

WEB PROGRAMLAMA I ■ HAFTA 6

CSS Box Model ve Görsel Biçimlendirme

Content–Padding–Border–Margin ■
Taşma yönetimi

türleri ■ Margin çökmesi ■

Prof. Dr. Özal YILDIRIM

Fırat Üniversitesi — Teknoloji Fakültesi

Yazılım Mühendisliği Bölümü

✓ Hafta 5 Özeti

- ▶ CSS sözdizimi ve dahil etme yöntemleri
- ▶ Seçici türleri ve birleştiriciler
- ▶ Cascade, özgüllük, kalıtım
- ▶ Renkler, birimler, değişkenler

Artık **ne seçeceğimizi** biliyoruz.

→ Bu Hafta

CSS'in **en temel ilkesi**: **Web** sayfasındaki **her HTML elemanı, diktörtgen bir kutudur**. Bir başlık, paragraf, resim veya buton — hepsi aslında birer kutudur.

Bu haftadan sonra sayfadaki her boşluğu, her hizalamayı ve her taşmayı **açıklayabilir** hale geleceksiniz.

- ▶ Kutu Modeli'nin dört katmanını (content, padding, border, margin) açıklayabilecek,
- ▶ `content-box` ile `border-box` farkını hesaplayarak gösterebilecek,
- ▶ Kısa yazım (shorthand) kurallarını doğru sırayla yazabilecek,
- ▶ `display` değerlerinin (`block`, `inline`, `inline-block`, `none`) davranış farkını ayırt edebilecek,
- ▶ Margin çökmesini (collapse) tanıyıp önleyebilecek,
- ▶ `width`, `max-width`, `min-height` ile esnek kutular kurabilecek,
- ▶ `overflow` ile taşmayı yönetebilecek,
- ▶ DevTools kutu modeli diyagramını okuyup hata ayıklayabileceksiniz.

Bu Haftanın İçeriği

- 1 Kutu Modeli
- 2 Boşluk Yönetimi
- 3 `display` Özelliği
- 4 Konumlandırma (`position`)
- 5 Yapay Zeka ile Kutu Modeli
- 6 Özet ve Değerlendirme



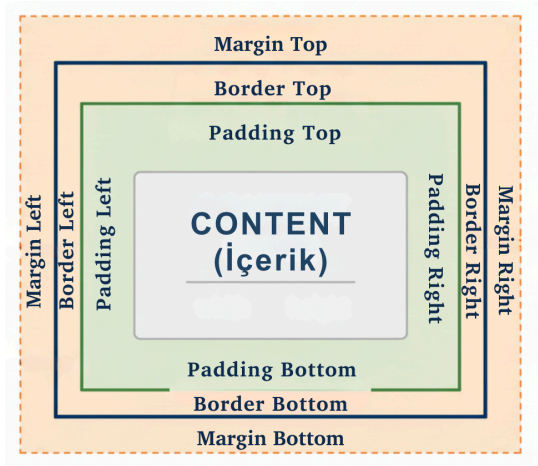
Kutu Modeli

BÖLÜM 1

Her Şey Bir Kutudur

Kutu, içten dışa doğru **dört katmandan** oluşur:

- ▶ **Content (İçerik)**
Metin, resim veya medya. `width / height` ile kontrol edilir.
- ▶ **Padding (İç Boşluk)**
İçeriği çevreleyen **şeffaf** boşluk. Arka plan rengini alır.
- ▶ **Border (Kenarlık)**
Padding'i saran çizgi. Renk, kalınlık, stil.
- ▶ **Margin (Dış Boşluk)**
Kenarlığın dışındaki boşluk. Kutuyu komşularından uzaklaştırır. **Her zaman şeffaftır.**



