Bir program kullanıcıdan bilgiyi nasıl alır?



www.yzokul.com.tr

Bir program kullanıcıdan bilgiyi nasıl alır?



Bir program kullanıcıdan bilgiyi nasıl alır? Kısa yanıt, bilginin klavye gibi giriş araçlarıyla geldiği, programın da onu denetleyip dönüştürdüğüdür. Örneğin yaşını yazdığında program bu metni alır ve sonra hesaplamalar için sayıya çevirir. Bu kitap bu sürecin her parçasını adım adım açıklar, böylece girdinin senden programa tam olarak nasıl gittiğini görürsün.



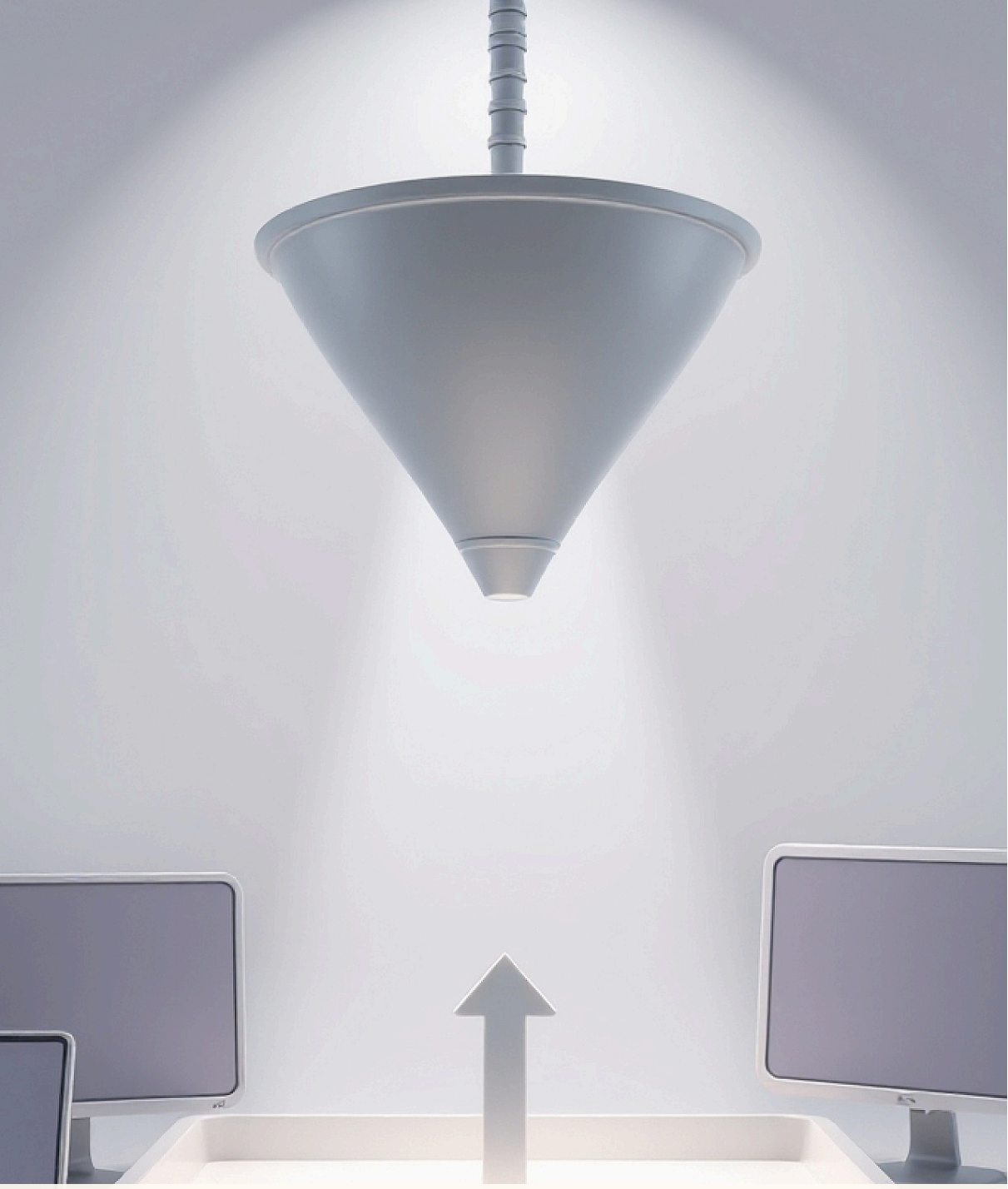
Girdi, bir programa verdiğin her türlü bilgi demektir. Girdi olmadan program yalnızca zaten bildikleriyle, yani yerleşik ayarlar veya kayıtlı verilerle çalışabilir. Örneğin girdisi olmayan bir hesap makinesi sadece sıfır gösterir; sayıları toplayabilmesi için önce senin sayı yazman gerekir. Yani girdi, seninle programın eylemleri arasındaki köprüdür.



