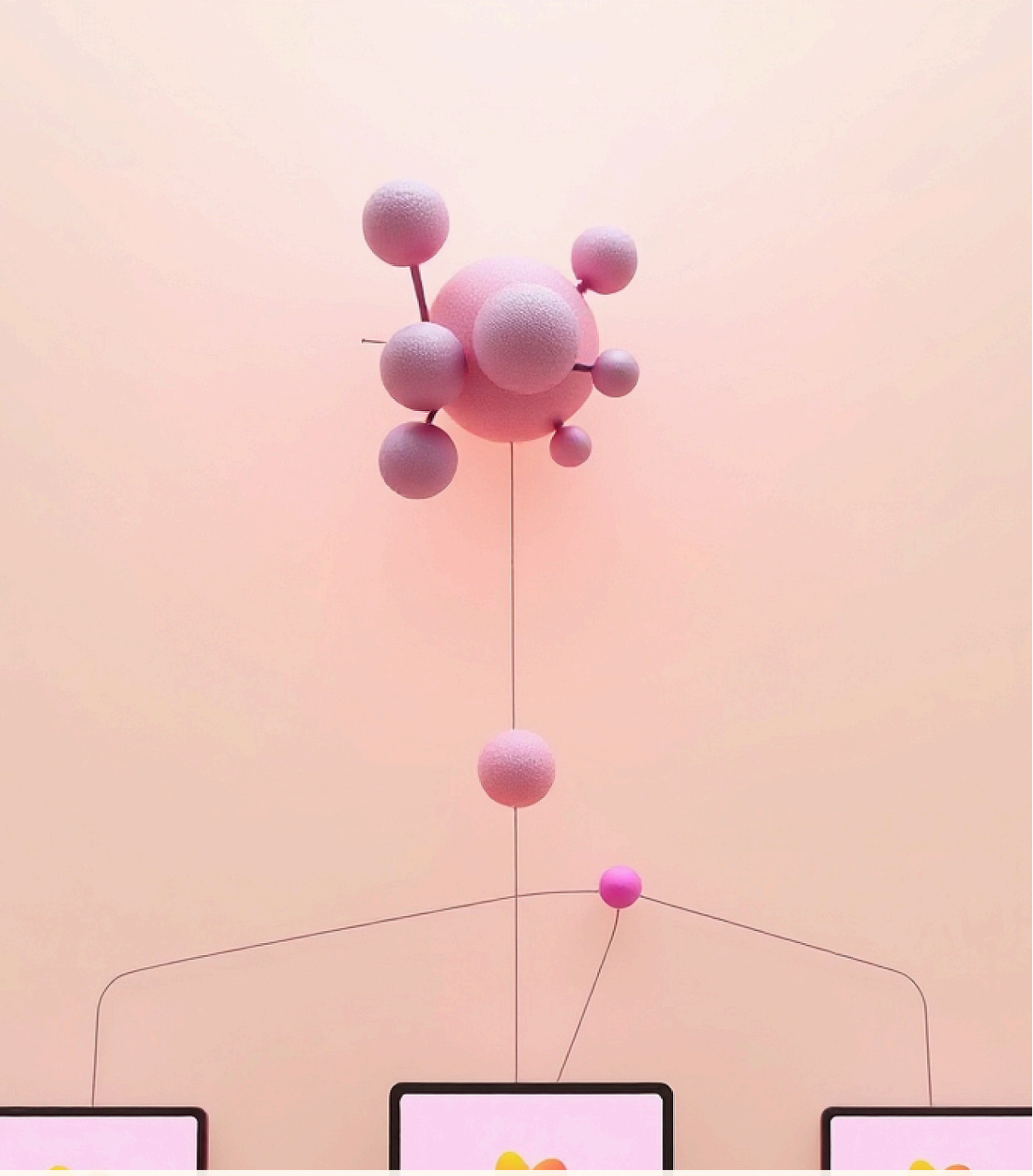
# Ağaç veri yapısı nedir?



