Unicode nedir?



www.yzokul.com.tr

Unicode nedir?



Şöyle soruyorsun: Unicode nedir? Kısa yanıt, Unicode'un her karaktere ortak bir kimlik veren bir standart olduğu, böylece bilgisayarların metni güvenilir biçimde saklayıp değiştirebildiğidir. Bu kitap bu fikri adım adım açıklar; bilgisayarların harfler için neden sayılara ihtiyaç duyduğuyla başlar ve ekrandaki tek bir işaretin nasıl birden fazla Unicode öğesi olabileceğiyle biter. Örnekler görecek ve yaygın hatalardan kaçınacaksın.



Bir bilgisayar bir harfi doğrudan tutamaz; yalnızca sayıları saklayabilir ve işleyebilir. Bu yüzden bir harf, rakam veya simge olsun, her karakterin bir makinenin kaydedebilmesi veya gönderebilmesi için önce bir sayıya dönüştürülmesi gerekir. Örneğin, A harfini saklamak için bilgisayar A'yı temsil eden bir sayı kaydeder. Bu sayıdan karaktere eşleme, tüm metin işlemenin temelidir.



Ortak bir standart yokken iki bilgisayar farklı sayı-harf tabloları kullanabilirdi. Aynı sayı bir makinede A, diğerinde başka bir şey olarak görünebilirdi. Metin sistemler arasında taşındığında bozulup okunamaz hale gelebilirdi. Hangi sayının hangi karakteri ifade ettiğine dair ortak bir kural olmadan metnin güvenilir biçimde değişimi zordu. Bir standart, eşlemeyi her yerde aynı yaparak bunu çözer.



Dünyada birçok yazı sistemi vardır: Latin, Türk harfleri, Arap, Kiril, Çince ve daha birçokları. Her sistemin kendi karakterleri vardır ve her birinin temsil edilmesi gerekir. İnsanlar ayrıca noktalama işaretleri, rakamlar, para birimi simgeleri, oklar ve emoji kullanır. Herhangi bir dilden metnin çakışma olmadan saklanabilmesi ve paylaşılabilmesi için bunların hepsinin tek bir sistemde bir arada bulunması gerekir.



Unicode, her yazı sistemindeki her karakterin kendi benzersiz kimliğini aldığı tek büyük bir listedir. Hiçbir iki karakter aynı kimlik numarasını paylaşmaz. Bu ortak liste, belirli bir karakterin Unicode'un kullanıldığı her yerde aynı kimliğe sahip olduğu anlamına gelir. Örneğin, Latin büyük harfi A'nın sabit bir kimliği vardır ve Türk harfi ş'nin de öyledir. Liste tüm yazıları ve birçok simgeyi kapsar.



ş, ğ, ı ve İ gibi Türk harfleri kendi kimlikleriyle ortak listeye yerleştirilir. Diğer harflere sıkıştırılmazlar; her birinin sabit bir yeri vardır. Örneğin, noktasız ı noktalı i'den farklıdır ve her birinin kendi kimlik numarası vardır. Bu, Türkçe metnin doğru temsil edilmesini ve anlam kaybı olmadan değiştirilebilmesini sağlar.



Unicode yalnızca harflerle ilgili değildir. Noktalama işaretleri, rakamlar, para birimi simgeleri, oklar ve emoji de aynı ortak listede kimliklere sahiptir. Örneğin, virgül, 7 rakamı, dolar işareti ve gülen yüz emojisinin her birinin benzersiz bir kimliği vardır. Hepsi tek bir sistemde bir arada durur, böylece bir belge metni ve simgeleri sorunsuzca karıştırabilir.



Bir karakterin kimliği soyut bir etikettir, diskte saklanan baytlar değildir. Kimlik karakterin ne olduğunu söyler, baytlar ise bir dosyada nasıl yazıldığını veya bir ağ üzerinden nasıl gönderildiğini söyler. Örneğin, A harfinin kimliği bir sayıdır, ancak onu saklamak için kullanılan baytlar kodlamaya bağlıdır. Kimliği baytlardan ayrı tutmak Unicode'u anlamak için önemlidir.



