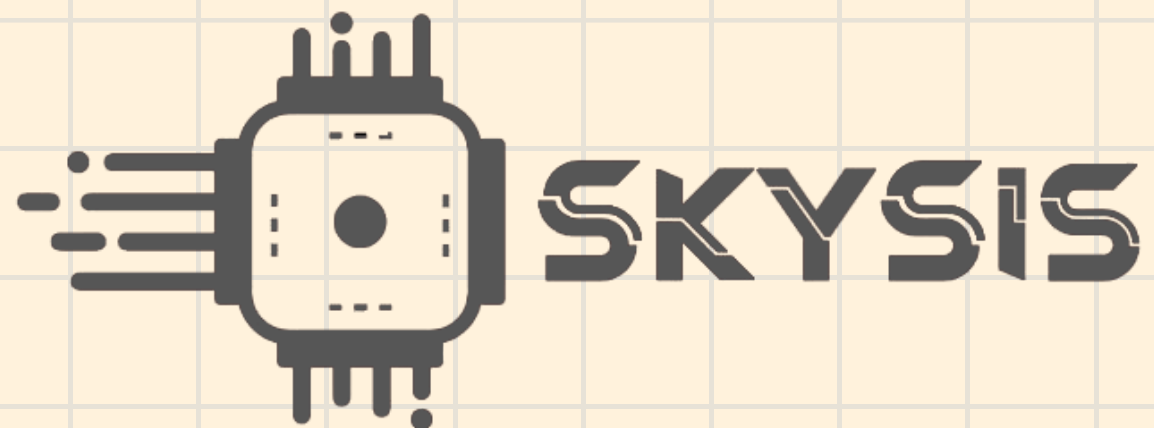




# Veri Yapıları Data Structures



# İçerik

No one:  
Beginner programmers when they make  
the compiler print "Hello World":



- Diziler - Matrisler
- Linked List
- Stack
- Queue

- Tree
- Heap
- Graph

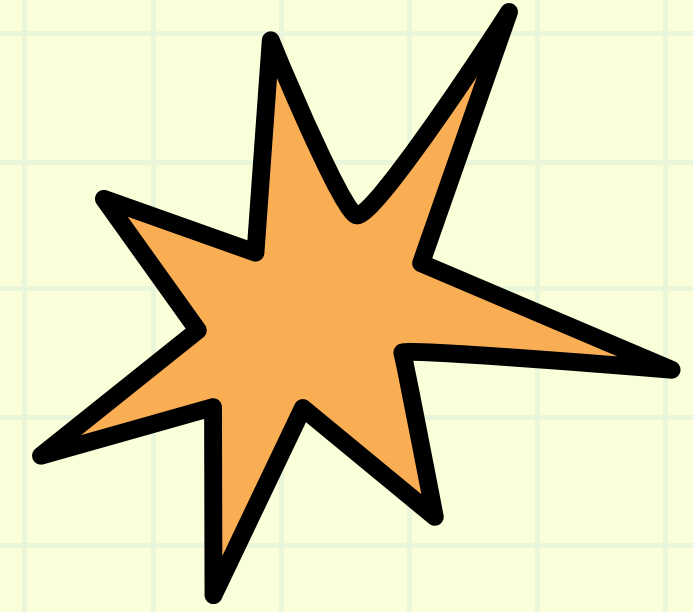
# Diziler ve Matrisler

Diziler, aynı tip verilerin bellek içerisinde ardışıl alanlarda tutulduğu veri tipidir.

