



CSS Temelleri



CodeCraft Bootcamp^{'25}

Bölüm 1: CSS'e İlk Adım

CSS'in ne olduğunu, neden önemli olduğunu anlamak ve temel CSS kurallarını kullanarak bir HTML sayfasını görsel olarak şekillendirebilmek.

1. CSS Nedir ve Neden Önemli?

Cascading Style Sheets - Ardışık Stil Şablonları

Temel görevi:

- Renkler, yazı tipleri, arka planlar ve genel düzenlemeler ekleyerek HTML'i güzelleştirmek.
- Her cihazda uyumlu çalışacak tasarımlar oluşturmak.

Önemi:

- İyi bir CSS, kullanıcı deneyimini doğrudan artırır ve sitenin görsel kimliğini oluşturur.

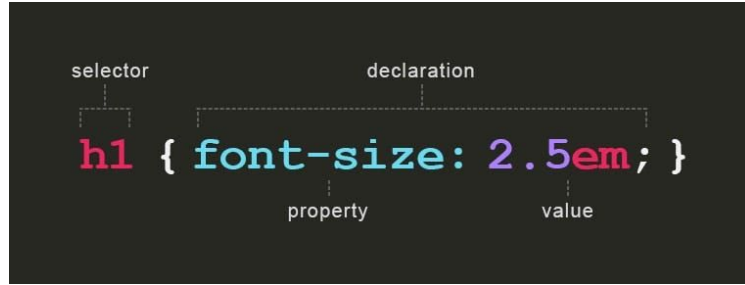
CSS olmadan bir sayfası inceleyelim:

<https://www.amazon.com.tr/>



2. CSS'in Temel Yapısı

Seçici { Özellik: Değer; }



- Seçici (Selector): "Kime stil vereceğiz?" (örn: h1 başlığının)
- Özellik (Property): "Neyini değiştireceğiz?" (örn: yazı boyutunu)
- Değer (Value): "Nasıl değiştireceğiz?" (örn: 2.5em yaparak)

3. CSS Ekleme Yöntemleri ve İlk Kodlama

Inline CSS:

- Sadece tek bir elemana özel stil vermek için kullanılır.
- HTML elemanın içine yazılır

Uygulama:

Dünkü HTML sheet'imiz açarak üzerinde style değişiklikleri yapalım. Başlığımızın rengini değiştirelim.

```
<h1 style="color: blue;"></h1>
```

3. CSS Ekleme Yöntemleri ve İlk Kodlama

Internal CSS:

- Tek bir sayfaya özel stiller için **<head>** içine **<style>** etiketiyle yazılır.

Uygulama:

Sayfadaki tüm paragrafların rengini tek seferde değiştirelim.

```
<style>
  p {
    color: green;
  }
</style>
```

3. CSS Ekleme Yöntemleri ve İlk Kodlama

External CSS (En İyi Yöntem):

- Tüm siteyi tek bir dosyadan yönetmemizi sağlar.
- Her HTML sayfası, başlık bölümündeki **<link>** etiketi içinde harici stil sayfası dosyasına bir referans içermelidir.

Uygulama:

- Şimdi **style.css** adında yeni bir dosya oluşturalım.
- `p { ... }` stilimizi oraya taşıyalım.
- HTML dosyamızın `<head>` bölümüne **stil dosyamızı** bağlayalım.

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



Bölüm 2: CSS Seçicileri ve Box Modeli

CSS elemanlarını neye göre seçeceğimizi kavramak, web tasarımının en temel konseptini öğrenmek

4. Seçiciler (Selector) ile Hedef Belirleme

CSS nasıl yazılır öğrendik, şimdi eleman seçimini öğreneceğiz.

- **Element Seçici:** **p**, **h2** gibi etiket isimleri.
- **Class Seçici:** Birden çok elemanı gruplamak için kullanılır (**.card**, **.button**)
- **ID Seçici:** Sayfada benzersiz, tek bir elemanı hedeflemek için kullanılır (**#header**, **#menu**)

```
.button {  
  font-family: Arial, sans-serif;  
  line-height: 1.6;  
}
```

```
#header {  
  font-size: 24px;  
  color: #333;  
}
```

