

Bilgisayar Programlama IV
HTML & CSS ile Web Tasarımı Dersi
CSS Ders Notları

İçindekiler Tablosu

CSS Nedir?	4
CSS ile neler yapılır?	4
CSS Nasıl Çalışır?	4
CSS'in Çalışma Yapısı	5
CSS Nereye Yazılır?	5
1. Inline CSS:	5
2. Internal CSS:	5
3. External CSS:	5
CSS Selectors (Seçiciler)	6
1. Element Selector	6
2. Class Selector	6
3. ID Selector	6
4. Group Selector (Grup Seçici)	6
CSS Renkler ve Arka Planlar	7
1. Text Color (Yazı Rengi)	7
2. Background Color (Arka plan rengi)	7
3. Renk Yazma Yöntemleri	7
4. Background Image (Arka plan resmi)	7
5. Background Repeat	7
CSS Text ve Font Özellikleri	8
1. Font Size	8
2. Font Family	8
3. Text Align (Yatay Hizalama)	8
4. Font Weight (Kalınlık)	8
5. Font Style (Eğik Yazı)	8
6. Line Height (Satır Aralığı)	8
7. Text Decoration (Altı Çizgili vb.)	8
CSS Box Model	9
1. Margin (Dış Boşluk)	9
2. Border (Çerçeve, Kenarlık)	9
3. Padding (İç Boşluk)	9
4. Content (İçerik)	9
5. Width ve Height	9
CSS Display ve Position (Yerleşim Sistemi)	10
1. Display Özelliği	10
a) block:	10
b) inline:	10
c) inline-block:	10
d) none:	10
2. Position Özelliği	10
a) static (varsayılan):	10
b) relative:	10
c) absolute:	10

c) fixed:	10
d) sticky:	11
CSS Flexbox	12
1. Flex Kullanımı	12
2. flex-direction	12
3. justify-content	12
4. align-items	12
5. gap	13
6. flex-wrap	13

CSS Nedir?

CSS, yani Cascading Style Sheets, web sayfalarının görünümünü ve tasarımını düzenlemek için kullanılan bir stil dilidir.

Basitçe söylemek gerekirse:

- HTML sayfanın iskeletini oluşturur (başlık, paragraf, buton vb.)
- CSS ise bu yapının renklerini, boyutlarını, düzenini ve görünümünü belirler

CSS ile neler yapılır?

- Metin renklerini ve yazı tiplerini değiştirme
- Arka plan rengi veya görseller ekleme
- Sayfa düzenini (layout) ayarlama
- Buton ve menü tasarımları oluşturma
- Mobil ve masaüstü uyumlu (responsive) tasarım yapma

Küçük bir örnek

```
p {  
  color: blue;  
  font-size: 18px;  
}
```

Bu kod, sayfadaki tüm paragrafları mavi renkte ve 18px boyutunda yapar.

Kıscası: CSS, web sitelerinin güzel görünmesini sağlayan teknolojidir.

CSS Nasıl Çalışır?

CSS, HTML elemanlarını seçip onlara stil uygulayarak çalışır.

Mantık şu şekildedir:

1. HTML'de bir eleman vardır
2. CSS bu elemanı "seçer"
3. Belirlenen stilleri uygular

```
<p>Merhaba Dünya</p>
```

Bu HTML kodunda bir paragraf vardır.

Şimdi CSS yazalım:

```
p {  
  color: red;  
  font-size: 24px;  
}
```

Burada:

- p → seçici (selector)
- color → yazı rengi
- font-size → yazı boyutu

Sonuç olarak tarayıcı paragrafı: kırmızı renkte ve 24 piksel boyutunda gösterir.

CSS'in Çalışma Yapısı

CSS genel olarak şu yapıyı kullanır:

```
secici { ozellik: deęer; }
```

Örneęin:

```
h1 {color: blue; }
```

Tüm h1 başlıklarını seçer ve hepsinin rengini mavi yapar.

CSS Nereye Yazılır?

CSS, 3 farklı şekilde kullanılabilir:

1. Inline CSS:

HTML etiketinin içerisine yazılır.

```
<p style="color:red">Yazı </p>
```

2. Internal CSS:

<style> etiketi içine yazılır.

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

3. External CSS:

Ayrı .css dosyasında tutulur.

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

style.css:

```
p {color:red; }
```

CSS Selectors (Seiciler)

CSS'te seiciler, hangi HTML elemanına stil uygulanacađını belirler.

1. Element Selector

HTML etiket adına gre seim yapar.

```
p {color:red; }
```

Tm <p> etiketleri kırmızı olur.

2. Class Selector

Belirli bir sınıfa sahip elemanları seer. CSS de sınıf (class) iin . kullanılır.