Klavyede yazmak, bir programa bilgi göndermenin en yaygın yoludur. Her tuşa basış, programın alabileceği metne dönüşür. Örneğin bir arama kutusuna adını yazdığında bastığın her harf programa karakter olarak gönderilir. Program, işinin bittiğini bildirmek için Enter gibi bir tuşa basana kadar bu karakterleri sırayla toplar.



Bilgi formlar, dokunmatik ekranlar ve diğer giriş araçlarıyla da gelebilir. Form, bir web sitesindeki ad ve e-posta gibi doldurduğun alanlar kümesidir. Dokunmatik ekran düğmelere dokunmanı veya sanal klavyede yazmanı sağlar. Bütün bu araçlar veriyi programa aynı temel yolla iletir: eylemlerini programın alabileceği metne veya sinyallere dönüştürürler.



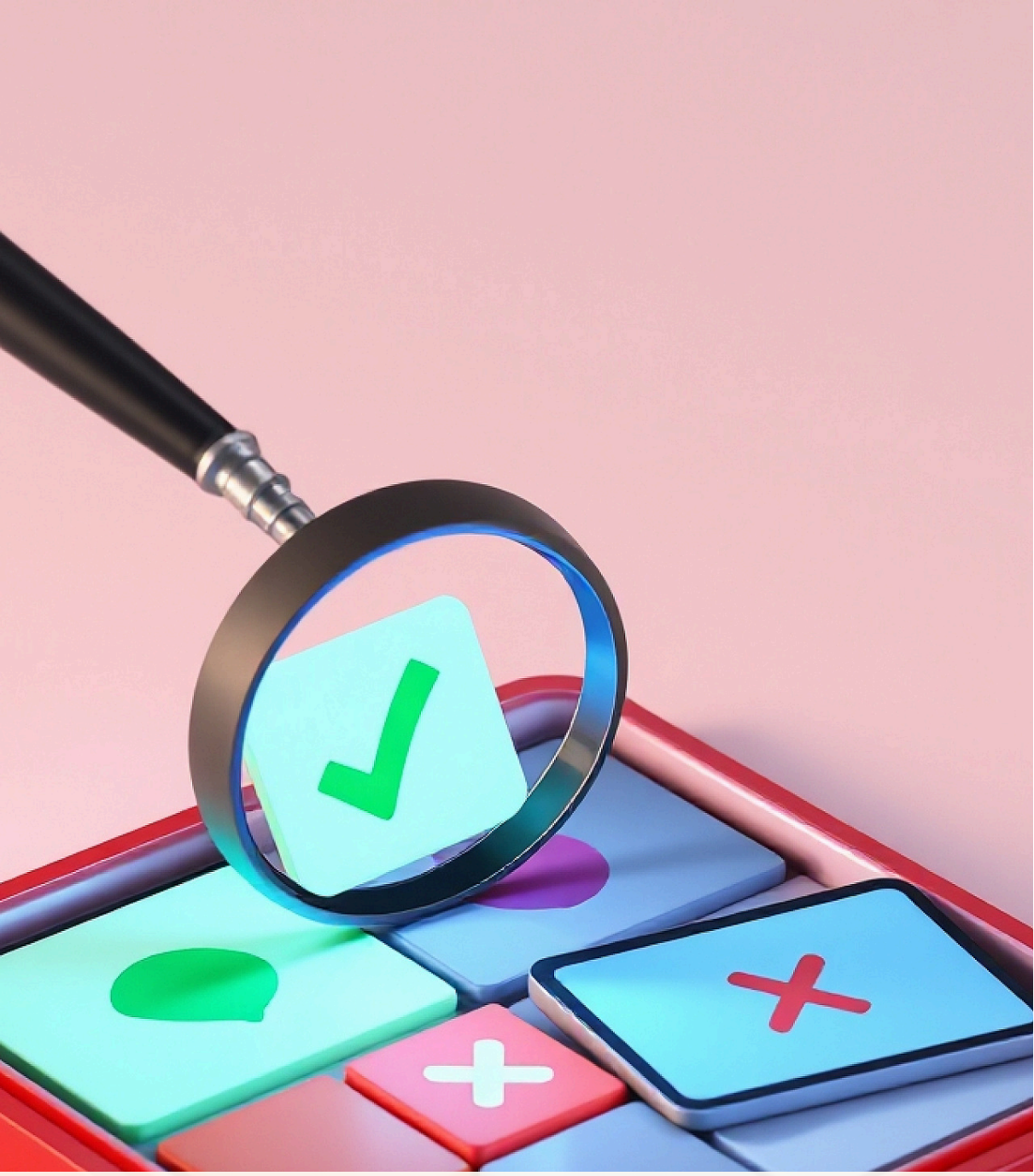
Bir program çoğu zaman durur ve bir şey girmeni bekler. Ancak girdi geldikten sonra devam eder. Örneğin bir giriş ekranı, kullanıcı adını ve parolanı yazıp giriş düğmesine basana kadar bekler. O zamana kadar başka bir şey yapmaz; yalnızca girdinin gelmesini bekler.



