Hata ayıklama nedir?

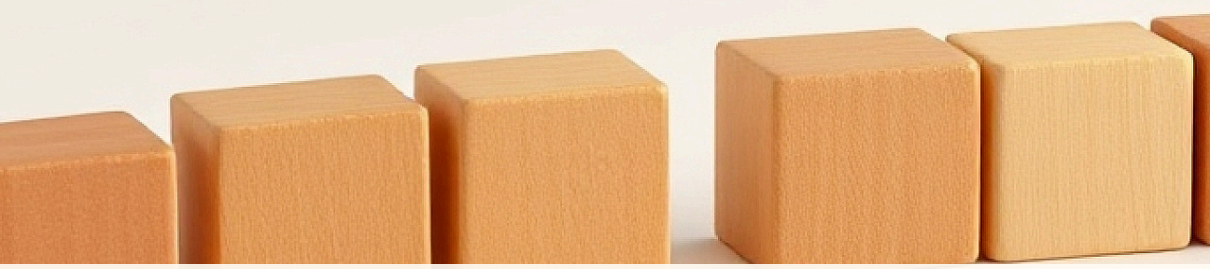


www.yzokul.com.tr

Hata ayıklama nedir?



Hata ayıklama, bir programdaki bir sorunun nedenini bulmak ve düzeltmektir. Program, bilgisayarın çalıştırdığı bir dizi komuttur ve bazen yanlış bir şey yapar. Hata ayıklama, bunun neden olduğunu bulma ve düzeltme işidir. Başlamak için ön bilgiye ihtiyacın yok; yalnızca sabır ve her adıma net bir bakış açısı yeterlidir.



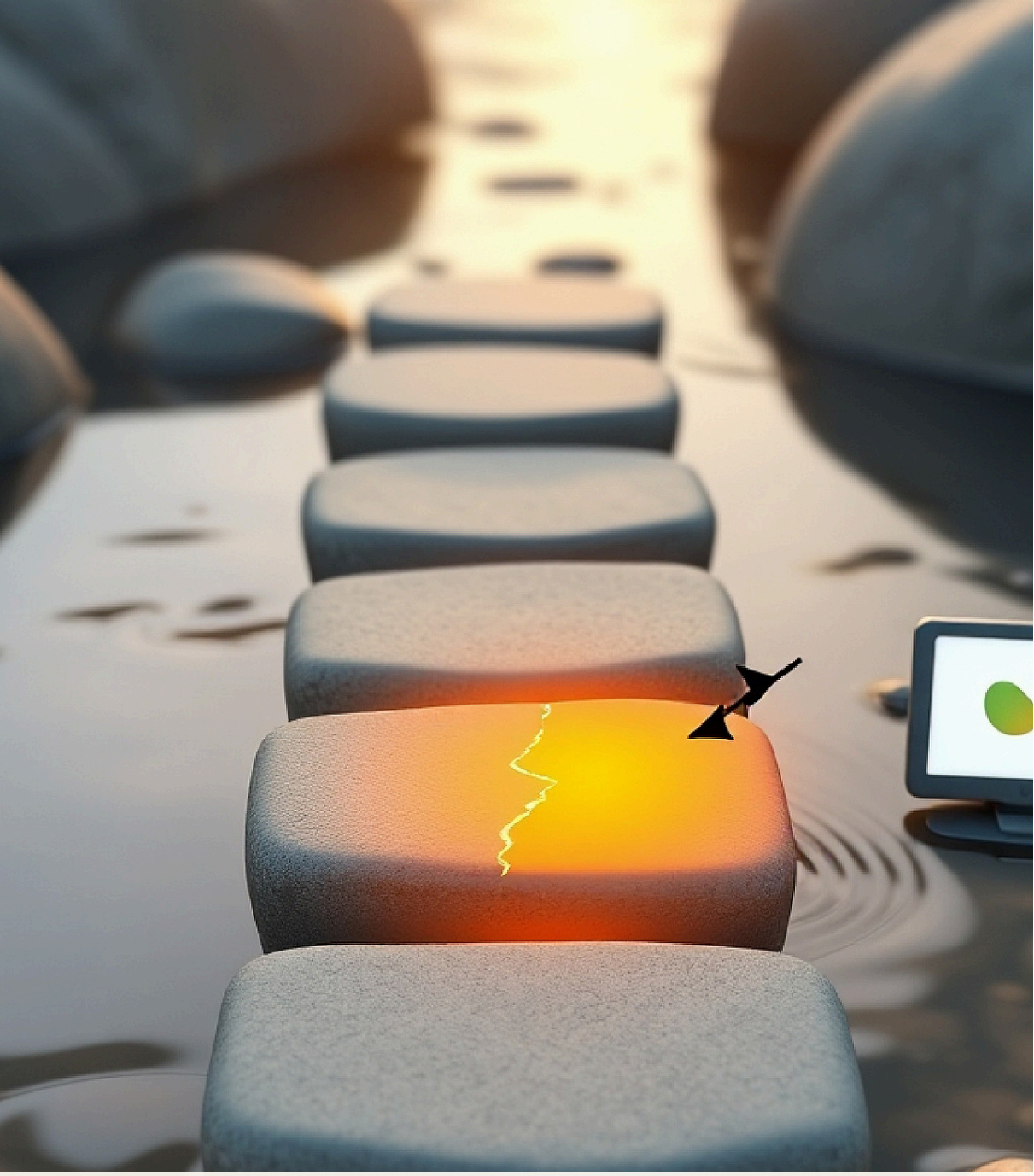
Program, bilgisayarın sırayla, birbiri ardına izlediği adımlar listesidir. Her adım, iki sayıyı toplamak veya bir değeri saklamak gibi küçük bir iş yapar. Hata ayıklama ancak bir programı adımlar dizisi olarak gördüğünde anlam kazanır, çünkü bir sorun her zaman bu dizideki belirli bir adımda bulunur. Adımları yukarıdan aşağıya doğru oku.