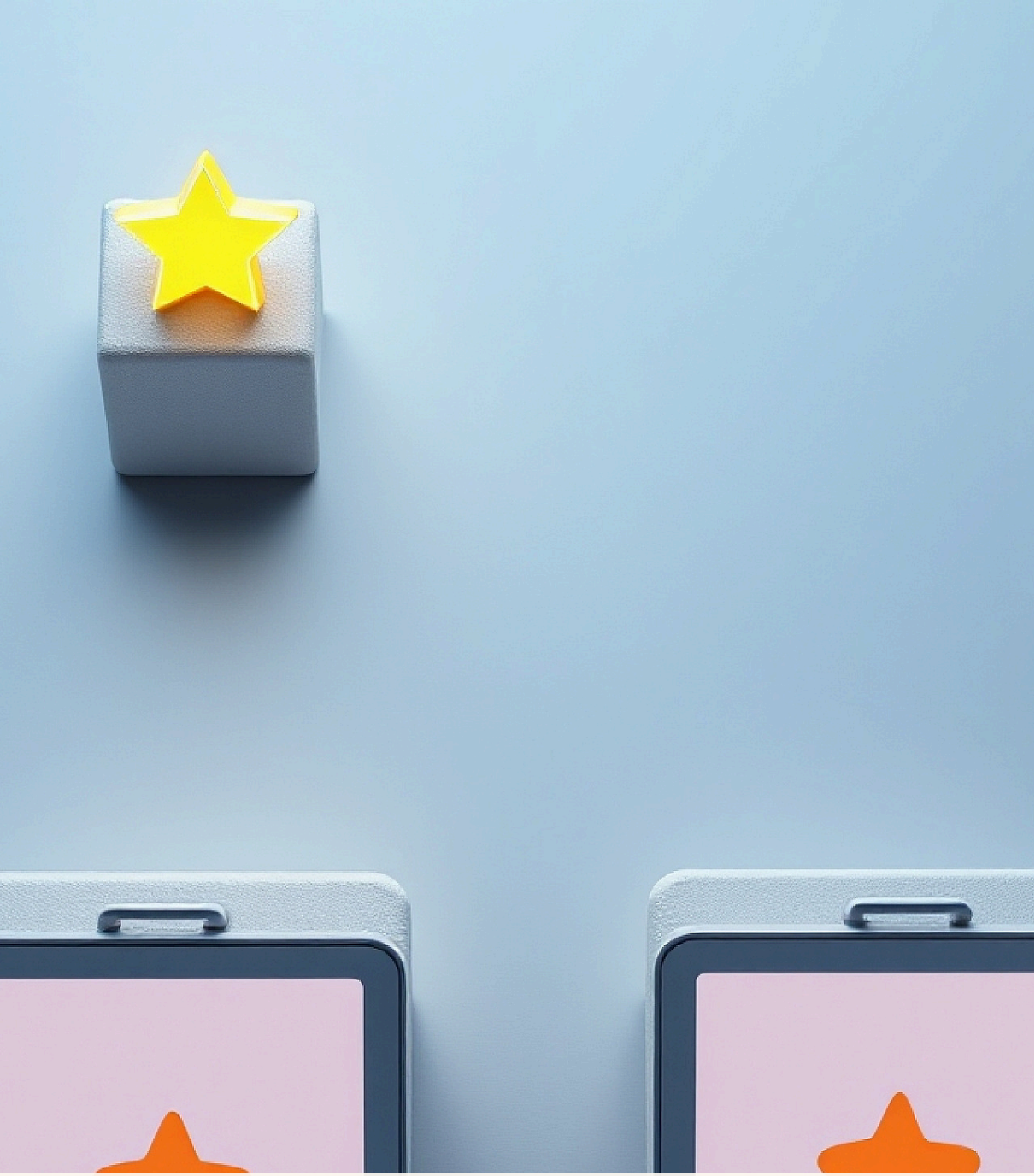
Kodlama, bir karakterin kimliğini bir bayt dizisine dönüştüren kuraldır. Kimlik aynı kalır, ancak baytlar hangi kodlamanın kullanıldığına bağlıdır. Örneğin, bir kodlama A'nın kimliğini tek bir bayta dönüştürebilirken bir diğeri farklı bir bayt dizisi kullanabilir. Kodlamalar, Unicode metnini bilgisayarlarda saklamayı ve iletmeyi mümkün kılar.



