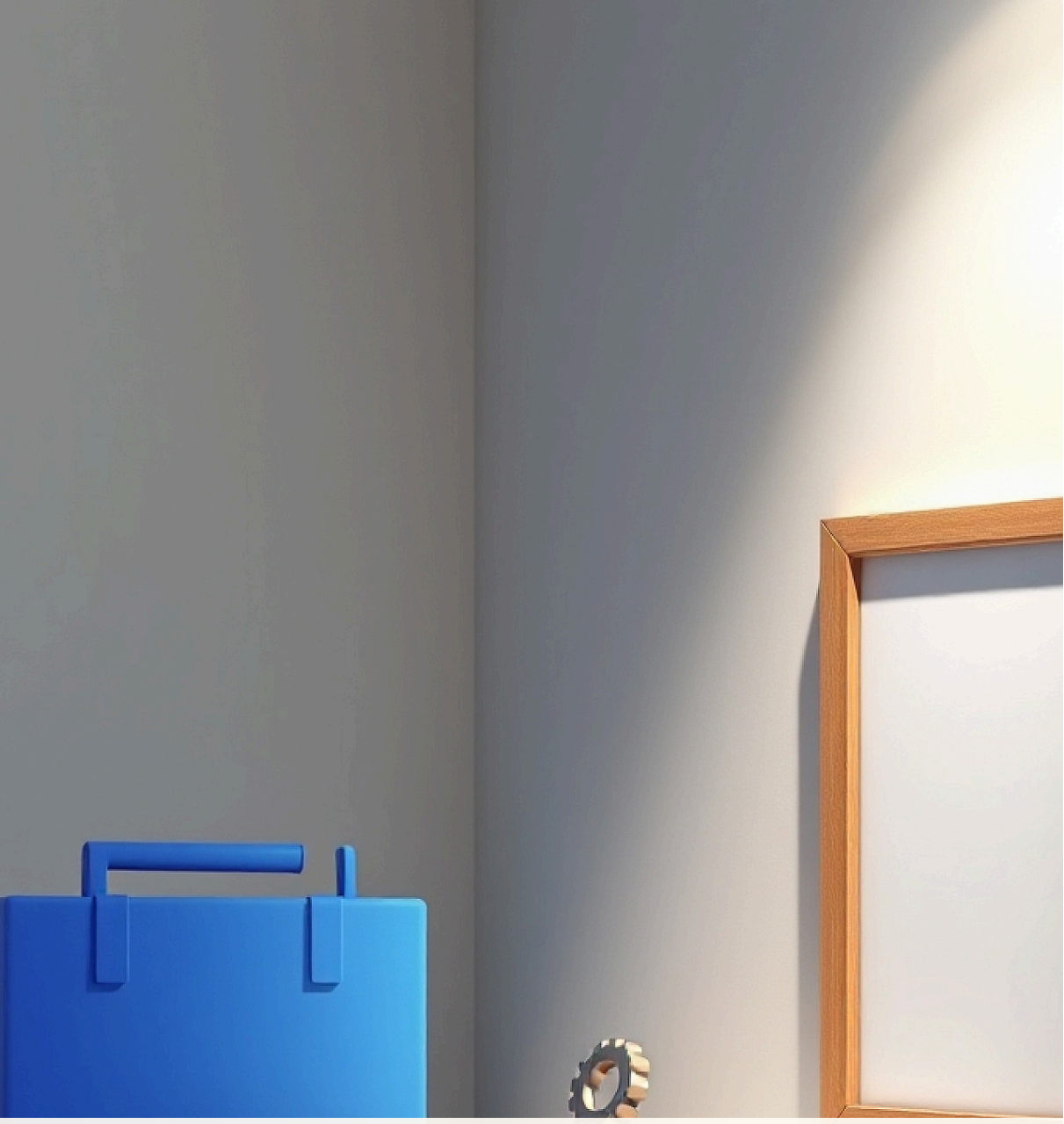
Temel fark, programın akışını kimin düzenlediğidir. Bir kütüphanede programın akışı düzenler: ihtiyaç duyduğunda kütüphanenin araçlarını sen çağırırsın. Bir framework'te akışı framework düzenler: kodunu doğru zamanlarda o çağırır. Bütün sorunun özü budur. Unutma: eylem sırasının sorumlusu kim? Bu, bir şeyin daha çok kütüphaneye mi yoksa framework'e mi benzediğini söyler.