Hata (bug), programın amaçlanandan farklı bir şey yaptığı adımdır. Bu bir gizem ya da lanet değildir; yalnızca beklenen davranış ile gerçek davranış arasındaki uyumsuzluktur. Bir hatanın somut bir uyumsuzluk olduğunu kabul ettiğinde, onu sakince arayabilirsin. Her hata, adımlar içinde bulunabilir bir yerde yaşar.



Beklenen ile gerçek arasındaki fark, hata ayıklamanın kalbidir. Programın ne yapması gerektiğini yaz, sonra gerçekte ne yaptığını gözlemle. İkisi arasındaki fark, bir hatanın var olduğunu ve kabaca nereye bakman gerektiğini söyler. Hata ayıklama bu karşılaştırmayla başlar, bu yüzden gerçek sonucu incelerken beklenen sonucu her zaman gözünün önünde tut.



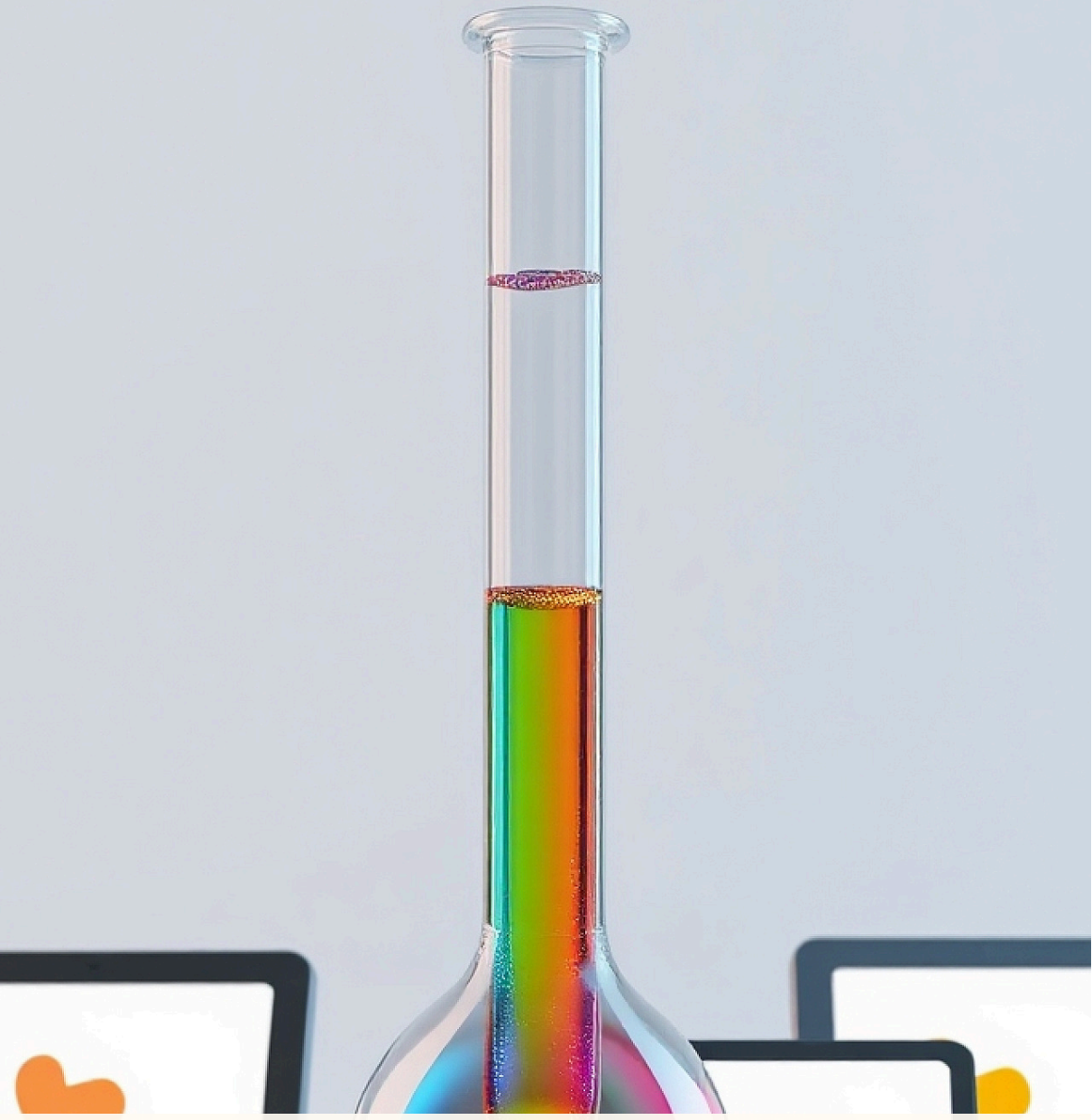
Davranışın ilk kez beklediğinden farklılaştığı adımı bul. Tüm programa birden değil, adım adım bak, çünkü bütünü aklında tutmak fazla gelir. Önce ilk adımı, sonra ikinciyi ve böyle devam ederek sonuç beklentinle eşleşmeyi bırakana kadar kontrol et. İlk farklılaşan adım, nedenleri test etmeye başlayacağın noktadır.



Üç fiyatı toplayıp toplam olarak 30 göstermesi gereken küçük bir program düşün. Bunun yerine 27 gösteriyor. Amaç tahmin etmek değil, yanlış sayıyı hangi adımın ürettiğini bulmaktır. Fiyatları okundukları anda, sonra toplamayı, sonra da gösterimi kontrol et. 30'un sessizce 27'ye dönüştüğü yer bu adımlardan biridir.



Hiçbir şeyi değiştirmeden önce hatayı yeniden üret. Programı aynı girdilerle aynı şekilde çalıştırarak hatayı bilerek yeniden oluştur. Hatayı yeniden üretemiyorsan, düzeltmenin işe yarayıp yaramadığını sonra test edemezsin. Yeniden üretilebilir bir hata güvenilir bir hedeftir, bu yüzden onu tetikleyen tam adımları yaz.