4. Seçiciler (Selector) ile Hedef Belirleme

ID ve Class Selector Farkı

- **#id selector:** Bir HTML ögesini benzersiz olarak tanımlar. Sayfa içinde bir öğeye özel olmalıdır.
- **.class selector:** Birden fazla öğeye atanabilir. Aynı sınıfa sahip birden fazla öğeyi **gruplamak** için kullanılır.

```
<!DOCTYPE html>
<html lang="tr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ID ve Class Farkı</title>
  <style>
    /* ID seçici */
    #ozel-kutu {
      background-color: blue;
      color: white;
      padding: 10px;
    }

    /* Class seçici */
    .kutu {
      border: 2px solid black;
      padding: 10px;
      margin: 5px;
    }
  </style>
</head>
<body>
  <!-- ID sadece bir öğede kullanılmalı -->
  <div id="ozel-kutu">Bu ID ile seçilmiş bir div.</div>

  <!-- Aynı class birden fazla öğeye uygulanabilir -->
  <div class="kutu">Bu bir kutu.</div>
  <div class="kutu">Bu başka bir kutu.</div>
</body>
</html>
```

4. Seçiciler (Selector) ile Hedef Belirleme

- **Özgüllük (Specificity):** Tarayıcının hangi CSS kuralını uygulayacağını belirlemesine yardımcı olan bir **öncelik** sistemidir.
- "ID her zaman Class'tan, Class da her zaman elementten daha güçlüdür!".
 - Bu, "kodum neden çalışmıyor?" sorusunun en yaygın cevabıdır.

CSS'te Özgüllük/Specificity

id



class



Element

Mini Test

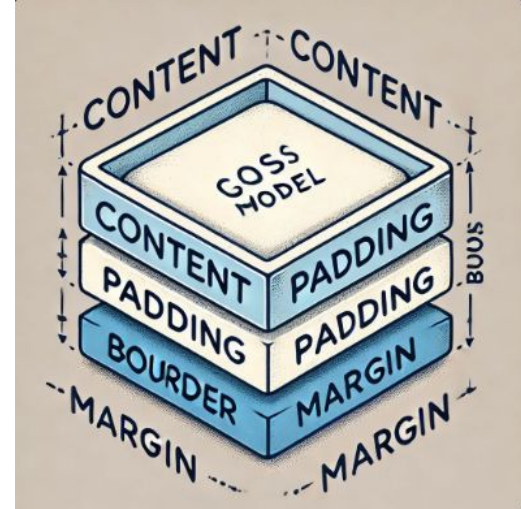
- Bir h1 etiketine hem **class** hem **id** verip ikisine de farklı renk değeri verirsek, hangisi kazanır?

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <style>
      h1#title {
        color: red;
      }
      h1.title {
        color: blue;
      }
      h1 {
        color: orange;
      }
      h1#title {
        color: green;
      }
    </style>
  </head>
  <body>
    <h1 id="title" class="title">Hello World!</h1>
  </body>
</html>
```

4. Her Şey Bir Kutu: CSS Box Model

Web tasarımının en temel konseptidir. HTML öğelerinin tarayıcıda nasıl düzenlendiğini anlamamızı sağlar. Bir öğenin çevresindeki alanları ve nasıl yer kapladığını belirleyen dört ana bileşenden oluşur:

- **Content (İçerik):** Metin veya resmin kendisi.
- **Padding (İç Boşluk):** İçerik ile kenarlık arasındaki boşluktur.
- **Border (Kenarlık):** Kutunun çerçevesi. Kalınlığı ve rengi değiştirilebilir.
- **Margin (Dış Boşluk):** Kutunun diğer kutularla arasına koyduğu mesafedir.



Örnekle İnceleme

- Bir **div** elemanı oluşturalım.
- width, height, background-color, padding, border ve margin verelim.

```
.box {  
  width: 150px;  
  height: 150px;  
  background-color: #4CAF50;  
  border: 2px solid #000;  
  margin: 10px;  
  padding: 5px;  
}
```



CodeCraft Bootcamp^{'25}

Bölüm 3: Sık Kullanılan CSS Özellikleri

En sık kullanılan CSS özelliklerinin neler olduğu öğrenmek ve kullanım yöntemlerini örneklerle incelemek

5. Sayfa Canlandırma

Renkler (Colors)

CSS'te renkler, metinlerin, arka planların, kenarlıkların ve diğer öğelerin görünümünü değiştirmek için kullanılır.

- Renkleri **isim** (red), **hex** (#FF0000) veya **rgb** (rgb(255,0,0)) ile tanımlayabiliriz.
- Bazı renk property'leri:
 - **background-color:** Arka plan rengi ekleme
 - **color:** Text rengini değiştirme
 - **border-color:** Kenarlığı renklendirme

5. Sayfa Canlandırma

CSS'te renk örnekleri

