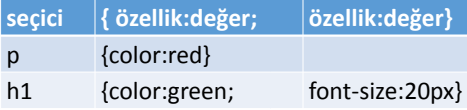
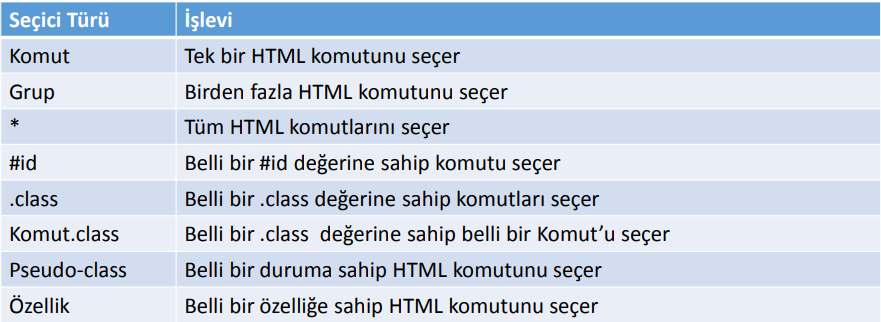
## Seçici İşlemeleri

* Biçimlendirilecek HTML komutlarını seçmeye yarar.
* CSS kodlarının en başına koyulur.
* Hangi HTML elemanları seçilirse CSS kodları onlara uygulanır. Tipik bir CSS komutunun yapısı aşağıdaki gibidir:



## Seçici Türleri

CSS komutundaki seçici biçimlendirme yapılacak HTML komutunun seçilmesini sağlar. Seçici türleri aşağıdaki gibidir:



## Etiket Seçici

* Tek bir HTML komutunun seçilmesini sağlar.
* Bir ya da daha fazla özellik kullanabilirsiniz.
* Özellikleri alt alta veya yan yana yazabilirsiniz.

## Tekil Seçici (#id)

* Belli bir #id değerine sahip HTML komutunu seçer.
* Tek bir tane etikete verilir.

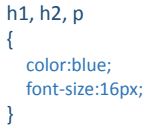
## Sınıf Seçici (.class)

* Belli bir .class değerine sahip HTML komutlarını seçer.
* CSS kodlarını birden fazla HTML komutuna uygulamak isterseniz sınıf seçiciyi kullanmalısınız.

## Evrensel Seçici

* Tüm HTML komutlarının seçilmesini sağlar.
* Yazılan kodlar sayfadaki tüm komutlara uygulanır.

## Grup Seçici

* Birden fazla HTML komutunun seçilmesini sağlar.
* Yazılan CSS kodları tüm HTML komutlarına uygulanır.

## Komut & Sınıf Seçici

* Belli bir class değerine sahip belli bir komutu seçer.
* Örneğin, class değeri .buyuk olan paragrafları seçmek için yandaki CSS kodu kullanılabilir:

## Bitişik Kardeş (A + B) Seçici

* (p + div) : p etiketinden sonra gelen ilk div elemanını seçer.

## Genel Kardeş (A ~ B) Seçici

* (p ~ div) : p etiketinden sonra gelen tüm div elemanlarını seçer.

## Nitelik [nitelik] Seçici

* Bu seçici, etikette nitelik olup olmadığına bakıyor. Olanları seçiyor. Örneğin title niteliği olan tüm etiketleri seçmek isteseydim;
  + abbr[title] : abbr etiketlerinde niteliği title olanı seçecektir.
  + [title] : niteliği title olan tüm etiketleri seçer.

## Nitelik [nitelik=”değer”] Seçici

* Bir önceki seçiciden farklı olarak, niteliğin değerine eşit etiketleri seçmek için kullanılır. Örneğin a etiketinde target niteliği \_blank olanları seçmek istersek;
  + a[target=”\_blank”]

## Nitelik [nitelik~=”değer”] Seçici

* Nitelik içinde belirtilen değer geçen etiketleri seçmek için kullanılır.
  + <div class="box box-lg">kutu 1</div>
  + <div class="box-lg box">kutu 2</div>
  + <div class="box-lg">kutu 3</div>
* Ve class niteliğinde box geçen etiketleri seçmek için şöyle yazabiliriz;
  + [class~="box"]

## Nitelik [nitelik|=”değer”] Seçici

* Nitelik belirtilen değerle başlayan etiketleri seçer. Ancak burada önemli olan, belirtilen değerin tüm kelimeyle eşleşiyor olmasıdır. Ya tüm kelimeyle ya da tire işareti ile ayrılmış kelimenin başlangıcı ile eşleşiyor olması gerekir. Örneğin html yapımız şöyle olsun;
  + <div class="box-sm">kutu 1</div>
  + <div class="box-lg">kutu 2</div>
  + <div class="boxlg">kutu 3</div>
* Ve box ile başlayan etiketleri seçecek olsaydık;
  + [class|="box"]
* Sonuç olarak en alttaki hariç diğerlerini seçmiş olacaktık. Çünkü en alttaki başlangıç olarak tüm kelimeyi baz aldığımızda box değil boxlg olduğu için. Ayrıca boşluk ile ayrılmış nitelik değerlerinde de bu seçici işe yaramaz. Yani başlangıcı seçtiğimiz değer bile olsa boşluk varsa o etiketi seçmeyecektir.

