

WEB PROGRAMLAMA I ■ HAFTA 10

JavaScript Temelleri

■ Değişkenler ■ Veri tipleri ■ Operatörler ■ Koşullar ■ Döngüler

Prof. Dr. Özal YILDIRIM

Fırat Üniversitesi — Teknoloji Fakültesi

Yazılım Mühendisliği Bölümü

Üçüncü Katman: Davranış

</> HTML

Yapı

Evin iskeleti, duvarları, odaları.

Hafta 2–4

🖌 CSS

Görünüm

Duvar kâğıdı, boya, mobilya, aydınlatma. **Hafta 5–9**

⚡ JavaScript

Davranış

Elektrik tesisatı, zil, garaj kapısı, alarm sistemi. **Hafta 10–14**

*JavaScript, siteyi **statik bir broşür** olmaktan çıkarıp **yaşayan, interaktif bir uygulamaya** dönüştüren her şeydir.*

HTML ve CSS'in aksine JavaScript, koşullar, döngüler ve fonksiyonlar gibi özelliklere sahip **gerçek bir programlama dilidir**.

- ▶ JavaScript'i harici dosya olarak sayfaya doğru şekilde bağlayabilecek,
- ▶ `console.log()` ve tarayıcı konsolunu hata ayıklamada kullanabilecek,
- ▶ `let` ve `const` arasında doğru seçim yapabilecek,
- ▶ Temel veri tiplerini ve tip dönüşümlerini açıklayabilecek,
- ▶ Aritmetik, karşılaştırma ve mantıksal operatörleri kullanabilecek,
- ▶ `==` ile `===` arasındaki kritik farkı açıklayabilecek,
- ▶ `if/else` ve `switch` ile karar yapıları kurabilecek,
- ▶ `for`, `while` ve `for...of` döngülerini yerinde kullanabileceksiniz.

Bu Haftanın İçeriği

- 1 Başlangıç
- 2 Değişkenler ve Veri Tipleri
- 3 Operatörler
- 4 Koşullar
- 5 Döngüler
- 6 Yapay Zeka ile JavaScript
- 7 Özet ve Değerlendirme



Başlangıç

BÖLÜM 1

BookVerse projemiz şu an ne kadar güzel görünürse görünsün, **sınırları** vardır:

✗ Yapamadıklarımız

- ▶ Butona tıklayınca **hiçbir şey olmaz**
- ▶ Form bilgilerini sunucuya göndermeden kontrol edemeyiz
- ▶ Yeni içerik için sayfanın **tamamen yeniden yüklenmesi** gerekir
- ▶ Canlı saat, galeri, sayac gibi dinamik bileşen yapamayız

🔑 JavaScript'in Gücü: DOM

Hafta 1'de öğrendiğimiz **DOM ağacına tam erişim** sağlar:

- ▶ Yeni element **ekleyebilir/silebilir**
- ▶ İçeriği **değiştirebilir**
- ▶ CSS stillerini **değiştirebilir**
- ▶ Kullanıcı **olaylarını dinleyebilir**

Dönem hedefimiz: Mobil ekranda gizlenen bir menüyü “hamburger” butonuyla açıp kapatmak (Hafta 12).

JavaScript'i Sayfaya Bağlamak

Aynen CSS gibi, en iyi yöntem **ayrı bir .js dosyası** kullanmaktır.

```
1   </footer>
2
3   <script src="js/main.js"></script>
4 </body>
5 </html>
```

