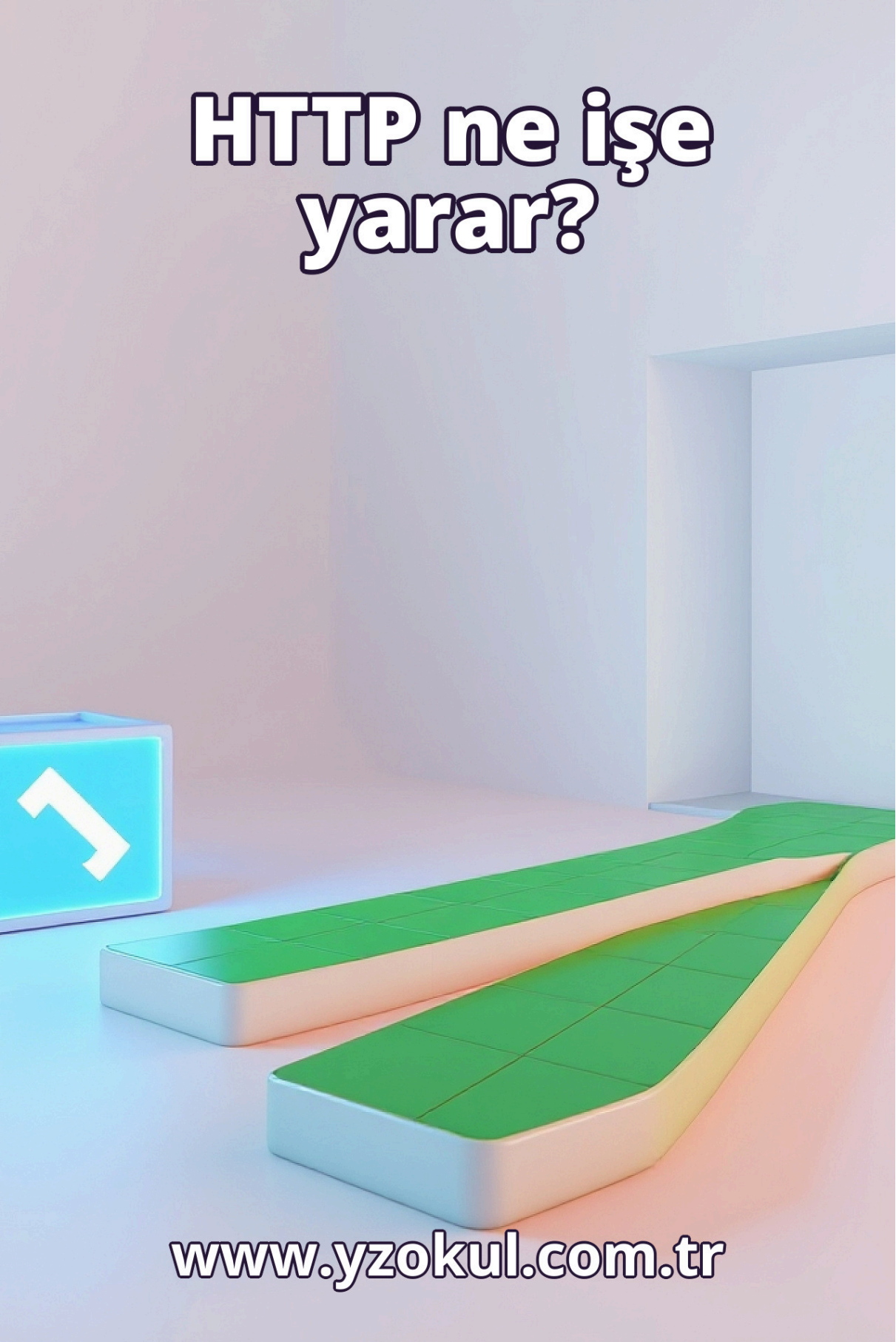
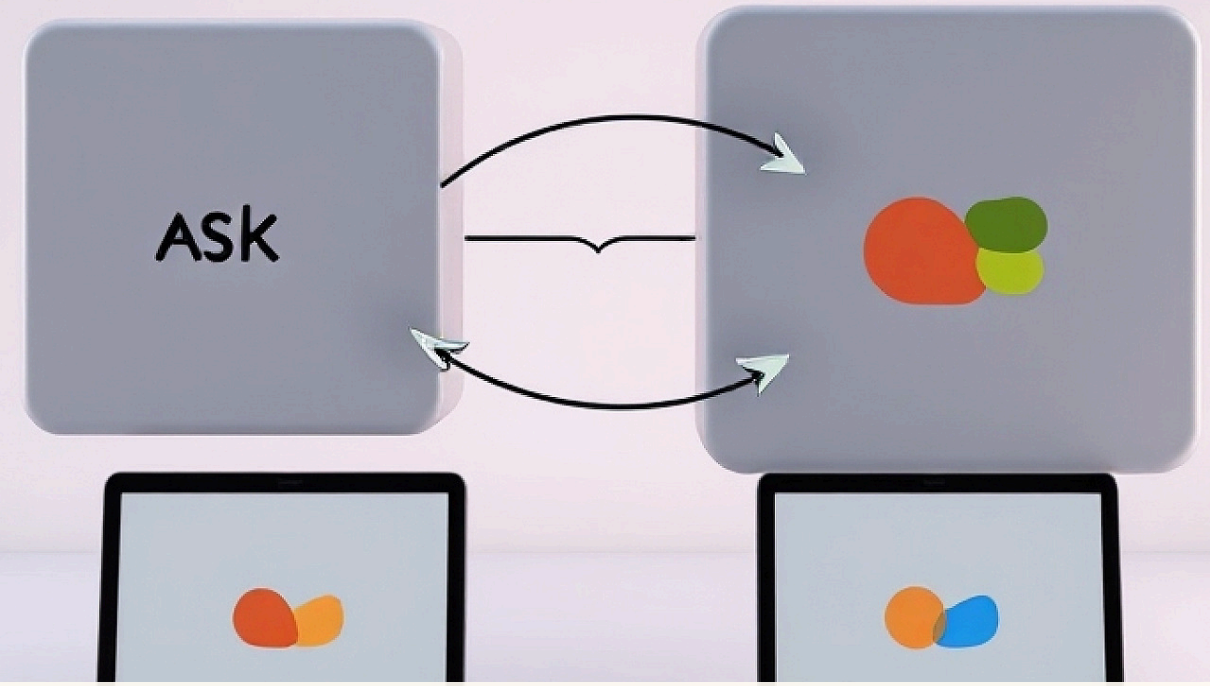
HTTP ne işe yarar?



