

## CSS Selectors

```
*{  
} bu evrensel seçici, yani her şeyi etkiler
```

```
div *{  
} div içindeki tüm etiketleri etkiler
```

```
h1, p, div, .paragraf, #test{  
} hepsini içerir, bu şekilde virgülle ayırabiliyoruz
```

```
div b{  
} div in içindeki b etiketini seçti
```

```
.box b{  
} .box in içindeki b etiketini seçti
```

```
p > b {  
} b nin ebeveyni p olacak, arada bir span vs varsa onu değiştirmez
```

```
h3 + p {  
} h3 ten sonra gelen ilk p yi seçti, diğerlerini seçmez
```

```
img[alt] {  
} img lerden alt niteliği olanı seçiyor
```

```
a[target="alt"] {  
} a lardan target ı "alt" olanı seçiyor.
```

```
img[alt=""] {  
} img lerden alt değeri boş olanı seçiyor
```

```
div[class="box"] {  
} divlerden class ı box olanları ve box- ile başlayanları seçer
```

```
div[class^="box"] {  
} box ile başlayanların tamamını seçer
```

```
div[class$="box"] {  
} divlerden classı box ile bitenlerin tümünü seçer
```

```
div[class*="top"]{  
} divlerden classının içinde top geçenleri seçer, Top seçmez, harfe duyarlı
```

```
div[class*="top" i]{  
} buradaki i flagı ile büyük küçük harf duyarlılığını kaldırıyoruz, artık Top u da seçer
```

## Sözde Öğeler

sözde sınıflar : tek 2 noktalı, sözde öğeler :: iki 2 noktalı

```
div::after {  
    content = "sözde öğe";  
    color = yellow;  
}
```

seçilen divden sonra bir şey ekler ve onu etkiler.

☆ after ı veya before u bir content ile kullanmamız gerekiyor: content boş bile olsa, yoksa kullanılmaz

```
.box::before{  
    content: "önceki yazı";  
}
```

.box classından önce gelir

attr() bir fonksiyon, herhangi bir nitelikteki değeri almak istiyorsak bunu kullanırız. Mesela yukarıdaki .box classından sonra title="bu baslık" diye bir şey var. before içine content: attr(title) yazarsak değeri otomatik olarak alır. ☆

::first-letter komutu, seçtiğimiz etiketin ilk harfini seçmeyi sağlar.

::first-line ilk satırı seçer

```
::selection {  
    background: yellow;  
}
```

kullanıcının seçtiği yeri (paragrafta bir bölüm gibi) seçip etkiler

```
::placeholder {  
}
```

text alanı boşken var olan yazı

## Sözde Sınıflar

```
div:first-child {  
}
```

ilk çocuk etiketi olan div i seçer

:last-child son çocuk etiketini seçer

```
div:first-of-type{  
}
```

tipi div olan ilk çocuk etiketi seçer

```
div:last-of-type{  
}
```

en son ki div çocuk etiketini seçer

## :nth-child

```
ul li:nth-child(3){
```

```
} 3. li yi seçer, çocuk etiketleri 1den başlar
```

```
ul li:nth-child(odd){
```

```
} li lerden tek sayıdakileri seçer
```

```
ul li:nth-child(even){
```

```
} çift sayıdan oluşan li leri seçer
```

```
ul li:nth-child(3n){
```

```
} her 3 elemanda 1 tanesini seçer
```

```
ul li:nth-child(3n+2){
```

```
} 2. elemandan başlayarak 3 elemanda 1 tanesini seçer (2. elemanı da seçiyor)
```

```
ul li:nth-child(n + 4){
```

```
} 4. elemandan itibaren tüm li leri seçer
```

```
ul li:nth-child(-n + 8){
```

```
} 8. eleman ve öncesini seçer
```

```
ul li:nth-child(n+4):nth-child(-n+8){
```

```
} 4 - 8 arasındakileri seçer, 4 ve 8 dahil
```

```
ul li:nth-child(n+4):nth-child(even):nth-child(-n+8){
```

```
} 4 - 8 arasındaki çiftleri seçer
```

:nth-of-type(3) araya hangi etiket girerse girsin 3.yü seçer

:nth-last-child(3) sondan 3.yü seçer

:hover tarayıcıda -seçilen- etiketin üzerine gelince ne olacağını belirler

```
button:hover + ul {
```

```
display: none;
```

```
} butonun üzerine gelince butondan sonra gelen ul etiketini gizler
```

```
<input type="checkbox">
```

```
<input type="radio">
```