⚠ Dikkat

Önemli fark: CSS'in `<link>` etiketi `<head>`'e konur; `<script>` etiketi ise genellikle `</body>` kapanışından **hemen önce** yerleştirilir. **Neden?** Tarayıcının JavaScript'i çalıştırmadan önce **tüm DOM yapısını hazırlaması** için. Aksi halde JS, henüz var olmayan elementlere ulaşmaya çalışır ve hata verir.

```
<!-- Alternatif: head içinde ama ertelenmiş -->
<script src="js/main.js" defer></script>
```

console.log(): En İyi Dostunuz

```
1 // Bu bir yorum satiridir. Kodun calistigini test edelim.
2 console.log("JavaScript dosyasi basariyla calisti!");
3
4 /* Çok satirli
5    yorum */
6
7 console.log("Kitap:", kitapAdi, "Fiyat:", fiyat); // birden fazla deger
8 console.error("Bir hata olustu!");                // kirmizi
9 console.warn("Dikkat: eski fonksiyon");           // sari
10 console.table(kitaplar);                          // dizi/nesne icin TABLO
11 console.log({ kitapAdi, fiyat });                 // etiketli goruntuleme
```

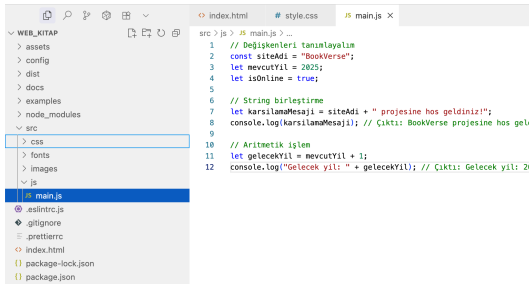
💡 İpucu

Sayfayı açın, **F12** → **Console** sekmesi. Mesajınızı orada görmelisiniz — bu, her şeyin doğru bağlandığını teyit eder.

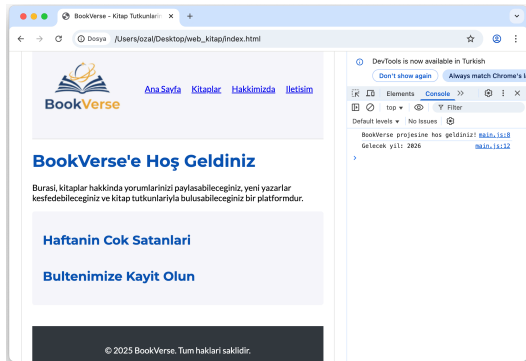
>_ Konsol Bir REPL'dir

Console sekmesine **doğrudan kod yazabilirsiniz**. Değişken tanımlayın, hesap yaptırın, DOM'u sorgulayın — anında sonuç alırsınız. Bu hafta **konsolda çok vakit geçireceğiz**.

VS Code'da Kurulum ve Konsol Çıktısı



main.js dosyası ve `<script>` etiketi



Konsolda console.log() çıktısı



Değişkenler ve Veri Tipleri

BÖLÜM 2

let ve const

```
1 const siteAdi = "BookVerse";    // DEĞİSMEZ - sabit
2 let mevcutYil = 2026;          // degistirilebilir
3 let isOnline = true;
4
5 mevcutYil = 2027;              // OK
6 siteAdi = "Kitapla";           // HATA: Assignment to constant variable
```

Anahtar	Ne zaman?	Kapsam (scope)
const	Varsayılan tercihiniz. Değer bir daha atanmayacaksa	Blok ({ }
let	Değer sonradan değişecekse (sayac, durum)	Blok ({ }
var	Kullanmayın. Eski kodlarda görürsünüz	Fonksiyon — kafa karıştırıcı

İpucu

Kural: Her zaman `const` ile başlayın. Kod yazarken “bunu değiştirmem gerekiyor” dersiniz o zaman `let`’e çevirin. Bu alışkanlık, **istmeden değer değiştirme** hatalarının çoğunu önler.

