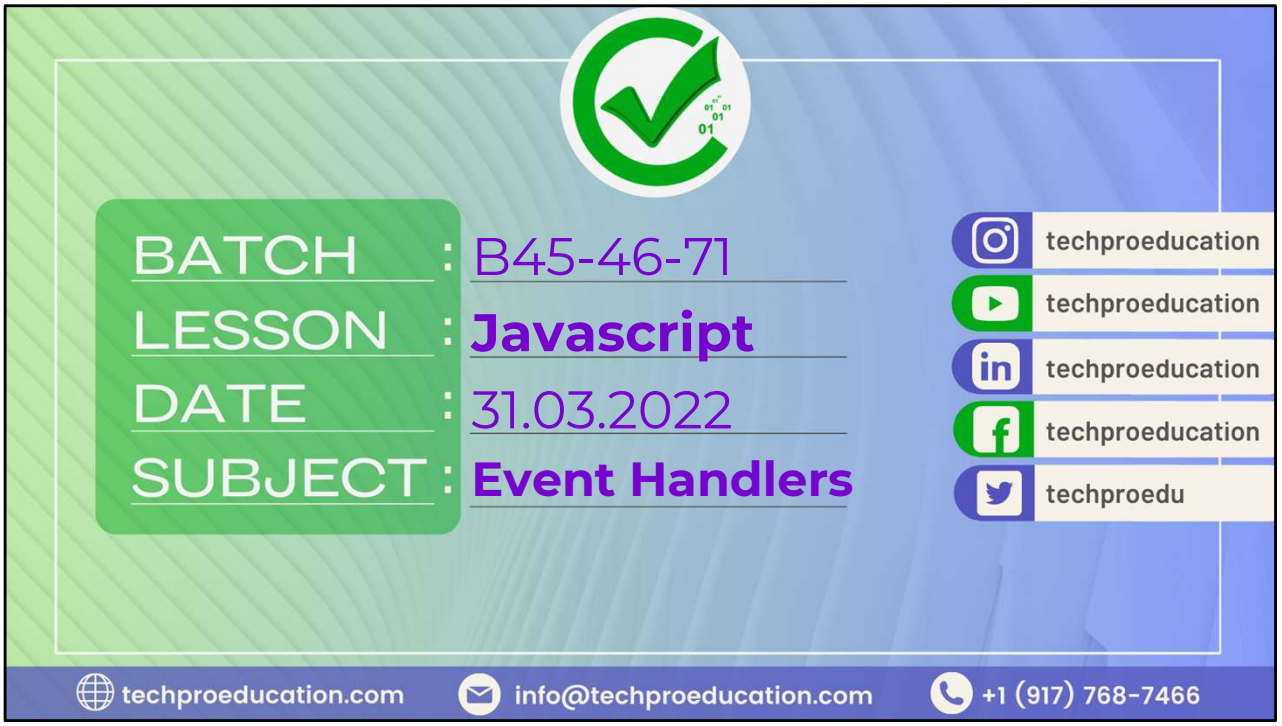

BATCH : B45-46-71
LESSON : Javascript
DATE : 31.03.2022
SUBJECT : Event Handlers
techproeducation
techproeducation
techproeducation
techproeducation
techproedu
techproeducation.com
info@techproeducation.com
+1 (917) 768-7466

Event Handlers

- Event handler nedir?
- Nasıl tanımlanır?
- Handler çeşitleri

Kullanıcı veya object kaynaklı bir olayın tetiklenmesi işlemine event triggering denir.
Bu işlemden hemen sonra DOM bir event oluşturur, bu işleme event throwing denir.
Ortaya çıkan event i yakalamak için oluşturulan yapıya event listener denir.
Yakalanana event içinde neler yapılacağını belirleyen yapıya ise event handler denir.

Tanımlama (1.Yöntem)

```
<button onclick="sayHello()">Tıkla</button>
```

event listener

event handler

```
function sayHello(e){
    alert("Hello");
}
```

Tanımlama (2.Yöntem)

```
<button id="btnMsg">Tıkla</button>
```

event listener

event handler

```
document.querySelector("#btnMsg").onclick = sayHello;

function sayHello(e){
    alert("Hello");
}
```

Tanımlama (3.Yöntem)

```
<button id="btnMsg">Tıkla</button>
```

event listener

event handler

```
document.querySelector("#btnMsg").onclick = function(e){
    alert("Hello");
}
```

```
document.querySelector("#btnMsg").onclick = (e) => {
    alert("Hello");
}
```

arrow function

Tanımlama (4.Yöntem)

```
<button id="btnMsg">Tıkla</button>
```

event listener

event handler

```
document.querySelector("#btnMsg").addEventListener("click", sayHello);

function sayHello(e){
    alert("Hello");
}
```