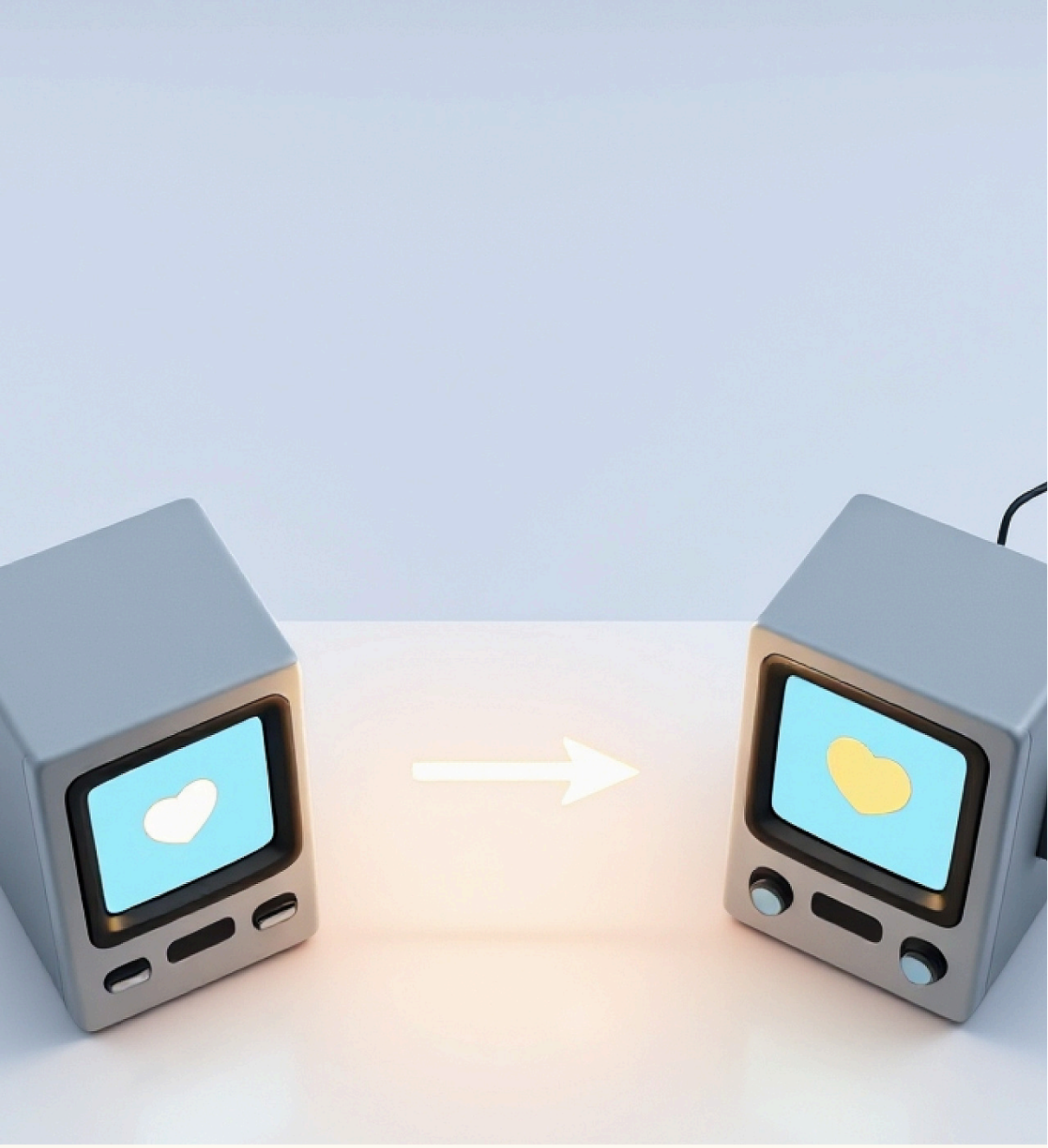
www.yzokul.com.tr

HTTP ne işe yarar?

