Kurs Linki: [Ücretsiz Eğitimi - 1 Saatte CSS Temellerini öğrenin ! | Udemy](https://www.udemy.com/course/bir-saatte-css-temellerini-ogrenin-web-css3/)

. ile class tanımlıyoruz. classları 1 den fazla etiket için kullanabiliriz.

Örneğin;

.mavi{

color: blue;

}

<p class="mavi">Bu yazı mavi yazılacak!</p>

# ile id tanımlaması yapıyoruz. id leri sadece 1 etiket için kullanabiliriz.

#yazi{

color: blue;

}

<p id="yazi">Bu yazı mavi yazılacak!</p>

: ile hover falan ekleyebiliyoruz. buna pseudo sınıf denir. Hover dışında, active (tıklanıldığında), focus (odaklandığında), visited (zaten ziyaret edilmiş linkler) gibi pseudo sınıfları da bulunur.

a:hover{

color: yellow;

}

<a href="www.google.com">Bu linkin üstüne imleç getirilince yazı rengi sarı olacak!</a>

bir div elementinin altındaki değerlerin stillerini değiştirmek için şu şekilde yazmalıyız.

div h5 #blue{

color: blue;

}

Yukarıdaki ifadenin anlamı div elementinin içindeki h5 elemenlerinin içindeki etiketlerde, id'si blue olanın rengini mavi yap.

<div>

<h5>Seçenek

<strong id="1"></strong>

</h5>

</div>

div h5 #blue{

color: blue;

}

#blue{

color: yellow;

}

<div>

<h5>Seçenek

<strong id="1"></strong>

</h5>

</div>

Yukarıdaki kodda yazı rengi mavi olur. Css in açılımdaki cascade kademeli demektir. Bu yüzden ne kadar detaylı yazarsak öncelik oranı yükselir.

Eğer aynı kod bloğu içerisinde aynı özelliğin değerini değiştirirsek örneğin;

h5{

color: red;

color: blue;

}

aşağıdaki özellik üsttekini ezecektir. Bunun sebebi spesificity(özgüllük) değeridir.

h5{

color: red;

}

div h5{

color: purple;

}

.header{

color: yellow;

}

#header{

color:

}

<div>

<h5 class="header" id="header">Merhaba</h5>

</div>

Yukarıdaki kodlarda özgüllük değerlerine göre en düşükten en yükseğe doğru sıraladık. Yazının rengi aqua olacaktır. Çünkü en detaylısı ve spesificity değeri en yüksek olan o.

h5{

color: red !important;

}

div h5{

color: purple;

}

.header{

color: yellow;

}

#header{

color:

}

<div>

<h5 class="header" id="header">Merhaba</h5>

</div>

Eğer bütün stilleri ezmesini istiyorsak "!important" kullanmalıyız. Bu durumda yazı tipi red olacaktır.

"text-align: justify;" özelliği, metindeki kelimelerin arasına farklı boyutlarda boşluk koyar. Bu özellik ile metinler daha düzgün gözükür. Diğer özellikler(left, right vb.) her kelimenin arasına aynı boşluğu koyar.

"font-family: 'Courier New', Courier, monospace;" özelliği yazı tipini belirlememize yarar. Burada 3 tane yazı tipinin olmasının sebebi birisi çalışmazsa diğerine geçer. İlk yazı tipinin tırnaklar içinde yazılmasının sebebi ise 2 kelimeden oluşması.

Eğer farklı bir yazı tipi eklemek istersek şu adımları uygulamalıyız;

1) Google Fonts sitesine git

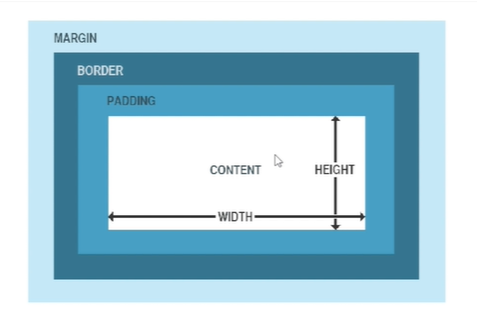
2) Beğendiğin bir yazı tipi seç

3) Ekranda çıkan "Select Regular" a bas

4) Verilen linki kopyala

5) HTML dosyasındaki head kısmının altına yapıştır

6) Linki aldığımız yerin altındaki yazı tipini kullanma kodunu kopyala ve istediğin yerde kullan

