

WEB PROGRAMLAMA I ■ HAFTA 5

CSS'e Giriş: Sözdizimi, Seçiciler ve Cascade

İçerik–stil ayrımı ■ Seçici türleri ■ Cascade ■ Özgüllük (specificity) ■ Kalıtım ■ Renk ve birimler

Prof. Dr. Özal YILDIRIM

Fırat Üniversitesi — Teknoloji Fakültesi

Yazılım Mühendisliği Bölümü

✓ Hafta 2–4 Özeti

- ▶ HTML belge iskeleti ve etiketler
- ▶ Tablolar ve formlar
- ▶ Semantik yapı, erişilebilirlik, SEO

Sayfamızın **iskeleti hazır** — ama hâlâ çirkin.

→ Bu Hafta

Artık **CSS** başlıyor. Önümüzdeki **5 hafta** boyunca aynı HTML'i, tek satır değiştirmeden tanınmayacak hale getireceğiz.

*Eğer HTML bir evin iskeleti, duvarları ve odalarıysa; **CSS** o evin duvar kâğıdı, boyası, mobilyaları, aydınlatması ve genel dekorasyonudur.*

- ▶ CSS'in ne olduğunu ve “içerik–stil ayrımı” ilkesini açıklayabilecek,
- ▶ CSS'i sayfaya dahil etmenin üç yöntemini karşılaştırabilecek,
- ▶ CSS kural sözdizimini (seçici, özellik, değer) doğru yazabilecek,
- ▶ Etiket, sınıf, `id`, öznitelik ve birleşik seçicileri kullanabilecek,
- ▶ Sözde-sınıf (`:hover`) ve sözde-elementleri (`::before`) uygulayabilecek,
- ▶ Cascade, özgüllük ve kalıtım kurallarıyla çakışmaları çözebilecek,
- ▶ CSS renk ve uzunluk birimlerini yerinde seçebilecek,
- ▶ DevTools ile hangi kuralın neden kazandığını teşhis edebileceksiniz.

- 1 CSS Nedir?
- 2 Seçiciler
- 3 Cascade, Özgüllük ve Kalıtım
- 4 Renkler ve Birimler
- 5 Yapay Zeka ile CSS
- 6 Özet ve Değerlendirme



CSS Nedir?

BÖLÜM 1

CSS: Basamaklı Stil Şablonları

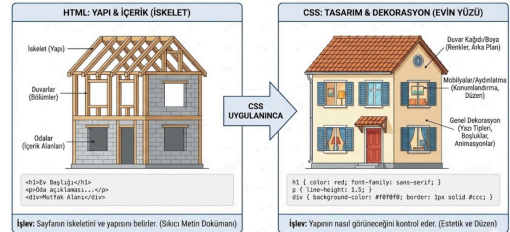
Tanım

CSS (Cascading Style Sheets), bir web sayfasının **görünümünü ve düzenini** tanımlamak için kullanılan stil dilidir.

HTML sayfanın **yapısını** belirlerken, CSS bu yapının **nasıl görüneceğini** kontrol eder:

- Renkler ve yazı tipleri
- Boşluklar ve hizalama
- Konumlandırma ve düzen
- Geçişler ve animasyonlar

CSS olmadan Web, sıkıcı metin dokümanlarından ibaret kalırdı.



İçerik–Stil Ayrımı Neden Devrimdi?

Web'in ilk yıllarında stil bilgisi **doğrudan HTML'in içine** yazılırdı:

```
<font color="red" size="5" face="Arial">Baslik</font>  
<table bgcolor="#eeeeee" cellpadding="10" align="center">
```

⚠ Dikkat

Sorun: Sitedeki tüm başlıkların rengini değiştirmek için **yüzlerce, hatta binlerce** HTML dosyasını tek tek düzenlemeniz gerekiyordu.

✂ Bakım Kolaylığı

Tüm renk şemasını değiştirmek için **tek bir dosya** yeterli.