Tanımlama (5.Yöntem)

```
<button id="btnMsg">Tıkla</button>
```

event listener

event handler

```
document.querySelector("#btnMsg").addEventListener("click", function(e){
    alert("Hello");
});
```

```
document.querySelector("#btnMsg").addEventListener("click", (e) => {
    alert("Hello");
});
```

arrow function

Function

PRACTISE

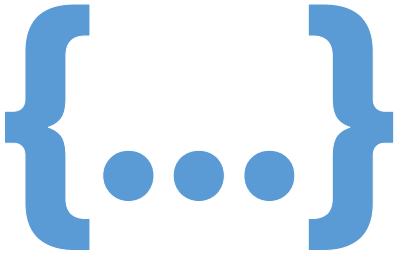
Butona basıldığında, iki textbox içinde 100 lük sistemde girilen notları kontrol edip, ortalamasını bulan ve bunu harf sistemine çeviren fonksiyonu yazınız.

Aralık	Not
[90-100]	A
[80-90)	B
[70-80)	C
[50-70)	D
[0-50)	F

Objects & Arrays

- Object nedir?
- Object tanımlama
- Array nedir?
- Array tanımlama
- Array of objects
- Array Methods
- Array iterations & search
  - for-in
  - forEach
  - map
  - filter
  - reduce
  - every
  - some
  - includes

Object nedir?



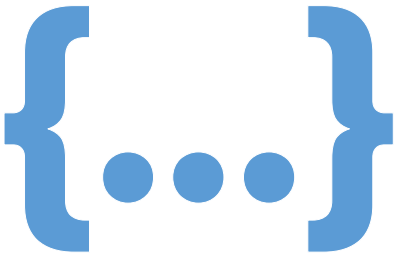
- Programımızda bir **araba** ya ait çeşitli özellikleri saklamak istiyoruz. Bu durumda, değişken kullanırsak;

```
const marka = 'Mercedes';
const model= 'S500';
const renk = 'Bej';
const vites = 'Otomatik';
```

Eğer birden fazla arabaya ait bilgiler saklanacksa?

Object yapıları kullanılarak bir nesneye ait özellikler ve değerleri gruplandırılabilir.

Object nedir?



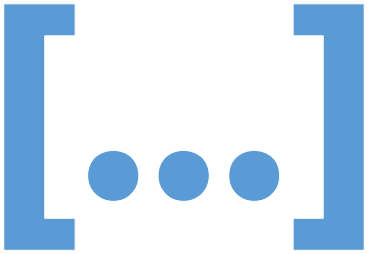
Javascript te basit object tanımı { } ile yapılır.

```
const araba = {
    marka:'Mercedes',
    model:'S500',
    renk:'Bej',
    vites:'Otomatik'
}
```

```
console.log(araba);

console.log(araba.marka);
```

Nesne yönelimli programlama (Object Oriented Programming), tasarımı ve soyutlamayı nesneler aracılığı ile gerçekleştiren bir yaklaşımdır. Bu yaklaşımda her şey bir nesnedir. (object) Her nesnenin çeşitli özellikleri (property) vardır.
Nesneler sayesinde tekrar kullanılabilir (reusable) yapılar çok daha kolay oluşturulur ve yönetilir.

Array nedir?



- Programımızda **araba** isimlerini saklamak istiyoruz. Bu durumda, değişken kullanırsak;

```
const araba1 = 'Mercedes';
const araba2 = 'TOFAŞ';
const araba3 = 'Anadol';
const araba4 = 'Ferrari';
```

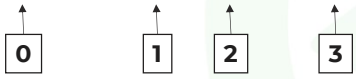
Saklanacak yüzlerce araba ismi olsaydı?

Diziler, bir **veri yapısı** (data structure) olup bir veya çok daha fazla veriyi saklamak için kullanılır.

Array tanımlama

Javascript te basit dizi tanımı [ ] ile yapılır.

```
let kurslar = ["Javascript","HTML","CSS","Bootstrap"];
```



Dizi tanımlama için bir diğer yöntem

```
let kurslar = new Array("Javascript","HTML","CSS","Bootstrap");
```

Dizilerden veri okumak

```
var kurslar = ["Javascript","HTML","CSS","Bootstrap"];
```

Dizi elemanlarına erişim

```
kurslar[1];
```

Dizi uzunluğu

```
kurslar.length
```

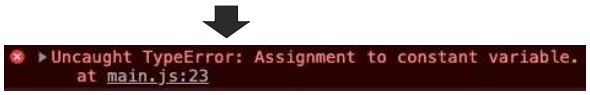
```
console.log(kurslar[1]);
console.log(kurslar[kurslar.length-1]);
```

Dizi elemanları **sıralı** olarak belleğe kaydedilirler.

