Listelerde konumlar neden bazen sıfırdan başlar?



www.yzokul.com.tr

Listelerde konumlar neden bazen sıfırdan başlar?



Şu soruyla karşılaşırsın: liste konumları neden bazen sıfırdan başlar? Kısa yanıt, ilk öğenin indeksinin sıfır olmasıdır çünkü bu, öğenin başlangıçtan kaç adım uzakta olduğunu gösterir. Bu fikre sıfır tabanlı indeksleme denir. Bu, bir listedeki konumları numaralandırmanın bir yoludur ve birçok programlama dilinde kullanılır. Bunu anlamak, listelerle çalışırken kafa karışıklığından kaçınmana yardımcı olur.



Bir liste, bir sıra öğedir; içinde şeyler olan beş kutu gibi. Her kutunun bir konumu vardır ve her kutuyu işaret etmek için bir sayı kullanırız. Örneğin, bir meyve listesinde ilk kutuda bir elma, ikincide bir muz olabilir ve bu böyle devam eder. Konum numarası, belirli bir öğeyi nasıl bulduğumuzdur. Listeler programlamada ve günlük yaşamda yaygındır.



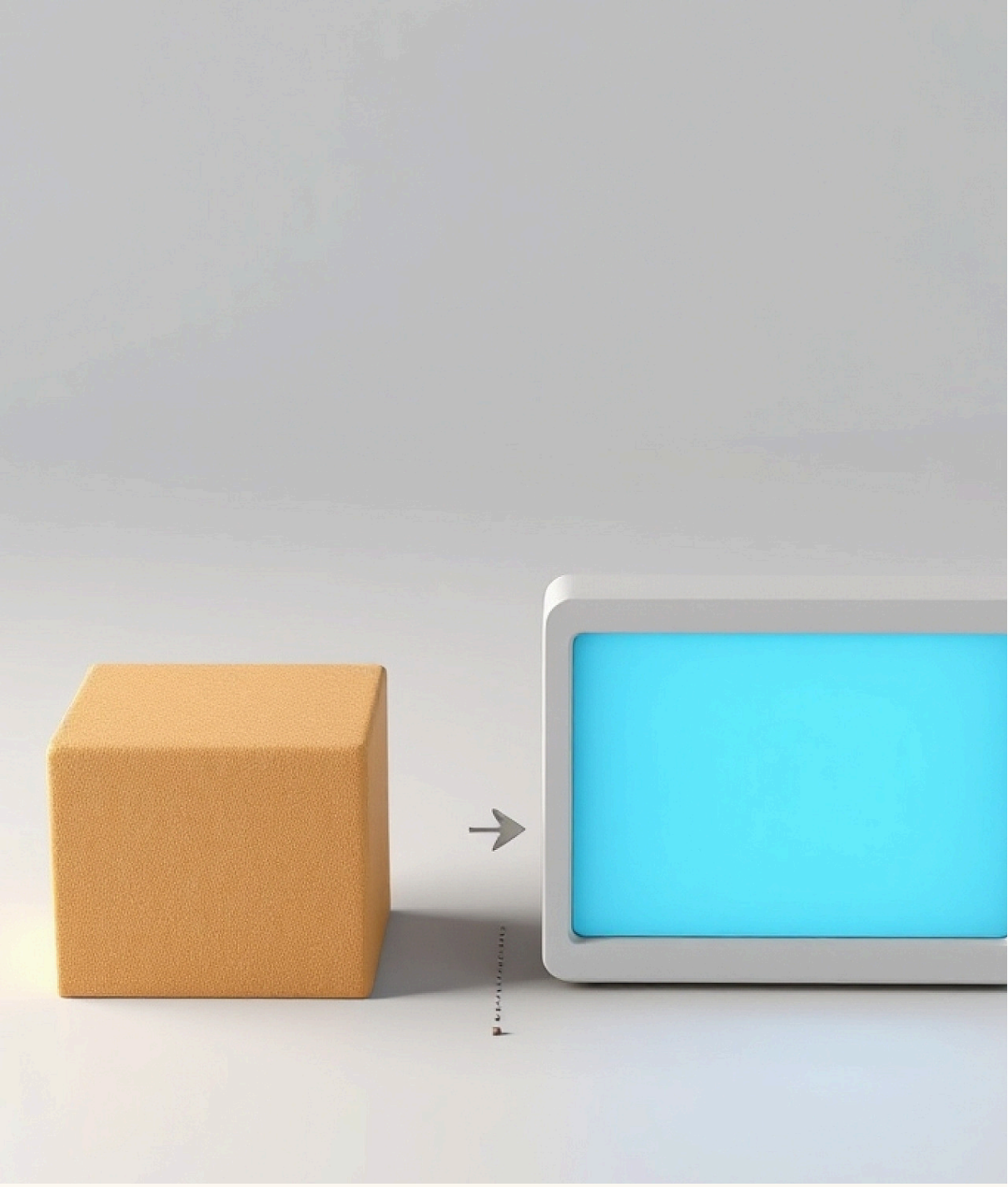
Bir listedeki her öğe indeks adı verilen bir sayı alır. İndeks, listedeki belirli bir öğeyi adlandırma ve bulma yoludur. Örneğin, bir renk listesinde ilk rengin indeksi sıfır, ikincinin indeksi bir olabilir ve bu böyle devam eder. İndeks, öğenin nerede olduğunu sana söyleyen bir etiket gibidir. Öğeye erişmek veya öğeyi değiştirmek için onu kullanırsın.



