

WEB PROGRAMLAMA I ■ HAFTA 2

HTML Temelleri ve Geliştirme Ortamı

VS Code & Live Server ■ Etiket Anatomisi ■ Belge İskeleti ■ Metin, Bağlantı, Görsel ve Listeler ■ BookVerse'e Başlangıç

Prof. Dr. Özal YILDIRIM

Fırat Üniversitesi — Teknoloji Fakültesi

Yazılım Mühendisliği Bölümü

✓ Hafta 1 Özeti

- ▶ İnternet = altyapı, Web = servis
- ▶ **HTTP**: istek-cevap döngüsü
- ▶ **DNS**: alan adı → IP
- ▶ **İstemci-Sunucu** mimarisi
- ▶ Tarayıcı: DOM + CSSOM → Render Ağacı
→ Layout → Paint

→ Bu Hafta

Geçen hafta sunucunun bize **HTML** gönderdiğini öğrendik. Bu hafta o HTML'i **biz yazmaya başlıyoruz**. Teoriden **pratige** geçiş: ilk satır kodumuzu yazacağız.

Hatırlatma: Tarayıcının oluşturduğu **DOM ağacı**, bizim yazdığımız HTML'in birebir yansımasıdır.
Kötü HTML → kötü DOM → kötü sayfa.

- ▶ VS Code'u kurup Live Server ile canlı önizleme ortamı hazırlayabilecek,
- ▶ HTML'in bir *işaretleme* dili olduğunu, programlama dilinden farkını açıklayabilecek,
- ▶ Bir HTML etiketinin anatomisini (açılış, kapanış, öznitelik) çözümleyebilecek,
- ▶ Standart HTML5 belge iskeletini ezberden yazabilecek,
- ▶ Başlık hiyerarşisini doğru kurabilecek,
- ▶ Bağlantı, görsel ve liste etiketlerini özniteliklerle birlikte kullanabilecek,
- ▶ Görelî ve mutlak dosya yollarını ayırt edebilecek,
- ▶ Yapay zekaya HTML ürettirip çıktıyı eleştirel biçimde denetleyebileceksiniz.

- 1 Geliştirme Ortamını Kurmak
- 2 HTML Nedir?
- 3 HTML5 Belge İskeleti
- 4 Metin Etiketleri
- 5 Bağlantılar ve Görseller
- 6 Listeler
- 7 Uygulama: BookVerse Ana Sayfası
- 8 Yapay Zeka ile HTML
- 9 Özet ve Değerlendirme



Geliştirme Ortamını Kurmak

BÖLÜM 1

Neden Bir Kod Editörü?

HTML aslında **düz metindir**. Not Defteri ile de yazabilirsiniz. Peki neden yazmıyoruz?

✗ Not Defteri ile

- ▶ Renklendirme yok → hata görünmez
- ▶ Otomatik tamamlama yok
- ▶ Kapanmayan etiket fark edilmez
- ▶ Girinti (indentation) elle
- ▶ Her değişiklikte elle yenileme

✓ VS Code ile

- ▶ Sözdizimi renklendirme
- ▶ **Emmet** ile saniyede iskelet
- ▶ Etiket eşleştirme ve otomatik kapatma
- ▶ Hata altı çizgisi
- ▶ **Live Server** ile anında önizleme

Editör, marangozun tezgâhıdır. Tahtayı elde de kesebilirsiniz, ama tezgâhla daha düzgün ve hızlı olur.

Adım 1: Visual Studio Code Kurulumu

- 1 <https://code.visualstudio.com> adresine gidin.
- 2 **Download** butonuna tıklayın — site işletim sisteminizi (Windows / macOS / Linux) otomatik algılar.
- 3 **Windows:** .exe dosyasını çalıştırın. Kurulumda **“Add to PATH”** ve **“Open with Code”** seçeneklerini işaretleyin.
- 4 **macOS:** .zip dosyasını açıp uygulamayı **Applications** klasörüne sürükleyin.
- 5 VS Code’u açın. Sol taraftaki dikey şeritte **Extensions** (kare ikon, **Ctrl/Cmd + Shift + X**) sekmesini bulun.