👁 Okunabilirlik

HTML, stil tanımlarından arındığı için **çok daha temiz** hale gelir.

⚡ Verimlilik

CSS dosyası bir kez indirilip **önbelleğe alınır**; diğer sayfalarda tekrar indirilmez.

Bu ilkenin adı: **Separation of Concerns** — ilgilerin ayrılması.

CSS'i Sayfaya Dahil Etmenin Üç Yolu

1. Satır içi (Inline) — ✗ Kaçının

```
<h1 style="color: blue; font-size: 32px;">BookVerse</h1>
```

Yeniden kullanılamaz, HTML'i kirletir, özgüllüğü çok yüksektir.

2. Dahili (Internal) — ⚠ Sınırlı

```
<head>
  <style> h1 { color: blue; } </style>
</head>
```

Sadece o sayfada geçerli. Tek sayfalık demolarda kabul edilebilir.

3. Harici (External) — ✓ En İyi Uygulama

```
<head>
  <link rel="stylesheet" href="css/style.css">
</head>
```

Tüm sayfalar paylaşır, önbelleklenir, HTML temiz kalır.

💡 İpucu

`rel = "stylesheet"` bu dosyanın bir **stil şablonu** olduğunu, `href` ise **konumunu** belirtir. Bu derste **her zaman** harici dosya kullanacağız.



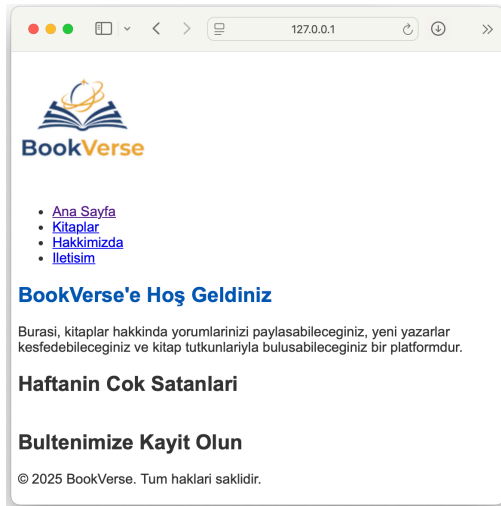
```
h1 {  
  color: #0056b3;  
  font-size: 32px;  
}
```

- ▶ Seçici (Selector): `h1` — hangi elemana?
- ▶ Bildirim bloğu: `{ ... }`
- ▶ Özellik (Property): `color`
- ▶ Değer (Value): `#0056b3`
- ▶ Her bildirim **noktalı virgülle** biter

css/style.css

```
1 body {  
2   font-family: Arial, Helvetica, sans-serif;  
3   color: #333333;  
4 }  
5  
6 h1 {  
7   color: #0056b3;  
8 }
```

- ▶ body ile **tüm sayfayı** hedefledik
- ▶ font-family'de **yedekleme zinciri** var: Arial yoksa Helvetica, o da yoksa sistem **sans-serif**
- ▶ h1 ile sadece ana başlığı renklendirdik



Tarayıcının Varsayılan Stilleri

Hiç CSS yazmasanız bile sayfanız **tamamen stilsiz değildir**. Her tarayıcının kendi **User Agent Stylesheet** dosyası vardır.

```
h1 -> font-size: 2em; bold;
      margin: 0.67em 0
p   -> margin: 1em 0
ul  -> padding-left: 40px;
      list-style: disc
a   -> color: blue; underline
body-> margin: 8px
```

👁 Kendiniz Görün

DevTools → **Styles** paneli → en altta **“user agent stylesheet”** başlığı. Sayfanızdaki beklenmedik boşlukların kaynağı genellikle **buradadır**.

⚠ Dikkat

Bu varsayılanlar tarayıcıdan tarayıcıya **az da olsa farklıdır**. Aynı sayfa Chrome ve Safari’de biraz farklı görünebilir — çözüm bir sonraki slaytta.

