

Meta Android Developer Professional Certificate - JavaScript Notlarım

Meta Android Developer Professional Certificate - JavaScript Notlarım

Tarih: 18.11.2025

Konular: Fonksiyonlar, Array'ler, String Manipülasyonu, Hata Yönetimi, Boş Değerler

Fonksiyonlar (Functions)

Fonksiyonlar aslında **DRY** prensibi için var - "Don't Repeat Yourself". Yani aynı kodu tekrar tekrar yazmamak için.

Temel fonksiyon yapısı:

```
function addTwoNums() {  
  let a = 10;  
  let b = 20;  
  let c = a + b;  
  console.log(c);  
}
```

Ama bu fonksiyon hep aynı sayıları toplar, çok kullanışlı değil. **Parametreler** ile daha esnek hale getirebiliriz:

```
function addTwoNums(a, b) {  
  let c = a + b;  
  console.log(c);  
}
```

Buradaki **a** ve **b** **parametre**, fonksiyonu çağırırken verdiğimiz değerler de **argüman** oluyor.

Fonksiyon çağırma:

```
addTwoNums(5, 10); // 15  
addTwoNums(7, 3); // 10
```

Array'ler (Diziler)

Array'ler bir grup veriyi sıralı şekilde saklamak için. Tren vagonları gibi düşün - her vagonun numarası var (index) ve içinde farklı yük var.

Array oluşturma:

```
let train1 = ["wheat", "coal", "wood"];
```

Array elemanlarına erişim:

```
console.log(train1[0]); // "wheat"
console.log(train1[1]); // "coal"
```

Array'lerin **index** numaraları 0'dan başlıyor!

String Manipülasyonu

String'ler aslında array gibi davranıyor - **iterable** yani üzerinde döngüyle gezinebiliyoruz.

```
let greeting = "howdy";
console.log(greeting.length); // 5
console.log(greeting[0]); // "h"
```

Ama string'ler array DEĞİL! Array method'larını kullanamıyoruz:

```
let greet = "hello";
// greet.pop() // HATA! Çalışmaz
```

String birleştirme için iki yol:

```
// + operatörü
console.log(greet + " Lisa"); // "hello Lisa"

// concat() methodu
console.log(greet.concat(" Lisa")); // "hello Lisa"
```

Math Object ve Random Sayılar

Random sayı üretme:

```
let decimal = Math.random(); // 0 ile 0.99 arası
let bigger = decimal * 10;    // 0 ile 9.99 arası
```

Yukarı yuvarlama:

```
let rounded = Math.ceil(0.0001); // 1
let rounded2 = Math.ceil(0.99);   // 1
```

```
let rounded3 = Math.ceil(2.01); // 3
```

Random integer üretme:

```
let decimal = Math.random() * 10;  
let rounded = Math.ceil(decimal);  
console.log(rounded); // 0-10 arası tam sayı
```

Hata Yönetimi (Error Handling)

Bug vs Error:

- **Bug:** Kod çalışıyor ama yanlış çalışıyor (beklenmeyen sonuç)
- **Error:** Kod tamamen duruyor

Common error types:

- **ReferenceError:** Tanımlanmamış değişken kullanma
- **SyntaxError:** Yazım hatası
- **TypeError:** Yanlış türde işlem yapma

Try-Catch ile hata yakalama:

```
try {  
  // Hata verebilecek kod  
  console.log(a + b);  
} catch(err) {  
  // Hata olursa buraya düşer  
  console.log("Hata oldu:", err);  
}  
console.log("Program devam ediyor");
```

Kendi hatamızı fırlatma:

```
try {  
  throw new ReferenceError("Bu benim hatam!");  
} catch(err) {  
  console.log("Yakalanan hata:", err);  
}
```

Boş Değerler (Empty Values)

null: Bilerek konulan boş değer

```
let result = "abc".match(/d/); // null - bulunamadı
```

undefined: Henüz değ er atanmamıř

```
let noise; // undefined
console.log(noise); // undefined

noise = "thunder";
console.log(noise); // "thunder"
```

Empty string: Boř string

```
let emptyString = "";
```

Ek String Methodları - Ne İře Yarar & Nasıl Kullanılır

Method	Ne İře Yarar?	Nasıl Kullanılır?
slice()	String'in bir b�l�m�n� alır	"hello".slice(1, 3) → "el"
push()	Array'e eleman ekler	[1,2].push(3) → [1,2,3]
charAt()	Belirtilen index'teki karakteri verir	"hello".charAt(1) → "e"
concat()	String'leri birleřtirir	"hello".concat(" world") → "hello world"
indexOf()	Aranan değ�erin ilk bulunduęu index'i verir	"hello".indexOf("l") → 2
lastIndexOf()	Aranan değ�erin son bulunduęu index'i verir	"hello".lastIndexOf("l") → 3
split()	String'i b�l�p array yapar	"a,b,c".split(",") → ["a","b","c"]
toUpperCase()	String'i b�y�k harfe �evirir	"hello".toUpperCase() → "HELLO"
toLowerCase()	String'i k��k harfe �evirir	"HELLO".toLowerCase() → "hello"
match()	Regex ile eřleřenleri bulur	"abc".match(/a/) → ["a"]