:checked inputlardan tipi checkbox veya radio olanlar seçilince işlem yapıyor

```
<label>  
  <input type="radio" name="cinsiyet" value="kadin">  
  <span>Kadın</span>  
</label>
```

bunu label içine yazdık ki sadece radio butonuna tıklanınca değil kadın yazısına tıklayınca da çalışsın ☆

```
input[type="radio"]:checked + span{  
  color: red;
```

} tipi radio olan inputlar check olduğunda ondan sonra gelen span in yazısının rengini kırmızı yap

```
<input type="checkbox" id="cb" class="checkbox">
```

```
<label for="cb">Button</label>
```

burada Button a tıklanınca checkbox tik oluyo ☆

:focus bir öğeyi tıklayınca olacak işlemi belirliyor

tabindex kullanıcı sayfada tab a basınca hangi sırayla elementlerin seçileceğini belirtir, form etiketlerinde kullanıcı deneyimini artırmak için kullanılıyor mesela

:focus-within focus işlemi yaptığımız etiketin ebeveynine stil işlemi uygulamak için kullanılırız

:only-child ebeveynin içinde bir tane çocuk etiketi varsa onu seçer

:not belirlediğimiz değere sahip olmayan etiketleri seçer

:empty içi boş etiketleri seçer

:valid geçerli konumdaki değerler için

:invalid geçersiz değerler için

:enabled disabled niteliği olmayanları seçmek için

:disabled

:optional required niteliği olmayanları seçmek için

:required

:read-only sadece okuma işlemi olanları seçmek için

:read-write sadece yazma işlemi olanları seçmek için

:active o an seçilmiş olan linki seçiyor

:link henüz ziyaret edilmemiş linkleri seçer

:visited ziyaret edilmiş linkleri seçiyor

:any-link geçerli a etiketleri olanları seçiyor

:target hedeflendiğimiz etiketi değiştirmeye yarar

:in-range , :out-of-range bu iki sözde sınıfı tipi number olan input umuza bir aralık yani range verdiğimizde kullanabiliyoruz

:root etiketlerdeki kök elemanı seçmek için kullanılıyor, html in üzerindedir yani root>html

div block seviyesinden elemandır, block seviyesinden elemanlar otomatik olarak sayfanın genişliği kadar uzun olur (100vw) ve ondan sonra gelen eleman direkt bunun altından başlar. ☆

satır içi seviyesinden elemanlar içeriğindeki eleman kadardır. mesela span in-line dir.

min-content, max-content block seviyesinden elemanların içeriği ne kadarsa onu etkiliyor

## padding (iç boşluk)

etiketin içine verilen, etikete verilen arka plan rengi boşluğu da etkiler mesela

padding-top, padding-bottom, padding-right, padding-left bunların birleştirilmiş hali: padding

padding: 100px dersek her kenara 100px boşluk verir, özellikler saat yönünde sıralanıyor

padding: 20 px 50 px; yukarı ve aşağı boşluğu 20px, sağ ve sol boşluğu 50px olur

padding: 10px 20px 50px; yukarıdan 10px, sağ ve sol 20px, aşağısı 50px olur.

Satır içi seviyede elemanlarda sağ ve sol boşluk düzgün çalışırken yukarıdan ve aşağıdan taşma oluyor

## margin(dış boşluk)

etiketin dışına etki eden boşluk.

margin-top, margin -bottom, margin -right, margin -left, margin

margin-top: -100px; yukarı doğru çeker

margin-left: auto; sağa yaslar, auto ortalamak için

outline (dış çizgi) : etiketin dışında bir çizgi çiziyor

outline-offset: dış çizgi ile kenarlık arasındaki mesafeyi ayarlamak için kullanılıyor

box-sizing: border-box; box a verilen padding, border vs kutunun genişliği ve yüksekliği sabit kalacak şekilde eklenir

border-radius: köşeleri yuvarlar

border-top-left-radius: sol üst köşeyi yuvarlar

border-top-right-radius:

border-bottom-left-radius:

border-bottom-right-radius:

rgb() - red green blue yani kırmızı yeşil mavi tonuna göre renk ayarlıyor. hsl() ise hue-saturation-lightness yani ton-doygunluk-hafiflik değerlerine göre renk tanımlamak için kullanılıyor