Kısa Yazım (Shorthand) Kuralları

```
1 padding: 20px;           /* dort yon: 20 */
2 padding: 10px 20px;      /* dikey 10 | yatay 20 */
3 padding: 10px 20px 30px; /* ust 10 | yatay 20 | alt 30 */
4 padding: 10px 20px 30px 40px; /* ust | sag | alt | sol */
5
6 margin-top: 10px; margin-right: 20px;
7 margin-bottom: 30px; margin-left: 40px; /* tek tek de yazilabilir */
8
9 /* Mantiksal ozellikler (modern, dil yonunden bagimsiz) */
10 margin-block: 1rem; /* ust + alt */
11 margin-inline: auto; /* sol + sag */
12 padding-inline: 1.5rem;
```

🕒 Hafıza Notu

Dört değer **saat yönünde** okunur:

Üst → Sağ → Alt → Sol (TRBL — “TRouBLe”)

🏠 border Kısa Yazımı

```
border: 1px solid #ccc;
/*      kalinlik stil renk      */

border-bottom: 2px dashed red;
border-radius: 8px;
```

Kritik Konu: box-sizing

```
.kutü { width: 300px; padding: 20px; border: 5px solid #333; }
```

✗ content-box (varsayılan)

width, **sadece içeriği** kapsar. Gerçek genişlik:
 $300 + 20 + 20 + 5 + 5 = 350\text{px}$ *300 yazdınız, 350 elde ettiniz.*

✓ border-box

width, **padding ve border dahil** toplam genişliktir.
Gerçek genişlik: **300px**
İçerik alanı: $300 - 50 = 250\text{px}$ *300 yazdınız, 300 elde ettiniz.*

💡 İpucu

Her projeye şu satırla başlayın:

```
*, *::before, *::after { box-sizing: border-box; }
```

Bu tek satır, CSS düzen hesaplarının **büyük kısmını** sezgisel hale getirir. Endüstri standardıdır.

1. Adım — HTML

```
<body>
  <div class="container">
    <header> ... </header>
    <main>   ... </main>
    <footer> ... </footer>
  </div>
</body>
```

2. Adım — CSS

```
.container {
  width: 960px;
  margin: 20px auto;
  padding: 20px;
  border: 1px solid #cccccc;
}
```

- ▶ width: 960px — içerik alanı sabitlendi
- ▶ margin: 20px auto — **en yaygın ortalama hilesi**: üstten/alttan 20px, yatayda kalan boşluk eşit bölünür
- ▶ padding: 20px — içerik kenarlara yapışmaz
- ▶ border — kutunun sınırı görünür olur

⚠ Dikkat

margin: auto **yatayda çalışır**, dikeyde çalışmaz. Dikey ortalama için Flexbox gerekir (Hafta 8).

DevTools Kutu Modeli Diyagramı

← → ↺ Dosya /Users/ozal/Desktop/web_kitap/index.html ☆ ⓘ

DevTools is now available in Turkish
Don't show again Always match Chrome's layout