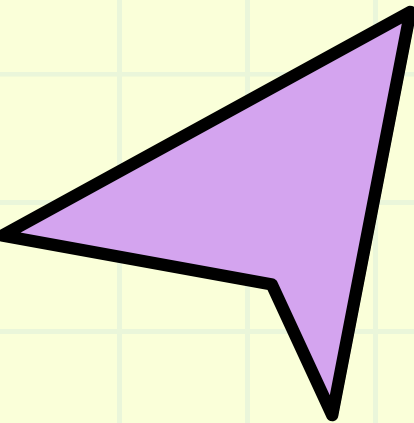
Matrisler ise satırlar ve sütunlar halinde tutulan aynı tip verilerin oluşturduğu bir veri topluluğudur. Diziler dizisi veya 2 boyutlu dizi denilebilir

Memory Location									
200	201	202	203	204	205	206	▪	▪	▪
U	B	F	D	A	E	C	▪	▪	▪
0	1	2	3	4	5	6	▪	▪	▪
Index									
				Column 0	Column 1	Column 2			
Row 0				x[0][0]	x[0][1]	x[0][2]			
Row 1				x[1][0]	x[1][1]	x[1][2]			
Row 2				x[2][0]	x[2][1]	x[2][2]			

# Bunu Biliyor Muydunuz?



Görüntüler de R(Red),G(Green)  
ve B(Blue) renk matrislerinin  
birleşiminden oluşan 3 boyutlu  
bir matrisdir.



JPG 260 X 194



260 X 194 X 3



8,11,0, 55,13,25,19

15,241,2,155,13,35,65

14,211,0,255,23,45,11

05,255,1,255,10,17,23

77,167,9,112,56,16,90

45,245,0,145,22,55,48



# Linked List Bağlı Liste

Dizilerden farklı olarak Linked List elemanları belleğin ardışıl alanlarında değil, farklı bölümlerinde tutulur.

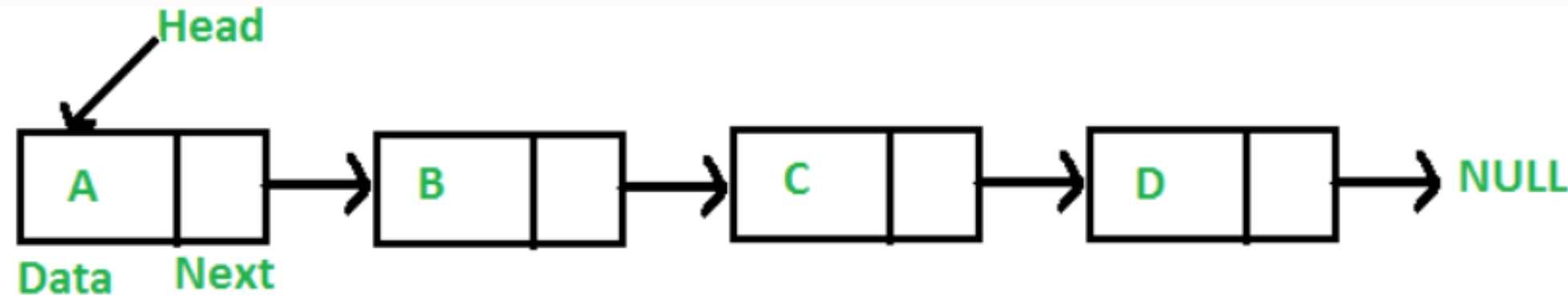
## Avantajları

- Eleman eklemek ve silmek çok kolaydır.
- Bellekte ardışıl büyük bir alan yoksa bile elemanlar ayrı yerlerde tutulabilir