`<input type="color">` bunu tıklayınca renklerin rgb değerlerine bakabiliriz

`font-family: Verdana, Geneva, Tahoma ...` birkaç font belirlemek ilk font alfabeyi desteklemiyorsa ikinci fonta geçmesi açısından önemli

`em`: bu ölçü birimi ebeveyninin ölçü birimine göre ayarlanıyor, mesela 1em ebeveyninin ölçü birimine eşit, 2em iki katı olur

`rem`: html etiketinin font-size ına bakar mesela

`line-height: 150%`; satır yüksekliğini belirler

`font: 30px/1.7 Arial, sans-serif`; yazı boyutu 30px, satır yüksekliği 1.7

`letter-spacing`: harfler arasındaki mesafeyi ayarlamak için kullanılır

`word-spacing`: kelimeler arasındaki mesafeyi ayarlamak için kullanılır

`text-align`: yatay hizalama için

`text-align-last`: son satırı hizalamak için

`text-decoration-line: underline`; yazının altına çizgi çizer

`text-decoration-line: overline`; yazının üstüne çizgi çizer

`text-indent`: metin girintisini ayarlar

`white-space`: boşlukları yönetmek için kullanılır

`white-space: nowrap`; alt satıra inmeden her şeyi yazar

`white-space: pre`; yazdığımız her şeyi boşluklarıyla olduğu gibi korur

`white-space: pre-wrap`; aynı düzeni korur ama taşıdığı zaman alt satırdan devam eder

`text-transform`: yazıların her harfi büyük mü küçük mü vs olacağını belirliyoruz

`text-shadow`: metinlere gölge vermek için kullanılır

`background-image: url(foto.png)`; arka plana görüntü ekler

`background-repeat`: tekrar edip etmemesi ile ilgili, `no-repeat` dersek tekrarlamaz, `repeat-x` sadece yatay olarak tekrarlar, `repeat-y` sadece dikeyde tekrarlar, `space` ise görseli kırpmadan sığdırmaya çalışır.

`background-position`: arka plan görselinin konumu

`background-clip`: arka planın nereleri kapsadığını belirliyoruz

`background-image: url(foto.png);`

`background-clip: text;`

`color: transparent`; diyelim ki YAZILIM diye bir başlığımız var. Harflerin içini dolduracak şekilde foto.png görseli başlıktaki harflere eklenir.

**background-origin;** nereden başlayacağı

**background-size: contain;** görselin yüksekliğini tam sığdırır

**background-size: cover;** görseli tamamen sığdırır genişlik olarak da

**background-attachment;** ekranı kaydırırken arka plan görselinin nasıl hareket edeceğini belirler

**background-attachment: fixed;** kaydırırken arka plan sabit kalır

**background:** tüm background özellikleri tek satırda yazılabilir

## Gradient (renk geçişleri)

**background-image:linear-gradient(blue, pink);** maviden pembeye yukarıdan aşağı bir renk geçişi yapar. Default olarak yukarıdan aşağı ama **(to right, blue, pink)** dersek soldan sağa doğru gider mesela. **to left, to top** vs kullanılabilir. **to top right:** sağ üst köşeye doğru gider, **to bottom left** sol alt köşeye doğru iner. İki'den fazla renk de kullanılabilir.

**(blue 33%, White 33% 66%, red 66%)** dersek maviyi %33'e kadar götürür, beyazı 33-66 arası kadar ve kırmızıyı da 66'dan 100 kadar doldurur

**radial-gradient(...)** merkezi bir noktadan renk yayılması yapıyor, bunda **at center,** at top, at bottom, at left, at right kullanılabilir

**conic-gradient()** merkezden yayılıyor koni şeklinde

bunların başına **repeating** ekleyerek tekrarlanması sağlanabiliyor: **repeating-linear-gradient()**

**display** ile css'de tüm etiketlerin görünüm değerini değiştirebiliyoruz, mesela **span** ı **block** yapabiliriz

**inline:** satır içi, **inline-block:** hem satır içi hem blok yani **block** özelliklerini alıyor ama **inline** seviyesinden olmaya devam ediyor