Elements Console >> ⚙️ ⋮ ✕

html body div.container

Styles Computed Layout Event Listeners >>

margin 20
border 1
padding 20
0 1 20 960x530.031 20 1 -552
20
1
20

Filter Show all Group

border-bottom-color solid
border-bottom-style solid

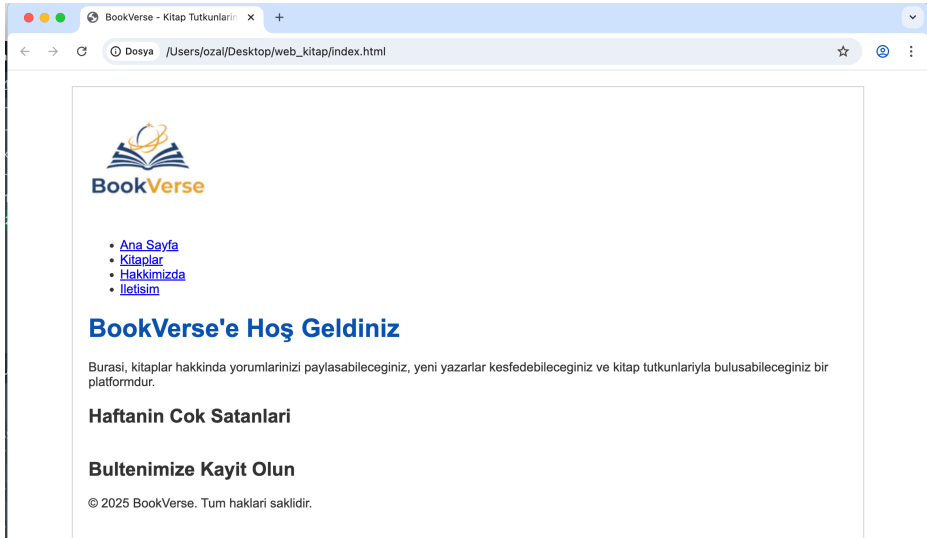
BookVerse

- [Ana Sayfa](#)
- [Kitaplar](#)
- [Hakkımızda](#)
- [İletişim](#)

BookVerse'e Ho

Burasi, kitaplar hakkında yorumlarinizi paylasabileceginiz, yen platformdur.

Geri
İleri
Yeniden Yükle
Farklı Kaydet...
Yazdır...
Yayınla...
Google Lens ile ara
Okuma Modu'nda Aç
Türkçe diline çevir
Sayfa Kaynağını Görüntüle
İncele





Boşluk Yönetimi

BÖLÜM 2

Padding mi, Margin mi?

Özellik	padding	margin
Nerede?	Kenarlığın içinde	Kenarlığın dışında
Arka plan rengi alır mı?	Evet	Hayır — her zaman şeffaf
Negatif değer alır mı?	Hayır	Evet (<code>margin: -10px</code>)
<code>auto</code> değeri	Anlamsız	Ortalama için kullanılır
Çöker mi (collapse)?	Hayır	Evet (dikeyde)
Tıklama alanını büyütür mü?	Evet	Hayır

İpucu

Pratik kural: Bir kutunun **içindeki** nefes alanı için `padding`; kutuların **arasındaki** mesafe için `margin`. Butonlarda **her zaman** `padding` kullanın — tıklanabilir alanı büyütür, dokunmatik ekranda kritiktir (min. 44×44 piksel).

Margin Çökmesi (Collapsing)

Dikey komşu marginler toplanmaz — **büyük olan kazanır**.

```
h2 { margin-bottom: 30px; }  
p  { margin-top:    20px; }
```

✗ Beklenen: $30 + 20 = 50\text{px}$

✓ Gerçek: **30px Üç çökme türü:**

- 1 Komşu kardeşler arası
- 2 Ana element ile ilk/son çocuk arası
- 3 Boş bir kutunun kendi margin'leri

Nasıl Önlenir?

Ana ile çocuk arasında **herhangi bir engel** koyun:

- ▶ `padding` ekleyin (1px bile yeter)
- ▶ `border` ekleyin
- ▶ `display: flex` veya `grid` yapın
- ▶ `overflow: hidden` verin

İpucu

Modern çözüm: Flexbox ve Grid'de margin **çökmez**. `gap` özelliğiyle boşluk vermek her zaman öngörülebilirdir (Hafta 8–9).

```
1 .kart {  
2   width: 100%;           /* ana elementin genisligi kadar */  
3   max-width: 720px;      /* ama en fazla 720px          */  
4   min-width: 280px;      /* ve en az 280px            */  
5  
6   min-height: 200px;     /* icerik kisaysa bile 200px    */  
7   height: auto;          /* icerige gore uzasin (varsayilan) */  
8 }  
9  
10 /* Modern, tek satirlik cozum */  
11 .container { width: min(960px, 100% - 2rem); margin-inline: auto; }
```

⚠ Sabit width Tuzağı

`width: 960px` yazan bir kutu, 375 piksellik telefonda **yatay kaydırma çubuğu** oluşturur. Bunun yerine `max-width` kullanın: küçük ekranda küçülür, büyükte sınırlı kalır.