Temel Veri Tipleri

```
1  const metin      = "Simyaci";           // string - metin
2  const sayi       = 89.90;               // number - tam veya ondalikli
3  const dogruMu    = true;                // boolean - true / false
4  let tanimsiz;                             // undefined - deger atanmamis
5  const bosDeger   = null;                // null - BILEREK bos birakilmis
6  const buyukSayi  = 9007199254740993n;   // bigint - cok buyuk tam sayi
7  const benzersiz  = Symbol("id");        // symbol - benzersiz tanimlayici
8
9  // Nesne tipleri
10 const dizi       = ["Simyaci", "Kurk Mantolu", "Tutunamayanlar"];
11 const nesne      = { ad: "Simyaci", yazar: "Paulo Coelho", yil: 1988 };
12
13 console.log(typeof metin);               // "string"
14 console.log(typeof sayi);                // "number"
15 console.log(typeof dizi);                // "object" (!)
16 console.log(Array.isArray(dizi));        // true - dizi kontrolu Boyle yapilir
```

Dikkat: `typeof null` → `"object"` döner. Bu, JavaScript'in ilk sürümünden kalma **bilinen bir hatadır** ve geriye uyumluluk için düzeltilmemiştir.

Metin (String) İşlemleri

```
1  const yazar = "Paulo Coelho";
2
3  yazar.length;           // 12
4  yazar.toUpperCase();    // "PAULO COELHO"
5  yazar.toLowerCase();    // "paulo coelho"
6  yazar.includes("Coelho"); // true
7  yazar.startsWith("Paulo"); // true
8  yazar.slice(0, 5);      // "Paulo"
9  yazar.split(" ");      // ["Paulo", "Coelho"]
10 yazar.trim();           // bastaki/sondaki boslugu siler
11 yazar.replace("Paulo", "P."); // "P. Coelho"
12
13 // SABLON LITERALI (template literal) - ters tirnak `
14 const kitap = "Simyaci";
15 const fiyat = 89.90;
16 const mesaj = `${kitap} kitabının fiyatı ${fiyat} TL'dir.`;
17 const cokSatir = `Birinci satir
18   Ikinci satir`;
```



İpucu

Tip Dönüşümleri

```
1 // Formdan gelen her deger METINDIR!  
2 const girdi = "25";  
3  
4 Number(girdi);           // 25           (sayiya cevir)  
5 parseInt("25.9");        // 25           (tam sayi)  
6 parseFloat("25.9");      // 25.9        (ondalikli)  
7 Number("abc");           // NaN          (Not a Number)  
8  
9 String(42);              // "42"  
10 (42).toString();         // "42"  
11 (89.9).toFixed(2);       // "89.90"   (2 basamak, METIN doner)  
12  
13 Boolean("");             // false  
14 Boolean("merhaba");      // true  
15 Boolean(0);              // false
```

⚠ Klasik Hata

```
const a = "5", b = "3";  
a + b;           // "53" (birlestirme!)  
Number(a) + Number(b);
```

Web Programlama (a) | HTML5 (b) | JavaScript Temelleri

👻 “Falsy” Değerler

Koşullarda **yanlış** sayılan 6 değer:
false, 0, "", null, undefined, NaN Diğer her şey doğrudur — “0” [1-3] dahil

Prof. Dr. Özal YILDIRIM

11 / 32



Operatörler

BÖLÜM 3

Aritmetik ve Atama Operatörleri

```
1 let a = 10, b = 3;
2
3 a + b;    // 13   toplama
4 a - b;    // 7    cikarma
5 a * b;    // 30   carpma
6 a / b;    // 3.333... bolme
7 a % b;    // 1    MOD - kalan
8 a ** b;   // 1000 us alma
9
10 // String ile + BIRLESTIRIR
11 "Book" + "Verse";    // "BookVerse"
12
13 // Kisa atama
14 let sayac = 0;
15 sayac += 5;    // sayac = sayac + 5 -> 5
16 sayac -= 2;    // 3
17 sayac *= 4;    // 12
18 sayac++;       // 13 (bir artir)
19 sayac--;       // 12 (bir azalt)
```

% (mod) operatörü çok işe yarar: $i \% 2 === 0 \rightarrow$ çift sayı kontrolü; $i \% 3 === 0 \rightarrow$ her üçüncü öğe.