## Nitelik [nitelik^=”değer”] Seçici

* Yukarıdaki seçicinin aksine belirtilen değer ile başlayan tüm etiketleri seçmek için kullanılır. Boşluk tire işareti vs. hiçbirine bakmaz. Tek baktığı, niteliğin değeri belirlenen değerle başlıyor mu başlamıyor mu? Yine aynı yapıda bir html yapımız olsaydı;
  + <div class="box-sm">kutu 1</div>
  + <div class="box-lg">kutu 2</div>
  + <div class="boxlg">kutu 3</div>
* Ve şu şekilde seçme işlemi yapsaydık;
  + [class^="box"]
* Sonuç, hepsini seçecektir.

## Nitelik [nitelik$=”değer”] Seçici

* Bir üstteki seçicinin tam tersi olarak, niteliği belirtilen değerle biten tüm etiketleri seçmemizi sağlıyor. Örneğin html yapımız şöyle olsun;
  + <div class="box-lg box">kutu 1</div>
  + <div class="bbox">kutu 2</div>
  + <div class="lg-box">kutu 3</div>
  + <div class="kutu">kutu 4</div>
* Şimdi class niteliği box ile biten etiketleri seçelim.
  + [class$="box"]
* Sonuç ise, son div hariç hepsini seçecektir.

## Nitelik [nitelik\*=”değer”] Seçici

* Sonu ve başlangıcı belirlediğimiz değer ile bitenleri seçmeyi başardık. Bu seçici ile, niteliklerde belirlediğimiz değer geçen tüm etiketleri seçmek için kullanılıyor.
  + <div class="box">kutu 1</div>
  + <div class="tox">kutu 2</div>
  + <div class="hoxxx">kutu 3</div>
* Şimdi class niteliği içerisinde ox değeri geçen etiketleri seçelim.
  + [class\*="ox"]

## Sözde Öğeler (Pseudo-elements)

* Bir elemanın belli bir kısmını seçmeye yarayan seçicilerdir. Örneğin, bir tablonun ilk satırı, bir paragrafın ilk harfi veya tablonun tek satırları seçilebilir. Sıklıkla kullanılan kısmi seçiciler aşağıdaki gibidir:

### ::after Sözde Öğesi

* ::after sözde elemanı ile, seçilen etiketin sonrasına bir değer ekler ve stil işlemi uygulamamızı sağlar. Bu sözde eleman ile birlikte content özelliği kullanılır, content özelliği olmadan bu sözde elemanın bir etkisi olmaz.

Örnek verecek olursak;



### ::before Sözde Öğesi

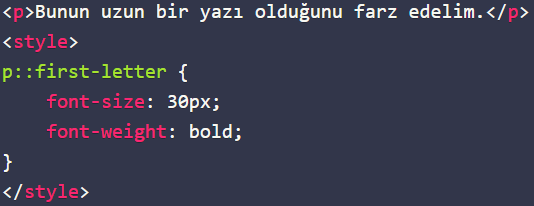
* ::before sözde elemanı ile, seçilen etiketin öncesine bir değer ekler ve stil işlemi uygulamamızı sağlar. Bu sözde eleman ile birlikte content özelliği kullanılır, content özelliği olmadan bu sözde elemanın bir etkisi olmaz.

Örnek verecek olursak;



### ::first-letter Sözde Öğesi

* Sıkça haber sitelerinde ilk harfin daha büyük boyutta ya da renkte olduğunu görürüz. İşte ilk harfi seçerek stil işlemi uygulamak için bu sözde elemanı kullanıyoruz.

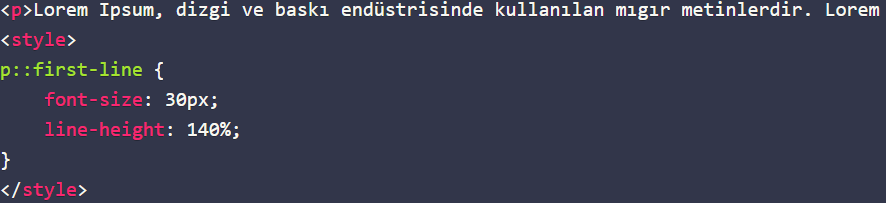


Sonuç;

**B**unun uzun bir yazı olduğunu farz edelim.

### ::first-line Sözde Öğesi

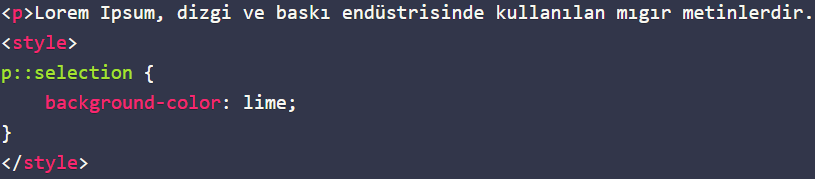
* Seçilen etikette ilk satırı seçerek işlem yapmak için bu sözde eleman kullanılır. İlk satır uygulanan stil işlemine göre farklılık gösterebilir, çünkü her zaman görünen ilk satır ne ise ona stil işlemi uygulanır. Bazen yazının boyutunu büyüttüğümüzde görünen ilk satır daha az kelimeden oluşabilir.