Çoğu girdi, rakam yazsan bile önce metin olarak gelir. Program bu aşamada sayıları değil karakterleri alır. Örneğin yaşın olarak 25 yazarsan program önce yirmi beş sayısını değil, 2 ve 5 karakterlerini görür. Bu ayrım önemlidir çünkü program bu karakterleri sayı olarak kullanmak için sonra yorumlamalıdır.



Veriyi almak ve onu anlamak iki ayrı adımdır. Önce program girdiyi alır, sonra girdinin ne anlama geldiğini çözer. Örneğin bir program senden merhaba metnini alabilir ama bunun bir selamlama olduğuna ancak sonra karar verebilir. Bu adımları ayrı tutmak, programın girdiyi doğru işlemesine ve karışıklıktan kaçınmasına yardımcı olur.



Veriyi kullanmadan önce program, girdinin ihtiyacı olan şey için mantıklı olup olmadığını denetler. Bu denetim, program veriyi kullanmadan önce yapılır. Örneğin program yaş için bir sayı bekliyorsa girdinin yalnızca rakam içerdiğini ve harf olmadığını denetler. Denetim başarısız olursa program kötü veriyi kullanmak yerine doğru girdi isteyebilir.



Yaşını metin olarak yazdığın bir örneği ele alalım. Program bu metni bir hesaplamada kullanılabilmesi için sayıya dönüştürür. Örneğin 25 yazarsan program karakterleri 25 sayısına çevirir. Bu dönüşüm gereklidir çünkü metin karakterlerini toplayamaz veya çıkaramazsın; matematik yapmak için gerçek sayılara ihtiyacın vardır.



Dönüştürüldükten sonra sayı, yıl eklemek gibi gerçek bir hesaplamada kullanılabilir. Bu, dönüşüm adımının neden önemli olduğunu gösterir. Örneğin program 25 olan yaşını alıp 10 ekleyerek on yıl sonraki yaşını bulabilir. Metni önce sayıya çevirmeden program bu toplamayı doğru yapamazdı.



