

Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS)

- Cascading Style Sheets, web sayfalarının **görüntüsü ile içeriğinin** birbirinden ayrı biçimde ele alınmasını sağlar
- HTML elemanlarına etki eden bir takım **CSS kurallarından** oluşurlar
- Bu kurallar ayrı CSS dosyalarında, yada ilgili HTML sayfası içerisinde, veya HTML elemanı üzerinde tanımlanabilir

CSS'in Temelleri

Her bir kural bir veya daha fazla **selector** içerir. Selector'ler HTML elemanlarını tespit etmek içindir

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

Property ifadeleri ise **label:value** şeklindedir. ; ile sona erer. HTML elemanının görüntüsünü değiştirirler

```
td {  
  background: yellow;  
  font-weight: bold;  
}
```

Her bir kural yine bir **deklerasyon bloğundan** oluşur. Bir deklarasyon bloğu içerisinde de bir veya daha fazla **property ifadesi** vardır

CSS'in Temelleri

- Her bir HTML elemanı **belirli bir takım property'leri** kabul eder
- **Inheritance** kabiliyeti sayesinde HTML elemanları içinde yer aldığı parent HTML elemanının CSS property ifadelerine sahip olabilirler

```
table {
  color: blue;
  background: yellow;
}
```

Örneğin, <table> elemanları için tanımlı bu property ifadeleri otomatik olarak HTML sayfasındaki <table> elemanı içerisindeki <tr>, <td> child elemanları için de geçerli olacaktır

Attribute

- Eleman tipi yanında HTML elemanın **class attribute** değeri de selector ifadesinde yer alabilir
- Class attribute içinde boşlukla ayırarak **birden fazla değer** de tanımlanabilir

```
<html>
<body>
  <p class="red">This paragraph's color is red</p>
  <p class="blue">This is blue</p>
  <table>
    <tr>
      <td class="red">This is cell has red background</td>
      <td class="blue">This has blue background</td>
    </tr>
  </table>
</body>
</html>
```

```
p.red {color: red;}
p.blue {color: blue;}
td.red {background: pink;}
td.blue {background: yellow;}
```

Bu sayede aynı tür elemanlara farklı kurallar uygulanabilir

Universal Selector: *

- HTML elemanı yerine * kullanılarak **class attribute değeri aynı olan bütün elemanlara** aynı CSS kurallarının uygulanması mümkündür

* kullanılması zorunlu değildir

He iki tanım da class attribute değeri eşleşen bütün HTML elemanları için geçerlidir

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

.yellow {
    background: yellow;
}
```

Containment İlişğine Göre Selector'ler

```
<body>
  <p class="mytext">Here is some text inside a
    paragraph element</p>
  <table class="mytable">
    <tr>
      <td class="mytext">Here is text inside
        a table and inside a td element.</td>
    </tr>
  </table>
</body>
```

```
.mytable .mytext {color: blue;}
```

Bu kural sadece <table> elemanı altındaki <td> elemanına uygulanacaktır

<p> elemanı <table> altında olmadığı için uygulanmayacaktır

Parent-child ilişkisi herhangi bir düzeyde olabilir. Hemen bir alt eleman olmak zorunda değildir

CSS Kurallarının Sırası

- Aynı HTML elemanına etki eden, farklı yerlerden gelen CSS kuralları olabilir
- Eğer elemanın **style attribute**'unda tanım varsa bu geçerlidir
- Yoksa, ilgili HTML elemanına selector ile match eden **en spesifik kural** geçerlidir
- Eğer birden fazla aynı düzeyde match eden kural varsa o zaman **yükleme sırasına göre sonra gelen önce geleni ezer**

CSS Kurallarının Sırası

```
.v-button {}  
.v-verticallayout .v-button {}  
.v-app .v-verticallayout .v-button {}
```

Üçüncü sıradaki kural en spesifik match olduğu için bu geçerli olacaktır

```
.v-button {color: white}  
.v-button {color: black}
```

İki kural da aynı düzeyde match ettikleri için sıralamada sonra gelen geçerli olacaktır.

```
.v-app .v-button {color: white}  
.v-button {color: black}
```

En spesifik eşleşme, sıralamada en yüksek önceliğe sahiptir. Dolayısı ile ilk kural geçerli olacaktır

CSS !important Direktifi

- Bazı durumlarda daha spesifik kuraldaki property ifadesi yerine **daha genel veya önceki kuralın property ifadesinin geçerli olması** istenebilir

```
.v-button {height: 30px !important;}
```

```
<div class="v-button" style="height: 20px;">
```

Normalde style attribute'undaki ifadenin geçerli olması gerekirken !important'dan dolayı .v-button kuralındaki ifade geçerli olacaktır

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

