

# Web Programming 20/21

---

**Modül:** Web Programing

**Konu:** Research Js 2

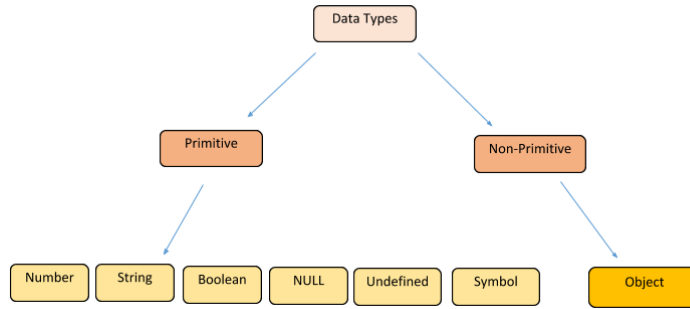
**Eğitmen:** Mehmet Doğan

**Hazırlayan:** Erhan Kudretli

# ICINDEKILER

<b>ICINDEKILER</b>	<b>2</b>
<b>Number</b>	<b>3</b>
<b>BigInt</b>	<b>4</b>
<b>Boolean</b>	<b>4</b>
<b>String</b>	<b>4</b>
<b>Null</b>	<b>5</b>
<b>Undefined</b>	<b>5</b>
<b>Object</b>	<b>5</b>
<b>Symbol</b>	<b>5</b>

# Javascript Data Yapilari



Program geliştirirken farklı farklı veri türleri kullanırız. Örneğin isim kolonu için Erhan, Ali gibi metinler yazarken, yaşları ya da telefon numaralarını yazarken 12345 gibi numaralardan faydalanırız. Fakat bilgisayar bunların arasındaki farkı anlayamaz. Bizim önceden hangi türde bir veri olduğunu bilgisayara söylememiz gerekir. Bunun için de veri yapılarını bilmemiz gerekir. Veri türleri 2 temel başlığa ayrılır. Primitive ve non-primitive. Önce bunları açıklayarak başlayalım.

**Primitive Data Types:** Bu veri türünde değişken sonradan değiştirilemez değerler alır.

**Non-Primitive Data Types:** Nesneler, diziler ve fonksiyonlar bu veri türündedir, Bu tür veriler program içerisinde değiştirilebilir.

Aşağıda 8 tane veri yapısının tanımını yapıyoruz.

## 1)Number

Veri türünün rakam olduğu yapıdır. EcmaScript standartlarına göre,  $(2^{53} - 1)$  and  $2^{53} - 1$  arasında değer alır. Ayrıca +Infinity, -Infinity, and NaN(none a number) şeklinde temsili 3 değeri de bulunur. Verilen sayıların tek türü vardır. Ondalıkli ya ondaliksiz yazılabilir.

Ondalık kısım nokta (.) ile ayrılır.

let sayi=24;

## 2)BigInt

$2^{53} - 1$  den daha büyük sayıları ifade etmek için bu veri türü kullanılır. Keyfi olarak  $2^{53} - 1$  ten daha küçük sayılar için de kullanılabilir. Bunun kullanarak hafızada daha büyük bir yer tutmuş oluruz.

```
const büyükSayı= BigInt(9007199254740991);
```

## 3)Boolean

Program geliştirirken bazen sadece 2 değeri olan yapılar kullanmamız gerekebilir. Doğru yanlış gibi. Bu durumda kullanacağımız veri türünün adıdır ve yalnızca 2 değeri vardır.

True/False

örnek;

```
let iceride = true;
```

```
let disarida= false;
```

## 4)String

Harflerin yan yana gelmesiyle oluşan türdür. Örneğin, "Ahmet", "Gezegen1234" gibi değerler alır. Metin yazıtı iki tırnak ya da tek tırnak içerisine yazılır.

örnek;

```
let isim= "erhan";
```

## 5)Null

Javascript te null ifadesi hiçbir sey anlamina gelir. Yasamayan degisken manasindadir.

Fakat degerinin olmamasi nesneyi obje olmaktan kurtarmaz. Degeri null olsa bile öyle bir nesne hala vardir anlamindadir.

```
Let kisi = { isim:"Erhan", SoyIsim:"K", yas:25,};  
kisi= null; (kisinin degeri yoktur fakat nesne hala vardir.)
```

## 6)Undefined

Yeni olusturulmus ya da olusturulmus fakat deger atanmamis degiskenlere atanan degerdir.  
ornek;

```
le tx; //degeri olmayan bir x degiskeni olusturalim console.log("x'in degeri", x) //consol  
log da x in degerinin undefined görürüz.
```

## 7)Object

Nesne anlamina gelir, birden fazla degiskene sahip olabilir.

örneğin;

```
var kisi = { ad:"Ali", soyad:"Veli", yas: 40};
```

## 8)Symbol

Symbol, ECMAScript 2015'teki JavaScript güncellemesi ile birlikte gelmistir. Symbol, benzersiz ve değişmez ilkel bir değerdir ve bir nesne özelliğinin anahtarı olarak kullanılabilir