



BATCH : B45-46-71
LESSON : **Javascript**
DATE : 01.04.2022
SUBJECT : **Objects & Arrays**



 techproeducation.com

 info@techproeducation.com

 +1 (917) 768-7466



Array of Objects

```
const arabalar = [{  
  marka: 'Mercedes',  
  model: 'S500',  
  renk: 'Bej',  
  vites: 'Otomatik'  
},  
{  
  marka: 'BMW',  
  model: 'i8',  
  renk: 'Mavi',  
  vites: 'Otomatik'  
},  
...  
]
```

Elemanları object olan dizilere array of object denir.



Array Methods

Dizi elemanlarına erişmek, bunları manipüle etmek veya iterasyon yapmak için çeşitli hazır methodlar bulunmaktadır

mutator methods

Dizi içeriğinin değiştirilmesini sağlarlar

access methods

Dizi elemanlarına erişmek için kullanılırlar

iterator methods

Dizi elemanları döngü yoluyla dolaşmak için kullanılır



Mutator methods

Method	Açıklama
push	Dizinin sonuna bir veya daha fazla eleman ekler ve dizinin yeni eleman sayısını döndürür
pop	Dizinin son elemanını diziden siler ve bu elemanı döndürür
shift	Dizinin ilk elemanını diziden siler ve bu elemanı döndürür
unshift	Dizinin ilk indeksine yeni bir eleman ekler ve dizinin yeni eleman sayısını döndürür
sort	Diziyi yükselen sırada sıralar
reverse	Dizinin elemanlarını ters sıralar
splice	Dizi'nin içeriklerini, yeni öğeler ekleyerek, mevcut öğeleri silerek günceller.



Mutator methodlar

```
const meyveler = ["Elma", "Armut", "Muz", "Kivi"];  
  
meyveler.pop(); Kivi silindi  
  
meyveler.push('Ananas'); Sona Ananas eklendi  
  
meyveler.shift(); Elma silindi  
  
meyveler.unshift('Çilek'); Başa Çilek eklendi  
  
meyveler.reverse(); Tersine çevrildi  
  
meyveler.sort(); Harfe göre sıralandı  
  
meyveler.splice(1,0,'Kiraz'); 1.İndex e Kiraz eklendi  
  
meyveler.splice(3, 1 , 'Kayısı'); 3.İndex e Kayısı eklendi, Muz silindi
```



Access methods

Method	Açıklama
concat	Dizi ile bir başka diziye veya değeri birleştirerek yeni bir dizi döndürür.
includes	Dizinin belirtilen bir elemanı içerip içermediğine bakar. Eğer içeriyorsa true, içermiyorsa false döndürür.
indexOf	Belirtilen elemanın dizide ilk görüldüğü indeks numarasını döndürür.
lastIndexOf	Belirtilen elemanın dizide görüldüğü en son indeks numarasını döndürür. Bulunmazsa -1 döndürür.
join	Bir dizi içerisinde yer alan bütün elemanları birleştirerek String bir ifade olarak geri döndürür.
toString	Dizinin içerisindeki elemanları tek bir String olarak döndürür.
slice	Bir Dizinin elemanlarını, belirtilen başlangıç ve bitiş indeksine göre kopyasını oluşturarak ve döndürür.



Access methods

```
const hayvanlar = ['fil', 'kuş', 'deve', 'fare', 'kedi'];  
sayilar.includes('fil'); true  
sayilar.includes(3); false  
sayilar.includes("at"); false
```

```
console.log(elementler.join()); fil,kuş,deve,fare,kedi  
console.log(elementler.join("")); filkuşdevefarekedi  
console.log(elementler.join("-")); Fil-kuş-deve-fare-kedi
```

```
console.log(hayvanlar.slice(2)); ["deve", "fare", "kedi"]  
console.log(hayvanlar.slice(2, 4)); ["deve", "fare"]  
console.log(hayvanlar.slice(1, 5)); ["kuş", "deve", "fare", "kedi"]
```

```
console.log(hayvanlar.toString()); fil,kuş,deve,fare,kedi
```



Access methods

```
const hayvanlar = ['fil', 'kuş', 'deve', 'kuş', 'kedi', 'kuş'];

console.log(dizi2.indexOf('kuş')); 1
console.log(dizi2.lastIndexOf('kuş')); 5
console.log(dizi2.indexOf('at')); -1

const harfler = ['a', 'b', 'c'];
const rakamlar = [1, 2, 3];
const birlesik = harfler.concat(rakamlar, 4, [5, 6]);
console.log(birlesik); ['a', 'b', 'c', 1, 2, 3, 4, 5, 6];
```




Iteration - for

ÖRNEK:

Bir dizideki sayıların toplamını hesaplayan uygulamayı For döngüsü ile yazınız.

```
const rakamlar = [-5, 15, 22, -4, 45, 78, -25];  
  
let toplam = 0;  
  
for (let i = 0 ; i < rakamlar.length ; i++) {  
    toplam += rakamlar[i];  
}  
  
console.log(toplam);
```



Iteration - for

PRACTISE

Bir dizideki pozitif ve negatif sayıların toplamını hesaplayıp, bu toplamların farkını bulan uygulamayı for döngüsü ile yazınız.

hesapla adında bir fonksiyonda hesaplamayı yapınız.



Iteration – for in

For döngüsünün kısaltılmış halidir. (sayaç ve koşul kullanmaya gerek yok.) Özellikle **dizi** ve **nesnelerin iterasyonu** için geliştirilmiştir. Değişken içine otomatik olarak elemanın indis numarası gelir.

```
for ( degisken in diziAdi) {  
    // Döngü içi  
}
```



Iteration – for in

```
const adlar = ["Ahmet", "Can", "Mustafa", "Ayşe", "Elif"];
const soyAdlar = ["Öztürk", "Yılmaz", "Arı", "Çalı", "Yazı"];

const birlestir = (x, y) => {
  let adVeSoyadlar = [];
  for (let i in x) {
    adVeSoyadlar[i] = `${x[i]} ${y[i]}`;
  }
  return adVeSoyadlar;
};

console.log(birlestir(adlar, soyAdlar));
```

ÖRNEK:

İki ayrı dizideki eşleşen indis elemanları birleştirerek ayrı bir diziye saklayan uygulamayı **FOR IN** ile yazınız.



Iteration – for of

ES
6

FOR OF, **bir çok veri yapısı** üzerinde çalışabildiği için **FOR IN'e** göre daha geniş kullanım alanına sahiptir. Diziler, Stringler v.b bir çok veri yapısında kullanılabilir. Değişken içine otomatik olarak elemanın değeri gelir.

```
for ( degisken of diziAdi) {  
    // Döngü içi  
}
```



Iteration – for of

```
let arabalar = ["BMW", "Volvo", "Mini"];
let yazi = "";

for (let x of arabalar) {
  yazi += x + " ";
}
console.log(yazi);
```

ÖRNEK:

Dizideki elemanları birleştirerek tek bir **String** haline getiren uygulamayı **FOR OF** ile yazınız.



For - of

PRACTISE

Harici ülke listesi datasını alarak bir select içinde ülkeleri gösteriniz.