www.yzokul.com.tr

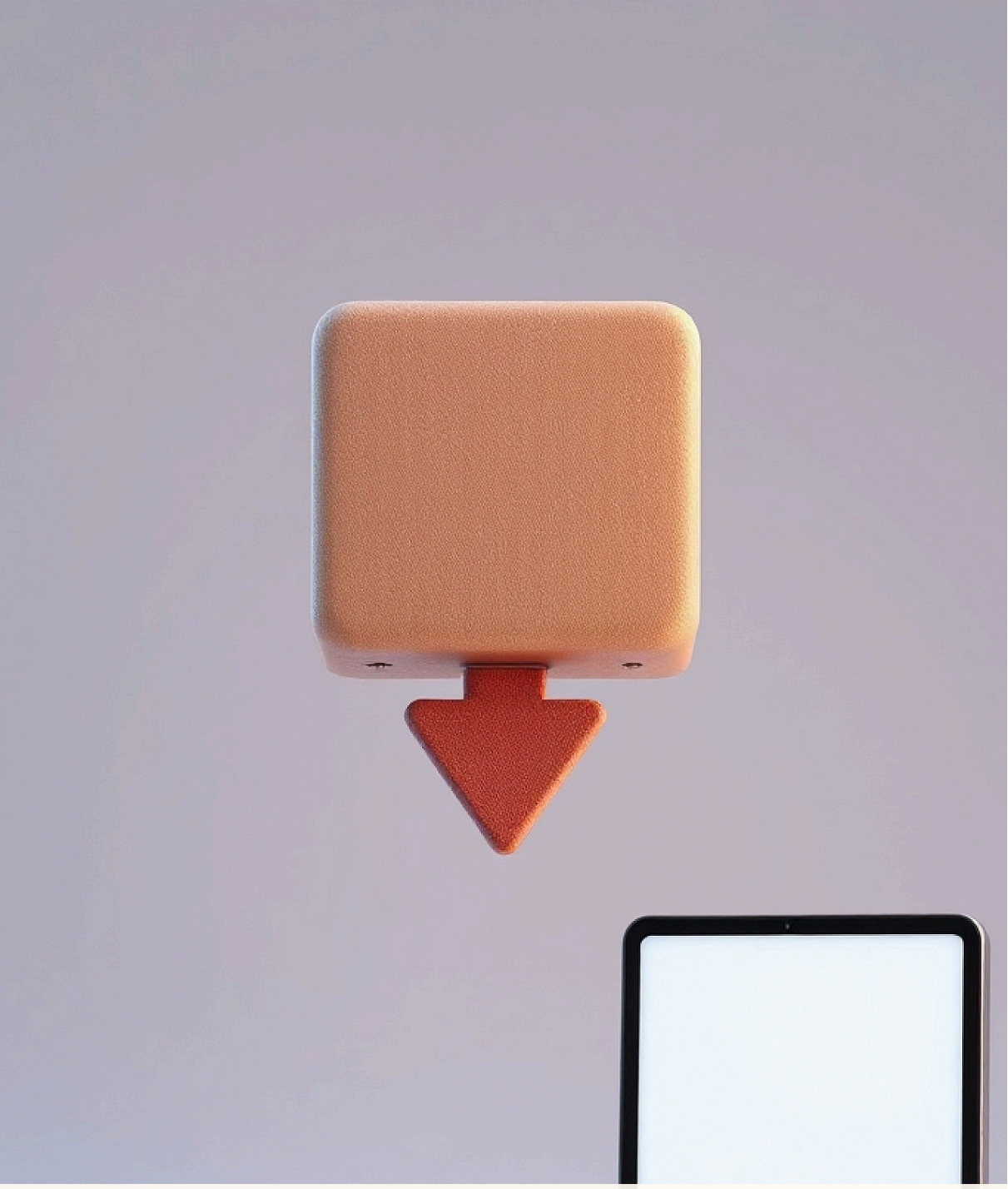
# Ağaç veri yapısı nedir?



Ağaç veri yapısı, öğeleri her biri diğerlerine ebeveyn ya da çocuk olarak bağlanacak şekilde düzenlemenin bir yoludur. Kısa yanıt: öğeler, kök adı verilen tek bir üst öğeden başlayarak ebeveyn-çocuk bağlantılarıyla birbirine bağlanır. Bu, ayrıntılara geçmeden önce sana bütün fikri verir. Bunu, net bağlantıları olan bir öğeler ailesi olarak düşün.