### ::selection Sözde Öğesi

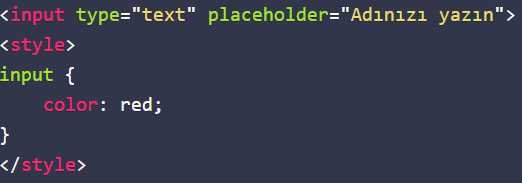
* Seçilen yazılara stil işlemi uygulamak için bu sözde eleman kullanılır. Genelde işletim sistemine göre farklılık gösteren bu seçme işleminin stillerini kendinize göre ayarlayabilirsiniz.

Örneğin aşağıdaki yazıyı seçtiğinizde varsayılan olarak ne ise o şekilde bir stil göreceksiniz. Muhtemelen windows'tan bakıyorsanız arkaplanı mavi yazı rengi beyaz olacak seçtiğiniz yerlerin.

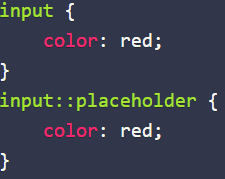


### ::placeholder Sözde Öğesi

* Form etiketlerinde placeholder niteliği bulunan etiketlere stil işlemi uygulamak için bu sözde eleman kullanılır. Şöyle ki;



* Bu örnekte “Adınızı yazın” değeri kırmızı olmayacaktır, ancak input'un içerisine bir şeyler yazarsanız bunun kırmızı renkte olduğunu görürsünüz. Yani placeholder değerine stil işlemi vermek için bu sözde öğeyi kullanabiliriz.



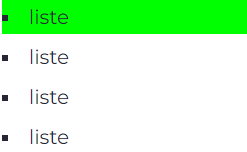
## Sözde Sınıflar (Pseudo-class)

### :first-child Sözde Sınıfı

* İlk çocuk etiketi bulmak için bu sözde sınıf kullanılır. Bu sözde sınıf kullanılan etiketin mutlaka bir ebeveyn etiketi olmak zorundadır. Örneğin;

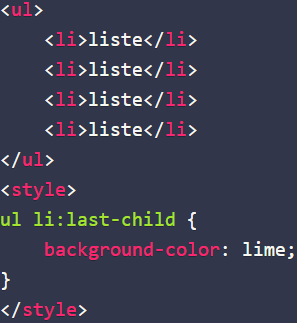


Bu kod ile ilk <li> etiketi seçilerek stil işlemi uygulanır. Çünkü li etiketinin ebeveyn etiketi <ul> etiketidir.

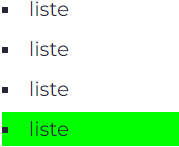


### :last-child Sözde Sınıfı

* Son çocuk etiketi bulmak için bu sözde sınıf kullanılır. Bu sözde sınıf kullanılan etiketin mutlaka bir ebeveyn etiketi olmak zorundadır. Örneğin;



Bu kod ile son <li> etiketi seçilerek stil işlemi uygulanır. Çünkü li etiketinin ebeveyn etiketi <ul> etiketidir.

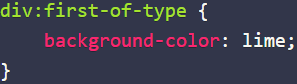


### :first-of-type Sözde Sınıfı

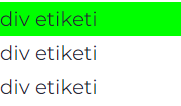
* :first-child sözde sınıfının yapamadığı şey, ilk eleman seçtiğimiz etiket değilse seçemiyordu. İşte bu durumda kullanılacak olan bu sözde sınıfıdır. Örnek vermek gerekirse;



Yukarıdaki CSS işlemi geçerli olmayacaktır çünkü ilk elemanımız div etiketi yerine p etiketidir. Fakat burada direk tipe bakarak işlem yaptırmış olsaydık, yani şöyle;



Sonuç başarılı olacaktı;



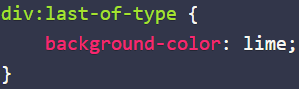
Yani burada önemli olan seçtiğimiz tipte bir etiketin olup olmadığı varsa onun ilk elemanını seçip stil işlemi uygulayabiliyoruz.

### :last-of-type Sözde Sınıfı

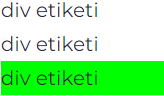
* :last-child sözde sınıfının yapamadığı şey, son eleman seçtiğimiz etiket değilse seçemiyordu. İşte bu durumda kullanılacak olan bu sözde sınıfıdır. Örnek vermek gerekirse;



Yukarıdaki CSS işlemi geçerli olmayacaktır çünkü son elemanımız div etiketi yerine p etiketidir. Fakat burada direk tipe bakarak işlem yaptırmış olsaydık, yani şöyle;



Sonuç başarılı olacaktı;



Yani burada önemli olan seçtiğimiz tipte bir etiketin olup olmadığı varsa onun ilk elemanını seçip stil işlemi uygulayabiliyoruz.