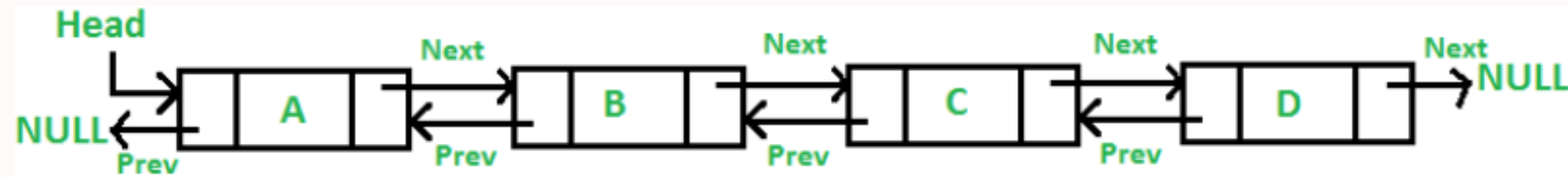
## Dezavantajları

- Hem veriyi hem de kendinden sonraki elemanın bellekteki yeri tutulduğundan daha çok alan kullanılır.
- Caching kullanılamaz. Bu sebeple elemanlara erişim dizi indexlerine göre daha yavaştır

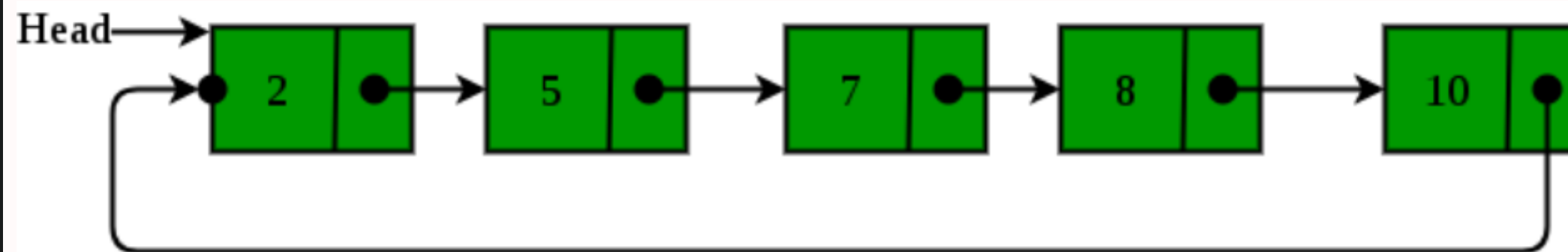
# LinkedList Türleri



Single Linked List



Double Linked List

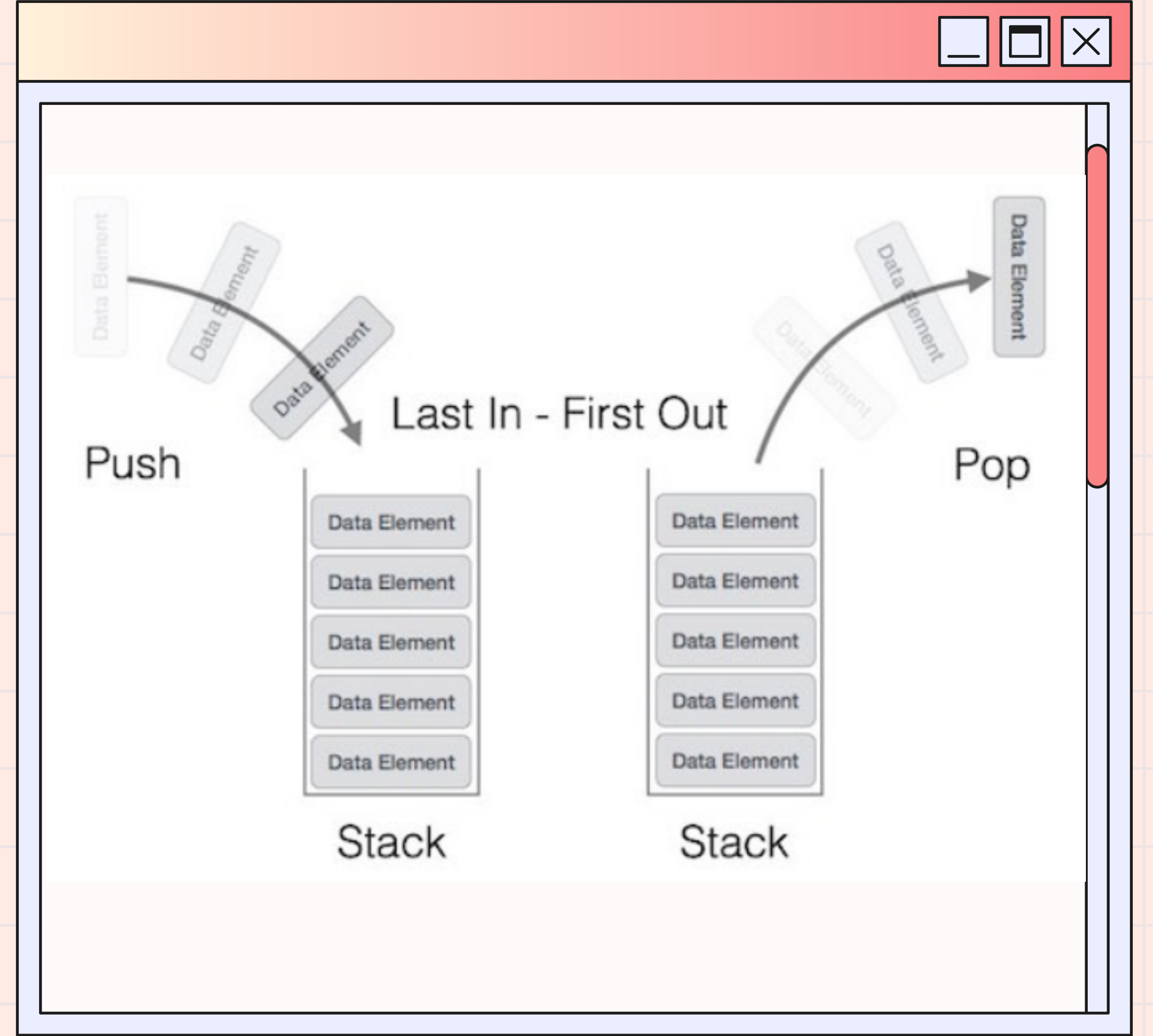


Circular Linked List



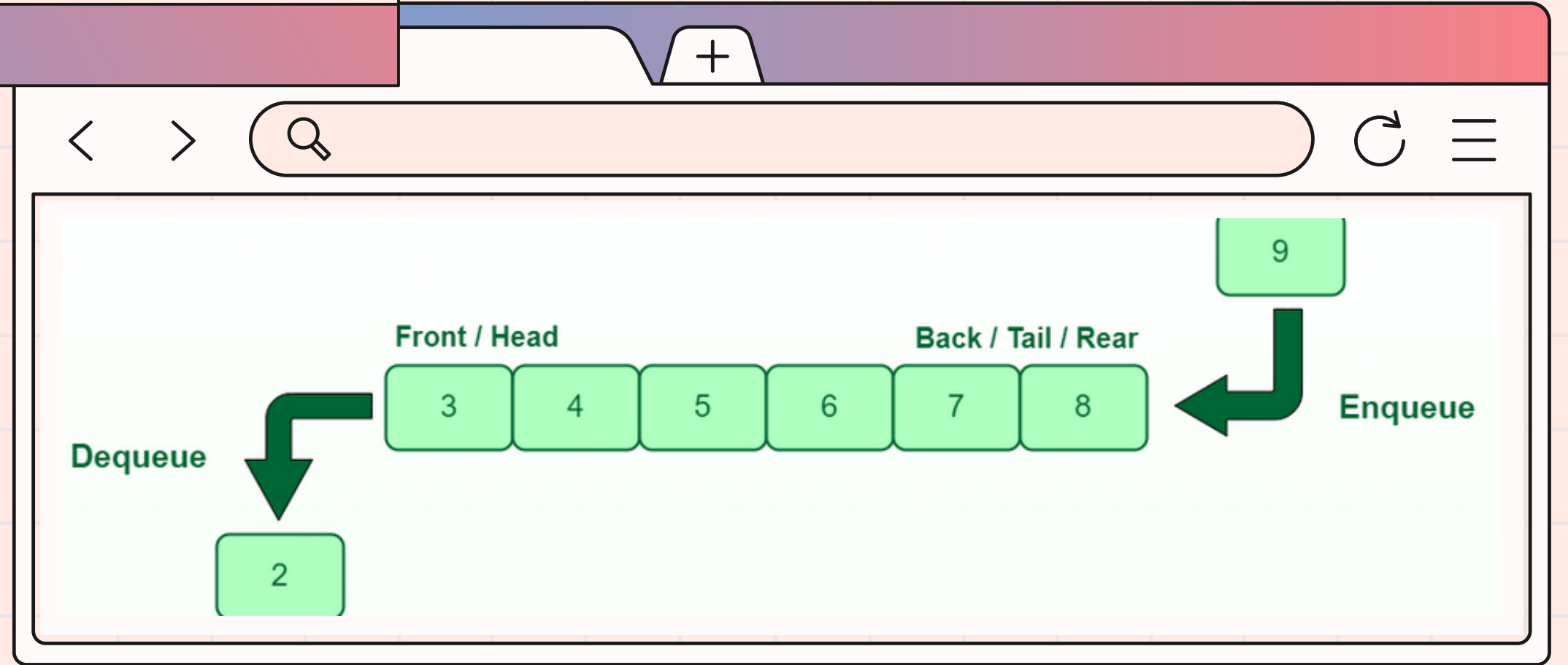
# Stack Yığın

- Lineer bir veri yapısıdır.
- Üst üste dizili tabaklar mantığıyla çalışır.
- Ekleme(push) ve çıkarma(pop) işlemleri en son dan gerçekleşir.
- ilk eklenen elemana erişebilmek için tüm verileri pop ile atmak gereklidir.