Bir framework, bir kütüphaneden çok daha fazla hazır yapı sunar. Bir kütüphane sana araçlar verir ama genel yapıyı kendin kurarsın. Bir framework, içinde inşa edeceğin bütün bir çerçeve verir; kodun için önceden tanımlanmış yerler ve şeylerin bağlanması için önceden tanımlanmış yollar sunar. Bu ek yapı zaman kazandırabilir ama aynı zamanda framework'ün kurallarına uyman gerektiği anlamına gelir. Yapı miktarı, farkın ikinci parçasıdır.



İki basit resmi yan yana karşılaştır. Bir tarafta kodun ana programdır; gerektiğinde bir kütüphane aracını çağırır ve kütüphane yanıt verir. Diğer tarafta framework ana programdır; belirli noktalarda senin kodunu çağırır ve kodun yanıt verir. İkisini aynı anda görmek farkı netleştirir: kim kimi çağırıyor? İlk resimde kütüphaneyi sen çağırırsın; ikincisinde framework seni çağırır.



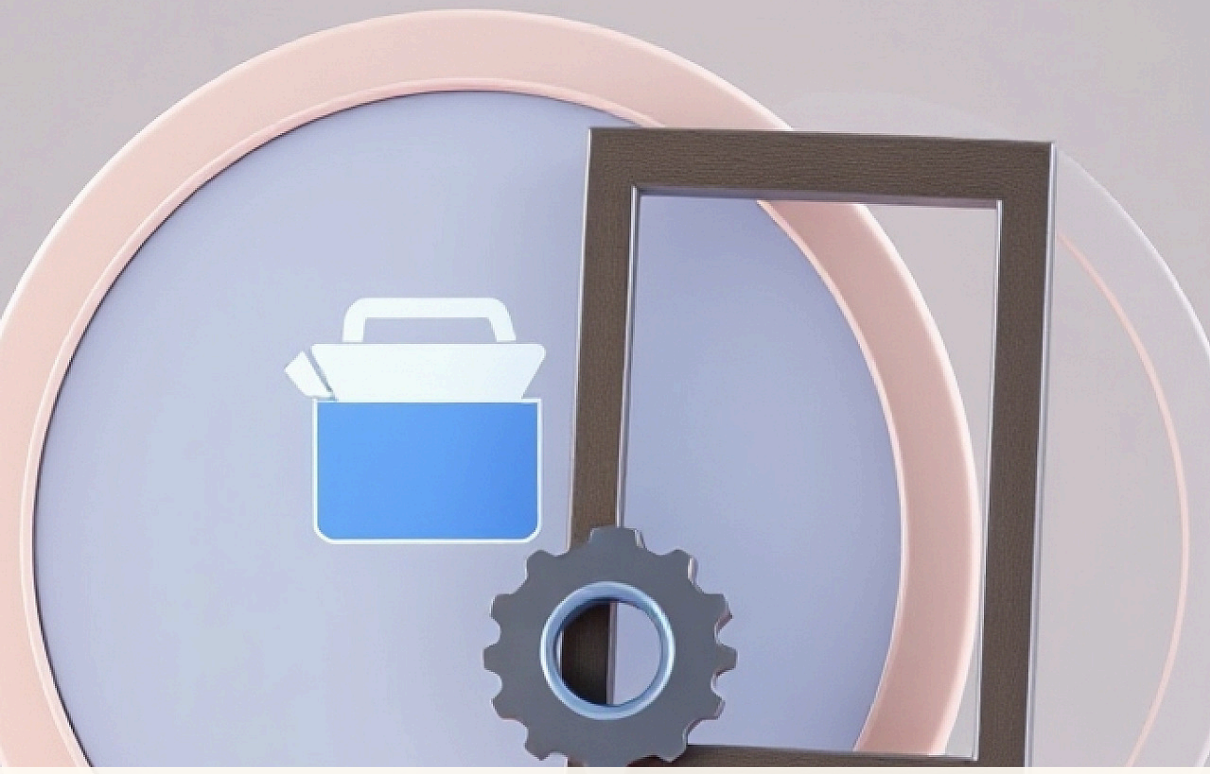
Kütüphane ile framework arasındaki çizgi her üründe her zaman net değildir. Bazı ürünler arada durur: araçlar sunarlar ama aynı zamanda bir miktar yapı dayatırlar veya kodunu belirli şekillerde çağırırlar. Bu yüzden etiketler tam olarak katı değildir. Framework diye adlandırılan ama kısmen kütüphane gibi davranan ürünler ya da tersi bulabilirsin. Yalnızca adına güvenmek yerine ürünün nasıl kullanıldığına bakmak daha yararlıdır.