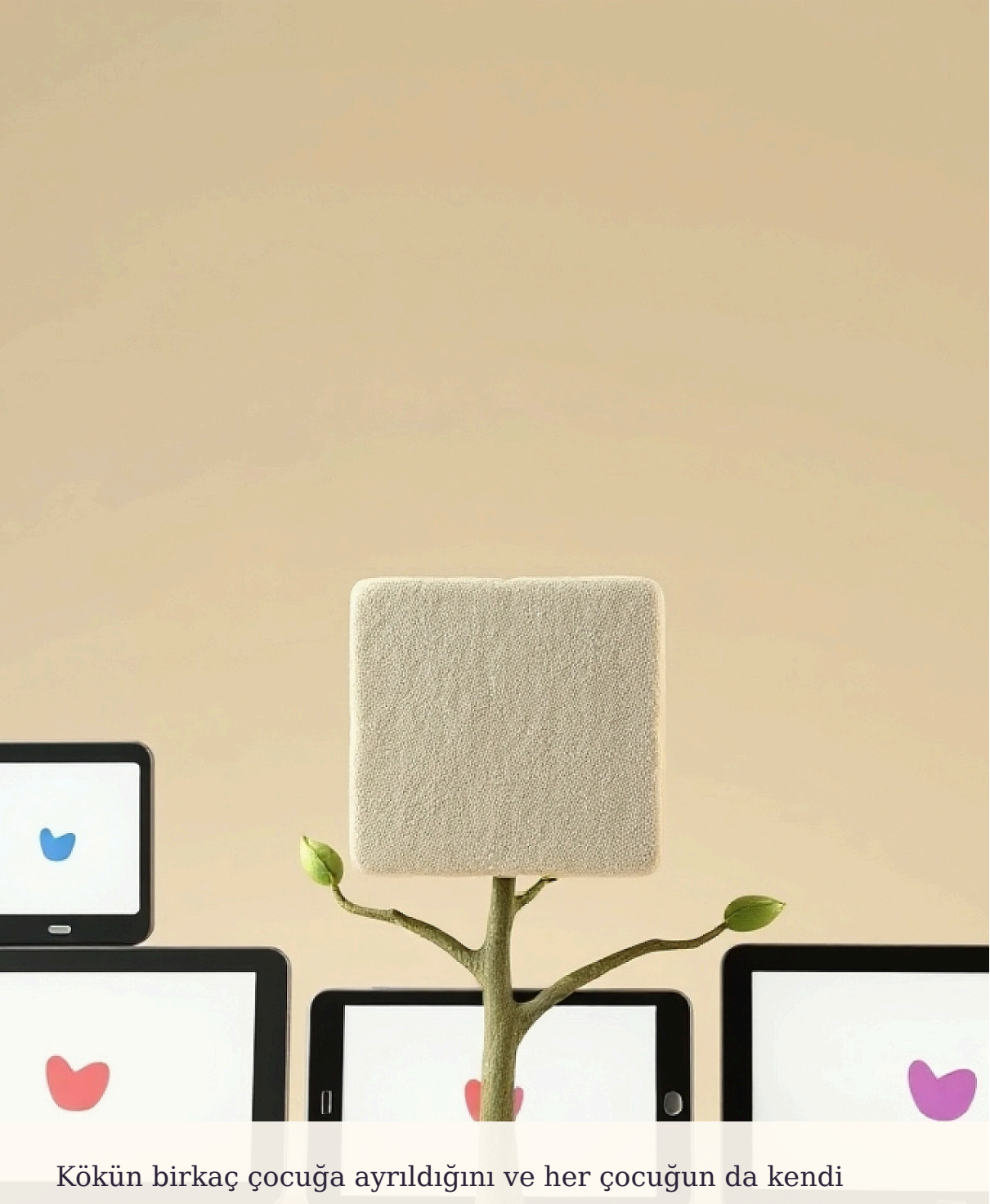
Birkaç öğenin çizgilerle birleştiğini hayal et; her çizgi bir öğeyi bir diğerine bağlar. Ağaç, öğelerin yalnızca bir listesi değil, nasıl bağlandıklarıyla ilgilidir. Bağlantılar bir şekil oluşturur ve bu şekil önemlidir. Örneğin, A öğesi B ve C öğelerine bağlanır, B de D öğesine bağlanır. Bağlantılar yapının kendisidir.



İki öğe arasındaki tek bir bağlantıya bak. Üstteki öğeye ebeveyn, alttaki öğeye çocuk denir. Her bağlantı tek yönlüdür, ebeveynden çocuğa doğru iner. Örneğin, A klasörü B klasörünü içeriyorsa A ebeveyn, B çocuktur. Aynı bağlantıdan asla yukarı çıkmazsın.