İpucu

Ücretsiz ve açık kaynaklıdır. Dünyada en çok kullanılan kod editörüdür (Stack Overflow anketlerinde yıllardır ilk sırada). İş hayatında da karşınıza çıkacak.

Adım 2: Kurulması Gereken Eklentiler

Eklenti	Yayıncı	Ne İşe Yarar?
Live Server	Ritwick Dey	Sayfayı yerel sunucuda açar, kaydettikçe otomatik yeniler . Zorunlu.
Prettier	Prettier	Kodu otomatik biçimlendirir, girintileri düzeltir.
Auto Rename Tag	Jun Han	Açılış etiketini değiştirince kapanış etiketini de değiştirir.
HTML CSS Support	ecmel	CSS sınıf adlarını HTML içinde önerir.
Turkish Language Pack	Microsoft	Arayüzü Türkçeleştirir (isteğe bağlı).
Path Intellisense	Christian Kohler	Dosya yollarını yazarken otomatik tamamlar.

Kurulum: Extensions sekmesinde adını arat → **Install** butonuna bas.

Adım 3: Proje Klasör Yapısı

Masaüstünde `bookverse` adında bir klasör oluşturun ve **VS Code ile açın** (*File → Open Folder*).

⚠ Dikkat

Klasör ve dosya adlarında:

- ▶ ❌ Türkçe karakter kullanmayın
- ▶ ❌ Boşluk kullanmayın
- ▶ ✅ Küçük harf + tire:
`ana-sayfa.html`

Sunucular büyük/küçük harfe **duyarlıdır**.

```
bookverse/
|
|-- index.html           <- Ana sayfa
|-- kitaplar.html
|-- hakkimizda.html
|-- iletisim.html
|
|-- css/
|   |-- style.css        <- Hafta 5'te
|
|-- js/
|   |-- script.js        <- Hafta 10'da
|
|-- images/
|   |-- bookverse_logo.png
|   |-- kapaklar/
```

index.html adı standarttır: sunucu, klasöre girildiğinde varsayılan olarak bu dosyayı arar.

Adım 4: Live Server ile Canlı Önizleme

Nasıl başlatılır?

- 1 `index.html` dosyasını açın
- 2 Sağ tıklayıp **“Open with Live Server”**
- 3 Veya sağ alttaki **“Go Live”** yazısına tıklayın

Tarayıcı `http://127.0.0.1:5500` adresinde açılır. Artık dosyayı her kaydettiğinizde

(Ctrl/Cmd + S) sayfa **kendiliğinden yenilenir**.



Neden 127.0.0.1?

Bu adres **“localhost”**, yani **kendi bilgisayarınız** anlamına gelir. Live Server, makinenizde küçük bir web sunucusu çalıştırır. **5500** ise **port numarasıdır** — geçen hafta öğrendiğimiz **“iletişim kapısı”**.



Dikkat

Dosyayı çift tıklayıp açmak (`file:///...`) ile **Live Server** arasında fark vardır. `fetch()`, form gönderimi ve bazı güvenlik özellikleri sadece `http://` üzerinden çalışır. **Her zaman Live Server kullanın.**

Emmet: Saniyede İskelet Oluşturmak

VS Code'da yerleşik gelen **Emmet**, kısa yazımları tam HTML'e çevirir. Yazdıktan sonra **Tab** tuşuna basın.

```
!  
ul>li*3  
div.kart  
p#giris  
a[href="#"]  
h1{Merhaba}  
nav>ul>li*4>a
```

```
<!DOCTYPE html> ... (tam iskelet)  
<ul><li></li><li></li><li></li></ul>  
<div class="kart"></div>  
<p id="giris"></p>  
<a href="#"></a>  
<h1>Merhaba</h1>  
<nav><ul><li><a></a></li>...</ul></nav>
```

İpucu

En çok kullanacağınız: Boş bir `.html` dosyasında `!` yazıp **Tab**'a basın — tüm HTML5 iskeleti bir anda gelir.



