

ARRAY METODLARI

`array.push(...[elemanlar])`

DİZİ SONUNA ELEMAN EKLER.

```
let array = [1, 8, 20, 33, 58, 67, 72];  
array.push(...[13, 27, 62]);  
[1, 8, 20, 33, 58, 67, 72, 13, 27, 62]
```

`array.pop()`

DİZİ SONUNDAKİ ELEMANI ÇIKARIR.

```
let array = [1, 8, 20, 33, 58, 67, 72];  
array.pop();  
[1, 8, 20, 33, 58, 67]
```

`array.shift()`

DİZİ BAŞINDAKİ ELEMANI ÇIKARIR.

```
let array = [1, 8, 20, 33, 58, 67, 72];  
array.shift();  
[8, 20, 33, 58, 67, 72]
```

`array.unshift(...[elemanlar])`

DİZİ BAŞINA ELEMAN EKLER.

```
let array = [1, 8, 20, 33, 58, 67, 72];  
array.unshift(...[13, 27, 62]);  
[13, 27, 62, 1, 8, 20, 33, 58, 67, 72]
```

`array.splice(index, kaç eleman silinip, hangi elemanlar eklenecek)`

BİR İNDEX TEN BAŞLAYARAK ELEMANLAR EKLEYEBİLİR YA DA SİLEBİLİR.

```
let array = [1, 8, 20, 33, 58, 67, 72];  
array.splice(2, 3, 13, 27)  
[1, 8, 13, 27, 67, 72]
```

`array.includes (eleman, index)`

BİR İNDEKSİNDEN BAŞLAYARAK ELEMAN ARAMAMIZI SAĞLAR. VARSA TRUE YOKSA FALSE DÖNER.

```
let array = [1, 8, 20, 33, 58, 67, 72];

//index=1 den sonra 58 elemanı var mı

console.log(array.includes(58, 1))
true

//index=3 den sonra 7 elemanı var mı

console.log(array.includes(7, 3))
false
```

`array.concat(dizi, eleman)`

BİR DİZİ İLE BAŞKA BİR DİZİ YA DA ELEMANLARI BİRBİRİNE EKLER.

```
let array = [1, 8, 20, 33, 58, 67, 72];

console.log(array.concat([27, 15], 9, 87))

[1, 8, 20, 33, 58, 67, 72, 27, 15, 9, 87]
```

`array.join("karakter")`

DİZİYİ BELİRTİLEN KARAKTERLERE GÖRE YENİDEN SIRALAR.

```
let array = ["Ahmet", "Mehmet", "Ali", "Veli"];

console.log(array.join("/-+*"))

Ahmet/-+*Mehmet/-+*Ali/-+*Veli
```

`array.reverse()`

DİZİNİN SIRALAMASINI TERSE ÇEVİRİR.

```
let array = ["Ahmet", "Mehmet", "Ali", "Veli"];

console.log(array.reverse())

['Veli', 'Ali', 'Mehmet', 'Ahmet']
```

array.fill (değer, başlangıç, son)

DİZİYİ BELİRTİLEN BAŞLANGIÇ INDEX TEN BAŞLAYARAK BELİRTİLEN SON INDEX TEN BİR ÖNCEKİ INDEX E KADAR TEKRAR EDEN YENİ BİR DEĞER İLE DOLDURUR.

```
let array = ["Ahmet", "Mehmet", "Ali", "Veli", "Selami"];

console.log(array.fill("Hasan", 1, 4))

['Ahmet', 'Hasan', 'Hasan', 'Hasan', 'Selami']
```

array.sort()

DİZİYİ KARAKTER SIRALMASINA GÖRE SIRALAR.

```
let array = ["Ahmet", "Mehmet", "Ali", "Veli", "Selami"];

console.log(array.sort())

['Ahmet', 'Ali', 'Mehmet', 'Selami', 'Veli']
```

array.map (fonksiyon)

DİZİDEKİ HER BİR ELEMAN İÇİN BİR FONKSİYONU ÇALIŞTIRIR VE BİR DİZİ ÜRETİLEREK DÖNDÜRÜLÜR.

```
let array = [16, 36, 64, 100];

console.log(array.map(Math.sqrt))

[4, 6, 8, 10]
```

array.flat(derinlik)

İÇ İÇE DİZİLERDE BELİRTİLEN DERİNLİĞE KADAR DÜZLEŞTİRME YAPAR.

```
let array = [[16, [36, [25, 49], 64], 100]];

console.log(array.flat(1))
[16, [36, [25, 49], 64], 100]

console.log(array.flat(2))
[16, 36, [25, 49], 64, 100]

console.log(array.flat(Infinity))
[16, 36, 25, 49, 64, 100]
```