

Meta Android Developer Professional Certificate

- JavaScript Ders Notları**

Meta Android Developer Professional Certificate - JavaScript Ders Notları

17 Mayıs 2024

Bu notlar, **Meta Android Developer Professional Certificate** programındaki JavaScript modüllerinden çıkarılmıştır. Konular:

- JavaScript'e giriş, kullanım alanları, tarihçe
- VS Code kurulumu, Node.js, Code Runner eklentisi
- Değişkenler, veri tipleri, operatörler
- Koşullu ifadeler (if/else, switch) ve döngüler (for, while, iç içe döngüler)

JavaScript Nedir & Neden Bu Kadar Popüler?

- JavaScript olmadan web'de yaptığımız çoğu şey (arama, alışveriş, sosyal medya) çalışmaz.
- **Dropdown menüler, otomatik tamamlama** gibi interaktif öğeler JS sayesinde çalışır.
- Sadece **front-end** değil, **back-end** ve **mobil uygulama** (React Native) geliştirmede de kullanılıyor.
- *Murtadha (Meta'da yazılım mühendisi) diyor ki: "JavaScript artık o kadar yaygın ki, kahve makinesinde bile çalışıyor olabilir."*
- Web geliştirmede **tek dil** olması onu benzersiz kılmıyor. Diğer alanlarda (oyun, masaüstü) seçenek var ama web'de yok.

Not: JavaScript'in garip davranışları olabilir (Java/C++'a kıyasla) ama öğrenmesi daha kolay. Pes etmemek lazım!

Kurulumlar

- **VS Code:** code.visualstudio.com'dan indir. Kurulumda "desktop icon" ve "Open with Code" seçeneklerini işaretle.
- **Code Runner eklentisi:** Extensions panelinden arat ve kur.
- **Node.js:** nodejs.org'dan indir. Terminalde `node -v` yazarak kontrol et.
- **Test kodu:**

```
console.log('Hello World');
```

Çıktıyı Code Runner ile al (Play ikonu veya Ctrl+Alt+N).

console.log nedir? → Konsola yazdırma yapar. Debug için sık kullanılır.

Değişkenler (Variables)

- `var` ile tanımlanır (sonra `let/const` göreceğiz).
- Örnek:

```
var person = "John";
var greeting = "Hello";
console.log(greeting, person); // Hello John
```

- Değişkenleri yeniden atayabiliriz:

```
greeting = "Hi";
person = "James";
console.log(greeting, person); // Hi James
```

- Aynı kod, farklı sonuç! Değişkenlerin gücü bu.

Veri Tipleri (Data Types)

7 tane **primitive** (ilkel) veri tipi var:

1. **String**: Metin. `' '` veya `'''` içinde.
2. **Number**: Sayı (tam veya ondalıklı).
3. **Boolean**: `true` veya `false`.
4. **Null**: Bilerek atanmış boş değer.
5. **Undefined**: Tanımlanmamış değişken.
6. **BigInt**: Çok büyük sayılar.
7. **Symbol**: Benzersiz identifier.

Örnek:

```
var guitarName = "Stratocaster"; // string
var price = 375; // number
var isAvailable = true; // boolean
```

Not: `typeof` ile tip kontrol edilebilir, sonra bakacağım.

Operatörler

- **Aritmetik**: `+`, `-`, `*`, `/`, `%` (modül - kalan), `**` (üs).
- **Karşılaştırma**: `>`, `<`, `==`, `===`, `!=`, `!==`.
- `==` ile `===` farkı:

```
100 == "100"    // true (değer aynı)
100 === "100"   // false (tip farklı)
```

- Mantıksal: `&&` (ve), `||` (veya), `!` (değil).
-

Koşullu İfadeler (Conditionals)

- `if/else`:

```
var result = 50;
if (result > 40) {
    console.log("You pass the test");
} else {
    console.log("You did not pass");
}
```

- `else if` ile çoklu koşul:

```
var place = "first";
if (place === "first") {
    console.log("Gold");
} else if (place === "second") {
    console.log("Silver");
} else {
    console.log("No medal");
}
```

- `switch`: Aynı işi daha düzenli yapar:

```
switch (place) {
    case "first":
        console.log("Gold");
        break;
    case "second":
        console.log("Silver");
        break;
    default:
        console.log("No medal");
}
```

Break koymazsan sonraki case'e geçer, dikkat!

Döngüler (Loops)

- `for loop`:

```
for (var i = 1; i <= 3; i++) {
    console.log(i);
}
// Çıktı: 1, 2, 3
```

- **while loop:**

```
var counter = 3;
while (counter > 0) {
    console.log(counter);
    counter--;
}
console.log("Happy New Year!");
```

- **İç içe döngüler (Nested loops):**

```
for (var i = 2023; i < 2025; i++) {
    console.log(i);
    for (var j = 6; j < 9; j++) {
        console.log("--" + j);
    }
}
```

Cıktı: 2023, --6, --7, --8, 2024, --6, --7, --8

Dikkat: İç içe döngüler performansı düşürebilir, gereksiz kullanmayalım.