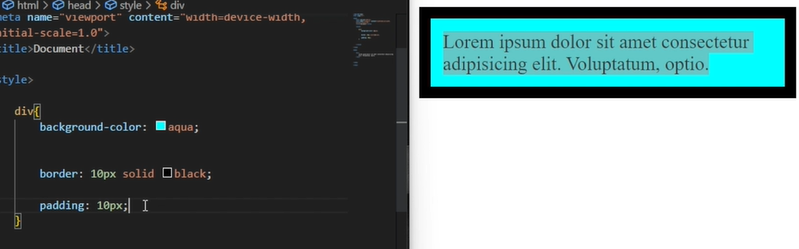


Css deki kutu modeli.

metin, yazılım, ekran görüntüsü, multimedya yazılımı içeren bir resim

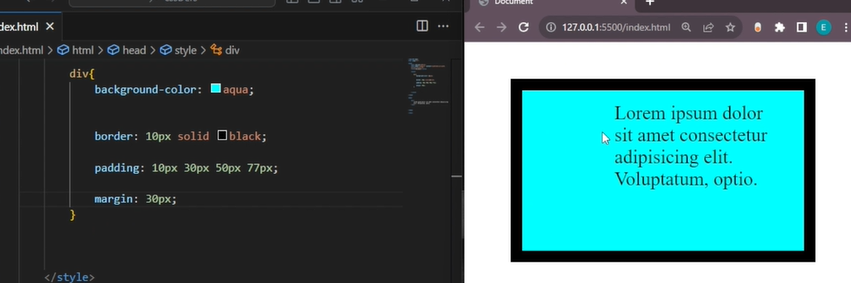
Açıklama otomatik olarak oluşturuldu

“border” özelliği kenarlık eklememize yarar.

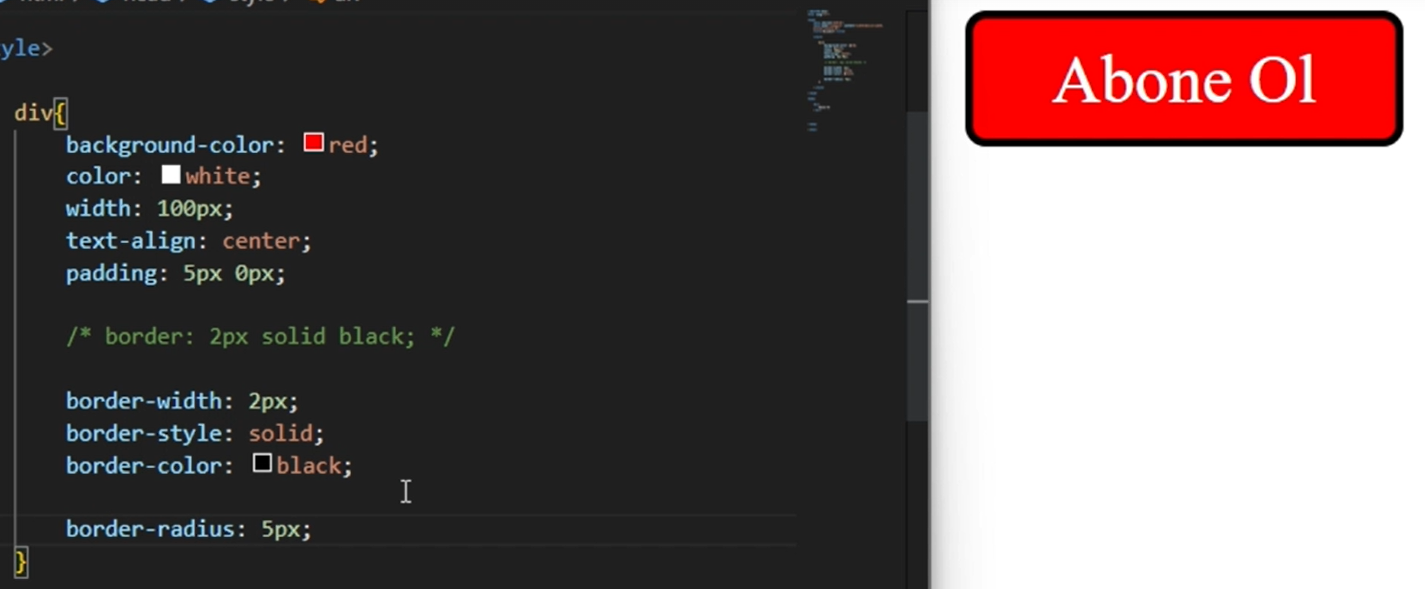


“padding” iç boşluk vermemize yarar. Bu özellik, metnin her tarafından boşluk bırakır. Eğer sadece 1 taraftan boşluk bırakılmasını istiyorsak “padding-left” veya “padding-top” gibi özellikleri kullanabiliriz.