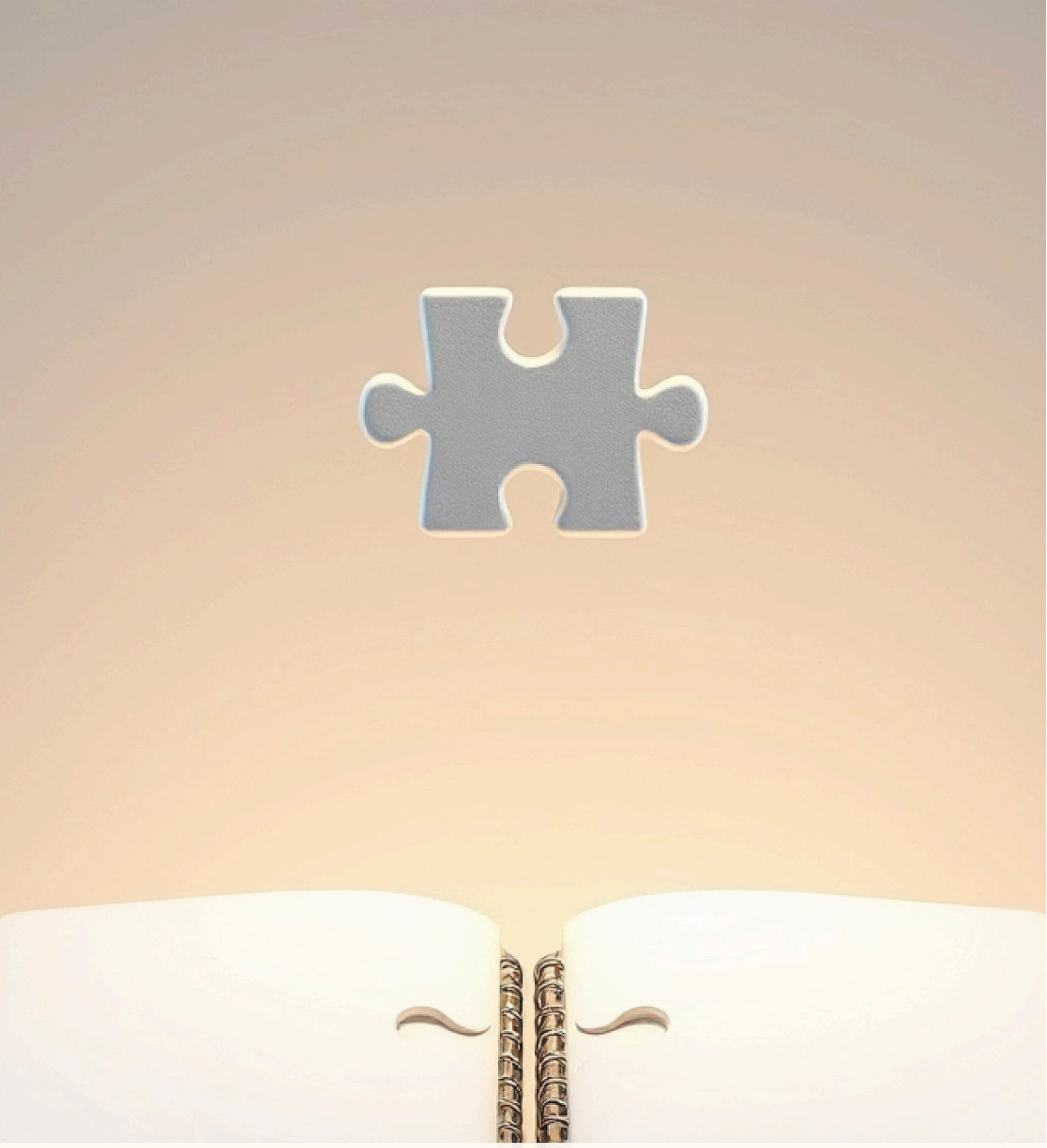
www.yzokul.com.tr



HTTP, bir istemci ile bir sunucu arasında istek ve yanıt alışverişi için kullanılan bir protokoldür. Bu tek cümle, kitabın küçük bir özetidir: istemci sorar, sunucu yanıtlar ve HTTP bu alışverişin kurallarını belirler. Bu kitap, HTTP'nin ne işe yaradığını görebilmen için her parçayı adım adım açıklar. Sonunda isteklerin, yanıtların ve durum kodlarının nasıl birlikte çalıştığını anlayacaksın.