CSS Reset: Temiz Başlangıç

Her projeye **aynı zeminden** başlamak için stil dosyanızın en üstüne ekleyin:

```
1  /* --- Modern minimal reset --- */
2  *, *::before, *::after { box-sizing: border-box; }
3
4  body, h1, h2, h3, h4, p, figure, blockquote, dl, dd {
5    margin: 0;
6  }
7
8  ul[class], ol[class] { list-style: none; padding: 0; }
9
10 img { max-width: 100%; display: block; }
11
12 input, button, textarea, select { font: inherit; }
13
14 body { min-height: 100vh; line-height: 1.5; -webkit-font-smoothing: antialiased; }
```

i Reset vs Normalize

Reset: Tüm varsayılanları **sıfırlar**.

Normalize: Varsayılanları koruyup tarayıcılar arası

💡 İpucu

```
input, button { font: inherit; }
```

satırı çok önemlidir, form elemanları varsayılan



Seçiciler

BÖLÜM 2

Tür	Yazım	Hedefi
Evrensel	<code>* { }</code>	Tüm elementler
Etiket (type)	<code>p { }</code>	Tüm <code><p></code> elementleri
Sınıf	<code>.kart { }</code>	<code>class = "kart"</code> olan tüm elementler
id	<code>#ustbar { }</code>	<code>id = "ustbar"</code> olan tek element
Öznitelik	<code>[type = "email"]</code>	O özneliğe sahip elementler
Gruplama	<code>h1, h2, h3 { }</code>	Birden çok seçiciye aynı stil

```
.kitap-kart      { border: 1px solid #ddd; } /* sınıf */
#ana-menu        { background: #16324f; }   /* id   */
input[type="email"] { border-color: #1f6fb2; } /* öznitelik */
h1, h2, h3      { font-family: Georgia, serif; } /* grup */
```

Birleşik Seçiciler (Combinators)

Yazım	Adı	Anlamı
<code>nav a</code>	Torun (descendant)	<code>nav</code> içindeki tüm <code>a</code> 'lar (kaç kat derinde olursa olsun)
<code>nav > ul</code>	Çocuk (child)	Sadece doğrudan çocuk olan <code>ul</code> 'ler
<code>h2 + p</code>	Bitişik kardeş	<code>h2</code> 'den hemen sonra gelen ilk <code>p</code>
<code>h2 ~ p</code>	Genel kardeş	<code>h2</code> 'den sonraki tüm kardeş <code>p</code> 'ler
<code>a.aktif</code>	Birleşik	Hem <code>a</code> olan hem <code>class = "aktif"</code> taşıyan

```
<nav>
  <ul>
    <li><a href="#">Ana Sayfa</a></li>
    <li><a href="#" class="aktif">Kitaplar</a></li>
  </ul>
</nav>
```

```
nav a      { color: white; }
nav > ul   { list-style: none; }
a.aktif    { font-weight: bold; }
```

Sözde-Sınıflar (Pseudo-classes)

Bir elementin **durumunu** veya **konumunu** hedefler.

Durum

```
a:hover      { color: #e8590c; } /* fare ustunde */
a:visited    { color: #6d28d9; } /* ziyaret edilmiş */
a:active     { color: #c62828; } /* tiklanırken */
input:focus  { outline: 2px solid #1f6fb2; }
input:disabled { background: #eee; }
input:checked { accent-color: #0e8a8a; }
input:invalid { border-color: red; }
```

Konum

```
li:first-child { font-weight: bold; }
li:last-child  { border-bottom: none; }
li:nth-child(2) { color: red; }
li:nth-child(odd) { background: #f7f7f7; }
li:nth-child(even) { background: #fff; }
tr:nth-child(3n) { background: #eef; }
p:not(.ozet)     { color: #555; }
```

İpucu

:hover sırası önemlidir: `a:link` → `a:visited` → `a:hover` → `a:active`. Hafıza notu: **LoVe HAtE** — LVHA.