HTML Nedir?

BÖLÜM 2

Tanım

HTML (HyperText Markup Language — Hipermetin İşaretleme Dili), bir web sayfasının **içeriğini yapılandıran** standart dildir.

HTML bir programlama dili **DEĞİLDİR**

- ▶ Değişken tanımlamaz
- ▶ Koşul (**if**) kuramaz
- ▶ Döngü (**for**) çalıştıramaz
- ▶ Hesaplama yapmaz

O bir **işaretleme (markup)** dilidir.

Tek ve En Önemli Amacı

Ham içeriği alarak, **etiketler (tags)** aracılığıyla ona **anlam ve yapı** kazandırmak. Anlamsız karakter yığını; tarayıcıların, arama motorlarının ve **ekran okuyucuların** anlayabileceği bir yapıya dönüştürür.

İçerik Nasıl Anlam Kazanır?

Düz metin — tarayıcı için anlamsız:

Blog Yazım
Bu benim ilk yazım.

Tarayıcı bunları yan yana, aynı boyutta yazar.
Hangisinin başlık olduğunu **bilemez**.

HTML ile işaretlenmiş:

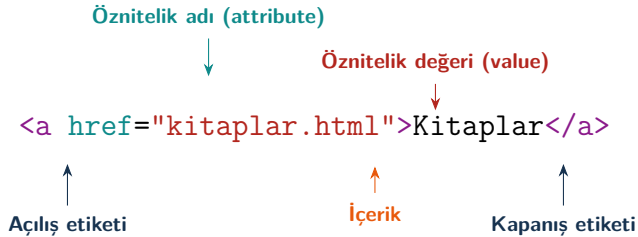
```
<h1>Blog Yazım</h1>  
<p>Bu benim ilk yazım.</p>
```

Artık tarayıcı `<h1>`'in bir **başlık**, `<p>`'nin ise bir **paragraf** olduğunu bilir.

Bu Neden Önemli?

- ▶ **Arama motorları** `<h1>` içeriğini sayfanın konusu olarak kabul eder.
- ▶ **Ekran okuyucular** görme engelli kullanıcıya “başlık seviyesi 1: Blog Yazım” diye okur.
- ▶ **CSS** daha sonra bu etiketleri hedef alarak stil verir.

Bir Etiketin Anatomisi



Kurallar

- ▶ Etiket adları **küçük harfle** yazılır
- ▶ Kapanış etiketinde / vardır
- ▶ Öznitelik değerleri **tırnak içinde** olur
- ▶ Bir element birden fazla öznitelik alabilir

</> Kendiliğinden Kapanan Etiketler

Bazı etiketlerin içeriği yoktur, kapanışı da yoktur (*void elements*): `<img>` `<br>` `<hr>`
`<input>` `<meta>` `<link>`

İç İçe Yerleştirme (Nesting)

Etiketler iç içe yerleştirilebilir — ancak **en son açılan en önce kapanır**.

✓ Doğru

```
<p>Bu <strong>\c{c}ok \"onemli</strong>  
    bir  
    duyurudur.</p>  
  
<ul>  
  <li><a href=\"#\">Ba\u{g}lantı</a></li>  
</ul>
```

✗ Yanlış

```
<p>Bu <strong>\c{c}ok \"onemli</p></strong>  
  
<ul>  
  <li><a href=\"#\">Ba\u{g}lantı</li></a>  
</ul>
```

💡 İpucu

Girinti (indentation) kullanın: her iç içe seviye için 2 boşluk. Kod çalışmasını etkilemez ama **okunabilirliği** inanılmaz artırır. VS Code'da **Shift + Alt + F** (Windows) / **Shift + Option + F** (macOS) otomatik düzenler.