En üstte, üzerinde ebeveyni olmayan tek öğeyi bul. Bu üst öğeye kök düğüm denir ve her ağaç ondan başlar. Kök, tüm yapının giriş noktasıdır. Örneğin, bir bilgisayarda ana sürücü klasörü kök görevi görebilir. Diğer tüm öğelere oradan ulaşılabilir.



Kökün birkaç çocuğa ayrıldığını ve her çocuğun da kendi çocuklarına ayrıldığını izle. Şekil dallar gibi yayılır; bu yüzden ona ağaç denir. Her dal, ebeveyn-çocuk bağlantılarından oluşan bir yoldur. Örneğin, kökün iki çocuğu vardır ve bunların her birinin ikişer çocuğu daha vardır; bu da küçük, dallanan bir ağaç oluşturur.



Dalları izleyerek en altta, çocuğu olmayan öğelere ulaş. Bu uç öğelere yaprak denir ve her yol orada durur. Yapraklar bir daldaki son öğelerdir. Örneğin, bir klasör ağacında en derin klasörlerin içindeki dosyalar yapraktır, çünkü başka hiçbir şey içermezler. Her ağacın en az bir yaprağı vardır.



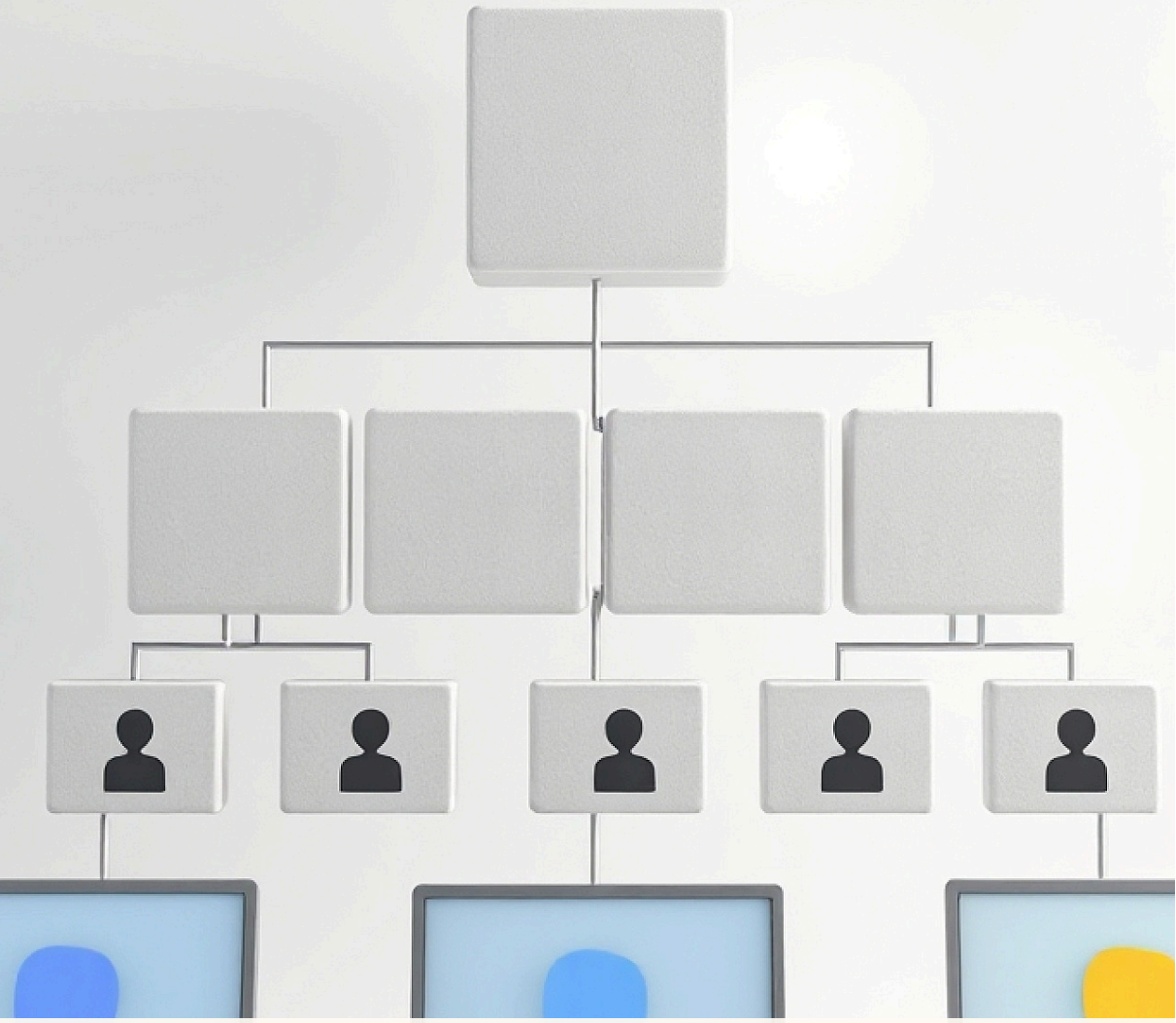
Ağacı bilgisayardaki klasörlerle karşılaştır: bir ana klasör alt klasörleri, onlar da daha fazla alt klasörü barındırır. Az önce öğrendiğin aynı ebeveyn-çocuk şeklini tanırsın. Ana klasör kök gibidir, alt klasörler de çocuk düğümler gibidir. Örneğin, bir Belgeler klasörü bir İş klasörünü, o da bir Raporlar klasörünü barındırır. Bu tanıdık bir ağaçtır.