Html:

```
<p class="kirmizi">Metin 1 </p>
<p>Metin 2</p>
```

Css:

```
.kirmizi { color: red; }
```

Sonu: Sadece Metin 1 yazısı kırmızı renkte olur.

3. ID Selector

Tek bir zel elemanı semek iin kullanılır. CSS de ID iin # kullanılır.

Html:

```
<p id="p1">Metin 1 </p>
<p id="p2">Metin 2</p>
<p>Metin 3 </p>
```

Css:

```
#p1 { color: red; }
```

Sonu: Sadece Metin 1 yazısı kırmızı renkte olur.

4. Group Selector (Grup Seici)

Birden fazla elemanı aynı anda seer.

```
h1, p, div { color: red; }
```

Tm h1, p ve div elemanları kırmızı renkte olur.

CSS Renkler ve Arka Planlar

Bu konu, web sayfasındaki yazı renklerini ve arka planları düzenlemek için kullanılır.

1. Text Color (Yazı Rengi)

color özelliği kullanılır.

```
p { color: blue; }
```

Tüm paragrafları mavi yapar.

2. Background Color (Arka plan rengi)

background-color özelliği kullanılır.

```
body { background-color: pink; }
```

Sayfanın arka plan rengini pembe yapar.

3. Renk Yazma Yöntemleri

CSS'te renkler farklı şekillerde yazılabilir.

- Renk ismi: Red, Green, Blue gibi hazır HTML renk isimleri kullanılabilir.
- HEX Kodları: #ff0000 gibi 6 haneli kod sistemi kullanılabilir.
- RGB: color: rgb(255,0,0) şeklinde Yeşil, Kırmızı ve Mavi renk değerleri verilebilir.
- RGBA (Saydamlık): background-color: rgba(0,0, 255,0.5); şeklinde RGB kodu ve sondaki saydamlık değeri ile kullanılabilir. Örnekte yüzde elli saydamlık verilmiştir.

4. Background Image (Arka plan resmi)

Arka plana resim eklemek için kullanılır.

```
body { background-image: url("resim1.jpg"); }
```

5. Background Repeat

Resmin tekrar edip etmeyeceği belirtilir.

```
background-repeat: no-repeat
```

Sonuç: Resim sadece 1 tane gösterilir. Arka planda döşeme/dizme yapılmaz.

CSS Text ve Font Özellikleri

Bu konu, yazıların nasıl görüneceğini kontrol eder (boyut, font, hizalama, kalınlık vb.).

1. Font Size

Yazı büyüklüğünü kontrol eder.

```
p {font-size: 20px;}
```

2. Font Family

Yazı tipini ayarlar.

```
p {font-family: Arial;}
```

3. Text Align (Yatay Hizalama)

Yazının yatayda nasıl hizalanacağını ayarlar.

- left: Sola
- right: Sağa
- center: Ortaya
- justify: İki yana yaslı

```
p {text-align: center; }
```

4. Font Weight (Kalınlık)

Yazının normal veya kalın gösterimi için kullanılır.

```
.baslik {font-weight: bold; }
```

5. Font Style (Eğik Yazı)

Yazının eğik gösterimi için kullanılır.

```
.baslik {font-style: italic; }
```

6. Line Height (Satır Aralığı)

Metinler arasındaki boşluğu ayarlar. Bu boşluğu arttırarak okunabilirliği arttırabiliriz.

```
p {line-height: 1.6 }
```

7. Text Decoration (Altı Çizgili vb.)

Yazının altını veya üzerini çizebiliriz. Veya varsa çizgileri kaldırabiliriz.

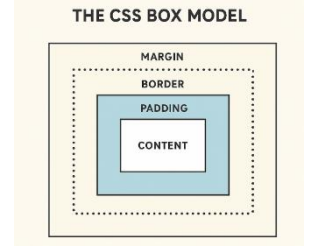
- none: çizgisiz
- underline: altı çizgili
- line-through: üzeri çizgili

```
a {text-decoration: none; }
```

CSS Box Model

CSS'te her HTML elemanı bir **kutu (box)** gibi düşünülür. Bu kutu 4 ana katmandan oluşur:

- Dış Boşluk
- Kenarlık
- İç Boşluk
- İçerik



1. Margin (Dış Boşluk)

Eleman ile diğer elemanlar arasındaki boşluktur.

- margin: 4 taraftan boşluk
- margin-top: üstten boşluk.
- margin-right: sağdan boşluk.
- margin-bottom: alttan boşluk.
- margin-left: soldan boşluk.

```
div {margin: 30px;}
```

2. Border (Çerçeve, Kenarlık)

Elemanın çerçevesidir.

```
div {border:solid 1px red;}
```

- Solid: Düz çizgi
- 1px: kalınlık
- Red: çerçevenin rengi.