```
<!DOCTYPE html>
<html>
<body>

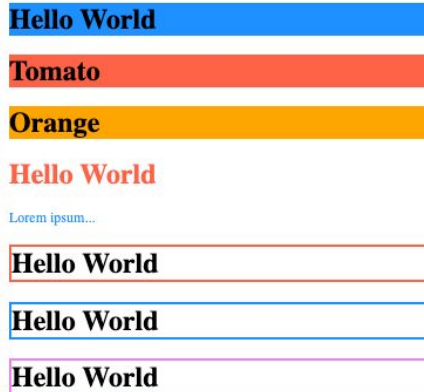
<h1 style="background-
color:DodgerBlue;">Hello World</h1>

<h1 style="background-
color:Tomato;">Tomato</h1>
<h1 style="background-
color:Orange;">Orange</h1>

<h1 style="color:Tomato;">Hello
World</h1>
<p style="color:DodgerBlue;">Lorem
ipsum...</p>

<h1 style="border:2px solid
Tomato;">Hello World</h1>
<h1 style="border:2px solid
DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid
Violet;">Hello World</h1>

</body>
</html>
```



5. Sayfa Canlandırma

Yazılar (Fonts)

- Doğru yazı tipini seçmek, okuyucuların bir web sitesini nasıl deneyimlediği üzerinde büyük bir etkiye sahiptir.
- Okunması kolay bir yazı tipi kullanmak önemlidir.
- Yazı tipi, metninize değer katar.
- Ayrıca, yazı tipi için doğru renk ve metin boyutunu seçmek de önemlidir.

Bazı yazı property'leri:

- **font-family:** Yazı tipi belirler.
- **font-size:** Yazı boyutu belirler.
- **font-weight:** Kalınlık ayarı yapar.

5. Sayfa Canlandırma

CSS'te font-family örnekleri

```
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
  font-family: "Times New Roman", Times, serif;
  font-size: 28px;
}

p.b {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 28px;
}

p.c {
  font-family: "Lucida Console", Courier, monospace;
  font-size: 28px;
}

p.d {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 28px;
}
</style>
</head>
<body>

<p class="a">This is a paragraph, shown in the Times New Roman
font.</p>

<p class="b">This is a paragraph, shown in the Arial font.</p>

<p class="c">This is a paragraph, shown in the Arial font.</p>

<p class="d">This is a paragraph, shown in the Arial font.</p>

</body>
</html>
```

This is a paragraph, shown in the Times New Roman font.

This is a paragraph, shown in the Arial font.

This is a paragraph, shown in the Arial font.

This is a paragraph, shown in the Arial font.

5. Sayfa Canlandırma

CSS'te font-size örnekleri

```
<!DOCTYPE html>
<html>
<head>
<style>
div.a {
  font-size: 15px;
}

div.b {
  font-size: large;
}

div.c {
  font-size: 150%;
}
</style>
</head>
<body>
<div class="a">This is some text.</div>
<div class="b">This is some text.</div>
<div class="c">This is some text.</div>
</body>
</html>
```

The font-size Property

This is some text.
This is some text.
This is some text.

5. Sayfa Canlandırma

CSS'te font-weight örnekleri

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-weight: normal;
  font-size: 28px;
}

p.light {
  font-weight: lighter;
  font-size: 28px;
}

p.thick {
  font-weight: bold;
  font-size: 28px;
}

p.thicker {
  font-weight: 900;
  font-size: 28px;
}
</style>
</head>
<body>

<p class="normal">This is a paragraph.
</p>
<p class="light">This is a paragraph.
</p>
<p class="thick">This is a paragraph.
</p>
<p class="thicker">This is a paragraph.
</p>

</body>
</html>
```

This is a paragraph.

This is a paragraph.

This is a paragraph.

This is a paragraph.

5. Sayfa Canlandırma

CSS'te border örnekleri

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
  border-style: solid;|
  border-color: green;
  font-size: 26px;
}

p.two {
  border-style: dotted;
  border-color: blue;
  font-size: 26px;
}
</style>
</head>
<body>

<p class="one">A solid green border</p>
<p class="two">A dotted blue border</p>

</body>
</html>
```

A solid green border

A dotted blue border

5. Sayfa Canlandırma

Liste Öğelerine Stil Verme

list-style-type: Liste işaretlerini değiştirir.