⚠ Sabit height Tuzağı

`height: 200px` verilen bir kutuya uzun metin girerse içerik **dışarı taşar**. Bunun yerine `min-height` kullanın: en az o kadar olsun, gerekirse uzasın.



Özellği

BÖLÜM 3

block ve inline Farkı

Davranış	block	inline
Yeni satıra geçer mi?	Evet	Hayır — yan yana dizilir
Varsayılan genişlik	Ana elementin tamamı	İçeriği kadar
width/height	Uygulanır	Uygulanmaz
Dikey margin/padding	Uygulanır	Görsel olarak taşar, komşuyu itmez
Yatay margin/padding	Uygulanır	Uygulanır
Örnek etiketler	div, p, h1, section, li	span, a, strong, em, img

```
span { width: 200px; }           /* ETKİSİZ - inline element */
span { display: inline-block;
      width: 200px; }           /* Şimdi çalışır */
```

inline-block: Yan yana dizilir (inline gibi) **ama** boyut verilebilir (block gibi).

display Değerleri

Değer	Davranış
block	Tam genişlik, alt alta
inline	Yan yana, boyut verilemez
inline-block	Yan yana, boyut verilebilir
none	Tamamen kaldırılır — yer kaplamaz, ekran okuyucu okumaz
flex	Esnek kutu düzeni (Hafta 8)
grid	Izgara düzeni (Hafta 9)
table	Tablo gibi davranır

Taşma Yönetimi: overflow

```
1 .kutu { height: 150px; overflow: visible; } /* varsayılan: dışarı tasar */
2 .kutu { overflow: hidden; } /* taşan kısım kirpilir */
3 .kutu { overflow: scroll; } /* her zaman kaydırma cubugu */
4 .kutu { overflow: auto; } /* gerekirse kaydırma cubugu */
5
6 .kutu { overflow-x: auto; overflow-y: hidden; } /* eksen bazlı */
7
8 /* Uzun kelimeleri kırmak için */
9 .metin { overflow-wrap: break-word; hyphens: auto; }
10
11 /* Tek satırda ucu uc noktali kesme */
12 .baslik {
13     white-space: nowrap;
14     overflow: hidden;
15     text-overflow: ellipsis;
16 }
```

Dikkat

overflow: hidden dikkatli kullanın: taşan içerik kullanıcıya **hiç ulaşmaz**. Uzun metinlerde **auto** tercih edin; **hidden**'i sadece *dekoratif* taşmaları kirpmak için kullanın.

Görsel Detaylar: Yuvarlatma ve Gölge

```
1 .kart {
2   border-radius: 8px;                /* dort kose */
3   border-radius: 8px 8px 0 0;        /* sadece ust koseler */
4   border-radius: 50%;                /* daire (kare kutuda) */
5
6   box-shadow: 0 2px 6px rgb(0 0 0 / 0.12);
7   /*      x y bulaniklik renk */
8
9   box-shadow: 0 1px 2px rgb(0 0 0 / 0.08),
10             0 4px 12px rgb(0 0 0 / 0.10); /* katmanli golge */
11
12   outline: 2px solid transparent;    /* odak halkasi icin yer */
13   outline-offset: 2px;
14 }
15
16 .kart:focus-visible { outline-color: var(--renk-vurgu); }
```