HTML5 Belge İskeleti

BÖLÜM 3

```
1 <!DOCTYPE html>
2 <html lang="tr">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport"
6       content="width=device-width,
7           initial-scale=1.0">
8   <title>BookVerse</title>
9 </head>
10 <body>
11
12   <!-- Gorunur icerik buraya -->
13
14 </body>
15 </html>
```

- ▶ `<!DOCTYPE html>`
Belgenin **HTML5** olduğunu bildirir. **İlk** satır olmalı.
- ▶ `<html lang="tr">`
Kök element. `lang`, sayfa dilini belirtir — ekran okuyucu ve çeviri için kritiktir.
- ▶ `<head>`
Ekranda görünmeyen meta veriler.
- ▶ `<body>`
Kullanıcının **gördüğü** her şey.

Emmet kısayolu: boş dosyada **!** + **Tab**

<head> İçindeki Meta Etiketleri

Etiket	Görevi
<code><meta charset ="UTF-8"></code>	Karakter kodlaması. Bu olmadan Türkçe karakterler bozulur (ş, ğ, ı, ö, ç, ü).
<code><meta name ="viewport" ...></code>	Mobil cihazlarda doğru ölçeklendirme. Hafta 9'da detaylandıracağız.
<code><title>...</title></code>	Tarayıcı sekmesinde ve arama sonuçlarında görünen başlık.
<code><meta name ="description" ...></code>	Arama motoru sonuçlarındaki açıklama metni (SEO).
<code><link rel ="stylesheet" href ="..."></code>	Harici CSS dosyasını bağlar (Hafta 5).
<code><link rel ="icon" href ="..."></code>	Sekmede görünen küçük ikon (favicon).

VS Code'da İlk İskelet

<> index.html ●

<> index.html > ...

```
1  <!DOCTYPE html>
2
3  <html lang="tr">
4  <head>
5  <meta charset="UTF-8">
6  <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  <title>BookVerse - Kitap Tutkunlarının Bulusma Noktası</title>
8  </head>
9  <body>
10
11 </body>
12 </html>
```



127.0.0.1



1 BookVerse - Kitap Tutkunlarının Bulusma Noktası

☆ Başlangıç Sayfası



Metin Etiketleri

BÖLÜM 4

Başlıklar: <h1> — <h6>

```
<h1>BookVerse</h1>
<h2>Haftanın Çok Satanlari</h2>
<h3>Roman</h3>
<h4>Klasikler</h4>
<h5>19. Yüzyıl</h5>
<h6>Rus Edebiyatı</h6>
```



Hiyerarşi Kuralları

- ▶ Sayfada **tek bir** <h1> bulunmalı
- ▶ Seviye **atlanmamalı**: h1 → h3 ✗
- ▶ Başlık, **yazıyı büyütmek için** kullanılmaz — onun için CSS vardır

İpucu

Başlık hiyerarşisi, sayfanın **içindekiler tablosudur**. Arama motorları ve ekran okuyucular sayfayı bu yapıya göre gezinir. Yanlış hiyerarşi = erişilebilirlik hatası.

Metin Biçimlendirme Etiketleri

Etiket	Görünüm	Anlamı (Semantik)
<code><p></code>	Paragraf, altında boşluk	Bir metin bloğu
<code></code>	Kalın	Önem belirtir — ekran okuyucu vurgular
<code></code>	<i>Eğik</i>	Vurgu belirtir — tonlamayla okunur
<code></code>	Kalın	Sadece görsel; anlam katmaz
<code><i></code>	<i>Eğik</i>	Sadece görsel; teknik terim/yabancı kelime
<code>
</code>	Satır sonu	<i>Void element</i> — kapanışı yok
<code><hr></code>	Yatay çizgi	Konu değişimi
<code><mark></code>	Vurgulu	Metin içinde işaretleme
<code><small></code>	Küçük	Yan not, telif metni

Özel Karakterler (Entities)

Bazı karakterler HTML'de özel anlam taşıdığı için doğrudan yazılamaz:

```
&lt;    -> <
&gt;    -> >
&amp;  -> &
&quot;   -> "
&nbsp;    -> bolunmez bosluk
&copy; -> telif isareti
&hellip;-> ...
```

Yorum Satırları

```
<!-- Bu bir yorumdur -->

<!--
  Çok satirli
  yorum da olur
-->

<!-- TODO: Logo eklenecek -->
```

Tarayıcıda **görünmez**, ancak **kaynak kodda okunabilir**
— gizli bilgi yazmayın!