```
ul.a {  
  list-style-type: circle;  
}  
  
ul.b {  
  list-style-type: square;  
}  
  
ol.c {  
  list-style-type: upper-roman;  
}  
  
ol.d {  
  list-style-type: lower-alpha;  
}
```

Unordered Lists:

- Coffee
- Tea
- Coca Cola

- Coffee
- Tea
- Coca Cola

Ordered Lists:

1. Coffee
2. Tea
3. Coca Cola

- I. Coffee
- II. Tea
- III. Coca Cola

5. Sayfa Canlandırma

Height

CSS'te height özelliği, bir HTML öğesinin yüksekliğini belirlemek için kullanılır. Bu özellik, piksel (**px**), yüzde (%), **vh**, **em**, **rem** gibi farklı birimlerle ifade edilebilir.

`height: 100px;` → Sabit yükseklik

`height: 50%;` → Kapsayıcının %50'si kadar yükseklik

`height: 100vh;` → Ekran yüksekliğinin tamamı

`min-height: 100px;` → En az 100px olur ama genişleyebilir

`max-height: 300px;` → En fazla 300px olur, fazlası kaybolabilir veya kaydırma eklenir

5. Sayfa Canlandırma

CSS'te height örnekleri

```
<!DOCTYPE html>
<html lang="tr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>CSS Height Örneği</title>
  <style>
    .box {
      width: 200px;
      background-color: lightblue;
      margin: 10px;
      text-align: center;
      line-height: 50px;
    }
    .fixed { height: 50px; } /* Sabit yükseklik */
    .percent { height: 50%; } /* Kapsayıcının %50'si kadar */
    .vh { height: 30vh; } /* Ekranın %30'u kadar */
  </style>
</head>
<body>

  <div class="box fixed">50px Yükseklik</div>
  <div class="box percent">%50 Yükseklik</div>
  <div class="box vh">30vh Yükseklik</div>

</body>
</html>
```

50px Yükseklik

%50 Yükseklik

30vh Yükseklik

5. Sayfa Canlandırma

Width

Width özelliği, bir HTML ögesinin genişliğini belirlemek için kullanılır. Piksel (**px**), yüzde (%), **vw**, **em**, **rem** gibi farklı birimlerle ifade edilebilir.

`width: 200px;` → Sabit genişlik

`width: 50%;` → Kapsayıcının %50'si kadar genişlik

`width: 100vw;` → Ekranın tamamı kadar genişlik

`min-width: 150px;` → En az 150px genişlik alır

`max-width: 300px;` → En fazla 300px genişlik alır

5. Sayfa Canlandırma

CSS'te width örnekleri

```
<!DOCTYPE html>
<html lang="tr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>CSS Width Örneği</title>
  <style>
    .box {
      height: 50px;
      background-color: lightblue;
      margin: 10px;
      text-align: center;
      line-height: 50px;
    }
    .fixed { width: 200px; } /* Sabit genişlik */
    .percent { width: 50%; } /* Kapsayıcının %50'si kadar */
    .vw { width: 80vw; } /* Ekranın %80'i kadar */
  </style>
</head>
<body>

  <div class="box fixed">200px Genişlik</div>
  <div class="box percent">%50 Genişlik</div>
  <div class="box vw">80vw Genişlik</div>