Bir kutu sırasının başında durduğunu hayal et. İlk kutu sıfır adım uzaktadır çünkü zaten oradasın. Ona ulaşmak için hareket etmene gerek yoktur. Ana fikir şudur: ilk öğe için başlangıçtan uzaklık sıfırdır. Bu yüzden indeksi sıfırdır. Adımları bu şekilde saymak, uzaklığı düşündüğünde doğaldır.



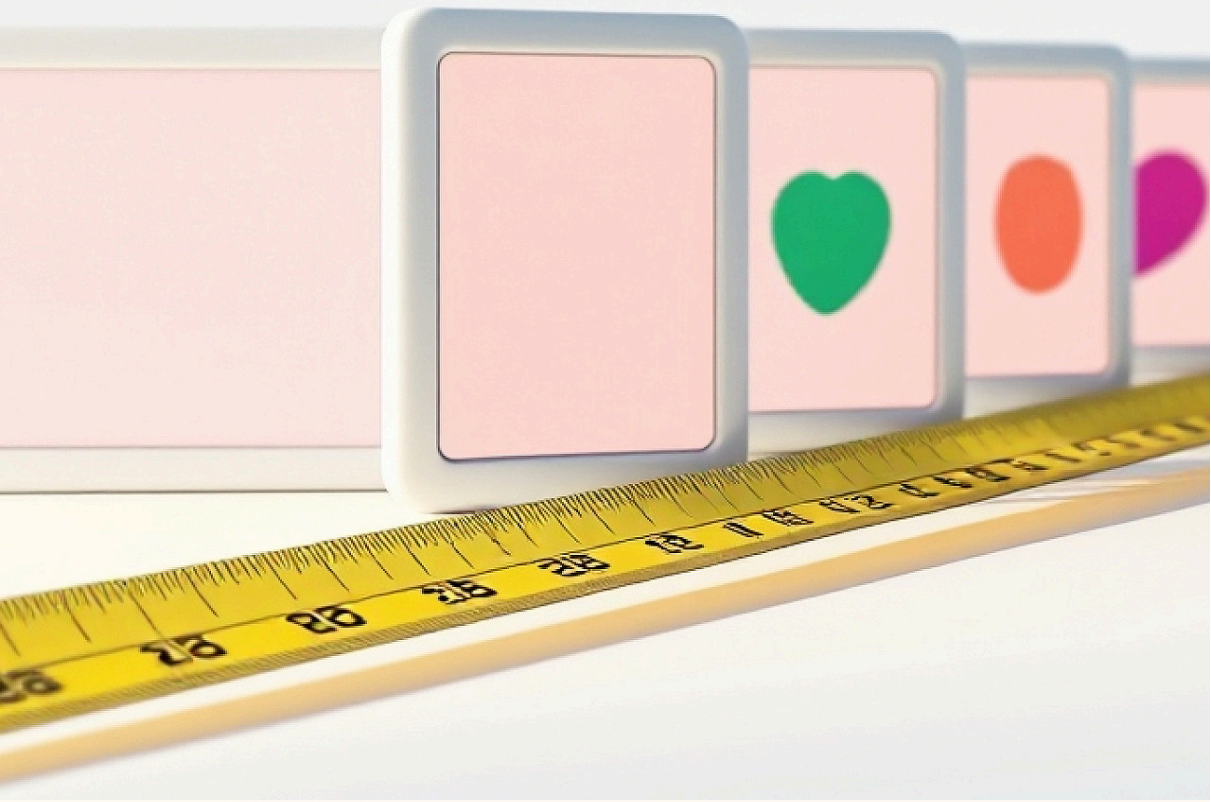
İlk öğenin indeksi sıfırdır çünkü başlangıç noktasından sıfır adım uzaktadır. Sıfır tabanlı indekslemenin özü budur. Örneğin, üç öğeli bir listen varsa ilk öğe sıfır indeksindedir. Başlangıçtan hareket etmediğin için onu bir olarak saymazsın. Bu, sıfırı doğal ilk sayı yapar.