İpucu

Kod örneği göstermek istediğinizde `<p>` yazmalısınız, yoksa tarayıcı onu gerçek bir paragraf sanar.

Bağlantılar ve Görseller

BÖLÜM 5

Bağlantılar: <a> Etiketi

Web'i "web" yapan şey bağlantılardır — **hipermetin** buradan gelir.

```
1 <!-- Baska bir siteye (mutlak yol) -->
2 <a href="https://www.mdn.org">MDN Web Docs</a>
3
4 <!-- Kendi sitemizdeki sayfaya (goreli yol) -->
5 <a href="kitaplar.html">Kitaplar</a>
6
7 <!-- Yeni sekmede acar -->
8 <a href="https://w3.org" target="_blank" rel="noopener noreferrer">W3C</a>
9
10 <!-- Sayfa icindeki bir bolumu -->
11 <a href="#iletisim">Iletisim bolumune git</a>
12
13 <!-- E-posta ve telefon -->
14 <a href="mailto:info@bookverse.com">Bize yazin</a>
15 <a href="tel:+904241234567">Bizi arayin</a>
```

Dosya Yolları: Göreli ve Mutlak

```
bookverse/  
|-- index.html  
|-- kitaplar.html  
|-- css/  
|   |-- style.css  
|-- images/  
|   |-- logo.png  
|   |-- kapaklar/  
|       |-- simyaci.jpg
```

Yazım

Anlamı

logo.png	Aynı klasördeki dosya
../logo.png	Aynı klasör (açıkça)
images/logo.png	Alt klasördeki dosya
../index.html	Bir üst klasör
/images/logo.png	Sitenin kökünden
https://...	Başka bir site (mutlak)

⚠ Dikkat

index.html içinden simyaci.jpg'ye ulaşmak için: images/kapaklar/simyaci.jpg.
css/style.css içinden aynı dosyaya: ../images/kapaklar/simyaci.jpg — çünkü önce bir

Görseller: Etiketi

```
1 
```

Öznitelik

Görevi

src	Görselin yolu (zorunlu)
alt	Alternatif metin (zorunlu)
width	Genişlik (piksel)
height	Yükseklik (piksel)
loading	lazy: görününce yükle Fare üzerine gelince ipucu
title	



alt Neden Zorunlu?

- ▶ Görsel yüklenemezse bu metin görünür
- ▶ **Ekran okuyucular** bunu sesli okur
- ▶ Arama motorları görseli bu metinle indeksler

✗ alt = "resim"

✓ alt = "Simyacı kitabının kapağı"

Dekoratif görselde: alt = "" (boş bırakın)

Görsel Formatları ve <figure>

Format	Ne Zaman?
.jpg	Fotoğraf, çok renkli görsel
.png	Şeffaflık gerekiyorsa, logo
.svg	Vektörel — ikon, logo, grafik. Büyütünce bozulmaz
.webp	Modern, %30 daha küçük . Tercih edin
.gif	Basit animasyon

Açıklamalı Görsel

```
<figure>
  
  <figcaption>
    Paulo Coelho - Simyaci (1988)
  </figcaption>
</figure>
```

`<figure>` görsel ile açıklamasını **anlamsal olarak** birbirine bağlar.

İpucu

Performans: 4000×3000 piksellik bir fotoğrafı 200 piksel genişliğinde göstermeyin. Görselleri **kullanacağınız boyuta küçülterek** yükleyin — sayfa açılma hızını doğrudan etkiler.