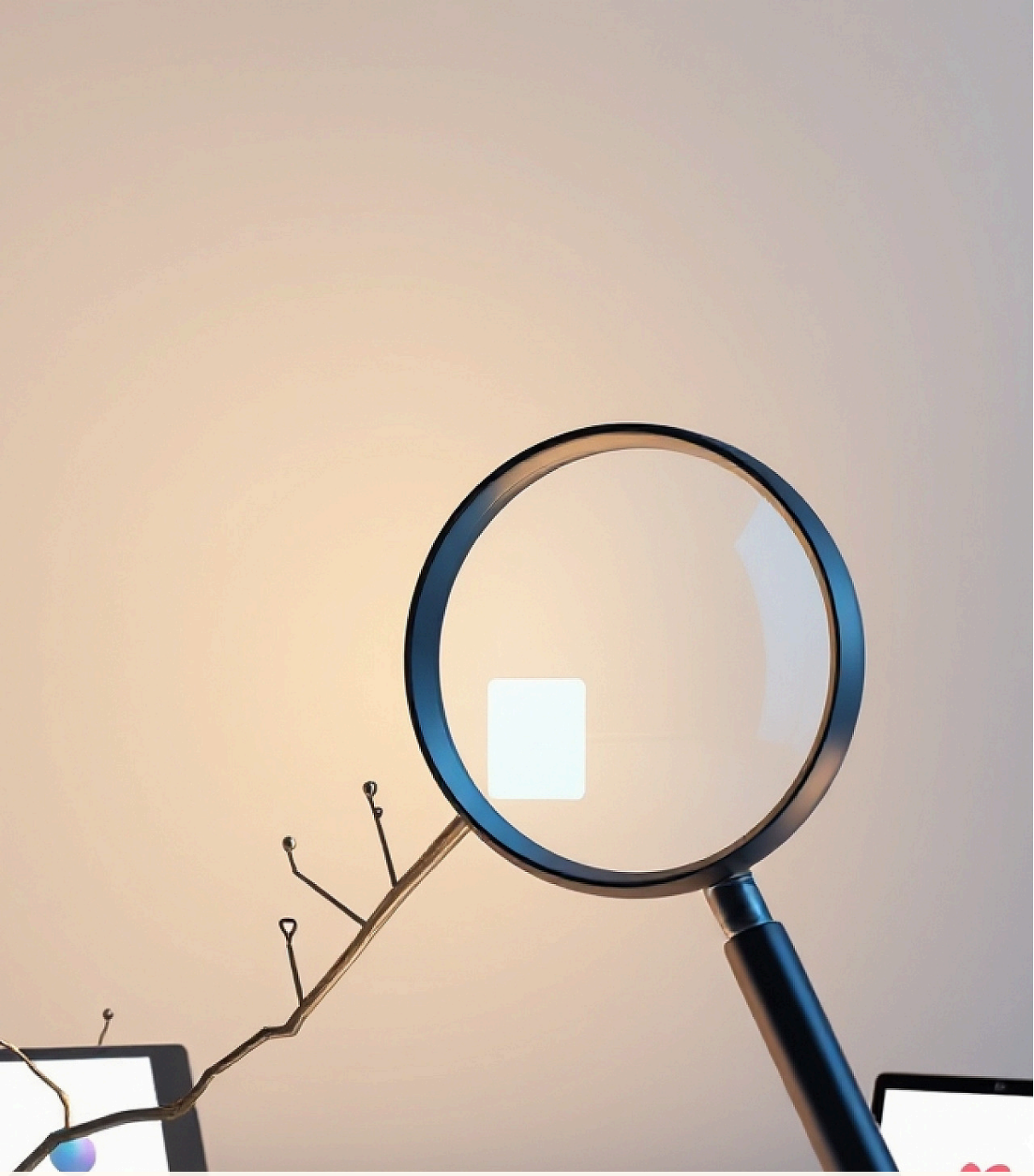
Bir klasörü, sonra içindeki bir alt klasörü, sonra onun içindekini aç. Tıpkı bir dalı izler gibi her seferinde bir seviye aşağı inersin. Her açma, o anki klasörün çocuklarını gösterir. Örneğin, Belgeler'i, sonra İş'i, sonra Raporlar'ı aç. Ebeveynden çocuğa, oradan bir çocuğa daha giden bir yolu izlemiş oldun.