Sözde-Elementler (Pseudo-elements)

Var olmayan bir kısmı **oluşturur** veya hedefler. **İki** iki nokta ile yazılır.

```
1  /* Icerikten once/sonra sanal element olusturur */
2  .zorunlu::after      { content: " *"; color: #c62828; }
3  .dis-link::after     { content: " (yeni sekme)"; font-size: 0.8em; }
4
5  /* Metnin ilk harfi / ilk satiri */
6  p::first-letter      { font-size: 2em; float: left; }
7  p::first-line        { font-weight: bold; }
8
9  /* Kullanıcının sectigi metin */
10 ::selection          { background: #f2b705; color: #16324f; }
11
12 /* Form ipucu metni */
13 input::placeholder   { color: #9aa5b1; font-style: italic; }
```

Dikkat

`::before` / `::after` ile eklenen içerik **DOM'da yoktur** — ekran okuyucuların çoğu okur ama **güvenilir değildir**. **Önemli bilgiyi** asla `content` ile eklemeyin, sadece **dekoratif** amaçla kullanın.

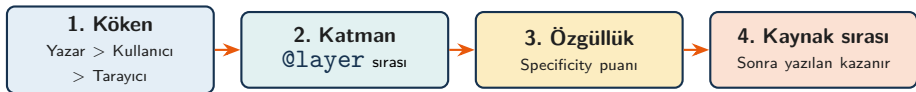


Cascade, Özgüllük ve Kalıtım

BÖLÜM 3

CSS'in "C"si: Cascade (Basamaklanma)

Aynı elemana birden fazla kural uygulandığında, tarayıcı **hangisinin kazanacağını** şu sırayla belirler:



İpucu

Pratikte günlük hayatınızda karşınıza çıkacak olanlar **3. ve 4. adımlardır**: **özgüllük** eşitse **sonra yazılan** kural kazanır.

Özgüllük (Specificity) Hesabı

Her seçiciye üç haneli bir puan verilir: (A, B, C)

Hane	Neyi sayar?	Örnek
A	id seçicileri	#ustbar
B	Sınıf, öznitelik, sözde-sınıf	.kart, [type], :hover
C	Etiket ve sözde-element	p, li, ::before

```
p                -> (0,0,1)
.kart            -> (0,1,0)
nav ul li a      -> (0,0,4)
.kart p.ozet     -> (0,2,1)
#ustbar .menu a:hover -> (1,2,1)
a[href^="https"]  -> (0,1,1)
style="..." (satir ici) -> (1,0,0,0) <- her
    seyi yener
!important       -> Cascade'i bozar
```

⚠ Dikkat

Karşılaştırma soldan sağa yapılır. (0,1,0)

> (0,0,10)

Yani **tek bir sınıf**, on tane etiketten daha güçlüdür. Hane taşımaları **yoktur** — 10 sınıf, 1 id etmez.

```
<div id="icerik">
  <p class="ozet vurgulu">BookVerse'e hos geldiniz.</p>
</div>
```

```
1 p          { color: black; } /* (0,0,1) */
2 .ozet      { color: blue;  } /* (0,1,0) */
3 .vurgulu   { color: green; } /* (0,1,0) */
4 div p      { color: orange; } /* (0,0,2) */
5 #icerik p  { color: purple; } /* (1,0,1) */
6 .ozet.vurgulu { color: teal;  } /* (0,2,0) */
```

? Metin hangi renkte olur?

Sırasıyla düşünün:

- ▶ En yüksek A hanesi hangisinde?
- ▶ Berabere kalan var mı?

✓ Cevap

Mor (purple) — #icerik p tek id içerdiği için A hanesi 1; diğer tüm kuralların A hanesi 0. .ozet ve .vurgulu berabere olsaydı, **sonra yazılan** (.vurgulu) kazanırdı.

!important: Son Çare, İlk Seçenek Değil

```
.uyari { color: red !important; }
```

⊘ Neden Kaçınılmalı?

