CSS Selectors (CSS Seçiciler)

CSS seçicileri (selector), HTML isimlerine, id'sine, class'ına, niteliğine ve daha fazlasına göre HTML öğelerini bulmak veya seçmek için kullanılır.

- **Temel seçiciler** (isim, kimlik, sınıfa göre öğeleri seçin)
- **Grup seçiciler** (aynı özelliğe sahip farklı seçicileri gruplandırın)
- **Birleştirici seçiciler** (aralarındaki belirli bir ilişkiye göre öğeleri seçin)
- **Sözde sınıf seçiciler** (belirli bir duruma göre öğeleri seçin)

*(1996 yılında **W3C** önerisi ile **CSS1 (Cascading Style Sheets, Level 1)** HTML etiketleri için basit bir görsel biçimlendirme modeli olarak duyuruldu.* Class as selector*,* ID as selector *ve* Contextual selectors *başlıkları altında kullanıma sunulan seçiciler ile tanımlarına uyan elementler bulunur ve barındırdıkları nitelikler üzerinden etiketler yeniden düzenlenir[3]. 1998 yılında duyurulan ve CSS1 temel alınarak geliştirilen **CSS2.1 (Cascading Style Sheets Level 2 Revision 1)** ile seçiciler çok daha kapsamlı hale gelmekte. Örneğin, pattern (şablon) tanımları kapsamında* type*,* class *ve* id *seçicilere ek olarak* universal*,* child*,* adjacent*,* attribute *kullanımları da bu sürüm itibariyle değerlendirilebilmektedir. 2001 yılında duyurulan CSS3 ile CSS2.1'e ek olarak* attribute*,* regular expression*,* elements contained*,* criteria*,* state *gibi kullanımlar da seçiciler arasına dahil edilmiştir.)*

Basic Selectors (Temel Seçiciler)

Bu yazıda bir çok seçici türüne yer verilecektir. Ancak ilk olarak temel bir kaç seçici ile işleme başlayalım.

En ilkel seçici türleri 3 seçici olmak üzere bu bölümde incelenebilir.

- HTML etiketleri seçmek için etiket seçici,
- id değerlerine göre seçim yapmak için kimlik seçici,
- class isimlerine göre seçim yapmak için sınıf seçici

*** (Universal selector) (Küresel Seçici)**

CSS tarafından sağlanan evrensel seçici, HTML sayfasındaki tüm elemanları veya benzer elemanları seçmek için kullanılır. Genellikle bir yıldız veya ardından bir seçici olarak yazılır. Bu yıldız, başka bir öğenin içindeki tüm öğeleri seçme yeteneğine de sahiptir.

Evrensel seçici, web sayfanızdaki tüm HTML öğelerine belirli bir stil eklemek istediğinizde yardımcı olur. Ayrıca, HTML sayfanızın öğesindeki her bir öğe için de kullanılabilir.

Bu seçicinin temel sözdizimi şöyledir:

```
* {
    ozellik: deger;
  }
```

Örnek :

```
* {
     margin : 0px;
     padding : 0px;
  }
```
Yukarıdaki CSS kodu tüm HTML öğelerinin dış ve iç boşluklarını “0” yapar.

Örnek :

```
<!DOCTYPE html>
<html lang="tr">
<head>
    <title>Tasarım Kodlama</title>
    <style>
        *{
            color:red;
        }
    </style>
</head>
<body>
    <h1>Tasarım Kodlama</h1>
    <p>Merhaba Dünya</p>
    <p lang="en">Hello World</p>
</body>
</html>
```
Yukarıdaki kodda tüm HTML öğlerinin yazı rengi kırmızı(red) yapılmıştır.

tag (Type selector) (Etiket Seçici)

Etiket seçici, tüm etiket adına dayalı ögeleri seçer. Örnek olarak sayfamızda bulunan tüm <p> etiketlerinin yani paragrafların **renginin kırmızı** ve **hizalama değerinin ortalı** olmasını istersek;

```
<!DOCTYPE html>
<html lang="tr">
<head>
    <title>Tasarım Kodlama</title>
    <style>
        p{
            color:red;
```

```
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Tasarım Kodlama</h1>
    <p>Merhaba Dünya</p>
    <p lang="en">Hello World</p>
</body>
</html>
```

idname (ID selector) (Kimlik Seçiciler)

ID seçici, belirli bir öğeyi seçmek için bir HTML öğesinin id özelliğini kullanır.
Bir öğenin kimliği bir sayfa içinde benzersiz olmalıdır, bu nedenle kimlik seçici bir benzersiz öğe seçmek için kullanılır!
Belirli bir kimliğe sahip bir eleman seçmek için, "#" karakteri ve ardından öğenin kimliği yazılır.
Not: Bir kimlik adı sayı ile başlayamaz!
Örnek: Aşağıdaki örnekte #ad öğesinin arkaplan rengi siyah, yazı rengi kırmızı olarak ayarlanmıştır.

```
<!DOCTYPE html>
<html lang="tr">
<head>
    <title>Tasarım Kodlama</title>
    <style>
        #ad{
            background-color: black;
            color:red;
        }
    </style>
</head>
<body>
    <input type="text" id="ad"><br>
    <input type="text" id="soyad"><br>
</body>
</html>
```