Eğer “padding: 10px 30px;” dersek üstten ve alttan 10px, sağdan ve soldan 30px boşluk bırakır. Bunu 4 taraf içinde kullanabiliriz. “padding 10px 30px 50px 77px” üstten 10px, sağdan 30px, alttan 50px ve soldan 77px boşluk bırakır.



“margin” özelliği ise border’ın dışından boşluk bırakır.



“border-radius” köşeleri yuvarlar.

metin, ekran görüntüsü, yazılım, işletim sistemi içeren bir resim

Açıklama otomatik olarak oluşturuldu

“box-shadow” özelliği ile gölge ekleyebiliyoruz. Buradaki ilk 4px dikeydeki gölgenin, ikinci 4px ise yataydaki gölgenin genişliğini belirtir.

metin, ekran görüntüsü, yazılım, işletim sistemi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sonraki 4px ise blur eklemek için kullanılır.

Getcssscan.com sitesinden css gölgelendirme örnekleri alabiliriz.

metin, diyagram, çizgi, yazı tipi içeren bir resim

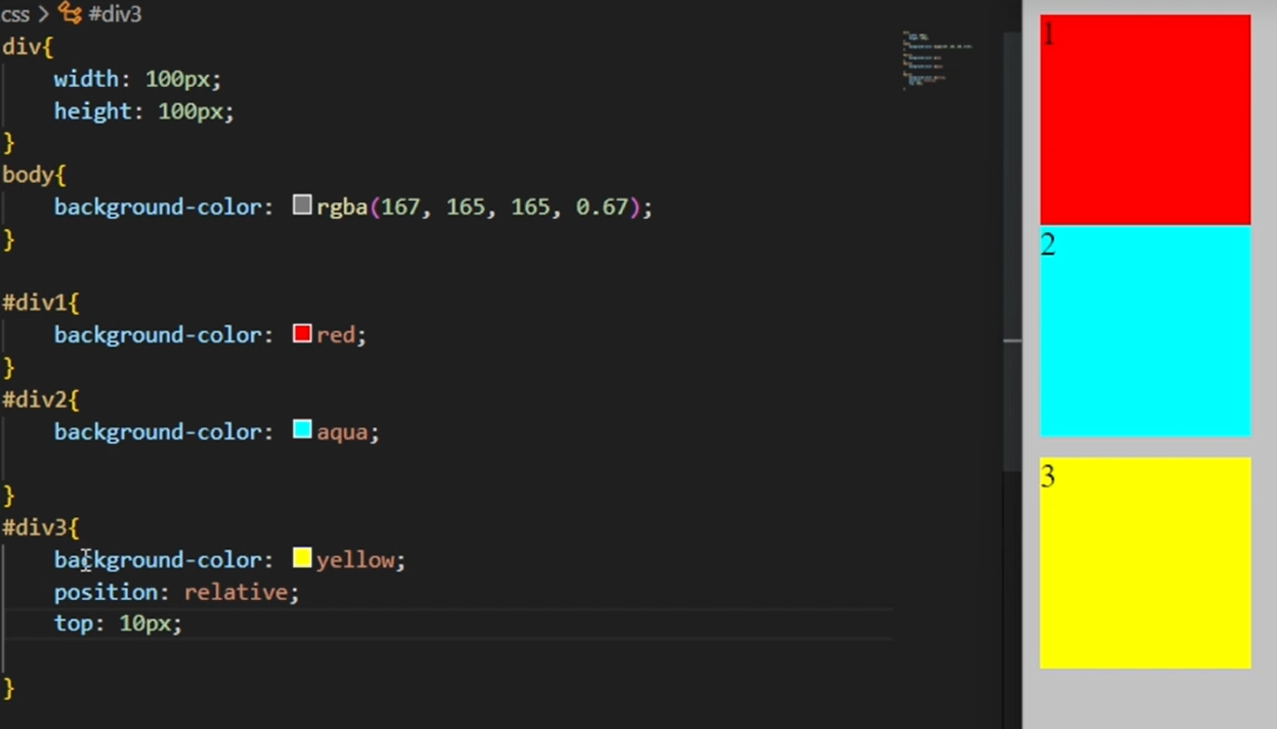
Açıklama otomatik olarak oluşturuldu

Css yerleştirme.