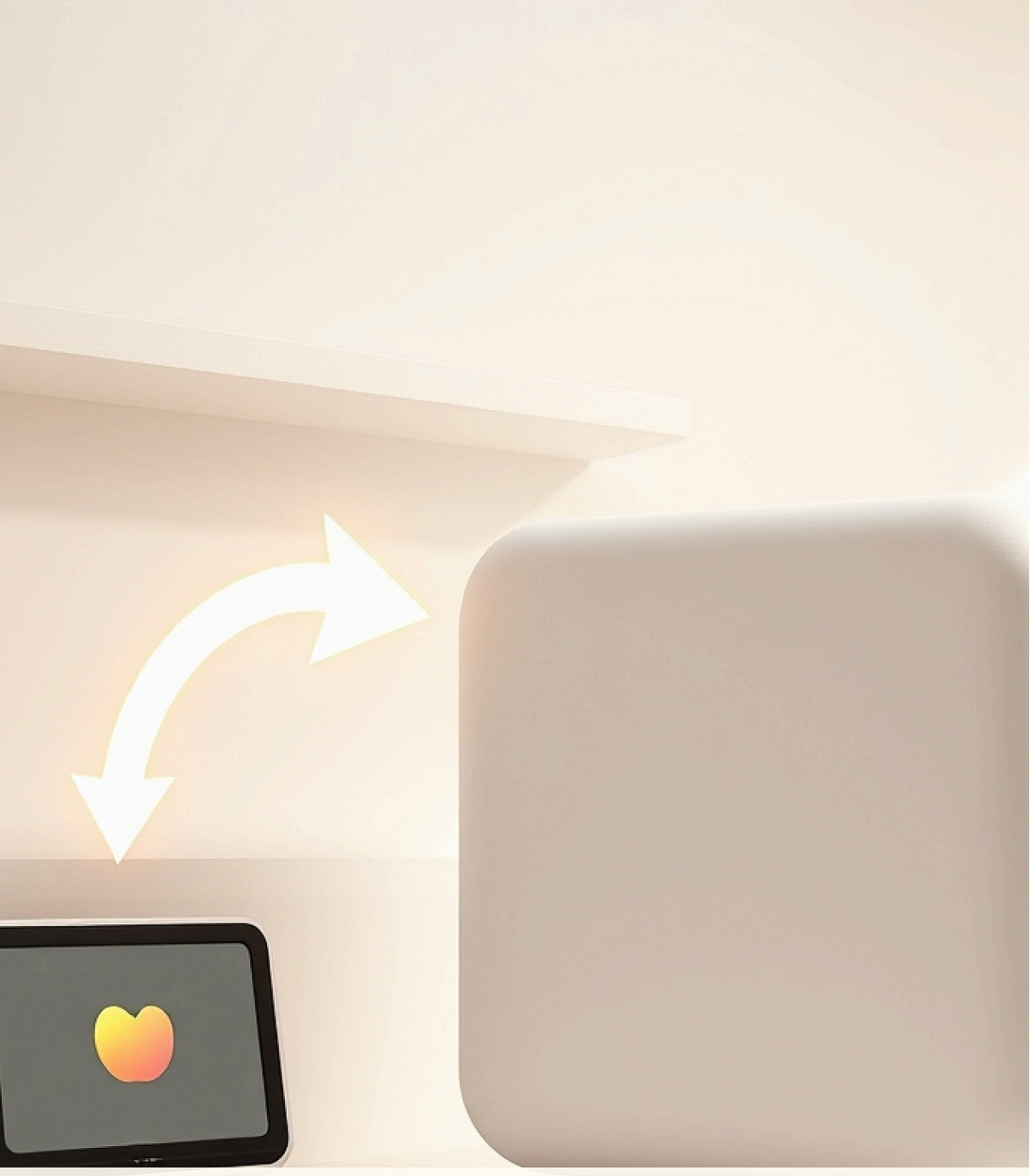
İki bilgisayar düşün: biri soru sorar, diğeri yanıtlar. Bu basit karşılıklı gidiş geliş, her HTTP alışverişinin özüdür. Soru soran bilgisayar bir mesaj gönderir, yanıtlayan bilgisayar da yanıt verir. HTTP, her iki tarafın da mesajları ikisinin de anlayacağı biçimde göndermesini ve okumasını sağlar. Bu düzen, bir sayfayı her yüklediğinde ya da bir uygulama veri çektiğinde tekrarlanır.