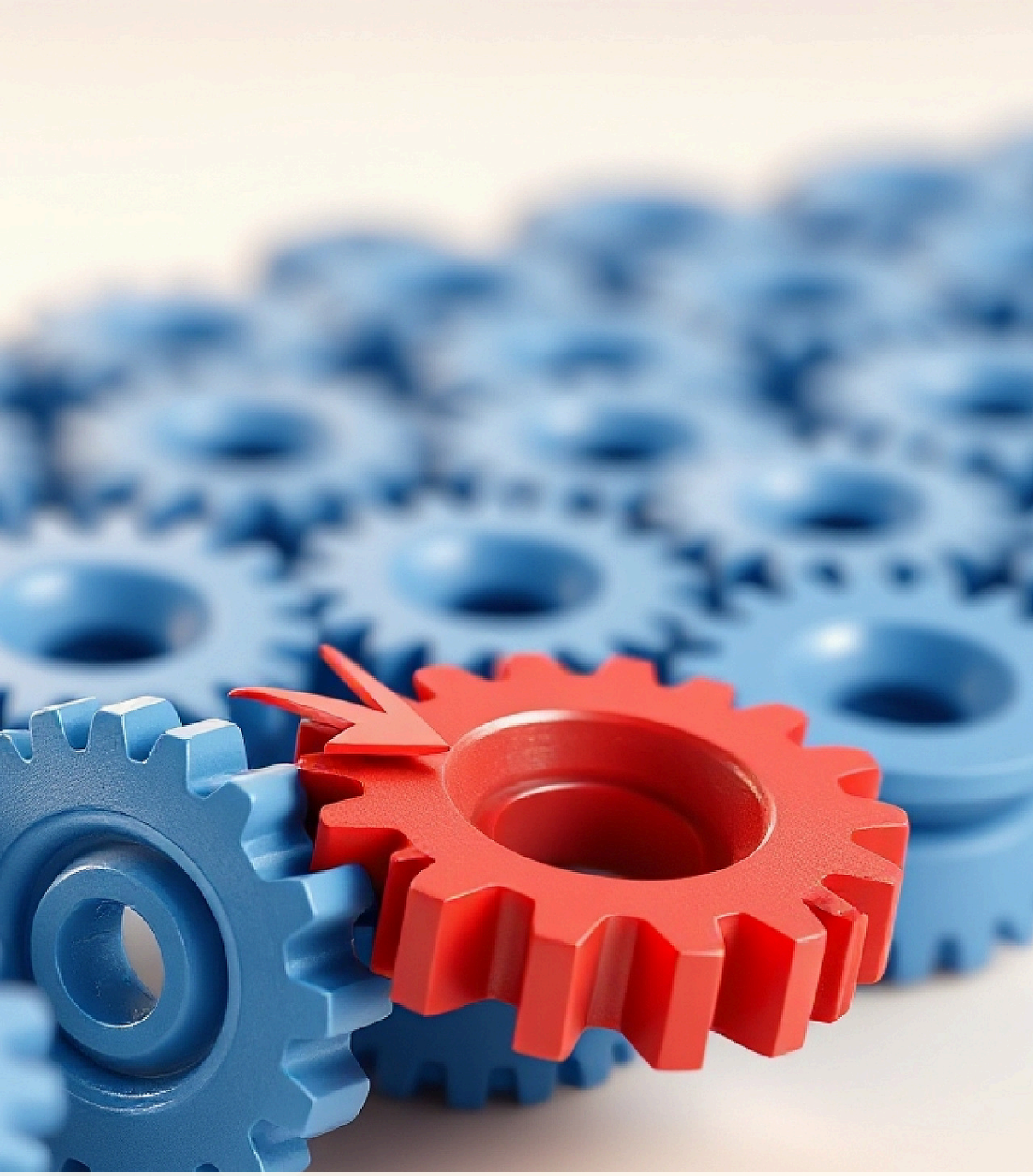
Değişkenleri yazdırarak farklı adımlardaki değerlerine bak. Değişken, bir toplam veya sayım gibi bir değeri tutan adlandırılmış bir kutudur. Her adımdan sonra bir yazdırma satırı ekle ve programı yeniden çalıştır. Değerleri görmek, bir sayının ilk kez tam olarak nerede yanlış gittiğini gösterir ve aramanı hızla daraltır.