**display:none;** sayfadan gizler, o alana başka şeyleri getirir

**display:flex;** görünüm esnek oluyor, elemanlar yan yana geliyor

**display: grid;** elemanlar alt alta geliyor

**visibility: hidden;** eleman görünmez ama kapsadığı alan boş kalır, **visible** dersek görünür

**border-collapse: collapse;** tablo hücrelerinin hepsini birleştirir

## Position

Etiket konumunu belirliyoruz

**position: absolute;** etiket sayfada yer kaplamıyormuş gibi kabul ediliyor yani ondan sonra yazılan şey onun altında kalıyor, bu etiket ön planda oluyor

**position: relative;** etiketleri bir üst ebeveynin içinde (görünüm olarak) kısıtlamak için de kullanılır

**z-index :** etiketlerin hangisinin ön planda olacağını bu değerle belirleyebiliyoruz

`position: fixed;` etiket sayfanın bir bölümünde sabit kalıyor

`position: sticky;` top değerini belirleyerek kullanabiliriz, top=0 ile

`column-count: 3;` yazıyı kaç sütuna böleceğimizi belirleyebiliyoruz

`column-width: 300px;` genişliğe göre böler

`columns: 5 250px;` 5 tane gözüktür 250px in altına düşmez

`column-rule-style:` burada sütun aralıklarına stil verilebilir

`column-rule-width:`

`column-rule-color:`

`column-gap: 50px;` sütunların arasını açabiliriz

`column-span: all;` başlık sütun ilişkisi

`filter: blur(1px);` bulanıklaştırır

`filter: brightness(1);` parlaklık ayarı

`filter: contrast(1.2);`

`filter: grayscale(.3);` grileştiriyor

`filter: hue-rotate(180deg);` renk dönüşümü

`filter: invert(1);` görüntüyü renk olarak tersine çeviriyor

`filter: opacity(.1);` opaklık ayarı

`filter: saturat();` doygunluk

`sepia()`

`filter: drop-shadow(0 0 10px red);` iç dolgulu gölgelendirme, (yatay dikey bulanıklık renk)

## Transform

`transform: rotateX(45deg);` x ekseninde 45 derece döndürür

`transform: rotateY(60deg);` y ekseninde 60 derece döndürür

`transform: rotate(50deg);` kendi içinde döndürme işlemi yapıyor

`transform: rotateY(-180deg);` arkaya döndürür

`transform: skewX(5deg);` öğeyi eğ

`transform: scale(1.1);` scale ölçeklendirme için kullanılıyor, bu örnekte öğemiz büyür

`transform-origin:` dönmenin nereden olacağını belirtiyor, varsayılan olarak ortadandır



`transform-origin: top-left;` dersek sol üstten döndürür

## Adem İter #CSSanlat

CSSde bir elemente verdiğimiz `top-right-bottom-left` değerini tarayıcının algılaması için etikete mutlaka bir position değeri vermemiz gerek. ☆

`position: relative;` ile konumlandığımız bir öğenin yerini `top-left` vs ile değiştirsek bile tarayıcı bize yaptığımız değişikliği gösterir ama aslında o öğe hayali olarak yine o yerinde kabul edilir, bu öğenin asıl kapladığı yer boş kalır yine. ☆

`position: absolute;` serbest demek, kapsayıcısının 0 noktasına göre hizalanması demek, kapsayıcısının `position: relative` özelliği yoksa onu kapsayıcı olarak görmez ☆

bir şeyin 0 noktası : sol üst köşesidir.

`position: fixed;` verdiğimiz elementin genişliği-yüksekliği olmalı, yoksa tarayıcıda görünmez ☆

`position: static;` özelliğini verdiğimiz an position değerlerini sıfırlar

CSSde elementler yukarıdan aşağı doğru sıralanır ama bunu z-index ile değiştirebiliriz, z-index i büyük olan değer ön planda görülür, z-index in çalışması için mutlaka position özelliğine `absolute-relative-fixed` değerlerinden biri verilmeli ☆

user agent stylesheet: tarayıcıların varsayılan olarak gelen css özellikleri

normalize.css: tarayıcıların varsayılan özelliklerini standarda çekmek için kullanılır

`block`: alt satıra geçiyorsa

`inline`: alt satıra geçmiyorsa

`display: initial;` elementin özelliklerini varsayılan haline getirir

`display: inherit;` kapsayıcısının `display` özelliği gibi davranır

`display: inline-block;` hem `block` hem `inline` seviyesinden element gibi davranır, hem `block` elementlerinin özelliklerini alır hem de alt satıra geçmez

htmldeki table cssde `display: table;` ile yapılabilir, tr `table-row` ile ve td `table-cell` ile yapılabilir.