Protokol, konuşmak için üzerinde anlaşılmış bir kurallar kümesidir. İki bilgisayar HTTP kullandığında, birbirini anlamaları için aynı kurallara uyar. Örneğin bir kural, bir isteğin nasıl biçimlendirileceğini söyler; bir başkası da bir yanıtın nasıl biçimlendirileceğini söyler. Ortak kurallar olmadan bilgisayarlar, diğerinin mesajlarının ne anlama geldiğini bilemezdi. HTTP işte o ortak kural kitabıdır.



İstemci, bir istek göndererek işi başlatan taraftır. Bilgisayarındaki tarayıcı yaygın bir istemcidir. Bir web adresi yazdığında tarayıcı bir HTTP isteği oluşturur ve gönderir. İstek, bir sayfa ya da görsel gibi belirli bir şeyi ister. İstemci, sunucunun önce konuşmasını beklemez; alışverişi o başlatır.



Sunucu, bekleyen, isteği alan ve geri bir yanıt gönderen taraftır. İstemcinin istediği içeriği, örneğin web sayfalarını, görselleri ya da verileri barındırır. Bir istek geldiğinde sunucu onu okur ve ne döndüreceğine karar verir. Sonra istemciye bir yanıt gönderir. Sunucu konuşmayı başlatmaz; sorulduğunda yanıtlar.



İstemci, bir web sayfası ya da görsel gibi belirli bir kaynak ister. İstek, neyin istendiğini belirtir; böylece sunucu tam olarak neyi arayacağını bilir. Örneğin istek, belirli bir adresteki sayfayı isteyebilir. Kaynak bir dosya, bir resim ya da bir uygulamadan gelen veri olabilir. Kaynağı adlandırmak, istemcinin sunucuya neyi geri göndermesi gerektiğini söyleme biçimidir.



Sunucu, istenen içeriği yanıtında geri gönderir. Bir sayfa ya da dosya tarayıcıya gerçekte böyle ulaşır. Yanıt, içeriğin kendisini, örneğin bir web sayfasının metnini ve görsellerini taşır. Ekranında bir sayfa belirdiğini gördüğünde, sunucu tam da o içeriği döndürmüştür. İstemci onu alır ve senin için görüntüler.