⚠ Dikkat

outline: none yazmayın! Klavye kullanıcısı nerede olduğunu göremez. Varsayılan odak halkasını beğenmiyorsanız kaldırmayın, değiştirin: **:focus-visible** ile kendi stilinizi verin.

```
1 .kitap-kart {
2   box-sizing: border-box;
3   max-width: 280px;
4   padding: var(--bosluk-m);
5   margin-bottom: var(--bosluk-m);
6
7   background: var(--renk-zemin);
8   border: 1px solid var(--renk-cizgi);
9   border-radius: var(--yuvarlak);
10  box-shadow: 0 1px 3px rgb(0 0 0 / 0.08);
11 }
12
13 .kitap-kart img {
14   width: 100%;
15   height: 220px;
16   object-fit: cover;          /* orani bozmadan kirp */
17   border-radius: 4px;
18 }
19
20 .kitap-kart h3 { margin: 0.75rem 0 0.25rem; }
21
22 .kitap-kart p {
23   margin: 0;
24   color: var(--renk-metin-soluk);
25   font-size: 0.9rem;
26 }
```

object-fit

Farklı boyutlardaki kitap kapaklarını aynı çerçeveye **oranını bozmadan** sığdırır. **cover**: çerçeveyi doldurur, taşanı kırpar
contain: tamamı görünür, boşluk kalabilir

Kartların Dizilimi

Kartları **yan yana** dizmeyi Hafta 8'de **Flexbox**, Hafta 9'da **Grid** ile öğreneceğiz. Şimdilik alt alta duruyorlar — **kutunun kendisi** doğru olsun yeter.

Kenarlık (border) Detayları

```
1 border: 1px solid #ccc;           /* kalinlik stil renk */
2
3 border-width: 1px 2px;           /* dikey | yatay */
4 border-style: solid dashed dotted double;
5 border-color: red blue green orange;
6
7 border-top: 2px solid var(--renk-ana);
8 border-bottom: 1px dashed var(--renk-cizgi);
9 border-left: 4px solid var(--renk-vurgu); /* vurgu cizgisi */
10
11 /* Ic golge ile "sahte kenarlik" - kutu boyutunu degistirmez */
12 box-shadow: inset 0 0 0 2px var(--renk-vurgu);
```

Stil	Görünüm	Stil	Görünüm
solid	Düz çizgi	double	Çift çizgi
dashed	Kesik çizgi	groove	Oyuk efekti
dotted	Noktalı	none	Kenarlık yok

Blok Biçimlendirme Bağlamı (BFC)

Bazı CSS değerleri, elementin içinde **yeni bir düzen adası** oluşturur. Buna **Block Formatting Context** denir.

🛡 BFC Ne Çözer?

- ▶ Margin çökmesini **engeller**
- ▶ `float` edilmiş öğeleri içeride tutar
- ▶ Taşmaları sınırlar

☰ Nasıl Oluşur?

```
overflow: hidden; /* veya auto */  
display: flow-root; /* modern, temiz */  
display: flex;  
display: grid;  
position: absolute;
```

💡 İpucu

`display: flow-root` bu iş için tasarlanmıştır — `overflow: hidden`'ın yan etkisi (taşan içeriği kesme) olmadan aynı sonucu verir. Modern ve tercih edilen yöntemdir.

Boşluk Ölçeği (Spacing Scale)

Rastgele boşluk vermeyin. **Sınırlı bir ölçek** tanımlayın:

```
1  :root {
2    --bosluk-3xs: 0.25rem;    /* 4px */
3    --bosluk-2xs: 0.5rem;     /* 8px */
4    --bosluk-xs:  0.75rem;    /* 12px */
5    --bosluk-s:   1rem;       /* 16px */
6    --bosluk-m:   1.5rem;     /* 24px */
7    --bosluk-l:   2rem;       /* 32px */
8    --bosluk-xl:  3rem;       /* 48px */
9    --bosluk-2xl: 4rem;       /* 64px */
10 }
11
12 .kart    { padding: var(--bosluk-m); }
13 .bolum   { margin-block: var(--bosluk-xl); }
14 .buton   { padding: var(--bosluk-2xs) var(--bosluk-s); }
```

Neden? 13px, 17px, 22px gibi rastgele değerler sayfayı **dağınık** gösterir. Tutarlı bir ölçek, tasarımcı olmayan birinin bile **düzenli** sayfa yapmasını sağlar.