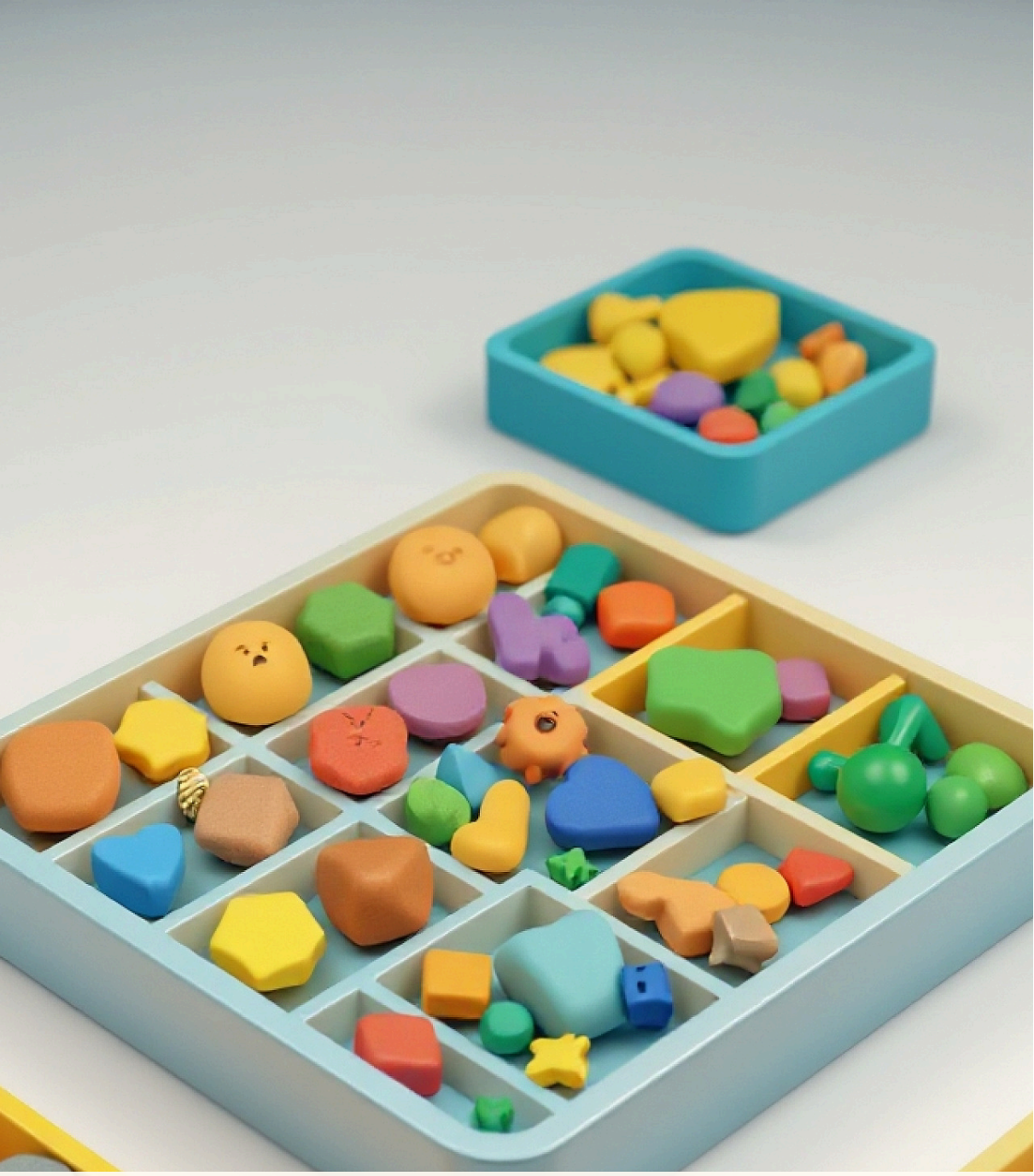
Öğeleri ne kadar derinde durduklarına göre sırala: kök birinci seviyede, çocukları ikinci seviyede ve böyle devam eder. Bir ağaç bilgiyi seviyeler halinde saklar ve bu düzenlemeye hiyerarşi denir. Her seviye, bir öğenin kökten ne kadar uzakta olduğunu gösterir. Örneğin, kök birinci seviyedir, çocukları ikinci seviyedir, onların çocukları da üçüncü seviyedir.