- ▶ Tüm özgüllük hesabını **devre dışı bırakır**
- ▶ Sonraki geliştirici geçersiz kılmak için **daha çok !important** yazar
- ▶ “!important savaşları” başlar
- ▶ Hata ayıklaması imkânsızlaşır

✓ Meşru Kullanımlar

- ▶ Üçüncü parti bir kütüphanenin stilini ezmek
- ▶ Yardımcı (utility) sınıfları: `.gizli { display: none !important; }`
- ▶ Kullanıcı erişilebilirlik tercihleri

Kural: !important yazmak zorunda kaldığınız anda, **CSS mimariniz** sorgulanmalıdır — seçiciyi düzeltin, !important eklemeyin.

Kalıtım (Inheritance)

Bazı özellikler **ana elementten çocuklara geçer**, bazıları geçmez.

✓ Kalıtımla Geçer

color ▪ font-family ▪ font-size
▪ font-weight ▪ line-height ▪
text-align ▪ letter-spacing ▪
visibility ▪ cursor Genelde **metinle ilgili** özellikler.

✗ Geçmez

margin ▪ padding ▪ border ▪ width
▪ height ▪ background ▪ display ▪
position Genelde **kutu ile ilgili** özellikler.

```
body { font-family: Georgia, serif; color: #333; } /* tum sayfaya yayilir */
.kutu { border: 1px solid #ccc; }                /* sadece .kutu'ya */

/* Zorlamak icin: */
a      { color: inherit; }                       /* ustunden devral */
.kart  { border: initial; }                      /* varsayilana don */
```



Renkler ve Birimler

BÖLÜM 4

CSS'te Renk Tanımlama

Yöntem	Örnek	Not
İsim	<code>color: red;</code>	147 tanımlı isim; sınırlı
Onaltılık (hex)	<code>color: #0056b3;</code>	En yaygın. #RGB kısa yazım da olur
Hex + alfa	<code>color: #0056b380;</code>	Son iki hane saydamlık
RGB	<code>rgb(0 86 179);</code>	0–255 aralığı
RGBA	<code>rgb(0 86 179 / 0.5);</code>	Son değer alfa (0–1)
HSL	<code>hsl(211 100% 35%);</code>	Ton, doygunluk, açıklık
<code>currentColor</code>	<code>border-color: currentColor;</code>	Mevcut <code>color</code> değerini kullanır

Uzunluk Birimleri

Birim	Tür	Anlamı ve Kullanımı
px	Mutlak	Piksel. Kenarlık, küçük boşluklar için uygun
%	Görelî	Ana elemente göre. Genişlik için yaygın
em	Görelî	Kendi yazı boyutuna göre. İç içe kullanımda katlanır
rem	Görelî	Kök (html) yazı boyutuna göre. En güvenli tercih
vw/vh	Görelî	Ekran genişliği/yüksekliğinin %1'i
ch	Görelî	"0" karakterinin genişliği. Metin satır uzunluğu için
fr	Grid	Kalan alanın payı (Hafta 9)

```
html { font-size: 16px; } /* varsayılan kök boyut */
h1   { font-size: 2rem; } /* 32px - köke göre */
.kart { padding: 1.5rem; } /* 24px */
meta { max-width: 65ch; } /* okunabilir satır uzunluğu */
```

CSS Değişkenleri (Custom Properties)

```
1 :root {
2   --renk-ana:      #16324f;
3   --renk-vurgu:    #e8590c;
4   --renk-metin:    #333333;
5   --bosluk:        1rem;
6   --yuvarlak:      6px;
7   --font-baslik:   Georgia, "Times New Roman", serif;
8 }
9
10 .kart {
11   background: var(--renk-ana);
12   padding: var(--bosluk);
13   border-radius: var(--yuvarlak);
14   color: var(--renk-vurgu);
15 }
16
17 .kart-genis { padding: calc(var(--bosluk) * 2); }
```

Kazanım: Marka rengini değiştirmek için **tek bir satır** düzenlemek yeterli. Karanlık tema (dark mode) için de temel oluşturur.