Listeler

BÖLÜM 6

Sırasız (ul)

```
<ul>
  <li>Roman</li>
  <li>\c{S}iir</li>
  <li>Deneme</li>
</ul>
```

Sıra önemli değilse.

Sıralı (ol)

```
<ol>
  <li>Simyaci</li>
  <li>K\"urk Mantolu</li>
  <li>Tutunamayanlar</li>
</ol>
```

Sıra anlam taşıyorsa.

Tanım (dl)

```
<dl>
  <dt>ISBN</dt>
  <dd>Kitap kimlik no</dd>
  <dt>Cilt</dt>
  <dd>Kapak turu</dd>
</dl>
```

Terim–açıklama çiftleri.

Kritik Kural

`<ul>` ve `<ol>` etiketlerinin **doğrudan çocuğu** yalnızca `<li>` olabilir. `<li>` içine ise istediğiniz her şeyi (paragraf, görsel, hatta başka bir liste) koyabilirsiniz.

Listeler Neden Menü İçin Kullanılır?

```
1 <nav>
2   <ul>
3     <li><a href="index.html">Ana Sayfa</a></li>
4     <li><a href="kitaplar.html">Kitaplar</a></li>
5     <li><a href="hakkimizda.html">Hakkımızda</a></li>
6     <li><a href="iletisim.html">İletisim</a></li>
7   </ul>
8 </nav>
```

✓ Neden Liste?

Bir gezinti menüsü **gerçekten de bir listedir**. Ekran okuyucu “4 öğeli liste” diye duyurur ve kullanıcı aralarında gezinebilir.

⚠ Dikkat

Madde işaretlerini ve alt alta diziliminden rahatsız olmayın — Hafta 8’de **Flexbox** ile bu listeyi **yatay bir menü çubuğuna** dönüştüreceğiz.

Uygulama: BookVerse Ana Sayfası

BÖLÜM 7

```
1 <!DOCTYPE html>
2 <html lang="tr">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>BookVerse - Kitap Tutkunlariinin Bulusma Noktasi</title>
7 </head>
8 <body>
9   
10  <nav>
11    <ul>
12      <li><a href="index.html">Ana Sayfa</a></li>
13      <li><a href="kitaplar.html">Kitaplar</a></li>
14      <li><a href="hakkimizda.html">Hakkimizda</a></li>
15      <li><a href="iletisim.html">Iletisim</a></li>
16    </ul>
17  </nav>
18  <h1>BookVerse'e Hos Geldiniz</h1>
19  <p>Burasi, kitaplar hakkında yorumlarinizi paylasabileceginiz, yeni yazarlar
20    kesfedebileceginiz ve kitap tutkunlariyla bulusabileceginiz bir platformdur.</p>
21 </body>
22 </html>
```





W3C Validator

`validator.w3.org` adresine gidin ve kodunuzu yapıştırın. Yakalayacağı tipik hatalar:

- ▶ Kapatılmamış etiket
- ▶ Eksik `alt` öznitelik
- ▶ Yanlış iç içe yerleşim
- ▶ Aynı `id`'nin iki kez kullanımı

DevTools Elements

F12 → **Elements** sekmesi. Burada gördüğünüz **DOM ağacıdır** — yani tarayıcının sizin HTML'inizden *anladığı şey*. Yazdığınız HTML ile buradaki yapı farklıysa, **bir hata yapmışsınız** demektir.



Yapay Zeka ile HTML

BÖLÜM 8

Yapay Zeka HTML'de Çok İyi — Ve İşte Tehlike Burada

HTML, yapay zeka modellerinin **en iyi bildiği** dillerden biridir. Saniyeler içinde çalışan bir sayfa üretir. Peki sorun ne?

⚠ Öğrenme Tuzağı

“Bana bir kitap sitesi yaz” deyip çalışan kodu aldığınızda **hiçbir şey öğrenmemiş olursunuz**. Bir sonraki hafta CSS'i o koda uygulayamazsınız, çünkü yapıyı siz kurmadınız.