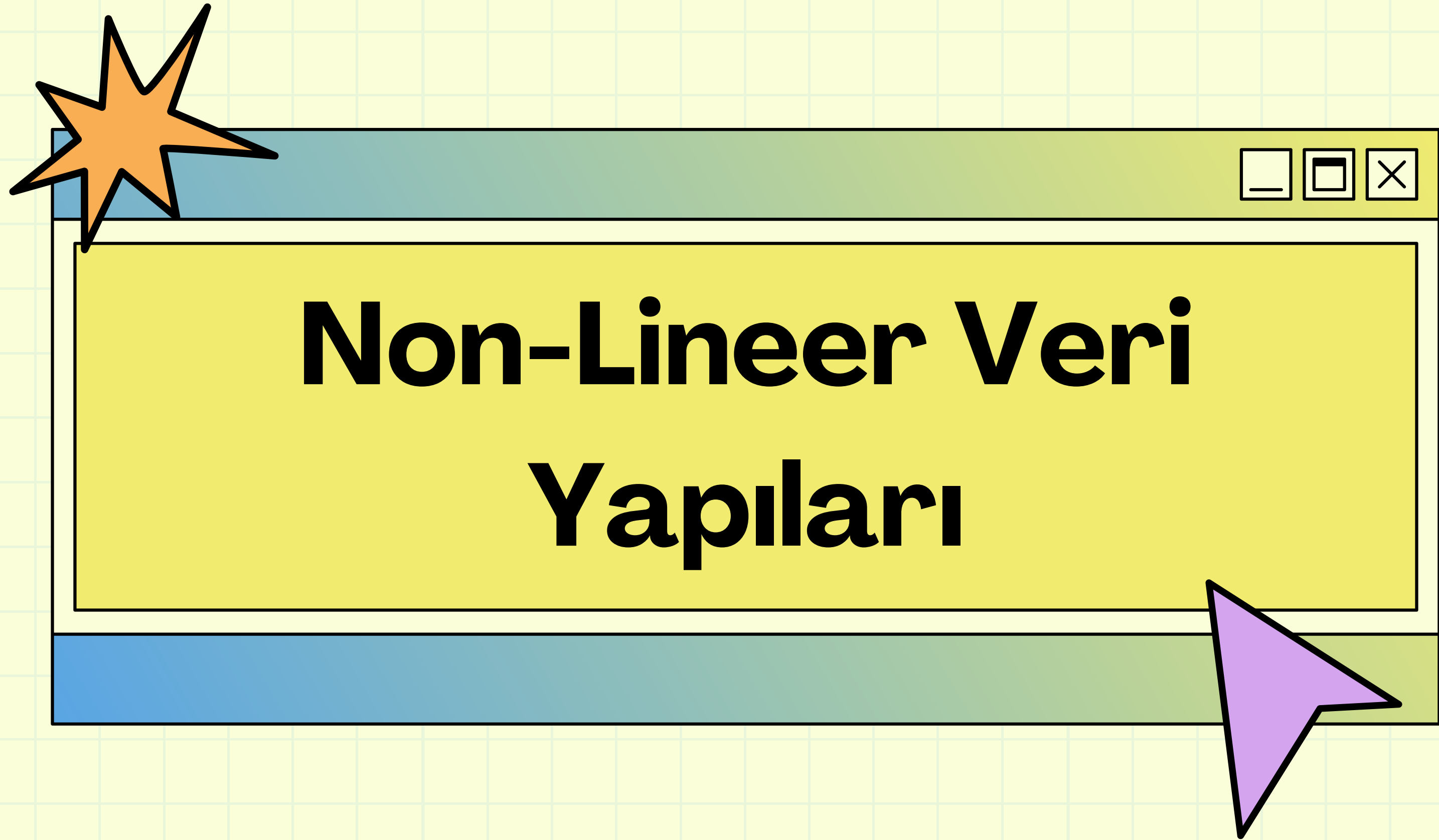


# Queue Kuyruk

- Stack yapısından farklı olarak giriş ve çıkış işlemleri ayrı uçlardan olur.
- First-in-first-out mantığıyla çalışır. Yani ilk eklenen eleman ilk çıkarılabilir.

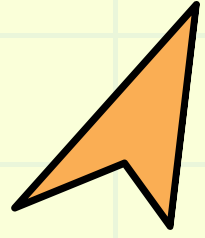
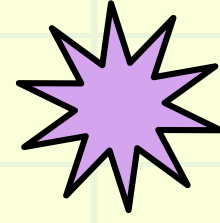