Yeni başlayanların yaygın bir hatası, farkın boyutla ilgili olduğunu, bir framework'ün yalnızca daha büyük bir kütüphane olduğunu düşünmektir. Ama boyut asıl mesele değildir. Küçük bir framework akışı kontrol edebilir ve büyük bir kütüphane hâlâ yalnızca bir araçlar kümesi olabilir. Gerçek fark, akışı kimin düzenlediği ve ne kadar yapı sunulduğudur. Bu yüzden araçların veya kod satırlarının sayısına göre değil, kontrole göre karar ver.



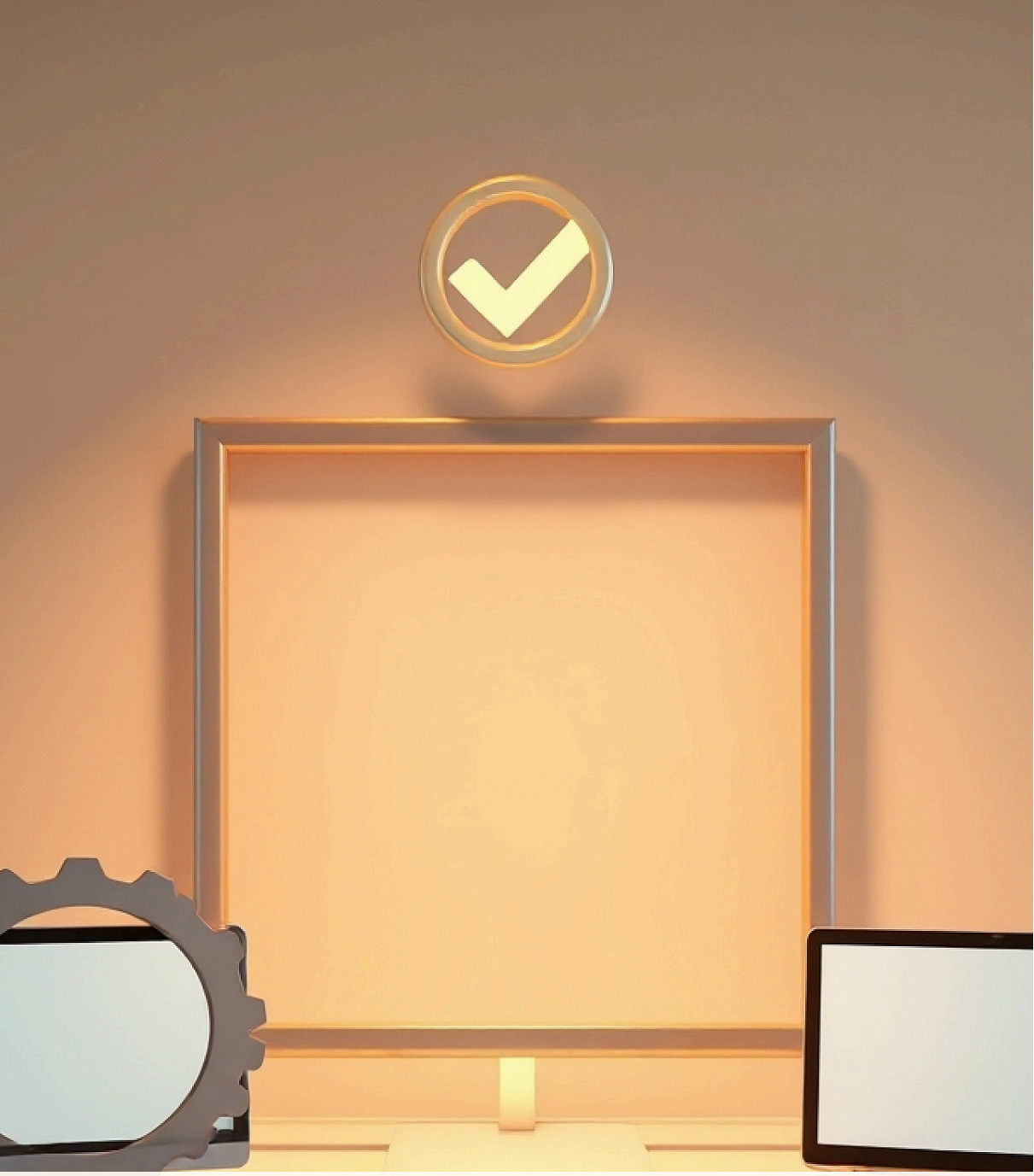
Bir başka hata, bir framework'ün yalnızca daha fazla aracı olan bir kütüphane olduğunu düşünmektir. Asıl değişiklik bu değildir. Temel değişiklik, programın akışının kontrolünün framework'e geçebilmesidir. Bir kütüphanede kontrol sende kalır; bir framework'te framework öne geçer ve senin kodunu çağırır. Yani bir framework yalnızca ek kod değil, programının nasıl çalıştığını düzenlemenin farklı bir yoludur.



İki sözcüğü her durumda tamamen ayrı şeyler olarak ele almak bir hatadır. Sınır bulanıklaşabilir, bu yüzden yalnızca ada göre değil, akışı kimin kontrol ettiğine göre