🔍 Styles Paneli

- ▶ Elemente uygulanan **tüm kurallar** özgüllük sırasıyla listelenir
- ▶ **Üstü çizili** olanlar: geçersiz kılınmış kurallar
- ▶ Sağdaki dosya adı: kuralın **kaynağı**
- ▶ Değerleri **canlı değiştirebilirsiniz** (ok tuşlarıyla artır/azalt)

🧮 Computed Paneli

- ▶ Tarayıcının **sonuçta hesapladığı** değerler
- ▶ **2rem** yerine **32px** görürsünüz
- ▶ Her özelliğin yanındaki oka basınca **hangi kuraldan geldiği** açılır
- ▶ Kutu Modeli diyagramı burada

💡 İpucu

“Stilim neden uygulanmıyor?” üç adımlı teşhis: **1)** Element seçili mi, kural listede görünüyor mu? **2)** Görünüyorsa üstü çizili mi? → özgüllük sorunu. **3)** Hiç görünmüyorsa → seçici yanlış veya CSS dosyası bağlanmamış.

CSS Dosyanızı Nasıl Düzenlemeli?

```
1  /* =====  
2    BookVerse - style.css  
3    Yazar: Ad Soyad / Guncelleme: 2026-09-06  
4    ===== */  
5  
6  /* --- 1. Degiskenler ----- */  
7  :root { --renk-ana: #16324f; }  
8  
9  /* --- 2. Reset ----- */  
10 *, *::before, *::after { box-sizing: border-box; }  
11  
12 /* --- 3. Temel elementler ----- */  
13 body { font-family: Georgia, serif; }  
14  
15 /* --- 4. Duzen (layout) ----- */  
16 .container { width: min(960px, 100% - 2rem); margin-inline: auto; }  
17  
18 /* --- 5. Bilesenler (components) ----- */  
19 .kitap-kart { border: 1px solid var(--renk-cizgi); }  
20  
21 /* --- 6. Yardimci siniflar (utilities) ----- */  
22 .gizli { display: none; }
```

@import mi, <link> mi?

✓ <link> — Tercih edin

```
<link rel="stylesheet" href="css/reset.css">
<link rel="stylesheet" href="css/style.css">
```

Tarayıcı dosyaları **paralel** indirir.

✗ @import — Kaçının

```
/* style.css içinde */
@import url("reset.css");
```

Zincirleme indirme: önce `style.css`, sonra `reset.css`. Sayfa daha geç boyanır.

A Web Fontu Bağlama (Hafta 7'de detaylı)

```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link rel="stylesheet"
      href="https://fonts.googleapis.com/css2?family=Merriweather:wght@400;700&display=
      swap">
```

`preconnect`, font sunucusuna bağlantıyı **önceden** kurarak yükleme süresini kısaltır.

Uygulama: BookVerse Renk Sistemi

```
1 :root {
2   /* Marka renkleri */
3   --renk-ana:      #16324f;
4   --renk-ana-acik: #1f6fb2;
5   --renk-vurgu:    #e8590c;
6
7   /* Metin */
8   --renk-metin:    #22303c;
9   --renk-metin-soluk:#5a6672;
10
11  /* Zemin */
12  --renk-zemin:     #ffffff;
13  --renk-zemin-alt: #f2f4f7;
14  --renk-cizgi:     #dde3ea;
15
16  /* Olcekler */
17  --bosluk-s: 0.5rem;
18  --bosluk-m: 1rem;
19  --bosluk-l: 2rem;
20  --yuvarlak: 6px;
21 }
22
23 body {
24   color: var(--renk-metin);
25   background: var(--renk-zemin);
26 }
27 h1, h2, h3 { color: var(--renk-ana); }
28 a { color: var(--renk-ana-acik); }
29 a:hover { color: var(--renk-vurgu); }
```



Neden Sistem?