Karşılaştırma: == vs ===

```
1 5 == "5";      // true   - TIP DÖNÜŞÜMÜ yapar (gevsek esitlik)
2 5 === "5";     // false  - tip de kontrol eder (KATI esitlik)
3
4 0 == false;    // true
5 0 === false;   // false
6
7 null == undefined; // true
8 null === undefined; // false
9
10 5 != "5";      // false
11 5 !== "5";     // true
12
13 10 > 5;  10 >= 10;  3 < 7;  3 <= 2;
```

⚠ Dikkat

Altın kural: Her zaman **===** ve **!==** kullanın. **==** operatörü tip dönüşümü yapar ve **öngörülemez sonuçlar** üretir. Profesyonel projelerde kod denetçileri (ESLint) **==** kullanımını **hata** olarak işaretler.

Mantıksal Operatörler

```
1  const yas = 20, uyeMi = true;
2
3  yas >= 18 && uyeMi;      // VE - ikisi de dogruysa true
4  yas < 18 || uyeMi;       // VEYA - biri dogruysa true
5  !uyeMi;                  // DEGIL - tersine ceviriir
6
7  // Kisa devre (short-circuit) degerlendirme
8  const ad = kullanıcıAdi || "Misafir";      // bos ise "Misafir"
9  const sehir = kullanıcı?.adres?.sehir;     // guvenli erisim (?)
10 const puan = gelenPuan ?? 0;               // null/undefined ise 0
11
12 // Ternary (uclu) operator - kisa if/else
13 const durum = yas >= 18 ? "Yetiskin" : "Cocuk";
```

?. — Güvenli Erişim

kullanıcı.adres.sehir yazarsanız
ve adres yoksa **program** **çöker.**
kullanıcı?.adres?.sehir yazarsanız
undefined döner — program devam eder.

? || vs ??

0 || 10 → **10** (0 "falsy")
0 ?? 10 → **0** (0 null değil) Sayısal değerlerde ??
kullanın — 0 geçerli bir değerdir.



Koşullar

BÖLÜM 4

if / else if / else

```
1  const puan = 8.4;
2
3  if (puan >= 9) {
4    console.log("Mukemmel kitap");
5  } else if (puan >= 7) {
6    console.log("Iyi kitap");
7  } else if (puan >= 5) {
8    console.log("Idare eder");
9  } else {
10   console.log("Onerilmez");
11 }
12
13 // Tek satirlik kontrol
14 if (!kullaniciGirisYapti) return;
15
16 // Ternary ile ayni is
17 const sonuc = puan >= 7 ? "Onerilir" : "Onerilmez";
```

💡 İpucu

Sıra önemlidir: Koşullar yukarıdan aşağıya kontrol edilir ve **ilk doğru olan** çalışır; geri kalanlara bakılmaz. **puan >= 5** koşulunu en üste yazsaydınız, **8 puanlık kitap da "Idare eder" olurdu**.

switch: Çok Dallı Seçim

```
1  const kategori = "roman";
2
3  switch (kategori) {
4      case "roman":
5          console.log("Roman bolumune yonlendiriliyorsunuz");
6          break;
7      case "siir":
8      case "deneme":           // birden fazla case birlestirilebilir
9          console.log("Edebiyat bolumu");
10         break;
11     case "bilim":
12         console.log("Bilim bolumu");
13         break;
14     default:
15         console.log("Kategori bulunamadi");
16 }
```

⚠ break Unutmayın

break yazmazsanız, eşleşen **case**'ten sonraki **tüm** bloklar da çalışır (*fall-through*). Bu, en sık yapılan