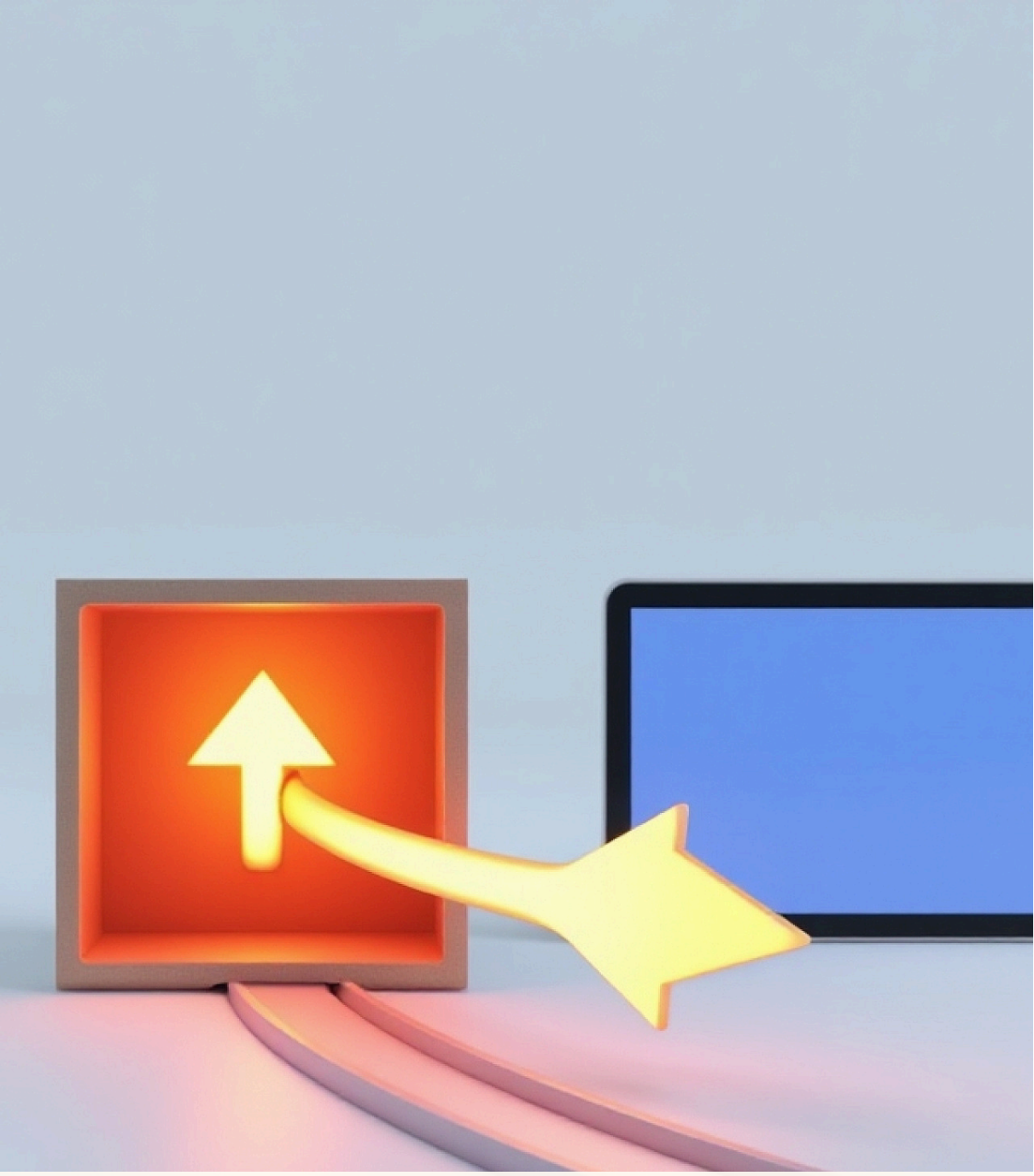
Her yanıt, işin nasıl gittiğini söyleyen bir durum kodu taşır. Kod, sonucu özetleyen kısa bir sayıdır. Örneğin bir kod, isteğin başarılı olduğunu ya da kaynağın bulunamadığını söyleyebilir. İstemci, içeriğe bakmadan önce ne olduğunu bilmek için kodu okur. Durum kodları, bir isteğin sonucunu bildirmenin hızlı bir yoludur.



Sayfa bulunduğunda sunucu, başarı durum koduyla yanıt verir. İçerik de onunla birlikte gelir. Örneğin sunucu, başarı anlamına gelen 200 sayısını döndürebilir ve ardından sayfanın kendisini gönderebilir. İstemci başarı kodunu görür ve içeriğin geçerli olduğunu bilir. Her şey beklendiği gibi çalıştığında bu normal durumdur.