Bir aile ağacı ya da bazı şeylerin diğerlerinin üzerinde durduğu bir şirket şeması gibi örneklere bak. Ağaçlar, kimin kime bağlı olduğunu ya da kimin kimden geldiğini gösterdiği için hiyerarşik bilgiyi saklamanın doğal bir yoludur. Örneğin, bir şirket şemasında en üstte bir patron, altında yöneticiler, onların altında da personel vardır. Seviyeler yapıyı netleştirir.



Bir öğeyi bulmak için kökten başla ve her seferinde bir dalı kontrol et. Bir ağaç, ihtiyacın olmayan bütün dalları atlayabildiğin için aramaya yardımcı olur. Öğe bir dalda yoksa bir sonrakine geçersin. Örneğin, bir dosyayı bulmak için önce olası klasörü açarsın ve ilgisiz klasörleri yok sayarsın; bu da zaman kazandırır.



Öğeleri gruplara, sonra bunların içine alt gruplara yerleştir; tıpkı hayvanları türlerine göre sınıflandırmak gibi. Ağaçlar, benzer şeyleri bir araya getirerek sınıflandırma için kullanılır. Her grup bir ebeveyndir, üyeleri de çocuklarıdır. Örneğin, hayvanlar memelilere ve kuşlara ayrılır, memeliler de köpeklere ve kedilere ayrılır. Bu gruplama bir ağaç oluşturur.