## Ai - Flexbox

`display: flex;` elementleri yan yana getirir, boyutunu büyütsek bile 2. bir satıra taşmaz, bunun için `flex-wrap: wrap;` dersek taşıdığı yerde alt satıra geçer. nowrap satır atlamaz

`flex-wrap: wrap-reverse;` sarmalamayı tersten başlatır

`flex-direction:` hangi doğrultuda sıralanacağını belirtir

`flex-direction: row;` varsayılan da budur, satırda sıralanır soldan sağa

`flex-direction: row-reverse;` sağdan sola sıralar

`flex-direction:` x-y koordinatlarının yerini değiştiriyor ve ona göre verdiğimiz diğer özelliklerin de yönü-yeri vs değişiyor ☆

`height: 100vh;` vh viewport height demek yani tarayıcıda gösterilen alanın yüksekliği kadar

`justify-content: space-evenly;` yatayda hizalayıp aralara eşit boşluk koyar

`align-items:` dikeyde hizalamak için

`align-content:` birden fazla satırı dikeyde hizalamak için

`align-content` atadığımız anda `align-items` etkinliğini kaybeder, atadığımız şey etki etmez

kapsayıcıya (container) verilen özellikler: `flex-direction`, `flex-wrap`, `justify-content`, `align-items`, `align-content`

içerideki alt elemanlara verilen özellikler: `align-self`, `flex-grow`, `flex-shrink`, `order`

`flex-grow:` ekranda boşluk kaldığında `flex-grow: 1;` verdiğimiz element genişlik olarak yayılıp boşluğu kapatabilir mesela.

`flex-shrink:` `flex-grow` un tam tersi, elementin sıkıştırılmasını sağlıyor

`order:` sıralamayı ayarlamayı sağlıyor, -1 verdiğimiz element en önce geliyor, varsayılan olarak tüm elementlerin değeri 0, elementlerin `order` değerini kendi aralarında değiştirip sıralamasını da değiştirebiliriz, 2 5 ten önce gelir mesela

`flex-shrink: 0;` dersek o element asla sıkışmaz

## FLEXBOX

`display: flex;` özelliğini parent a veririz ki, içindeki child lar yan yana gelsin

`flex: 1;` 1 birim genişliğinde, `flex: 2;` 2 birim genişliğinde olur, öncekinin iki katı

## Ai - HTML CSS Öğreniyoruz (SCSS)

`class="header main"` bu şekilde class a iki isim atanabilir

`normalize.css` diğer tüm dosyaların üstünde eklenir

`pointer-events` verdiğimiz elemanı Mouse hiçbir şekilde tıklayamıyor

mobilden giren kullanıcılar yaklaşık %80-90lık oranla daha fazla olduğu için her zaman ilk kodumuzu mobile göre yazmalıyız, `mobile-first` denen şey tam olarak bu

aynı site için desktopta ve mobilde farklı menü yaparsak seo açısından iyi olmaz

## SASS Dersleri

Şekilde border-radius CSS3 ile geldiği için tarayıcı uyumsuzluklarını önlemek amacıyla prefix kodlaması içinde kullanmamız gerekir, bunun için mixin oluşturalım, mixin fonksiyonlar gibi parametre alır

Sass ta elementleri ayırmak için yorum satırı olarak // kullanılır

Cebirsel işlemler de yapılabilir: + - / \*, bu css'te calc fonksiyonu ile yapılıyor ama sass ta buna gerek yok

```
@for $i from 1 through 5 {
```

```
  .item-#{ $i } {
```

```
    // Buraya css özellikleri gelir
```

```
  }
```

```
}
```

## z-index

z-index değeri ne kadar büyükse o kadar öndedir

z index in çalışması için position değeri olmalı (fehmi uyar relative dedi, ai herhangi bir değer olabilir demişti)

**z-index: -1;** sayfanın en arkasındadır, sayfadaki her şey bunun önünde

**position: absolute;** sayfadaki herhangi bir nesneyi bu şekilde sayfadan bağımsız hale getirebiliriz

**position: static;** default olan pozisyonudur, yazmaya bile gerek yok