💡 İpucu

switch, **katı eşitlik** (**===**) kullanır. Aralık kontrolü

(**puan >= 7**) gerekiyorsa **if/else if** kullanın

Prof. Dr. Özal YILDIRIM

16 / 32



Döngüler

BÖLÜM 5

for Döngüsü

```
1 // baslangic ; kosul ; artirim
2 for (let i = 1; i <= 5; i++) {
3     console.log("Sayfa " + i);
4 }
5
6 const kitaplar = ["Simyaci", "Kurk Mantolu", "Tutunamayanlar"];
7
8 // Klasik indeksli dongu
9 for (let i = 0; i < kitaplar.length; i++) {
10     console.log(`${i + 1}. ${kitaplar[i]}`);
11 }
12
13 // MODERN: for...of - degerler uzerinde gez
14 for (const kitap of kitaplar) {
15     console.log(kitap);
16 }
17
18 // for...in - nesne ANAHTARLARI uzerinde gez
19 const kitap = { ad: "Simyaci", yazar: "Coelho", yil: 1988 };
20 for (const anahtar in kitap) {
21     console.log(`${anahtar}: ${kitap[anahtar]}`);
22 }
```

while ve do...while

```
1 // Kosul ONCE kontrol edilir - hic calismayabilir
2 let sayac = 0;
3 while (sayac < 3) {
4     console.log("Sayac: " + sayac);
5     sayac++; // ARTIRMAYI UNUTMAYIN!
6 }
7
8 // Kosul SONRA kontrol edilir - en az BIR KEZ calisir
9 let cevap;
10 do {
11     cevap = prompt("Devam? (e/h)");
12 } while (cevap !== "h");
13
14 // Dongu kontrolu
15 for (const kitap of kitaplar) {
16     if (kitap === "Gizli") continue; // bu adimi ATLA
17     if (kitap === "Son") break; // dongudan CIK
18     console.log(kitap);
19 }
```



Dikkat

Uygulama: BookVerse Kitap Listesi

```
1  const kitaplar = [
2    { ad: "Simyaci",          yazar: "Paulo Coelho",    fiyat: 89.90,  puan: 8.4 },
3    { ad: "Kurk Mantolu",     yazar: "Sabahattin Ali",    fiyat: 45.00,  puan: 9.1 },
4    { ad: "Tutunamayanlar",   yazar: "Oguz Atay",        fiyat: 120.00, puan: 9.5 },
5    { ad: "Seker Portakali",  yazar: "J.M. Vasconcelos", fiyat: 65.50,  puan: 8.8 }
6  ];
7
8  console.log(`Toplam ${kitaplar.length} kitap listeleniyor:\n`);
9
10 let toplamFiyat = 0;
11
12 for (const kitap of kitaplar) {
13   // Puana gore etiket belirle
14   let etiket;
15   if (kitap.puan >= 9)      etiket = "COK BEGENILDI";
16   else if (kitap.puan >= 8) etiket = "Begenildi";
17   else                     etiket = "Ortalama";
18
19   console.log(`${kitap.ad} - ${kitap.yazar}`);
20   console.log(`  Fiyat: ${kitap.fiyat.toFixed(2)} TL | Puan: ${kitap.puan} | ${etiket}`);
21
22   toplamFiyat += kitap.fiyat;
23 }
24
25 console.log(`\nToplam: ${toplamFiyat.toFixed(2)} TL`);
26 console.log(`Ortalama: ${(toplamFiyat / kitaplar.length).toFixed(2)} TL`);
```

Hata Ayıklama: Konsolu Okumak

Hata Mesajı	Anlamı ve Çözümü
Uncaught ReferenceError: x is not defined	Tanımlanmamış değişken. Yazım hatası veya değişken henüz tanımlanmamış.
Uncaught TypeError: Cannot read properties of null	null üzerinde özellik okumaya çalıştınız. Genelde DOM elementi bulunamadı demektir.
Uncaught SyntaxError: Unexpected token	Eksik parantez, süslü parantez veya tırnak.
NaN	Sayısal olmayan bir değerle hesap yaptınız. Number() ile dönüştürün.
undefined	Değişken tanımlı ama değer atanmamış.

İpucu

Hata mesajının sağındaki `main.js:14` bağlantısına tıklayın — doğrudan hatalı satıra gider.