.classname (Class selector) (Sınıf Seçici)

Sınıf seçici, belirli bir sınıf özelliğine sahip öğeleri seçer. "**id seçici**"den faklı olarak **birden fazla** öğeye uygulanabilir.

id ile class Arasındaki Fark

Id Seçici benzersizdir. ID(Kimlik) seçici, web sayfasındaki tek bir öğeye uygulanan stil kurallarını tanımlamanıza olanak tanır. Her web sayfası, tek bir kimlik özniteliğine sahip yalnızca bir öğeye sahip olabilir. Kimlik seçiciler, bir hashtag(#) işareti kullanılarak tanımlanır. Hemen ardından, bir dizi stil kuralı uygulamak istediğiniz kimlik değeri gelir.
Class Seçici benzersiz değildir. Bir class(sınıf) seçici, belirli bir değere eşit bir sınıf özniteliğine sahip herhangi bir öğeye uygulanan stil kurallarını tanımlamanıza olanak tanır. Sınıflar, birden çok öğe için kullanılabilir. Bu nedenle, bir sınıf seçici kullanarak bir stil uygularsanız, o sınıfı paylaşan herhangi bir öğe tanımladığınız stillere tabi olacaktır.
Belirli bir sınıfa sahip öğeleri seçmek için, bir nokta (.) Karakteri ve ardından sınıfın adını yazın.
Aşağıdaki örnekte, **class = 'ortala'** olan tüm HTML öğeleri kırmızı ve orta hizalanmış olacaktır:

```
<!DOCTYPE html>
<html>
<head>
<style>
.ortala {
    text-align: center;
    color: red;
}
</style>
</head>
<body>
<h1 class="ortala">Kırmızı ve Ortalanmış Başlık</h1>
<p class="ortala">Kırmızı ve Ortalanmış Paragraf.</p>
</body>
</html>
```

Ayrıca, yalnızca belirli HTML öğelerinin bir sınıf tarafından etkilenmesi gerektiğini de belirtebilirsiniz. Örnek olarak yukarıdaki örnekte sadece sınıfı ortala olan öğelerden sadece paragraf olanları ortalayalım.

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
p.ortala {
    text-align: center;
    color: red;
}
</style>
</head>
<body>
<h1 class="ortala">Kırmızı ve Ortalanmış Başlık</h1>
<p class="ortala">Kırmızı ve Ortalanmış Paragraf.</p>
</body>
</html>
```

Html elemanlarına **birden fazla sınıf** özelliği uygulanabilir. Aşağıdaki örneği incelediğinizde en sondaki paragrafın ortala ve buyuk isimli sınıfa dahil olduğunu göreceksiniz.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.ortala{
    text-align: center;
    color: red;
}
p.buyuk{
    font-size: 300%;
}
</style>
</head>
<body>
<h1 class="ortala">Bu başlığa biçimlendirme
uygulanmamıştır.</h1>
<p class="ortala">Bu paragraf kırmızı renkli ve
ortalanmıştır..</p>
<p class="ortala buyuk">Bu paragraf iki tane sınıf özelliği
içerir. kırmızı, ortalanmış ve yazı boyutu arttırılmıştır.</p>
```

```
</body>
</html>
```

örnekte **h1** etiketinde bulunan **başlık** ortala sınıfına dahil olduğu halde herhangi bir biçimlendirme olmamıştır. Bunun sebebi css özelliğinin **p.ortala** şeklinde yazılmasıdır. Bu şekilde sadece sınıf adı ortala olan **<p>** etiketlerinin biçimlendirmeden etkilenmesi sağlanmıştır.

Not: Class isimleri rakam ile başlayamaz.

[attr] (Attribute selector)(Özellik Seçici)

Özellik seçici (attribute selector) ile belirli bir özniteliğe (attribute) sahip tüm öğeleri seçebiliriz. Öznitelikler farklı biçimlerde tanımlanabilirler;

```
 [attr] [attr=value] [attr~=value] [attr|=value] [attr^=value]
 [attr$=value] [attr*=value]
```

Örnek:

```
a[href$=".org"]
{
   font-style: italic;
}
* span {
   color: blue;
}
```

Grouping selectors (Grup Seçiciler)

Aşağıdaki gibi aynı stil tanımlarına sahip öğeleriniz varsa;

```
h1 {
    text-align: center;
    color: red;
}
h2 {
    text-align: center;
    color: red;
}
p {
    text-align: center;
    color: red;
```

Kodu küçültmek için seçicileri gruplamak daha iyi olacaktır. Seçici gruplamak için, her seçiciyi virgülle ayırın.