Rastgele renk seçmek yerine **sınırlı bir palet** tanımlamak:

- ▶ Sayfa **tutarlı** görünür
- ▶ Değişiklik **tek yerden** yapılır
- ▶ Karanlık tema kolaylaşır



Dikkat

Değişken adlarını **işlevine göre** verin (**--renk-vurgu**), rengine göre değil (**--turuncu**). Yarın turuncu yeşil olursa adı yanlış kalmaz.



Yapay Zeka ile CSS

BÖLÜM 5

CSS'te Yapay Zekanın Güçlü ve Zayıf Yönleri

👍 İyi Olduğu Yerler

- ▶ Seçici sözdizimini açıklama
- ▶ Özgüllük puanı hesaplama
- ▶ Renk paleti önerme
- ▶ “Bu kural neden çalışmıyor?” teşhisi
- ▶ Uzun CSS'i düzenleme/gruplama

🗨️ Zayıf Olduğu Yerler

- ▶ **Görsel sonucu öngörme** — sayfayı görmüyor
- ▶ Gereksiz `!important` ekleme
- ▶ Aşırı uzun seçici zincirleri (`div > div > ul > li > a`)
- ▶ Eski yöntemler (`float` ile düzen)
- ▶ Kontrast oranını **tahmin etme**

Kritik sınır: Yapay zeka sayfanızı **görmüyor**. Ekran görüntüsü vermeden “şu kısım bozuk” dersiniz **tahmin** eder — ve genellikle yanılır.

Prompt 1 — “Stilim Neden Uygulanmıyor?”

Yapay Zeka İstemi (Prompt) — Teşhis Yardımcısı

CSS kuralım uygulanmıyor. Asagida HTML ve CSS kodum var.

GOREV: 1) Once CSS secicimin bu HTML'de HANGI elemanlari sectigini yaz. Hicbirini secmiyorsa bunu acikca belirt. 2) Ayni elemani hedefleyen BASKA kurallar varsa hepsini listele ve her birinin ozgullugunu (A,B,C) formatinda hesapla. 3) Hangi kuralin kazandigini ve NEDEN kazandigini acikla. 4) Cozum onerisi ver ama !important KULLANMA. Kodu duzeltip vermeni istemiyorum; sadece teshis yap.

HTML: [KODUNUZ]

CSS: [KODUNUZ]

Cevabı DevTools **Styles** paneliyle doğrulayın: üstü çizili kurallar yapay zekanın dediğiyle uyuyor mu?

Prompt 2 — Seçici Egzersizi Üretimi

Yapay Zeka İstemi (Prompt) — Seçici Alıştırmaları

Asagidaki HTML yapisini kullanarak bana 12 adet CSS SECICI SORUSU hazirla.
Her soru sunun gibi olsun: "Sadece <su ozellikteki> elemanlari secen CSS secicisini yaz."
Zorluk sirasi: etiket -> sinif -> id -> oznitelik -> torun -> cocuk -> kardes ->
sozde-sinif -> sozde-element -> birlesik.
KURALLAR: - Once SADECE 12 soruyu yaz. - Cevap anahtarini en sona, ayri bir baslik altina
koy. - Her cevabin yanina ozgulluk puanini (A,B,C) yaz.
HTML: [KENDI HTML KODUNUZU YAPISTIRIN]

Cevapları DevTools Console'da `document.querySelectorAll("secici")` ile **gerçekten** test edin.

Prompt 3 — Renk Paleti ve Kontrast

Yapay Zeka İstemi (Prompt) — Erişilebilir Palet Tasarımı

"BookVerse" adında bir online kitap platformu için CSS renk paleti oluşturun. Marka hissi: güvenilir, sakin, okumaya davet eden.

Bana şu formatta bir :root blogu ver: --renk-ana, --renk-ana-koyu, --renk-vurgu, --renk-metin, --renk-metin-soluk, --renk-zemin, --renk-zemin-alt, --renk-cizgi

KURALLAR: - Her rengi HEX ve HSL olarak birlikte yaz. - Her metin/zemin çifti için KONTRAST ORANINI hesapla ve WCAG AA (4.5:1) eşimini geçip geçmediğini belirt. - Kontrast oranını nasıl hesapladığını kısaca açıkla. - Karanlık tema için ayrı bir blok daha ver.

