push () metodu: Dilimizde ittirmek anlamına gelmektedir. JS' de dizinin sonuna eleman eklemek için kullanılır.

```
let sayilar = ["sıfır", "bir", "iki", "üç"];

sayilar.push("dört", "beş");

console.log(sayilar);
```

çıktı => sıfır, bir, iki, üç , dört, beş

unshift () metodu: Dizinin başına eleman eklemek için kullanılır.

```
let sayilar = ["sıfır", "bir", "iki", "üç"];

sayilar.unshift("eksi iki", "eksi bir");

console.log(sayilar);
```

çıktı => eksi iki, eksi bir, sıfır, bir, iki, üç

splice () metodu: JS' de belirtilen indeks değerinden sonra silinecek eleman ya da elemanlar için kullanılır. Bu metotta iki tane parametre vardır. İlk parametre silinecek indeks konumunu, ikinci parametre ise indeksten sonra kaç adet elemanın silineceğini belirtir. Negatif indeksler de kullanılabilir. Bu da sondan başa doğru kullanımlar içindir. Aynı metotla diziye eleman da eklenebilir.

```
let sayilar = ["eksi iki", "eksi bir", "sıfır", "bir", "iki", "üç"];

sayilar.splice(0,2);

console.log(sayilar);
```

çıktı => sıfır, bir, iki, üç

```
let sayilar = ["eksi iki", "eksi bir", "sıfır", "bir", "iki", "üç"];

sayilar.splice(0,2,'' Bu eklenir'');

console.log(sayilar);
```

çıktı => Bu eklenir, sıfır, bir, iki, üç

pop () ve shift () metotları: pop metodu dizinin son elemanını, shift metodu ise dizinin ilk elemanını silmeye yarar.

```
let sayilar = ["Bu eklenir", "sıfır", "bir", "iki", "üç"];

sayilar.pop();

sayilar.shift();

console.log(sayilar);
```

çıktı => sıfır, bir, iki,

concat () metodu: Birden fazla diziyi birleştirmek için kullanılır ve yeni dizi tanımlanılır.

```
let sayilar = ["sıfır", "bir", "iki", "üç"];

let alfabe = ["a", "b", "c"];

let sayilar_ve_alfabe = sayilar.concat(alfabe);

console.log(sayilar_ve_alfabe);
```

çıktı => sıfır, bir, iki, üç, a, b, c

sort () metodu: Dizi elemanlarını küçükten büyüğe ve alfabetik sıraya göre sıralamak için kullanılır. Eleman indeks konumları da değişir.

```
let sayilar = ["sıfır", "bir", "iki", "üç"];

sayilar.sort();

console.log(sayilar);
```

çıktı => bir, iki, sıfır, üç

reverse () metodu: Dizi elemanlarını tersten sıralamak için kullanılır.

```
let sayilar = ["sıfır", "bir", "iki", "üç"];

sayilar.reverse();

console.log(sayilar);
```

çıktı => üç, iki, bir, sıfır

join () metodu: Normal şartlar altında çıktı alınırken elemanlar arasına varsayılan olarak virgül konulur. Bu metot ile dizi içindeki elemanların sıralama görünümünü değiştiririz.

```
let sayilar = ["sıfır", "bir", "iki", "üç"];

sayilar.join("-");

console.log(sayilar);
```

çıktı => sıfır-bir-iki-üç

includes () metodu: Dizide çok fazla eleman varsa ve aradığımız eleman dizinin içinde olup olmadığını kontrol etmek istiyorsak bu metot tercih edilir. İki değer döndürür true ya da false.

```
let sayilar = ["sıfır", "bir", "iki", "üç"];

sayilar.includes("bir");

console.log(sayilar);
```

çıktı => true

fill () metodu: Vereceğimiz değer, başlangıç indeksi ve bitiş indeksi ile dizideki elemanları değiştirir. Bitiş indeksi verilmezse dizideki başlangıç indeksinden itibaren bütün elemanları değiştirir.

```
let sayilar = [1, 2, 3, 4];

sayilar.fill(1,2,4);

console.log(sayilar);
```

çıktı => 1,2,1,1

```
let sayilar = [1, 2, 3, 4];

sayilar.fill(7,1);

console.log(sayilar);
```

çıktı => 1,7, 7, 7

flat () metodu: Dizide belirtilen derinliğe kadar alt dizi elemanları ile yeni dizi oluşturur ve dizilerdeki boş yuvaları kaldırır.

```
let sayilar = [1, 2, [3, 4]];

sayilar.flat ();

console.log(sayilar);

çıktı =>    [1, 2, 3, 4]
```

```
let sayilar = [1, 2, , 4, 5];

sayilar.flat ();

console.log(sayilar);

çıktı =>    [1, 2, 4, 5]
```

map () metodu: Dizideki her elemana verilen işlemi yaptıktan sonra yeni bir dizi oluşturur.

```
let sayilar = [1, 2, 3, 4];
let sonuc = sayilar.map(x => x * 2 );

console.log(sonuc);

çıktı => [2, 4, 6, 8]
```