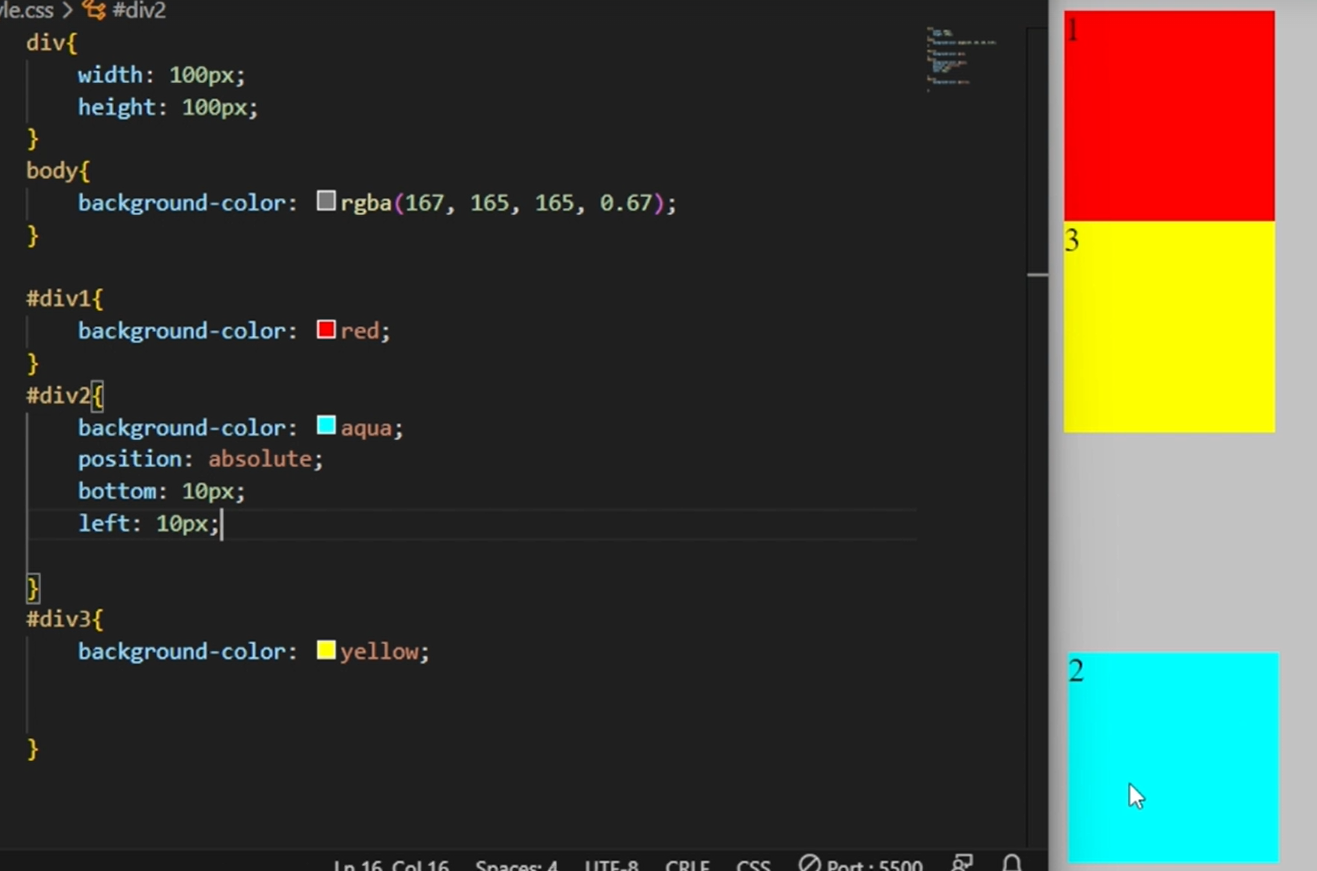
metin, ekran görüntüsü, yazılım, işletim sistemi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yukarıdaki örnekte “position: static;” olarak ayarlanmış. “top 10px;“ yazılmasına rağmen herhangi bir değişiklik olmadı.



“position: relative;” yapılmış ve “top: 10px;” kullanılınca sarı kutu aşağıya 10px indi.