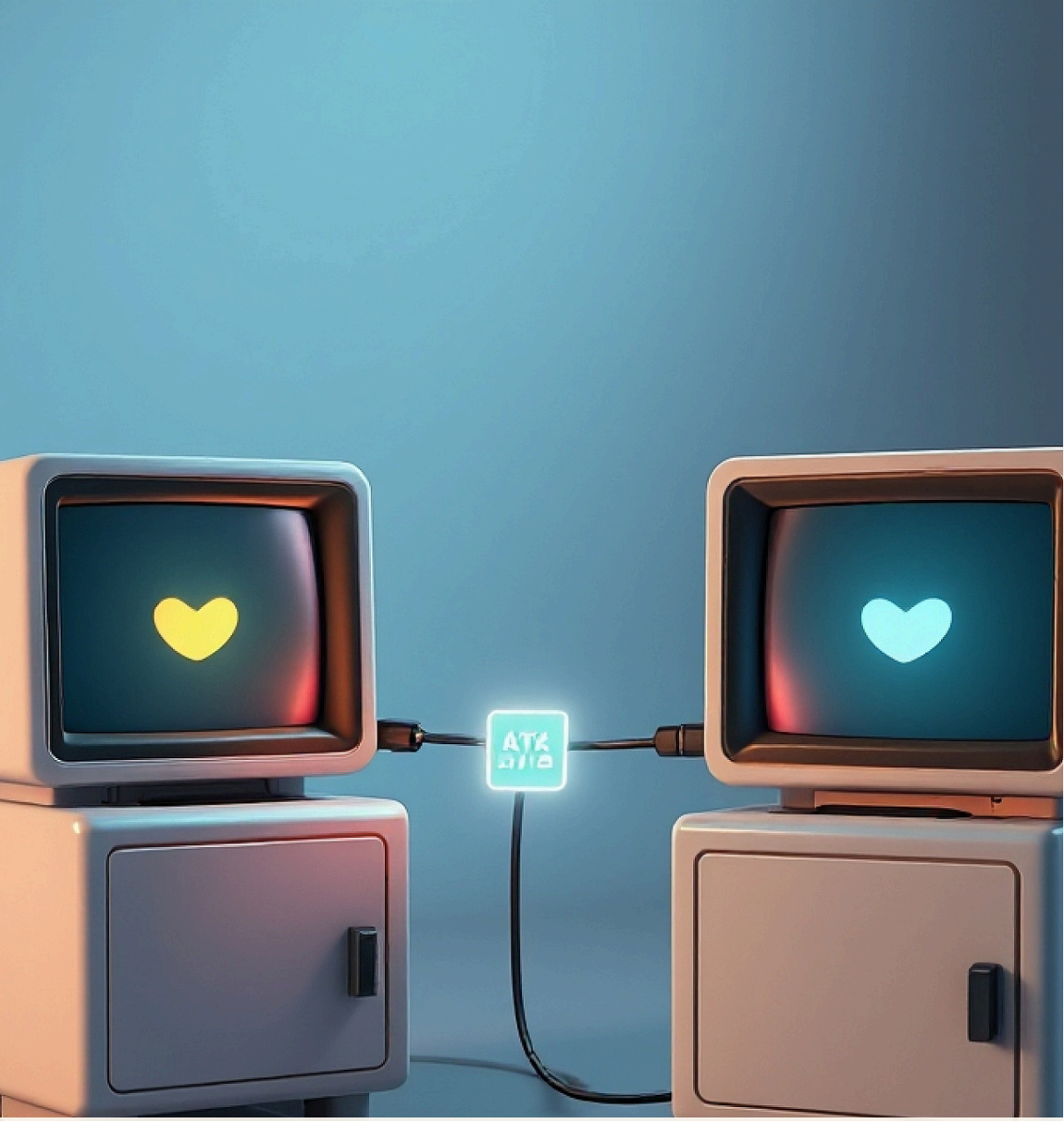
Sayfa yoksa sunucu, bulunamadı durum koduyla yanıt verir. Kod, istemciye ne olduğunu söyler. Örneğin sunucu, bulunamadı anlamına gelen 404 sayısını döndürebilir. İstemci bu kodu görür ve istenen kaynağın var olmadığını bilir. Sunucu yine de kısa bir mesaj gönderebilir, ama asıl işaret koddur.



Her alışveriş, bir isteği izleyen bir yanıttır. Bu eşleşme, HTTP'nin temel birimidir. İstemci tam olarak bir istek gönderir ve sunucu tam olarak bir yanıt gönderir. Sonra alışveriş tamamlanır. İstemcinin daha fazlasına ihtiyacı varsa başka bir istek gönderir. Bu basit düzen, yüklediğin her kaynak için tekrarlanır.