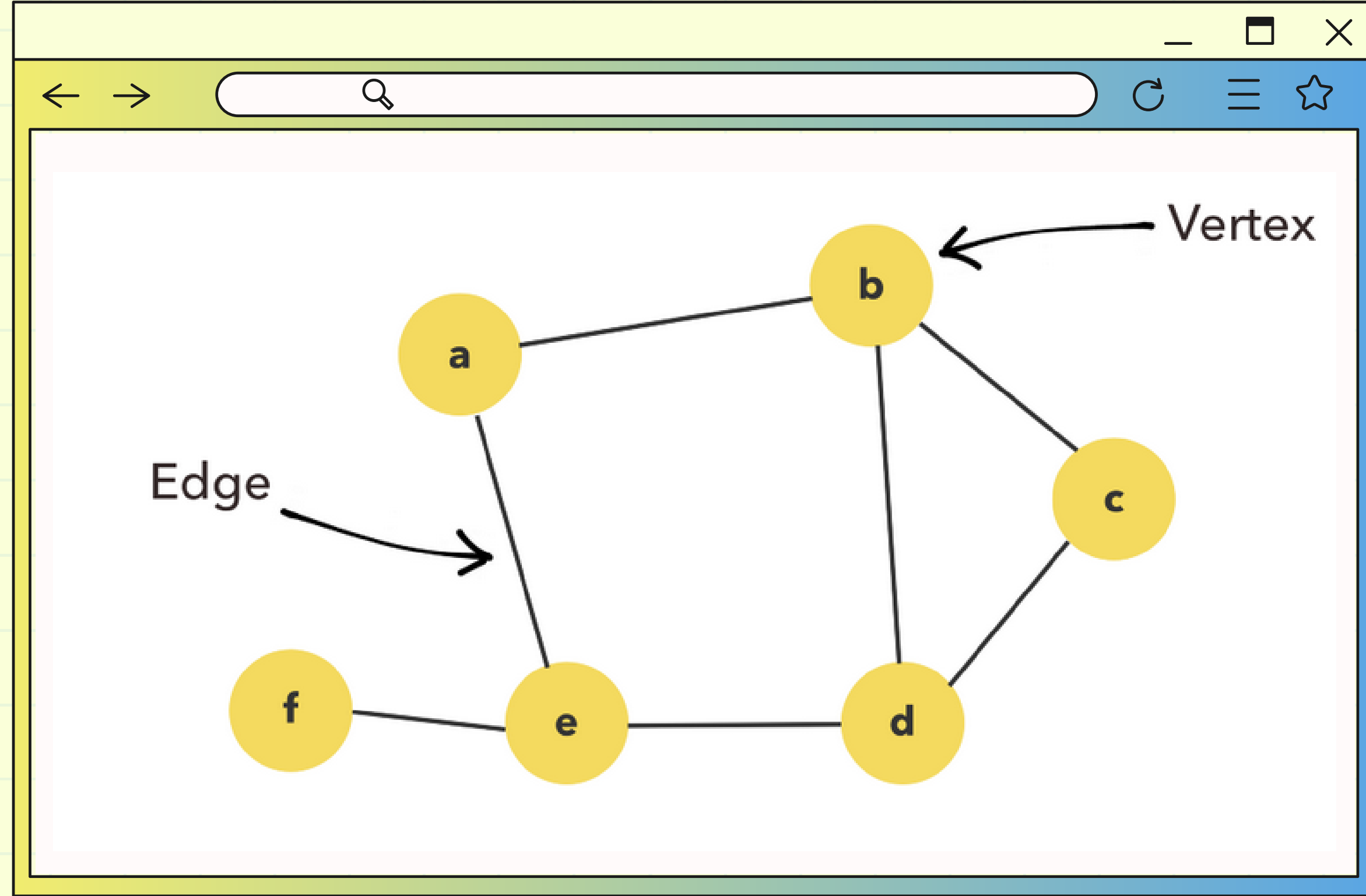


# Non-Linear Veri Yapıları

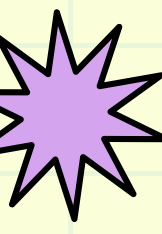
# Graph



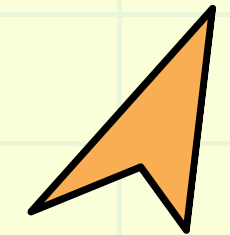
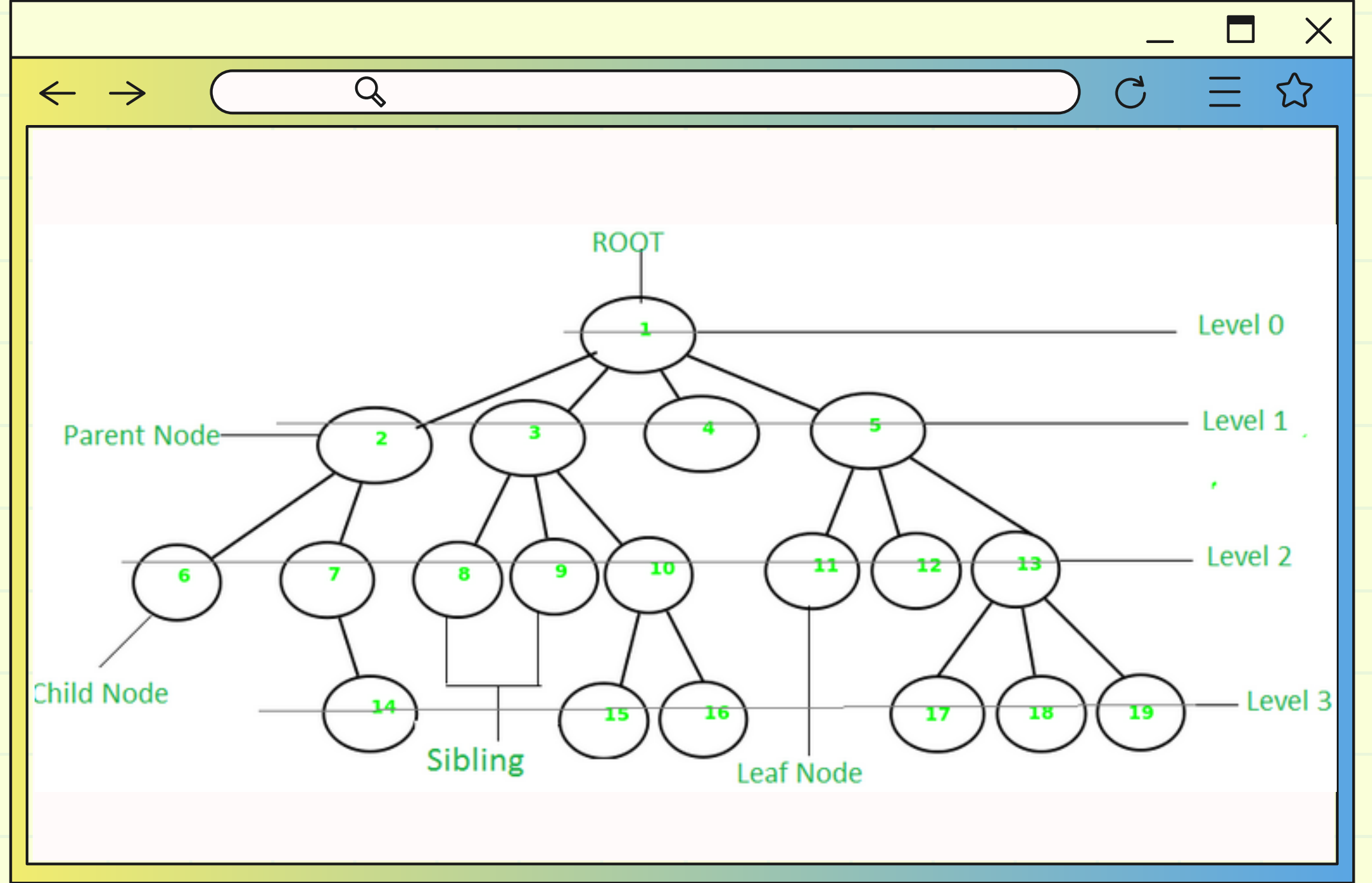
- Graflar, birbirine bağılı düğümlerden oluşan bir veri tipidir.
- İlişkiler "edge" adı verilen kenarlarla belirtilir.
- Vertice- $\rightarrow V$ , Edge- $\rightarrow E$
- Graflar, yön belirtme durumlarına göre yönlü ve yönsüz olarak ikiye ayrılırlar.



# Tree-Ağaç



- Özel bir tür graftır. Döngü içermezler.
- Her düğümün sıfır ya da daha fazla alt düğümü ve en fazla bir üst düğümü olabilir.
- Üst düğümü olmayan düğüme kök, alt düğümü olmayan düğümlere yaprak denir.



# Heap

- Özel bir tür Ağaçtır.
- Dengeli ağaçlardır.
- İki türü vardır->Min Heap ve Max Heap
- Min Heap'de root en küçük olmalıdır, Max heapde ise en büyüktür.