Sık Yapılan Kutu Hataları

Hata	Sonuç ve Çözüm
<code>width: 100%</code> + <code>padding</code> (content-box ile)	Kutu ana elementten taşar . Çözüm: <code>border-box</code> .
Sabit <code>height</code> verip uzun metin koymak	Metin dışarı taşar. Çözüm: <code>min-height</code> .
<code>margin: auto</code> ile dikey ortalamaya çalışmak	Çalışmaz. Çözüm: Flexbox (Hafta 8).
Her şeye <code>margin-bottom</code> verip son öğede fazladan boşluk	<code>:last-child { margin-bottom: 0; }</code> veya <code>gap</code> .
<code>inline</code> elemente <code>width</code> vermek	Etkisiz. Çözüm: <code>inline-block</code> veya <code>block</code> .
<code>outline: none</code> ile odak halkasını silmek	Klavye kullanıcısı kaybolur. Çözüm: <code>:focus-visible</code> ile değiştirin .
Görsele <code>width</code> verip <code>height</code> vermemek	Sayfa yüklenirken zıplayan düzen (layout shift). Çözüm: <code>aspect-ratio</code> veya ikisini de yazın.

Konumlandırma ()

BÖLÜM 4

1

Değer	Davranış	Tipik Kullanım
static	Varsayılan. Normal akışta. top/left etkisiz	—
relative	Normal yerinde kalır , görsel olarak kayar. Yeri korunur	Kapsül (absolute çocuklar için referans)
absolute	Akıştan çıkarak . En yakın konumlanmış atasına göre yerleşir	Rozet, kapat butonu, açılır menü
fixed	Akıştan çıkar, ekrana sabitlenir. Kaydırınca durur	Sabit üst bar, “yukarı çık” butonu
sticky	Normal akışta durur, eşiğe gelince yapışır	Yapışkan tablo başlığı, menü

```
.kart { position: relative; } /* referans olustur */
.kart .rozet { position: absolute; top: 8px; right: 8px; }
.ust-bar { position: sticky; top: 0; z-index: 10; }
```

⚠ Dikkat

`absolute` bir öge, en yakın `static olmayan` atasına göre yerleşir. Böyle bir öge ata yoksa sayfanın tamamına / 32

Uygulamalı: Kitap Kartına Rozet Ekleme

```
<article class="kitap-kart">
  <span class="rozet">Yeni</span>
  
  <h3>Simiyaci</h3>
  <p>Paulo Coelho</p>
</article>
```

```
.kitap-kart {
  position: relative;      /* REFERANS */
  overflow: hidden;
  border-radius: 8px;
}

.rozet {
  position: absolute;      /* karta gore */
  top: 10px;
  right: 10px;
  z-index: 1;

  padding: 4px 10px;
  background: var(--renk-vurgu);
  color: #fff;
  font-size: 0.75rem;
  border-radius: 999px;
}
```



Altın Kural

position: absolute kullanan **her öğenin**, atasında **position: relative** olmalıdır. Bu ikili **birlikte yazılır** — birini yazıp diğerini unutmak en sık hatadır.



border-radius: 999px

Çok büyük bir değer vermek, yüksekliği ne olursa olsun kutuyu **tam hap şekline** sokar. Rozet ve etiketlerde standart hiledir.

z-index ve Yığın Sırası

Öğeler üst üste geldiğinde hangisinin önde olacağını **z-index** belirler — ama **sadece konumlanmış öğelerde çalışır**.

```
.arka { position: relative; z-index: 1; }  
.onde { position: relative; z-index: 10; }  
.modal { position: fixed; z-index: 1000; }  
  
/* ETKİSİZ: position tanımlı değil */  
.kutu { z-index: 999; }
```

Ölçekli Değerler Kullanın

z-index: 99999 yazmayın. Katman ölçeği tanımlayın:

```
--kat-icerik: 1;  
--kat-menu: 10;  
--kat-ustbar: 20;  
--kat-modal: 100;
```

Yığın Bağlamı

opacity, transform, filter gibi özellikler **yeni bir yığın bağlamı** oluşturur. Bu durumda çocuğun **z-index: 9999** değeri bile ata kutunun dışına **çıkamaz**.