💡 Doğru Yaklaşım

Önce siz yazın, sonra yapay zekaya **denetletin**. Bu şekilde:

- ▶ Kendi hatalarınızı görürsünüz
- ▶ Neden yanlış olduğunu öğrenirsiniz
- ▶ Bir dahakine yapmazsınız

Bu haftaki prompt'ların hepsi “bana yaz” değil, “benim yazdığımı denetle” üzerine kuruludur.

Prompt 1 — Kendi Kodunuzu Denetletin

Yapay Zeka İstemi (Prompt) — Kod İnceleme (Code Review)

Sen deneyimli bir frontend geliştiricisisin. Asagida bir universite 1. sinif ogrencisinin yazdigi HTML kodu var.

Bu kodu su basliklar altinda incele: 1) SOZDIZIMI HATALARI - kapatilmamis veya yanlis ic ice etiketler 2) EKSİK ÖZNİTELİKLER - özellikle alt, lang, charset 3) SEMANTİK SORUNLAR - yanlis etiket secimi, baslik hiyerarsisi 4) ERİSİLEBİLİRLİK - ekran okuyucu kullanicisi icin sorunlar

KURAL: Duzeltilmis kodu HEMEN VERME. Once her sorunu numaralandirip "neden sorun oldugunu" acikla. Kodu ben duzeltecegim. Sonunda oncelik sirasina gore 3 maddelik bir eylem listesi ver.

Kod: [BURAYA KENDİ KODUNUZU YAPISTIRIN]

Kritik kısım: “Düzeltilmiş kodu hemen verme”. Bu cümle olmadan yapay zeka size hazır cevabı verir ve öğrenme biter.

Prompt 2 — Etiket Seçimini Sorgulatin

Yapay Zeka İstemi (Prompt) — Semantik Karşılaştırma

Bir web sayfasında "önemli bir uyarı metnini kalın göstermek" istiyorum. Aşağıdaki üç seçeneği karşılaştır:

A) `<b>Uyari</b>` B) `<strong>Uyari</strong>` C) `<span style = "font-weight:bold">Uyari</span>`

Her seçenek için: (1) tarayıcıdaki görünüm, (2) ekran okuyucunun davranışı, (3) arama motorunun yorumu, (4) ne zaman tercih edilmeli.

Sonucu bir tablo halinde ver ve hangisini neden önerdiğini tek cümleyle belirt. Türkçe yaz.

Yapay Zeka Çıktısını Doğrulama

Aldığınız cevabı bu slaytlardaki “**Metin Biçimlendirme Etiketleri**” tablosuyla karşılaştırın. Örtüşüyor mu? Fazladan bir bilgi veriyorsa MDN’de developer.mozilla.org/docs/Web/HTML/Element/strong adresinden teyit edin.

Yapay Zeka İstemi (Prompt) — Alıştırma Üretimi

Universite 1. sınıf web programlama öğrencisiyim. Bu hafta sunuları öğrendim: HTML5 belge iskeleti, başlık hiyerarsisi (h1-h6), p/strong/em, a etiketi ve göreceli-mutlak yollar, img ve alt özneliği, ul/ol/li listeleri.

Bana 5 adet KUCUK ALISTIRMA hazırla. Her alıştırma için: - Kısa bir görev tanımı (2-3 cümle) - Beklenen çıktının sözlü tanımı - KOD VERME, sadece görevi tanımla - Zorluk seviyesi: 1'den 5'e artan

En sonunda ayrı bir başlık altında "İpuçları" bölümü olsun ama çözüm kodu içermesin. Türkçe yaz.

Sık Karşılaşılan Hatalar

- ▶ **Eskimiş etiketler:** `<center>`, `<font>`, `<marquee>` — HTML5'te kaldırıldı
- ▶ **Eskimiş öznitelikler:** `align = "center"`, `bgcolor = "red"`
- ▶ **Uydurma etiketler:** `<container>`, `<text>`
- ▶ **Gereksiz `<div>` yığını** (“div çorbası”)
- ▶ **XHTML alışkanlığı:** `<br />` — HTML5'te gereksiz

Yapay Zeka Çıktısını Doğrulama

Nasıl tespit edersiniz?