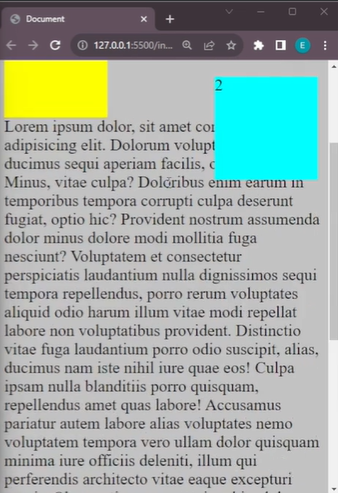


“position: absolute;” yaptığımız için 3 gitti 1 in altına yerleşti. 2’nin aşağıda olmasının sebebi girdiğimiz bottom ve left özellikleridir.

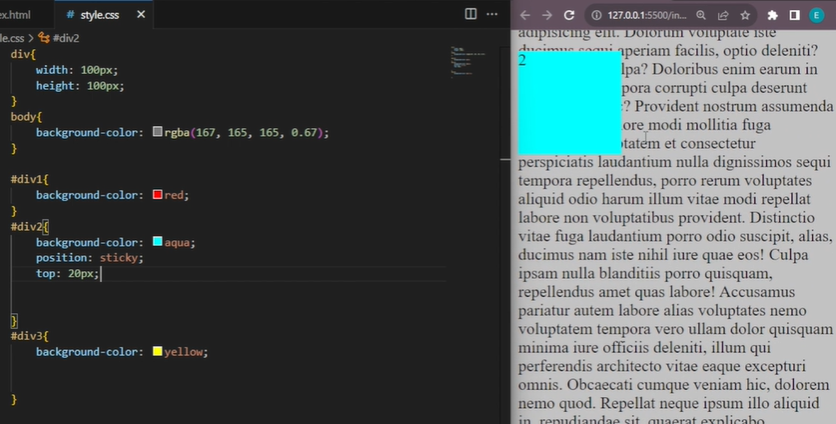
metin, ekran görüntüsü, yazılım, işletim sistemi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Eğer top deseydik gidip 1’in üstüne çıkacaktı.



“position: fixed;” özelliği sayfayı kaydırırken bizle gelmesini sağlar.



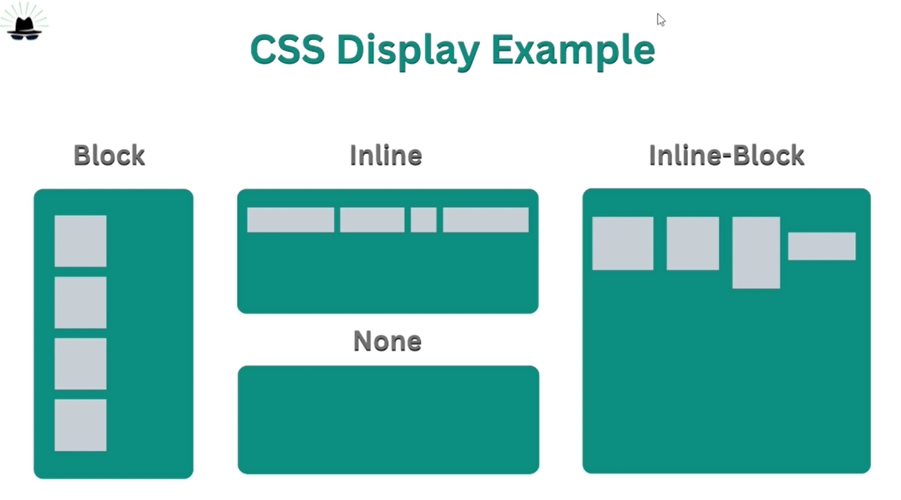
“position: sticky;” ekranı kaydırırken girilen özelliklere göre ekrana yapışır. Mesela burada “top: 10px;” kullanmışız. Yukarıda 10px kalınca ekrana yapışacak.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

“z-index” özelliği ekrandaki nesnelerin hangisinin üstte hangisinin altta olacağını belirler. Bu kodda z-index olmasaydı 2 üstte olacaktı. Çünkü en son o yazıldı.

“float: right;” ekrandaki nesneyi sağa kaydırmamıza yarar.