karar vermelisin. Bazı ürünler framework diye adlandırılabilir ama yalnızca araçlar sunabilir ve bazı kütüphaneler bir miktar yapı sağlayabilir. Her zaman sor: ürünü benim kodum mu çağırıyor, yoksa ürün benim kodumu mu çağırıyor? Bu soru etiketten daha güvenilirdir.



Herhangi bir ürün için basit bir test, kimin kimi çağırdığını sormaktır. Kodun, araçlarını kullanmak için ürünü çağırıyorsa o zaman ürün kütüphane gibi davranır. Ürün, kodunu belirli noktalarda çağırıyorsa o zaman framework gibi davranır. Bunu genellikle örneklere bakarak anlayabilirsin: ürünün çağıracağı fonksiyonları mı yazıyorsun, yoksa ürünün fonksiyonlarını çağıran kodu mu yazıyorsun? Bu yanıt rolü ortaya koyar.



Özetle, bir kütüphane programının çağırdığı araçlar sunar; bir framework ise programının yapısını kurabilir ve akışını kontrol edebilir. Temel fikri unutma: akışı kim düzenliyor ve ne kadar hazır yapı elde ediyorsun. Pratikte karar vermek için kim-kimi-çağırıyor testini kullan. Bu anlayışla projen için doğru aracı daha iyi seçebilir ve ne bekleyeceğini bilebilirsin.