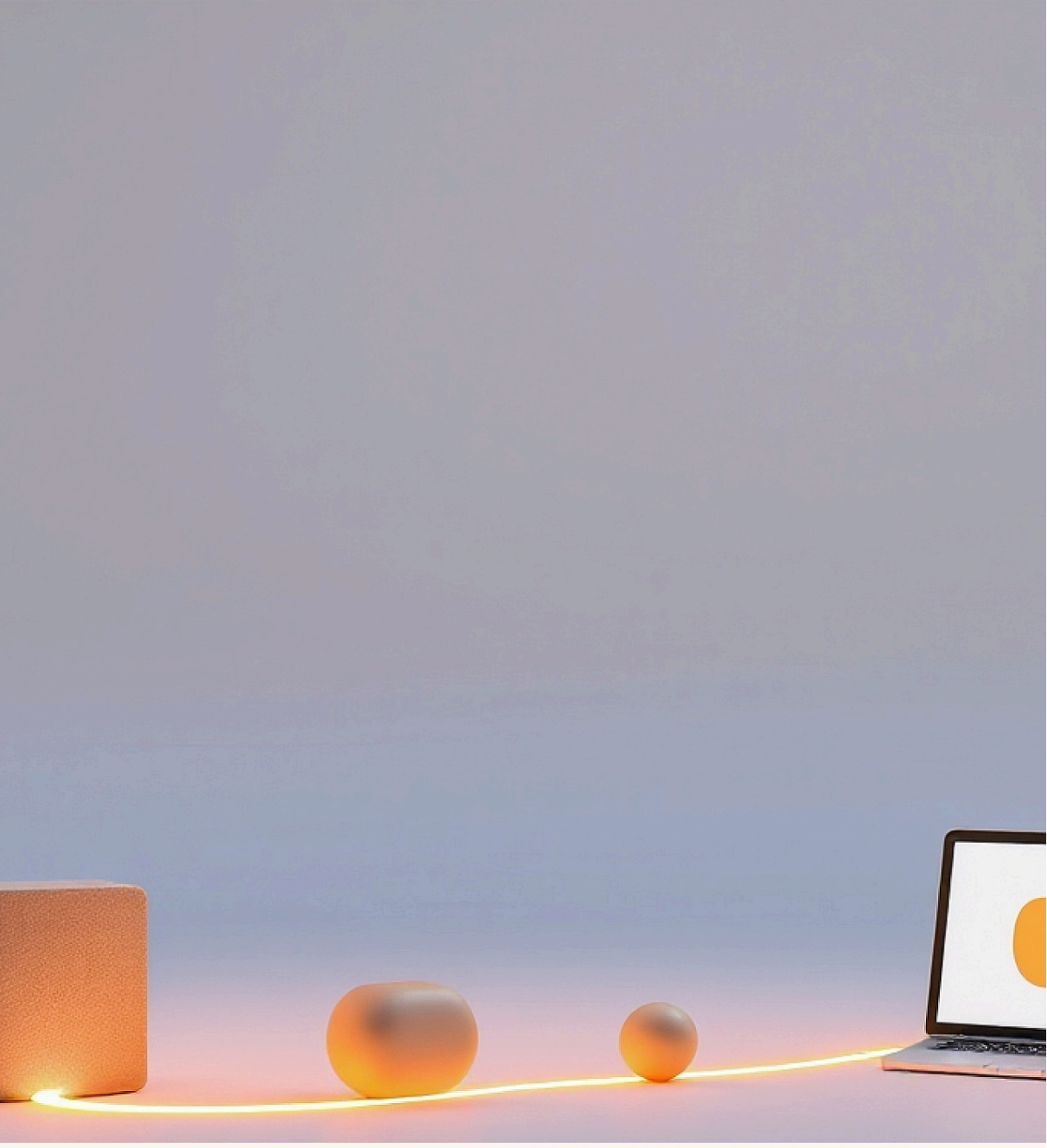
HTTP yalnızca sayfa göstermek için değildir. Uygulamalar arasında veri aktarmak için de yaygın olarak kullanılır. Örneğin telefonundaki bir hava durumu uygulaması, tahmini bir sunucudan çekmek için HTTP kullanabilir. Aynı istek ve yanıt düzeni geçerlidir, ama içerik bir web sayfası değil veridir. HTTP, sayfaları taşıdığı gibi veriyi de taşır.



Bir kişi ile bir tarayıcının değil, iki uygulamanın HTTP üzerinden veri alışverişi yaptığını düşün. Bir uygulama istemci olarak davranır ve bir istek gönderir; diğeri sunucu olarak davranır ve bir yanıt gönderir. Örneğin bir alışveriş uygulaması, bir mağazanın sunucusundan ürün bilgisi isteyebilir. Aynı kurallar geçerlidir: istek, yanıt ve durum kodları. HTTP, uygulamalar için de tarayıcılar için olduğu gibi çalışır.



Yaygın bir yanılgı, HTTP'nin bir web sitesi ya da bir sayfa olduğunu düşünmektir. HTTP, istekleri ve yanıtları taşımak için kullanılan kurallar kümesidir, içeriğin kendisi değil. Web sitesi içeriktir; HTTP onu taşıyan protokoldür. HTTP'yi yalnızca web sayfalarını değil, her tür içeriği taşımak için kullanabilirsin. Kuralları ve içeriği zihninde ayrı tut.



Bir başka yaygın yanılgı, HTTP'nin yalnızca tarayıcılarla çalıştığını düşünmektir. Kurallara uyan her istemci ve sunucu, veri paylaşan uygulamalar dahil, onu kullanabilir. Bir mobil uygulama, bir masaüstü programı ya da başka bir sunucuyla konuşan bir sunucu, hepsi HTTP kullanabilir. Tarayıcı yalnızca yaygın bir istemcidir. Protokol, kurallara uyduğu sürece istemcinin kim olduğuyla ilgilenmez.



HTTP, bir istemci ile bir sunucu arasındaki istekleri ve yanıtları düzenler. İstemci bir kaynak ister ve sunucu, sonucu söyleyen bir durum koduyla birlikte içerik döndürür. HTTP ayrıca yalnızca web sayfaları değil, uygulamalar arasında veri de taşır. Şu temel noktaları hatırla: istek, yanıt, durum kodu ve veri aktarımı. HTTP'nin işlevi budur.