Önemli alışkanlık: Hata mesajını **okuyun**. Çoğu öğrenci mesajı okumadan koda bakar — oysa

Kullanıcı ile Etkileşim: alert, prompt, confirm

```
1 alert("Kayıt başarıyla tamamlandı!");           // sadece bilgi verir
2
3 const ad = prompt("Adınız nedir?");             // metin ister -> string
4 const yas = Number(prompt("Yasınız?"));          // SAYIYA CEVIRIN!
5
6 const eminMi = confirm("Silmek istediğinize emin misiniz?");
7 if (eminMi) {
8     console.log("Silindi");
9 } else {
10    console.log("İptal edildi");
11 }
```

⚠ Öğrenme Aracı, Ürün Değil

Bu üç fonksiyon **sayfayı dondurur** (blocking). Tasarımları değiştirilemez, mobilde kötü görünür. Gerçek projelerde **kendi modal pencerelerinizi** yaparsınız. Bu hafta sadece **öğrenme amacıyla** kullanıyoruz.

💡 İpucu

prompt() her zaman **metin döner** — “25” girseniz bile. **Number()** ile çevirmezseniz:
"25" + 1 → "251"
Number("25") + 1 → 26 Kullanıcı “İptal” derse **null** döner.

Sayılarla Çalışmak: Math Nesnesi

```
1 Math.round(8.5);      // 9      en yakin tam sayi
2 Math.floor(8.9);      // 8      asagi yuvarla
3 Math.ceil(8.1);       // 9      yukari yuvarla
4 Math.trunc(8.9);      // 8      ondalik kismi at
5
6 Math.abs(-15);        // 15     mutlak deger
7 Math.max(3, 9, 1);    // 9
8 Math.min(3, 9, 1);    // 1
9 Math.pow(2, 10);      // 1024   (2 ** 10 ile ayni)
10 Math.sqrt(144);      // 12
11
12 Math.random();        // 0 ile 1 arasinda rastgele ondalikli
13
14 // 1-6 arasi rastgele tam sayi (zar)
15 const zar = Math.floor(Math.random() * 6) + 1;
16
17 // Diziden rastgele bir eleman
18 const rastgeleKitap = kitaplar[Math.floor(Math.random() * kitaplar.length)];
```



Dikkat

Tarih ve Saat: Date

```
1  const simdi = new Date();
2
3  simdi.getFullYear();    // 2026
4  simdi.getMonth();       // 8    -> DIKKAT: 0 = Ocak, 8 = Eylul!
5  simdi.getDate();        // 6    ayin gunu
6  simdi.getDay();         // 0    -> 0 = Pazar
7  simdi.getHours();       // 21
8  simdi.getMinutes();     // 45
9
10 // Turkce bicimlendirme
11 simdi.toLocaleDateString("tr-TR");
12 // "6.09.2026"
13
14 simdi.toLocaleDateString("tr-TR", {
15   day: "numeric", month: "long", year: "numeric", weekday: "long"
16 });
17 // "6 Eylul 2026 Pazar"
18
19 // Belirli bir tarih
20 const yayinTarihi = new Date(1988, 0, 15);    // 15 Ocak 1988 (ay 0-tabanlı!)
```

En sık hata: `getMonth()` 0'dan başlar. Eylül ayı için 8 döner. Ekranda göstermek için `getMonth() + 1`

Yazım Kuralları ve İyi Alışkanlıklar

Kural	Uygulama
camelCase kullanın	<code>kitapAdi, toplamFiyat, kullanıcıGirisYapti</code>
Sabitler için BUYUK_HARF	<code>const MAX_KITAP = 100;</code>
Anlamlı isim verin	<code>x</code> değil <code>kitapSayisi</code> ; <code>a</code> değil <code>aktifKullanici</code>
Boolean'lar oru gibi	<code>isAktif, hasIzin, kayitliMi</code>
Fonksiyonlar eylem gibi	<code>hesaplaToplam(), getKitap()</code>
Noktalı virgül	Tutarlı olun — ya hep koyun ya hiç
Girinti	2 boşluk (JS topluluğunun standardı)
Türkçe karakter	Değişken adlarında kullanmayın (<code>fiyat</code> , <code>fiyatı</code> değil)

İpucu

Neden önemli? 6 ay sonra kendi kodunuza döndüğünüzde `let a = 5;` satırının ne olduğunu



Yapay Zeka ile JavaScript

BÖLÜM 6

JavaScript, yapay zeka modellerinin **en çok eğitim verisi gördüğü** dillerden biridir. Bu hem fırsat hem **tehlike** anlamına gelir.