Yapay Zeka ile Kutu Modeli

BÖLÜM 5

Bu hafta öğrendikleriniz **görsel** sonuçlar üretiyor. Yapay zekanın en zayıf olduğu alan da tam olarak burası.

❌ “Şu kısım bozuk” demeyin

Model sayfanızı görmüyor. “Kartım kaydı” dediğinizde **tahmin üretir** ve genellikle alakasız bir çözüm verir. **Bunun yerine:** Kodu, DevTools'tan aldığınız **hesaplanmış değerleri** ve varsa ekran görüntüsünü verin.

🧮 İyi Olduğu Yer: Hesap

Kutu modeli **arimetiktir**. Yapay zeka:

- ▶ **content-box/border-box** hesabını yapar
- ▶ Margin çökmesini tespit eder
- ▶ Shorthand'i açar
- ▶ Neden taşıtığını adım adım gösterir

Prompt 1 — Kutu Hesabı Yaptırma

Yapay Zeka İstemi (Prompt) — Adım Adım Hesaplama

Asagidaki CSS icin, elemanin ekranda kapladigi GERCEK alani adım adım hesapla.
.kart width: 320px; padding: 16px 24px; border: 2px solid #333; margin: 12px 8px;
Hesabi İKİ DURUM icin ayrı ayrı yap: A) box-sizing: content-box (varsayılan) B)
box-sizing: border-box
Her durum icin su tabloyu doldur: | Katman | Sol | Sag | Toplam yatay | | Icerik alani
genisligi | ... | | Border kutusu genisligi | ... | | Margin dahil kaplanan alan | ... |
Sonunda: "İki durum arasindaki fark kac piksel?" sorusunu yanitla.

Cevabı **DevTools kutu diyagramıyla** doğrulayın — kodu gerçekten yazıp bakın.

Prompt 2 — “Neden Taşıyor?” Teşhisi

Yapay Zeka İstemi (Prompt) — Taşma Teşhisi — Veriyle

Sayfamda yatay kaydırma cubugu olustu ve nedenini bulamiyorum.
DevTools > Console'da su komutu calistirdim:
`document.querySelectorAll('*').forEach(e => if (e.offsetWidth >
document.documentElement.clientWidth) console.log(e.tagName, e.className, e.offsetWidth);
);`
Cikti: DIV container 1000 IMG kapak 1200 (viewport genisligi: 960)
Bu ciktiyi yorumla: 1) Hangi eleman(lar) tasmaya neden oluyor? 2) Her biri icin OLASI
CSS nedenini sirala (en olasidan baslayarak) 3) Her neden icin duzeltme onerisi ver 4)
"overflow: hidden" ile duzeltmenin neden KOTU bir cozum olabilecegini acikla
Kodu tamamen yeniden yazma; sadece degistirilecek satirlari goster.

Farkı gördünüz mü? Yapay zekaya **ölçüm verisi** verdik — tahmin etmesini istemedik.

Prompt 3 — Kutu Modeli Sınavı

Yapay Zeka İstemi (Prompt) — Kendinize Sınav Yaptırın

Bana CSS Kutu Modeli üzerine 10 soruluk bir sınav hazırla.

Soru dağılımı: - 3 soru: content-box / border-box hesaplama (sayısal) - 2 soru: margin çokmesi senaryosu - 2 soru: display block/inline/inline-block davranış farkı - 2 soru: padding mi margin mi seçimi (senaryo bazlı) - 1 soru: overflow ve erişilebilirlik

KURALLAR: - Sayısal sorularda net rakamlar ver, "yaklaşık" deme. - Önce SADECE soruları yaz. - Cevap anahtarını en sona koy ve her cevap için hesabın ADIMLARINI göster. - Zorluk: üniversite vize sınavı seviyesi. Türkçe.