- 1 Kodu validator.w3.org'a yapıştırın
- 2 Uyarı çıkarsa etiketi MDN'de arayın
- 3 “Deprecated” yazıyorsa **kullanmayın**

Bu üç adım 30 saniye sürer ve sizi **yıllarca sürecekt** kötü alışkanlıktan kurtarır.

Özet ve Değerlendirme

BÖLÜM 9

🔧 Ortam

VS Code + Live Server + Emmet. Proje klasörü: `index.html`, `css/`, `js/`, `images/`. Türkçe karakter ve boşluk yok.

</> Yapı

`<!DOCTYPE html>` → `<html lang = "tr">`
→ `<head>` (meta, title) + `<body>` (görünen içerik).

A Etiketler

`h1-h6` hiyerarşi ▪ `p`, `strong`, `em` ▪ `a + href`
▪ `img + src/alt` ▪ `ul/ol/li`

🤖 Yapay Zeka

Önce kendin yaz, sonra denetlet. "Düzeltilmiş kodu hemen verme" cümlesini prompt'a ekle. Çıktıyı W3C Validator + MDN ile doğrula.

Altın Kural: Görünüm için değil, **anlam** için etiket seçin.

Terim	Anlam	Terim	Anlam
Markup	İşaretleme (programlama değil)	Void element	Kapanışı olmayan etiket
Etiket (Tag)	<code><p></code> gibi işaret	Nesting	İç içe yerleştirme
Öznitelik	<code>href</code> , <code>src</code> , <code>alt</code>	Entity	<code>&amp;</code> gibi özel karakter
DOCTYPE	Belge türü bildirimi	Emmet	Kısa yazım → HTML
head	Görünmeyen meta veri	Live Server	Yerel canlı sunucu
body	Görünen içerik	localhost	127.0.0.1, kendi makineniz
Görelî yol	<code>images/logo.png</code>	Mutlak yol	<code>https://...</code>
Validator	W3C sözdizimi denetçisi	alt	Erişilebilirlik metni

- 1 HTML'in bir programlama dili olmadığını iki somut örnek ile açıklayınız.
- 2 `<meta charset = "UTF-8">` satırı silinirse ne olur? Neden?
- 3 `<strong>` ile `<b>` arasındaki fark nedir? Hangi durumda hangisi kullanılmalıdır?
- 4 Aşağıdaki kodda **üç hata** vardır. Bulunuz ve düzeltiniz:

```
<h1>Kitaplar<h1>  
  
<ul><p>Roman</p><li>Siir</li></ul>
```

- 5 `css/style.css` dosyasından `images/logo.png` dosyasına giden doğru görel yol nedir?
- 6 Bir gezinti menüsünü neden `<ul>` ile yazarız? `<div>` ile yazsaydık ne kaybederdik?

Laboratuvar Föyü 2

- ▶ VS Code + Live Server kurulumu
- ▶ BookVerse klasör yapısının oluşturulması
- ▶ `index.html` iskeletinin yazılması
- ▶ Başlık, metin, bağlantı, görsel, liste ekleme
- ▶ W3C Validator ile doğrulama
- ▶ Yapay zekaya kod denetlettirme görevi

Hazırlık

Okuma: Kitap Ünite 2.1 — “Temel Etiketler ve Sayfa Yapısı”, “Metin, Link, Resim ve Listeler”
Kaynaklar:

- ▶ developer.mozilla.org/docs/Web/HTML
- ▶ validator.w3.org
- ▶ docs.emmet.io/cheat-sheet

Önemli: `bookverse` klasörünüzü saklayın — dönem boyunca **aynı proje** üzerine ekleme yapacağız.

Haftaya Görüşmek Üzere!

Gelecek Hafta:

HTML: Tablolar, Formlar ve Multimedya

yapısı, form elemanları, türleri, doğrulama öznitelikleri, ses ve video