3. Padding (İç Boşluk)

Kenarlık ile içerik arasındaki boşluktur.

- padding: 4 taraftan boşluk
- padding -top: üstten boşluk.
- padding -right: sağdan boşluk.
- padding -bottom: alttan boşluk.
- padding -left: soldan boşluk.

```
div {padding: 30px;}
```

4. Content (İçerik)

Elemanın gerçek içeriğidir (yazı, resim vb.)

```
<div>Merhaba Dünya!</div>
```

5. Width ve Height

Elemanın genişlik ve yüksekliğini belirler.

```
div {width: 300px; height: 200px;}
```

Varsayılan olarak width elemanın sadece içerik (content) alanını kapsar. Bunu değiştirmek için:

```
box-sizing:border-box
```

Bu sayede, genişliğe padding ve border da dâhil edilir.

CSS Display ve Position (Yerleşim Sistemi)

Bu konu, HTML elemanlarının **sayfada nasıl duracağını ve konumlanacağını** belirler.

1. Display Özelliği

Bir elemanın **nasıl görüneceğini (blok mu, satır içi mi)** belirler.

a) block:

- Tüm satırı kaplar.
- Alt alta dizilir.
- Örnek: div, p, h1

```
div {display:block; }
```

b) inline:

- Sadece içeriği kadar yer kaplar.
- Yan yana durur.
- width/height tam çalışmaz.
- Örnek: span, a

```
span {display:inline; }
```

c) inline-block:

- Yan yana durur .
- Ama width ve height verilebilir.

```
div {display:inline-block; }
```

d) none:

- Elemanı tamamen gizler.
- Sayfa yer kaplamaz ve görünmez.

```
div {display:none; }
```

2. Position Özelliği

Elemanın sayfada nerede duracağını kontrol eder.

a) static (varsayılan):

- Normal akışta durur.
- Hareket ettirilemez.

```
div {position:static; }
```

b) relative:

- Kendi normal yerinden kaydırılır.
- Boşluk bırakmaya devam eder.

```
div {position:relative; top:20px; left:30px; }
```

c) absolute:

- En yakın "position verilmiş" elemana göre konumlanır (absolute, relative veya fixed)
- Sayfa akışından çıkar.
- Verilen pozisyonda konumlanır.

```
div {position:absolute; left:10px; top:10px }
```

c) fixed:

- Ekranı sabitlenir.
- Sayfa kaydırılsa bile sabit kalır.

```
div {position:fixed; bottom:10px; right:10px; }
```

d) sticky:

- Sayfa kaydırıldığı zaman belli bir noktada sabit kalır.
- Özellikle yatay menü çubuklarında çok kullanılır.

```
div {position:sticky; top:0; }
```

CSS Flexbox

Flexbox, web sayfalarında elemanları kolay hizalamak ve yerleřtirmek

için kullanılan modern CSS yerleřim sistemidir.

Özellikle:

- yan yana dizme
- ortalama
- boşluk dağıtma
- responsive tasarım

iřlerinde çok kullanılır.

1. Flex Kullanımı

Bir kapsayıcıya (container) uygulanır.

HTML

```
<div class="kutu">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

CSS

```
.kutu {
  display: flex;
}
```

Sonuç: Elemanlar yan yana dizilir.

2. flex-direction

Elemanların yönünü belirler.

```
flex-direction: row;
```

Seçenekler:

- row → yatay
- column → dikey

3. justify-content

Yatay hizalama yapar.

```
justify-content: center;
```

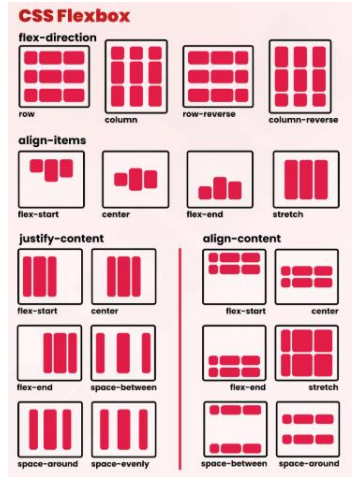
Seçenekler:

Değer	Açıklama
center	Ortala
flex-start	Başta
flex-end	Sonda
space-between	Araları eşit
space-around	Çevresinde boşluk

4. align-items

Dikey hizalama yapar.

```
display: flex; align-items: center;
```



5. gap

Elemanlar arası boşluk verir.

```
display: flex; gap: 20px;
```

6. flex-wrap

Elemanların alta geçmesini sağlar.

```
flex-wrap: wrap;
```

Örnek:

```
.kutu {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  gap: 20px;  
}
```