Bir adım ileri git ve ikinci öğeye ulaş. İndeksi birdir çünkü başlangıçtan bir adım uzaktadır. Örneğin, bir harf listesinde ikinci harfin indeksi birdir. Örüntü şudur: her ileri adım indekse bir ekler. Yani ikinci öğe her zaman başlangıçtan bir adım uzaktadır.



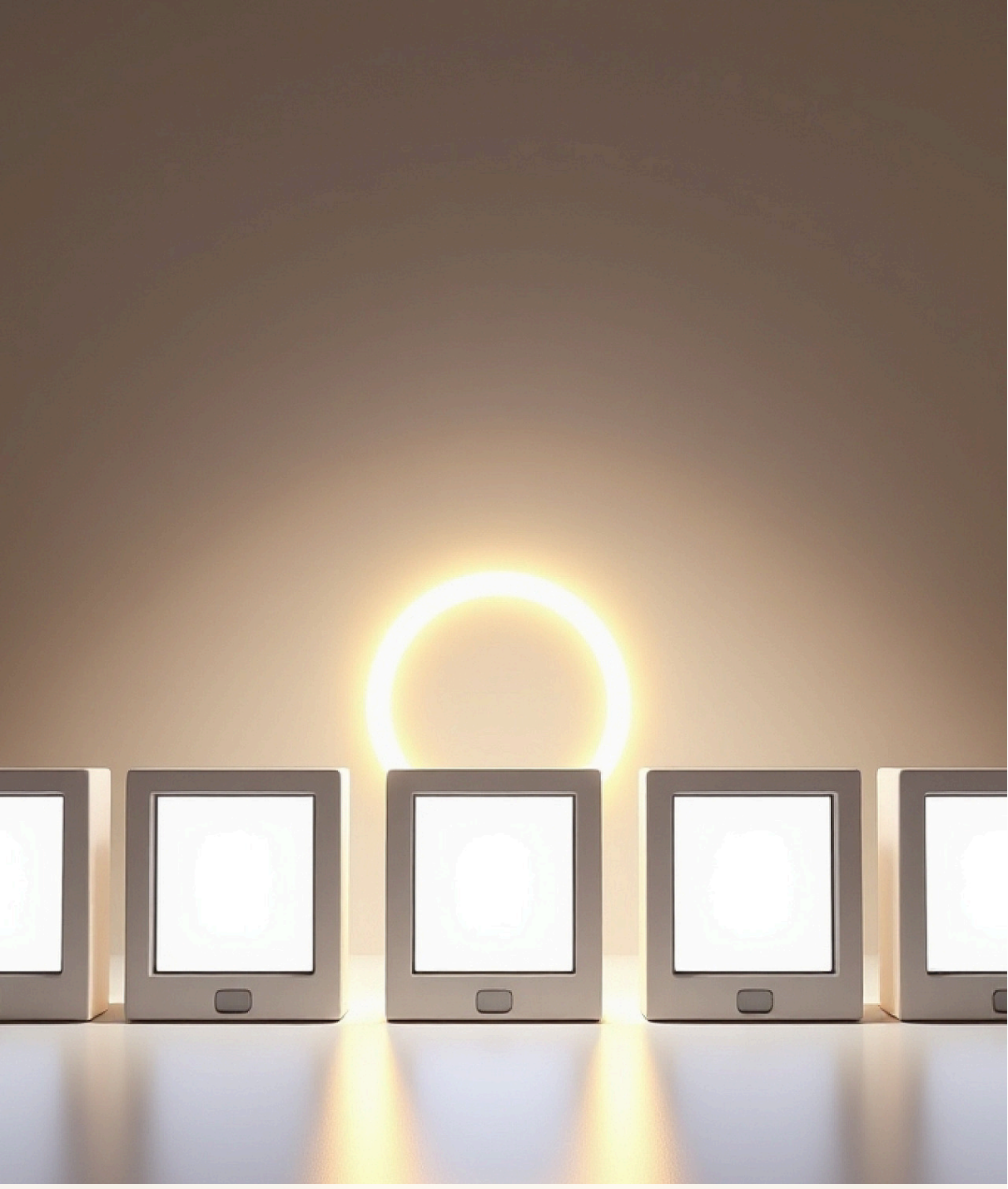
Bir adım daha at ve indeksi iki olan üçüncü öğeye ulaş. Örüntü açıktır: her ileri adım indekse bir ekler. Örneğin, bir listedeki üçüncü öğenin indeksi ikidir. Bu, tüm öğeler için böyle devam eder. İndeks, başlangıçtan kaç adım uzakta olduğunu söyler.