Q Yapay Zeka Çıktısını Doğrulama

Sayısal cevapları **kendiniz de hesaplayın**. Yapay zeka aritmetik hata yapabilir — özellikle 4 değerli shorthand (padding: 10px 20px 30px 40px) yorumlarken sık yanılır.

Özet ve Değerlendirme

BÖLÜM 6

Kutu Modeli

`content` → `padding` → `border` → `margin`.
Shorthand sırası: **üst, sağ, alt, sol**.
`padding` arka plan alır, `margin` almaz.

Boyutlandırma

Her projeye `box-sizing: border-box` **ile başlayın.**
Sabit `width/height` yerine
`max-width/min-height`.

`display`

`block`: tam genişlik, alt alta.
`inline`: yan yana, boyut verilemez.
`inline-block`: ikisinin birleşimi.
`none`: tamamen kaldırır.

Yapay Zeka

Hesap yapmada güçlü, **görsel teşhiste kör**. “Bozuk” demek yerine **ölçüm verisi** verin.

Altın Kural: Bir boşluğu açıklayamıyorsanız DevTools kutu diyagramını açın. Tahmin etmeyin, **ölçün**.

Terim	Anlam	Terim	Anlam
Box Model	Her elementin kutu yapısı	max-width	Üst sınır genişlik
content-box	<code>width</code> sadece içerik	min-height	Alt sınır yükseklik
border-box	<code>width</code> = toplam	overflow	Taşma davranışı
Shorthand	Kısa yazım (<code>padding: 10px 20px</code>)	object-fit	Görsel sığdırma
Margin collapse	Dikey marginlerin çökmesi	box-shadow	Dış/iç gölge
display	Kutunun akış türü	outline	Odak halkası (yer kaplamaz)
inline-block	Yan yana + boyutlu	text-overflow	<code>ellipsis</code> ile kesme

- 1 Aşağıdaki kutunun ekranda kapladığı **toplam yatay alanı** her iki `box-sizing` değeri için hesaplayınız:

```
.kutu { width: 400px; padding: 15px 25px; border: 3px solid #000; margin: 0 10px; }
```
- 2 `padding` arka plan rengini alırken `margin` neden almaz? Bu fark hangi durumda işinize yarar?
- 3 `h2 { margin-bottom: 40px; }` ve `p { margin-top: 25px; }` kuralları ile aralarındaki mesafe kaç piksel olur? Neden?
- 4 Bir `<span>` elementine `width: 300px` verdiniz ama değişmedi. Neden? İki farklı çözüm öneriniz.
- 5 `display: none`, `visibility: hidden` ve `opacity: 0` arasındaki farkları **erişilebilirlik** açısından yazınız.
- 6 `outline: none` yazmak neden sakıncalıdır? Doğru alternatif nedir?

Laboratuvar Föyü 6

- ▶ `.container` oluşturma ve ortalama
- ▶ `box-sizing` karşılaştırmalı ölçüm
- ▶ Margin çökmesi deneyi
- ▶ `display` davranış tablosu
- ▶ BookVerse kitap kartı tasarımı
- ▶ Taşma avcılığı (Console komutu ile)
- ▶ Yapay zekaya hesap yaptırma ve doğrulama

Hazırlık

Okuma: Kitap Ünite 2.2 — “The Box Model”
Kaynaklar:

- ▶ developer.mozilla.org/docs/Learn/CSS
- ▶ css-tricks.com/box-sizing
- ▶ web.dev/learn/css/box-model

Hatırlatma: `bookverse` klasörünüzü her hafta **yedekleyin**. Bu hafta dosyanıza `.container` ve `.kitap-kart` eklenecek.

Haftaya Görüşmek Üzere!

Gelecek Hafta:

CSS Renkler, Tipografi ve Arka Planlar

Web fontları ▪ yığını ▪ Tipografi
ölçeği ▪ özellikleri ▪ Gradyanlar