Yapay Zeka Çıktısını Doğrulama

Kontrast oranlarını mutlaka doğrulayın: webaim.org/resources/contrastchecker veya DevTools renk seçicisi. Yapay zeka bu oranları sıkça yanlış hesaplar — “4.8:1” dediği çift gerçekte 3.1:1 çıkabilir.

Özet ve Değerlendirme

BÖLÜM 6

Temel

İçerik–stil ayrımı. Harici dosya + `<link>`.
Sözdizimi: `seçici { özellik: değer; }`

Seçiciler

Etiket, `.sınıf`, `#id`, `[öznitelik]`, birleşik (`>`, `+`, `~`), sözde-sınıf (`:hover`), sözde-element (`::after`).

Cascade

Özgüllük (A,B,C) → eşitse **sonra yazılan** kazanır.
`!important`'tan kaçınin.
Metin özellikleri kalıtsal, kutu özellikleri değil.

Yapay Zeka

Seçici ve özgüllük açıklamada güçlü; **görsel sonuç ve kontrast hesabında zayıf**. Sayfayı görmüyor — ekran görüntüsü verin veya kendiniz doğrulayın.

Altın Kural: En **düşük özgüllükle** istediğiniz sonucu elde edin. Basit seçici = bakımı kolay CSS.

Terim	Anlam	Terim	Anlam
Cascade	Kural çakışmasını çözme düzeni	rem	Kök yazı boyutuna göreli birim
Specificity	Seçici gücü (A,B,C)	em	Kendi yazı boyutuna göreli
Inheritance	Kalıtım	ch	“0” karakteri genişliği
Selector	Seçici	vw/vh	Ekran boyutuna göreli
Declaration	özellik: değer;	Custom property	--degisken
Pseudo-class	:hover, :focus	var()	Değişken okuma
Pseudo-element	::before, ::after	calc()	Hesaplama fonksiyonu
Combinator	>, +, ~, boşluk	Kontrast oranı	Metin/zemin okunabilirliği

- 1 “İçerik–stil ayrımı” ilkesinin üç avantajını yazınız.
- 2 Aşağıdaki seçicilerin özgüllük puanlarını hesaplayınız:
`nav ul li a .menu .aktif #ust a:hover p::first-letter`
- 3 `nav a` ile `nav > a` arasındaki fark nedir? Hangi durumda ikisi aynı sonucu verir?
- 4 `em` ile `rem` arasındaki fark nedir? İç içe üç seviye `1.2em` tanımlı elementte son yazı boyutu ne olur?
- 5 `margin` özelliği neden kalıtımla geçmez? Geçseydi ne olurdu?
- 6 Bir kural `!important` olmadan uygulanmıyorsa, izlemeniz gereken üç adımlı teşhis yöntemini yazınız.

Laboratuvar Föyü 5

- ▶ `css/style.css` oluşturma ve bağlama
- ▶ 12 seçici alıştırmaları
- ▶ Özgüllük hesaplama tablosu
- ▶ Kalıtım deneyleri
- ▶ CSS değişkenleriyle renk paleti
- ▶ DevTools ile teşhis egzersizi
- ▶ Yapay zeka ile seçici doğrulama

Hazırlık

Okuma: Kitap Ünite 2.2 — “CSS: Web’in Görsel Dünyası”, “CSS Sözdizimi” **Kaynaklar:**

- ▶ developer.mozilla.org/docs/Web/CSS
- ▶ specificity.keegan.st — özgüllük hesaplayıcı
- ▶ flukeout.github.io — seçici oyunu
- ▶ webaim.org/resources/contrastchecker

Haftaya Görüşmek Üzere!

Gelecek Hafta:

CSS Box Model ve Görsel Biçimlendirme

-
-
-
-
- türleri
- Margin çökmesi