⚠ Gerçek Tehlike

Bu haftadan itibaren yapay zeka **ödevinizi tamamen yapabilir**. Ancak **Hafta 12'de** DOM manipülasyonuna geçtiğimizde, temel sözdizimini bilmeyen öğrenci **üretilen kodu düzeltemez** — ve o kod mutlaka düzeltme gerektirir.

💡 Bu Hafta İçin Kural

Kod ürettirmeyin. Kod okutun. Bu haftaki tüm prompt'lar şu kalıpta:

- ▶ “Bu kod ne yazıyor?”
- ▶ “Bu kodda ne hata var?”
- ▶ “Bana alıştırma üret”
- ▶ “Çözümümü değerlendir”

Prompt 1 — Kod İzleme (Trace) Egzersizi

Yapay Zeka İstemi (Prompt) — Çıktıyı Tahmin Et

Bana JavaScript temelleri üzerine 8 adet "KOD İZLEME" sorusu hazırla.
Her soruda kısa bir kod parçası olsun ve ben ÇIKTIYI tahmin edeyim.
Konu dağılımı: - 2 soru: let/const ve kapsam (blok scope) - 2 soru: tip dönüşümü ("5" + 3 gibi tuzaklar) - 1 soru: == vs === farkı - 2 soru: dongu ve kosul akisi - 1 soru: falsy degerler
KURALLAR: - Her kod parçası en fazla 8 satır olsun - Önce SADECE kodları yaz, çıktıları YAZMA - En sona ayrı bir başlık altında cevap anahtarı koy - Cevap anahtarında her çıktı için ADIM ADIM açıklama yap - Türkçe yaz

Önemli: Önce **kendiniz tahmin edin**, sonra kodu `console.log` ile **gerçekten çalıştırın**, en son cevap anahtarına bakın.

Prompt 2 — Hata Mesajı Çözümleme

Yapay Zeka İstemi (Prompt) — Konsol Hatasını Açıklat

Tarayıcı konsolunda su hatayı aldım:

[HATA MESAJINI AYNEN YAPISTIRIN]

İlgili kod satırları:

[KODUNUZU YAPISTIRIN]

GOREV: 1) Bu hata mesajı TAM OLARAK ne diyor? Kelime kelime çevir. 2) Hatanın en olası 3 sebebini, en olasidan başlayarak sırala. 3) Her sebep için, DOGRULAMAK için konsola yazacağım bir komut öner (örneğin console.log(...) veya typeof ...). 4) Çözümü HEMEN VERME. Önce ben kontrol edeceğim. Türkçe yaz.

3. madde öğretici kısımdır: Size **balık** değil, **olta** verir. Bir sonraki hatada aynı yöntemi kendiniz uygularsınız.

Prompt 3 — Kendi Çözümünüzü Değerlendirtme

Yapay Zeka İstemi (Prompt) — Kod Eleştirisi

Universite 1. sınıf öğrencisiyim. Su problemi cozdum:

PROBLEM: Bir kitap dizisindeki kitaplardan puani 8'in uzerinde olanlari konsola yazdir ve toplam fiyatlarini hesapla.

KENDI COZUMUM: [KODUNUZU YAPISTIRIN]

Kodumu su basliklar altinda degerlendir: 1) DOGRULUK - istenen sonucu uretiyor mu? 2) OKUNABILIRLIK - degisken isimleri anlamlı mı? 3) MODERN JS - let/const, for...of, sablon literalı kullanılmış mı? 4) HATA RISKİ - hangi girdide bozulur?