Bir öğenin indeksi, başlangıçtan adım cinsinden uzaklığına eşittir. Bu, sıfırı doğal ilk sayı yapar. Örneğin, üç adım uzaktaki bir öğenin indeksi üçtür. Bu kural, ne kadar uzun olursa olsun her liste için geçerlidir. Konumları numaralandırmanın basit ve tutarlı bir yoludur.



Beş öğeli bir listeye bak ve indekslerini say: sıfır, bir, iki, üç, dört. Sayıların her seferinde bir arttığını görürsün. Örneğin, beş adlı bir listenin indeksleri sıfırdan dörde kadar olur. İlk öğe sıfırdır, ikincisi birdir ve bu böyle devam eder. Bu, sıfır tabanlı indekslemenin pratikte nasıl çalıştığını gösterir.



Beş öğeli listedeki son öğenin indeksi dörttür çünkü başlangıçtan beş değil, dört adım uzaktadır. Örneğin, listenin beş öğesi varsa sonuncusu dört indeksindedir. Bu, yaygın bir kafa karışıklığı noktasıdır. Unutma: son indeks her zaman öğe sayısından bir eksiktir.



Tüm diller sıfır yerine birden saymaya başlamaz. Bazı diller, ilk öğenin indeksinin bir olduğu bir tabanlı indeksleme kullanır. Yani sıfır tabanlı indeksleme bir seçimdir, her dil için bir kural değildir. Örneğin, Python gibi bazı diller sıfır kullanırken Lua gibi diğerleri bir kullanır. Bu, kullandığın dile bağlıdır.



Yaygın bir hata, beş öğeli bir listenin beş indeksinde bittiğini varsaymaktır. Doğru son indeks dörttür. Bu karışıklığa bir kayma hatası denir. Örneğin, beş öğeli bir listede beş indeksine erişmeye çalışırsan hata alırsın çünkü indeksler sıfırdan dörde kadardır. Son indeksi her zaman kontrol et.