Aşağıdaki örnekte seçicileri nasıl gruplayabileceğinizi göreceksiniz.

```
h1, h2, p {
    text-align: center;
    color: red;
}
```

Yukarıda <h1>, <h2> ve <p> etiketlerine ortalanmış ve kırmızı renk özelliği verilmiştir.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1>Başlık!</h1>
<h2>Küçük Başlık!</h2>
<p>Bu bir paragraf.</p>
</body>
</html>
```

Yukarıdaki sayfayı çalıştırdığınızda tüm öğelerin ortalanmış ve kırmızı renkte olduğunu göreceksiniz.

Combinators (Bileştirici Seçiciler)

Bir CSS seçicisi, birden fazla basit seçici içerebilir. Basit seçiciler arasına bir birleştirici ekleyebiliriz.

CSS'de dört farklı birleştirici vardır:

- torun seçici seçici (boşluk)
- çocuk seçici (>)
- komşu kardeş seçici (+)
- genel kardeş seçici (~)

E F (Descendant combinator) (Torun Seçici - Boşluk)

CSS'deki torun seçici, birleştiricisi olmayan iki seçici arasında boşluk bulunan herhangi bir seçicidir.

İşte bazı örnekler:

```
ul li {  }
header h2  {  }
footer a  {  }
.bolum div {  }
#logo span  {  }
table td a  {  }
```

ul li { } örneğini alalım. **"Sırasız bir listenin soyundan gelen herhangi bir liste öğesi"** anlamına gelir.

Descendant, DOM ağacında kendi içinde yuvalanmış herhangi bir yer anlamına gelir. Doğrudan bir çocuk olabilir, beş seviye derinlikte bir torun olabilir, bu hala soyundan geldiği anlamına gelir. Bu seçim bir alt öğenin seçimini sağlayan çocuk seçicisinden(>) farklı olarak tüm alt öğlerde etkili olmasını sağlar.

Göstermek için, **div p{ }** eşleşecek:

```
<!DOCTYPE html>
<html>
<head>
    <style>
        div p{
            color:red;
        }
    </style>
</head>
<body>
    <p>Merhaba Dünya</p>
    <div>
        <p>Merhaba</p>
        <div>
            <p>Dünya</p>
        </div>
    </div>
</body>
</html>
```

A > B (Child combinator)(Çocuk Seçici)

"**>**" birleştirici (combinator) ile belirtilen elementin kapsadığı alt elemanlar (child) seçilir. Aşağıdaki örnek tanım `ol > li` ile `ul > li` için farklı nitelikler sağlayacaklardır.

```
li {
   color: blue;
}
ul > li {
   color: red;
}
```

A ~ B (General sibling combinator)(Kardeş Seçici)

"**~**" birleştirici (combinator) kardeş (sibling) elementleri seçer. İkinci elementin ilkini (hemen olmasa da) takip ettiği ve her ikisinin de aynı ebeveynleri (parent) paylaştığı anlamına gelir. Elementlerin yanı sıra `.classname` ve `#idname` ile birlikte de kullanılabilir. İkinci elementin her tekrarı aynı nitelikleri taşıyacaktır. `p ~ span` kullanımında aynı `parent` içerisinde olmak şartıyla `p` elementinden sonraki her `span` hedeflenir.

```
span {
   color: blue;
}
p {
   color: green;
}
p ~ span {
   color: red;
}
.featured-image ~ p {
   font-weight: bold;
}
```

A + B (Adjacent sibling combinator)

+ combinator bitişik kardeşleri (sibling) seçer. Bu, ikinci elementin doğrudan ilk elementi izlediği ve her ikisinin de aynı üst elementi (parent) paylaştığı anlamına gelir. `General sibling combinator` kullanımından farklı olarak, `adjacent sibling combinator` sadece ilk elementi takip eden kardeş elementi hedef alır. `li:first-of-type + li` kullanımında bir listedeki ikinci madde `blue` ile renklendirilir.

```
h2 + p {
```

```
   color: red;
}
li:first-of-type + li {
   color: blue;
}
```

Pseudo

: Pseudo classes (Sözde sınıflar)

: (pseudo class) bize belge ağacında (document tree) bulunmayan durum (state) bilgilerine göre elementleri seçme imkanı sağlar. Yaygın olarak bağlantıların ayrıştırılması sürecinde kullanılır. Örneğin, `a:visited` ile `a` elementinin kullanıcı tarafından ziyaret edildiği durum nitelendirilmektedir.

```
a {
   color: red;
}
a:visited {
   color: blue;
}
```

:: Pseudo elements (Sözde elementler)

:: (pseudo element) ile HTML'de bulunmayan varlıklar nitelendirilir. Örneğin, `p::first-line` ile bir paragrafın sadece ilk satırına nitelikler atayabiliriz.

```
p {
   color: red;
}
p::first-line {
   color: blue;
}
```

CSS3 ile birlikte kullanıma sunulan diğer sözde seçiciler ise şöyle:

:root	:nth-child()	:nth-last-child()
:nth-of-type()	:nth-last-of-type()	:last-child
:first-of-type	:last-of-type	:only-child
:only-of-type	:empty	:not
:target	:disabled	:enabled
:checked	:lang	