KURAL: Duzeltilmis kodu VERME. Her baslik icin 1-2 cumle yorum yap ve iyilestirme YONUNU tarif et. Kodu ben yeniden yazacagim.

En sonda: "Bu kod 1. sınıf seviyesine gore kac puan alir (10 uzerinden)?" sorusunu gerekcesiyle yanitla.

Özet ve Değerlendirme

BÖLÜM 7

Kurulum

`js/main.js` + `<script src>` (`</body>` öncesi veya `defer`). `console.log()` en iyi dostunuz.

Değişkenler

Varsayılan `const`, gerekirse `let`. `var` kullanmayın.
Şablon literalı: ``${değişken}``

= Operatörler

Her zaman `===`, asla `==`. `?.` güvenli erişim, `??` varsayılan değer. 6 “falsy” değeri ezberleyin.

Yapay Zeka

Bu hafta **kod ürettirmeyin, kod okutun**. Temel sözdizimini bilmeden Hafta 12’de üretilen kodu **düzeltemezsiniz**.

Altın Kural: Hata mesajını **okuyun**. Konsol size sorunun ne olduğunu ve hangi satırda olduğunu **açıkça söyler**.

Terim	Anlam	Terim	Anlam
<code>const / let</code>	Sabit / değişken tanımı	Falsy	Koşulda yanlış sayılan değer
<code>typeof</code>	Veri tipini döndürür	NaN	“Not a Number”
String	Metin verisi	===	Katı eşitlik (tip dahil)
Number	Sayısal veri	==	Gevşek eşitlik (kullanmayın)
Boolean	<code>true / false</code>	?.	Güvenli özellik erişimi
<code>null</code>	Bilerek boş	??	Null birleştirme
<code>undefined</code>	Değer atanmamış	Ternary	<code>koşul ? a : b</code>
Şablon literalı	<code>`\${x}`</code> yazımı	break	Döngü kontrolü
		continue	

1 Aşağıdaki kodun çıktısı nedir? Neden?

```
console.log("5" + 3);  
console.log("5" - 3);  
console.log(5 == "5", 5 === "5");
```

2 `const` ile tanımlanan bir dizinin içeriği değiştirilebilir mi? Test edip açıklayınız.

3 JavaScript'te **altı falsy değeri** yazınız.

4 `<script>` etiketi neden `</body>` kapanışından hemen önce yazılır? `defer` bunu nasıl değiştirir?

5 1'den 20'ye kadar olan sayılardan sadece **3'e tam bölünenleri** yazdıran bir döngü yazınız.

6 `while` döngüsünde sonsuz döngüye düşmemek için nelere dikkat edilmelidir?

7 `0 || "varsayılan"` ve `0 ?? "varsayılan"` ifadeleri ne döner? Farkı açıklayınız.

Laboratuvar Föyü 10

- ▶ `js/main.js` kurulumu ve bağlama
- ▶ Konsolda REPL alıştırmaları
- ▶ Tip dönüşümü tuzakları
- ▶ `==` vs `===` tablosu
- ▶ Kitap listesi programı (döngü + koşul)
- ▶ Hata mesajı çözümleme
- ▶ Yapay zekaya kod izleme soruları üretirme

Hazırlık

Okuma: Kitap Ünite 2.3 — “Temel Sözdizimi”,
“Koşullar ve Döngüler” **Kaynaklar:**

- ▶ developer.mozilla.org/docs/Web/JavaScript
- ▶ javascript.info
- ▶ jsfiddle.net / codepen.io

Gelecek hafta: Fonksiyonlar, kapsam, diziler ve nesneler.

Haftaya Görüşmek Üzere!

Gelecek Hafta:

Fonksiyonlar, Kapsam, Diziler ve Nesneler

Fonksiyon tanımlama ▪ Ok fonksiyonları ▪ Kapsam ▪
Dizi metotları (, ,) ▪ Nesneler