UTF-8, Unicode kimliklerini baytlara dönüştüren yaygın bir kodlamadır. Karaktere bağlı olarak karakter başına bir ila dört bayt kullanır. Örneğin, A harfi UTF-8'de tek bayt olan 0x41 olur. ş gibi bir karakter iki bayt alır. Bu değişken uzunluk, UTF-8'i çoğunlukla İngilizce olan metinler için verimli kılarken tüm Unicode karakterlerini de destekler.



Unicode ve UTF-8 aynı şey değildir. Unicode kimliklerin listesidir ve UTF-8 bu kimlikleri bayt olarak saklamanın bir yoludur. Bunları karıştırmak yaygın bir hatadır. Örneğin, bir dosyanın Unicode'da olduğunu söylemek belirsizdir; UTF-8'de olduğunu söylemek kesindir. İki fikri ayrı tut: Unicode karakterleri tanımlar, UTF-8 onları kodlar.



Aynı karakter farklı kodlamalar altında farklı bayt dizileri olarak saklanabilir. Örneğin, A harfi UTF-8'de bir bayt olabilirken UTF-16'da iki bayt olabilir. Bu, metin paylaşılırken kodlamanın her iki tarafta da eşleşmesi gerektiğini gösterir. UTF-8 baytları gönderip alıcı UTF-16 beklerse metin bozulur. Kodlamayı her zaman belirt.



Ekrandaki tek bir görünür işaret, birleştirilmiş birden fazla Unicode öğesinden oluşturulabilir. Örneğin, e harfinin ardından gelen birleştirici keskin aksan é olarak, tek bir şekil olarak görünebilir. Unicode listesi hem temel harfi hem de aksan işaretini ayrı öğeler olarak içerir. Yazılım bunları görüntülemek için birleştirir, ancak alttaki metin iki öğeye sahiptir.



Her görünür işaretin tam olarak bir Unicode öğesine eşit olduğunu varsayma. Bazı işaretler bir temel karakter ile bir veya daha fazla birleştirici işaretin birleşimidir. Örneğin, çengelli bir harf iki öğe olabilir: harf ve çengel. Öğeleri saymak, gördüğünü saymakla aynı şey değildir. Bu, metni programlı olarak işlerken önemlidir.



Bir baytın her zaman bir harf anlamına geldiğini varsayma. UTF-8'de tek bir karakter bir ila birkaç bayt alabilir. Örneğin, A harfi bir bayttır, ancak ş iki bayttır ve bir emoji dört bayt olabilir. Yani bayt sayıları ile harf sayıları farklıdır. Metin boyutunu ölçerken baytların karakter olmadığını unutma.



Üç temel noktayı hatırla: Unicode karakterlere tüm yazı sistemlerinde ortak kimlikler verir; UTF-8 gibi kodlamalar bu kimlikleri bayt olarak saklar; ve ikisi aynı şey değildir. Ayrıca tek bir görünür işaretin birkaç Unicode öğesi olabileceğini de hatırla. Bu bilgiyle metnin nasıl temsil edildiğini anlayabilir ve farklı diller ve simgelerle çalışırken yaygın hatalardan kaçınabilirsin.