</body>
</html>
```

200px Genişlik

%50 Genişlik

80vw Genişlik

6. Elementleri Konumlandırma (Position)

Position özelliği, bir eleman için kullanılan konumlandırma yönteminin türünü belirtir.

- **Static:** Varsayılan değerdir. Elementler HTML kodunda yazıldığı sırayla, normal akışta görüntülenir. top, left gibi özellikler işlemez.
- **Relative:** "Normalde olması gereken yere göre biraz kaydır" anlamına gelir. Bunu **top**, **left**, **bottom**, **right** özellikleriyle yaparız. En önemli özelliği, **position: absolute** ile konumlandırılacak elemanlar için bir "çapa" veya "ata" görevi görmesidir.
- **Absolute:** "Kendisine en yakın **position: relative** olan atasına göre konumlanır". Eğer öyle bir atası yoksa, kendini doğrudan **<body>** etiketine göre hizalar. Bu özellik, elementi sayfa akışından tamamen çıkarır; yani diğer elementler o yokmuş gibi davranır.

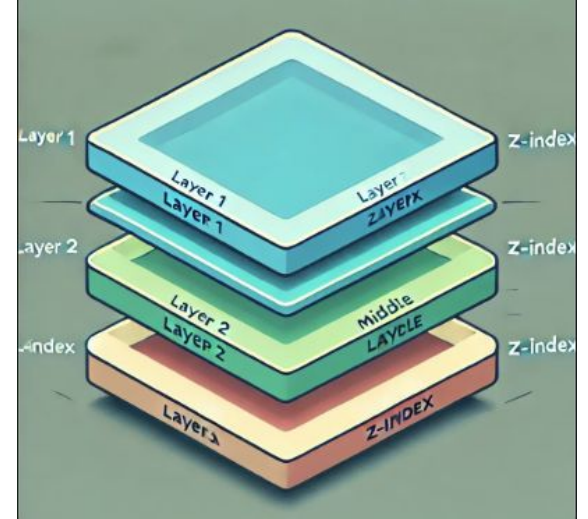
6. Elementleri Konumlandırma (Position)

- **Fixed:** "Sayfa ne kadar kaydırılırsa kaydırılsın, ekranda sabit kalır". Konumunu her zaman tarayıcı penceresine (viewport) göre alır.
 - Sıkça kullanılan örnekler: "Sayfanın sağ altındaki 'Yukarı Çık' butonu" veya "Ekranı yapışık kalan çerez bildirimi".
- **Sticky:** Relative ve fixed pozisyonlarının bir birleşimidir. Element, normalde akışta durur , ancak sayfa kaydırılıp belirlenen bir noktaya (örneğin top: 0) gelince ekrana fixed gibi yapışır.
 - Çalışması için **top**, **bottom** gibi bir yapışma noktası belirtilmesi **zorunludur**.
 - En Popüler Örnek: Sayfa aşağı kaydırıldığında yukarıda sabit duran menü.

6. Elementleri Konumlandırma (z-index)

Konumlandırılmış elemanların üst üste gelme sırasını (katmanını) belirler.

- Değer her kadar artar ise elemanı o kadar öne çıkarır.
- Eğer -1 değeri verir isek bütün elemanların arkasına geçecektir.
- En fazla ve en düşük z-index değeri:
 ± 2147483647



Örnekle İnceleme

- Bir ana kapsayıcı (card-container), bir kartın kendisi (card) ve bir "yeni" etiketinden (new-label) oluşan bir HTML oluşturalım.
- Position, z-index CSS özellikleri ile YENİ etiketini card'ın sol üst köşesinde konumlandıralım.

```
<div class="card-container">
  <div class="new-label">YENİ</div>

  <div class="card">
    <h3>Kart Başlığı</h3>
    <p>Bu kartın içeriğidir. Üzerindeki etiketin z-index değeriyle nasıl öne veya arkaya geçtiğini göreceğiz.</p>
  </div>
</div>
```

7. Göze Hoş Görünen Dokunuşlar

- **border-radius:** Kenarlıkları yuvarlatmak için kullanılır. Bir kareye **border-radius: 50%** vererek aşağıdaki gibi daire yapabiliriz:



Example

Rounded Image:

```
img {  
  border-radius: 8px;  
}
```



Example

Circled Image:

```
img {  
  border-radius: 50%;  
}
```


7. Göze Hoş Görünen Dokunuşlar

- **box-shadow:** Elementlerin kutusuna gölge ekleyerek onlara bir derinlik hissi verir.

Thumbnail Image:



Example

```
img {  
  border: 1px solid #ddd;  
  border-radius: 4px;  
  padding: 5px;  
  width: 150px;  
}  
  

```

Thumbnail Image as Link:



Example

```
img {  
  border: 1px solid #ddd;  
  border-radius: 4px;  
  padding: 5px;  
  width: 150px;  
}  
  
img:hover {  
  box-shadow: 0 0 2px 1px rgba(0, 140, 186, 0.5);  
}
```

7. Göze Hoş Görünen Dokunuşlar

- `img` etiketi ile ilgili bir tüyo



Bir görüntünün orijinal boyutundan daha büyük olmasını istemiyorsanız, ancak gerektiğinde küçültülmesini istiyorsanız, aşağıdakileri ekleyin:

Örnek

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

Insider | USEINSIDER.COM

CodeCraft
Bootcamp^{'25}



Soru & Cevap