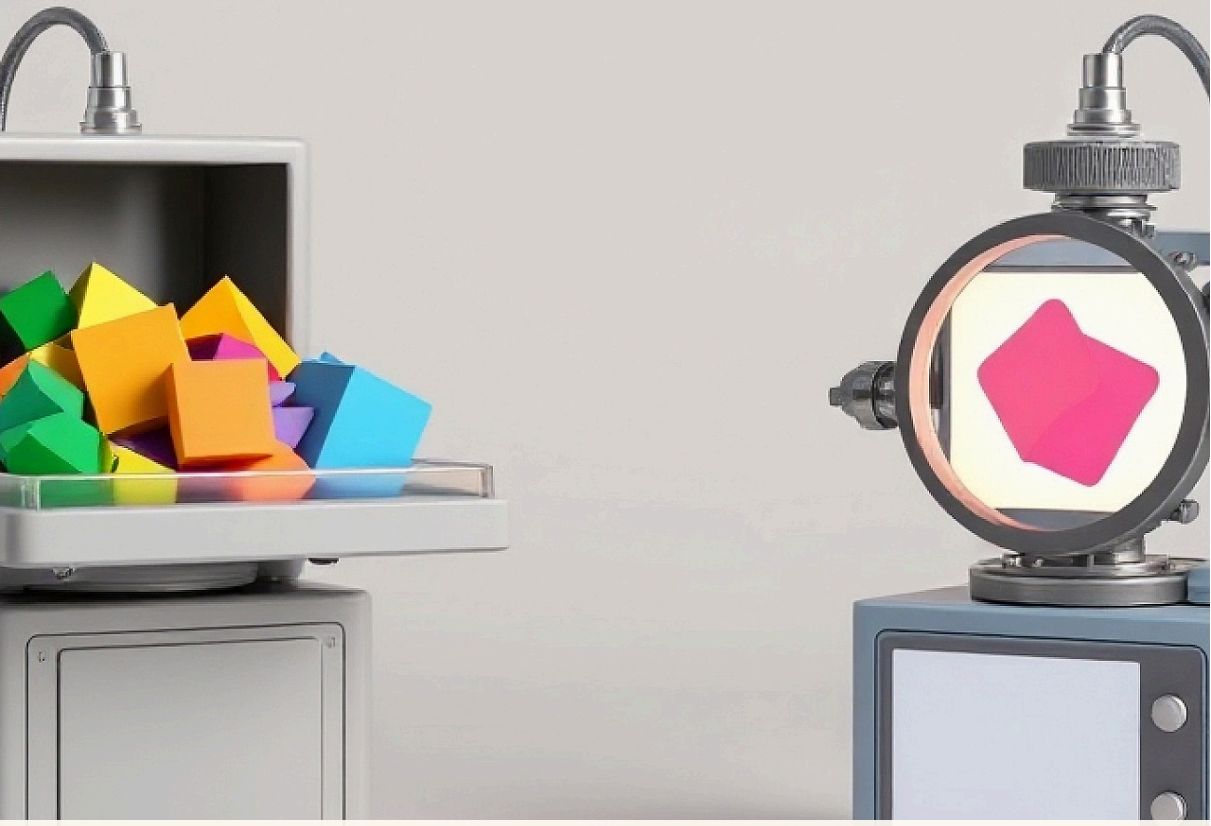
Kullanıcı hiçbir şey yazmayıp boş girdi gönderebilir. Program bunu fark etmeli ve çökmek yerine bununla başa çıkmalıdır. Örneğin bir form adını isteyip onu boş bırakıp gönder düğmesine basarsan program girdinin boş olduğunu algılamalıdır. Ardından alanı doldurmanı isteyen bir mesaj gösterebilir.



Kullanıcı, sayı beklenen yerde harf yazabilir. Program bu yanlış girdiyi algılamalı ve mantıklı bir yanıt vermelidir. Örneğin bir program yaşını sorup sen yirmi yazarsan program yirmi'nin rakamlarla geçerli bir sayı olmadığını fark etmelidir. Ardından sessizce başarısız olmak veya yanlış sonuç vermek yerine yalnızca rakam girmeni söyleyebilir.



Dikkatli bir program, girdi boş veya yanlış olduğunda tekrar denemeni isteyebilir. Bu, programın iyi veriyle çalışmasını sağlar. Örneğin bir hatayı algıladıktan sonra program lütfen geçerli bir yaş gir gibi bir mesaj gösterip girdi alanını yeniden gösterebilir. Sen de girdini düzeltebilirsin ve program ancak veri kabul edilebilir olduğunda devam eder.



Arayüzden veri almak ve onu doğru yorumlamak her zaman ayrı adımlardır. Bunları ayrı tutmak programları daha açık ve daha güvenli yapar. Örneğin programın tuş vuruşlarını okuyan kısmı, girdinin geçerli bir yaş olup olmadığına karar vermez; bunu başka bir kısım denetler ve dönüştürür. Bu ayrım hatalardan kaçınmaya yardımcı olur ve programı düzeltmeyi kolaylaştırır.



Yaygın bir hata, girdinin her zaman doğru olduğunu varsaymaktır. Bir diğeri denetimi atlamak veya metni sayıya çevirmeyi unutmaktır. Örneğin bir program 5 metni ile 3 metnini dönüşüm yapmadan toplarsa 8 elde etmek yerine onları 53 olarak birleştirebilir. Bu hatalardan kaçınmak programı güvenilir tutar ve yanlış sonuçları önler.



Özetle girdi klavye gibi araçlarla gelir, sonra program onu denetler ve dönüştürür. Girdinin boş veya yanlış olabileceğini ve almanın ile yorumlamanın ayrı adımlar olduğunu unutma. Örneğin yaşını metin olarak yazdığında program onu alır, rakam olduğunu denetler, sayıya çevirir ve ancak ondan sonra bir hesaplamada kullanır.