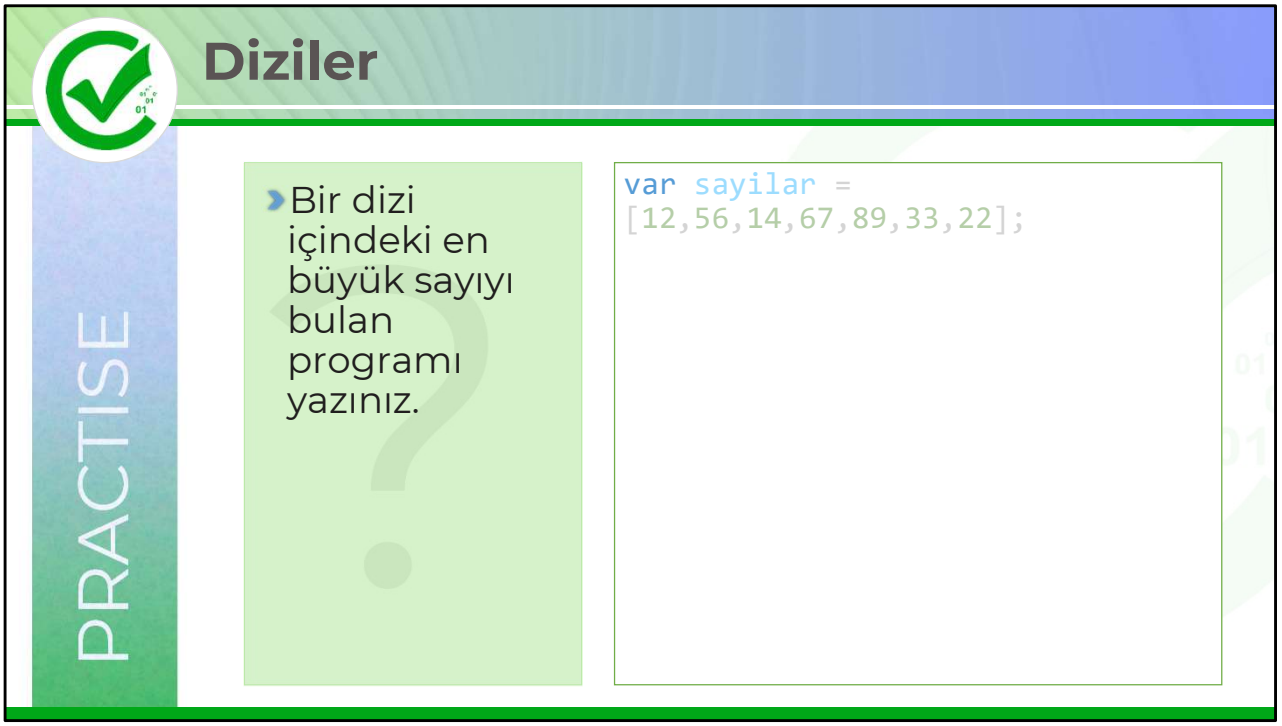
Sıra numaraları **0** ile başlar ve dizinin **eleman sayısının bir eksiğine** (**length-1**) kadar devam eder.

Dizilere veri yazmak

```
const kurslar = ["Javascript","HTML","CSS","Bootstrap"];


kurslar[1]="React";
kurslar[0]="Java";

console.log(kurslar);

kurslar = ["Ali","Veli];
```



Dizi **const** ile tanımlanmış olsa da dizinin elemanlarını değiştirebildik. Çünkü **non-primitive** veri türlerinin içerikleri değiştirilebilir. Ancak, tamamen bir başka dizi ile değiştirilemez.




```
var sayilar = [12,56,14,67,89,33,22];
var enbuyuk=sayilar[0];
for(var i=0; i<sayilar.length;i++){
    if(sayilar[i]>enbuyuk){
        enbuyuk = sayilar[i];
    }
}
console.log(enbuyuk);
```

Diziler

PRACTISE

const fiyatlar = [123, 5666, 126, 67];

Şeklindeki dizi elemanlarının değerini %20 artıran fonksiyonu yazınız.