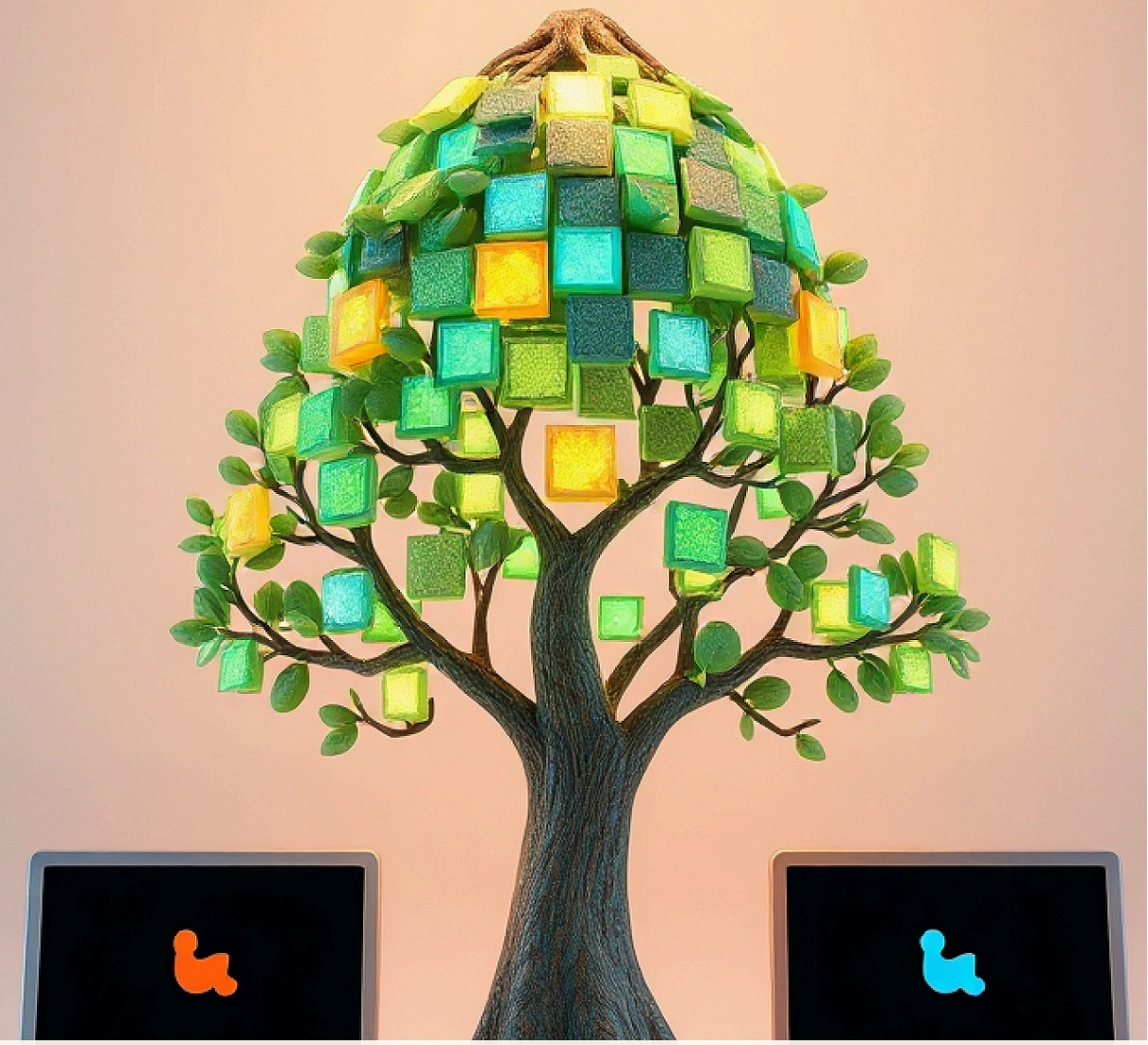
İki ağacı karşılaştır ve birinin çocukları sırayla tuttuğunu, diğerinin tutmadığını fark et. Her ağaç türünün kendi erişim kuralları vardır, bu yüzden davranış farklı olabilir. Bazı ağaçlar hızlı aramaya izin verir, bazıları ise gruplamaya odaklanır. Örneğin, sıralı bir ağaç sol çocukları daha küçük, sağ çocukları daha büyük tutar; sırasız bir ağaç ise sırayı önemsemez.



Birinin bir ağacı düz bir öğe çizgisi gibi ele aldığını gör. Bunu düzelt: bir ağaçta bir öğenin birçok çocuğu olabilir, bu yüzden bir sıra halinde kalmak yerine dallanır. Bir listede öğeler birbiri ardına gelir, ama bir ağaç yayılır. Örneğin, üç çocuğu olan bir kök bir liste değildir; dallanan bir yapıdır.



Bir öğenin üzerinde iki ebeveyni olduğu bir çizim gör; bu bir ağaç değildir. Bir ağaçta her öğenin tam olarak bir ebeveyni olduğunu, ebeveyni olmayan tek öğenin ise kök olduğunu öğren. İki ebeveyn bir ağaç değil, bir ağ oluşturur. Örneğin, B klasörü hem A klasörünün hem de C klasörünün içindeyse bu bir ağaç değildir.



Yanıtı özetle: bir ağaç, öğeleri tek bir kökten ebeveynler ve çocuklar olarak bağlar ve seviyelere dallanır. Örnek olarak bilgisayardaki klasörleri hatırla ve kuralların ağaç türüne göre değiştiğini unutma. Kök en üsttedir, yapraklar uçlardadır ve diğer her öğenin bir ebeveyni vardır. Bu yapı hiyerarşik bilgiyi saklar ve arama ile sınıflandırmaya yardımcı olur.