Programlar günlük (log) yazabilir; bunlar program çalışırken ne olduğunun kayıtlarıdır. Her günlük satırı genellikle bir mesajı ve bazen belirli bir andaki bir değeri gösterir. Günlükleri okumak, programın gerçekte izlediği yolu, beklemediğin adımlar dahil gösterir. Son doğru kaydı ve ilk yanlış kaydı ara.



Bir programı duraklatıp çalışırken incelemene olanak tanıyan özel hata ayıklama araçları vardır. Kesme noktası (breakpoint), programı seçilen bir satırda durduran ve oradaki değerlere bakmanı sağlayan bir işarettir. Bu araçlar, yazdırma satırları eklemeden her adımdaki durumu görmene yardımcı olur. Yazdırma işi karışık veya yavaş hale geldiğinde bunları kullan.



Her seferinde tek bir nedeni test et. Yanlış bir fiyatın okunması gibi, neyin yanlış olduğuna dair tek bir tahmin oluştur ve yalnızca o tahmini test et. Tek bir şeyi değiştir, programı çalıştır ve sonucu gözlemle. Her seferinde tek bir şeyi değiştirmek sonuçları net tutar, böylece gördüğün değişikliğe neyin yol açtığını her zaman bilirsin.



Belirtiyi değil, nedeni düzelt. Belirti, görünen yanlış toplamdır; neden ise onu üreten adımdır. Yanlış sayının üzerine yalnızca doğru sayıyı yazarsan hata kalır ve başka girdilerle yeniden ortaya çıkar. Doğru sonucun kendiliğinden çıkması için gerçekten bozuk olan adımı onar.



Yaygın bir hata, kodu rastgele değiştirip sorunun kendiliğinden geçmesini ummaktır. Rastgele değişiklikler gerçek nedeni gizler ve çoğu zaman başka adımlarda yeni hatalar yaratır. Neyin yardımcı olduğunu ve neyin olmadığını izleyemezsin. Hızlıca düzenleme dürtüsüne karşı koy; bunun yerine net bir nedeni, her seferinde tek bir değişiklikle test et.



Bir diğer yaygın hata, hatayı önce yeniden üretmeden düzeltme yapmaktır. Hatayı bilerek hiç görmediysen, değişikliğinin onu ortadan kaldırdığını doğrulayamazsın. Hiç bozuk olmayan bir şeyi düzeltiyor ya da gerçek nedeni kaçırıyor olabilirsin. Her zaman önce hatayı yeniden üret, sonra düzelt, sonra yeniden üret.



Düzeltmenin işe yaradığını, hataya yol açan aynı adımları yeniden çalıştırarak kontrol et. Daha önce yazdığın aynı girdileri ve aynı eylemleri kullan. Davranışın artık beklendiği gibi olduğunu ve yeni yanlış değerlerin olmadığını doğrula. Bu, hata ayıklama döngüsünü kapatır ve nedenin gerçekten ortadan kalktığını söyler.



Hata ayıklamanın, beklenen ile gerçek davranışı karşılaştırarak bir sorunun nedenini bulmak ve düzeltmek olduğunu unutma. Hatayı yeniden üret, ilk farklılaşan adımı bul ve yazdırma, günlükler veya hata ayıklama araçlarıyla nedenleri tek tek test et. Rastgele değişikliklerden kaçın ve düzeltmeyi her zaman özgün adımları yeniden çalıştırarak kontrol et.