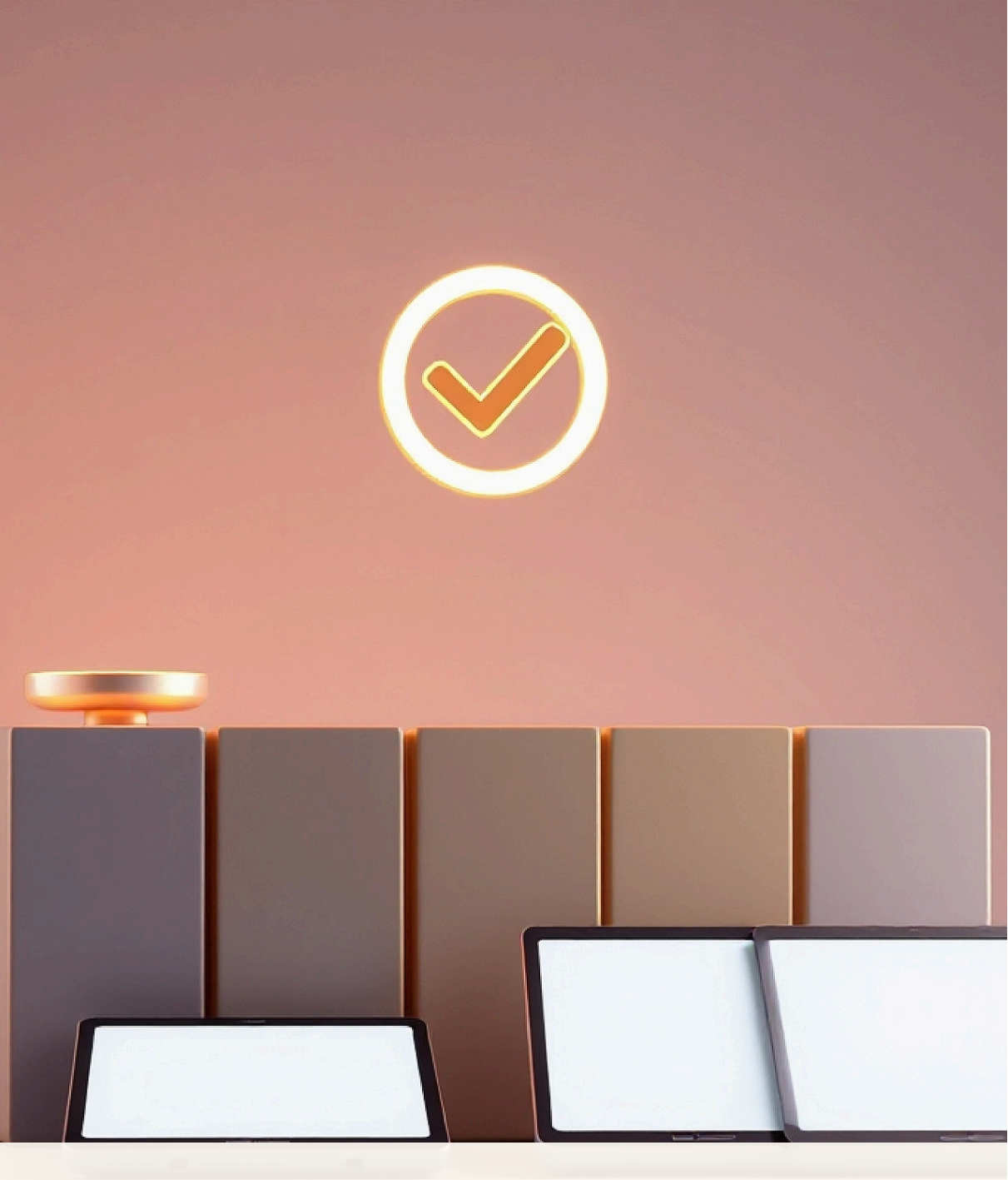
İnsanlar hata yapar çünkü genellikle öğeleri birden başlayarak sayarız. Sıfır tabanlı indekslemede öğe sayısı ile son indeks arasında bir fark vardır. Örneğin, beş öğen varsa son indeksin beş olduğunu düşünebilirsin, ama o dörttür. Bu, ilk öğeyi sıfır yerine bir olarak saymaya alışkın olduğumuz için olur.



Herhangi bir boyuttaki liste için son indeks, öğe sayısı eksi birdir. Bu hızlı kontrol, hatadan kaçınmana yardımcı olur. Örneğin, on öğeli bir listenin son indeksi dokuzdur. İki öğeli bir listenin son indeksi birdir. Yani öğe sayısını biliyorsan son indeksi kolayca bulabilirsin.



Bir öğenin başlangıçtan kaç adım uzakta olduğunu sorarak iki kez kontrol et. İlk öğeyse yanıt sıfırdır. Örneğin, bir indeks konusunda emin değilsen başlangıçtan adımları say. İlk öğe sıfır adımdır, ikincisi bir adımdır ve bu böyle devam eder. Bu basit soru seni doğru yolda tutar.



İlk indeks sıfırdır çünkü başlangıçtan uzaklığı gösterir. Beş öğeli bir listede son indeks dörttür. Tüm dillerin böyle çalışmadığını unutma ve bir kayma hatasına dikkat et. Örneğin, sıfır tabanlı bir dilde son indeksi bulmak için her zaman sayıdan bir çıkar. Bu özet, temel noktaları hatırlamana yardımcı olur.