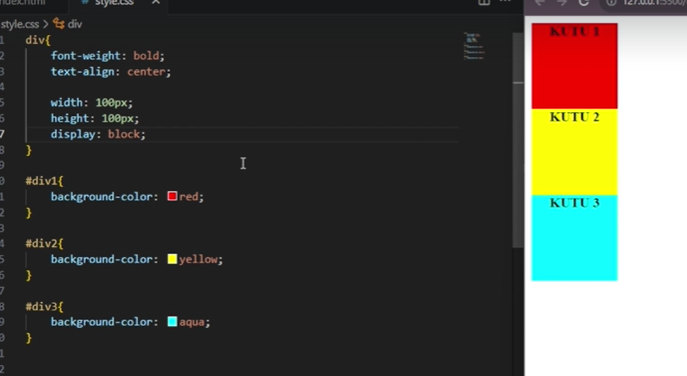


Block elementler tüm satırı kapsar.

Inline elementler satırı kapsamaz. Sadece kendi kapsadığı alan kadar yer tutar.

None direk elemanın görünürlüğünü yok eder.

Inline-Block ise elementlere genişlik ve yükseklik verebilmemizi sağlar. Inline’da bu özellik yok. Geniş ve yükseklik veremeyiz. Vermek istersek Inline-Block kullanmalıyız.

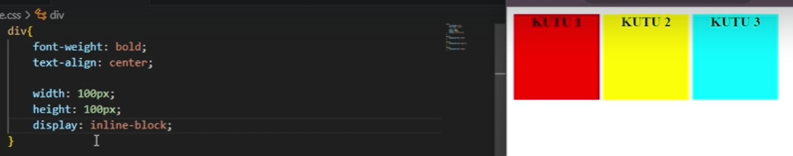


Genişlik ve yükseklik çalışıyor.

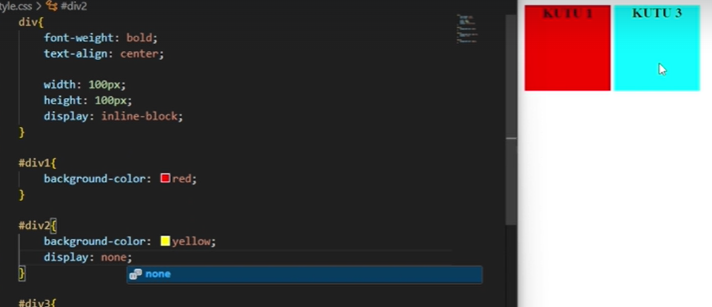
metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

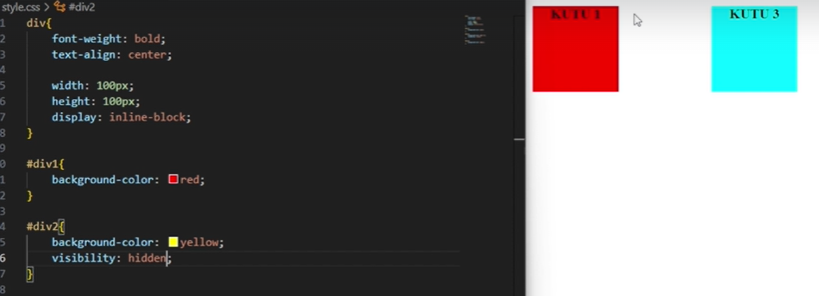
Genişlik ve yükseklik çalışmıyor.



Hem sıralı hem de genişlik ve yükseklik çalışıyor.



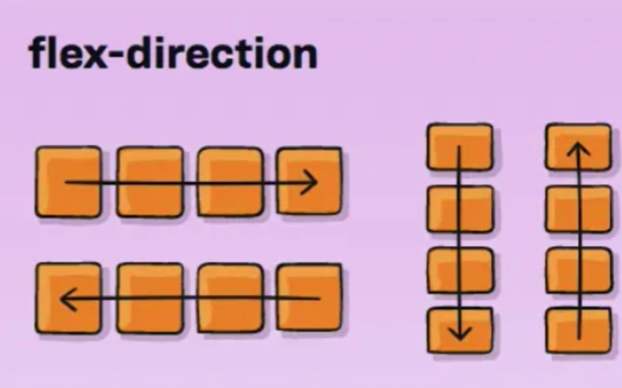
“visibility: none;” elementi yok sayar.



“visibility: hidden;” elementi gizler.



Buradaki “not”, div’in altındaki etiketlerde class’ı container olan etiket hariç diğerlerine bu özelliklerin uygulanmasını sağlar. Yani bu özellikleri, not kullanarak class’ı container olmayanlara uyguladık.

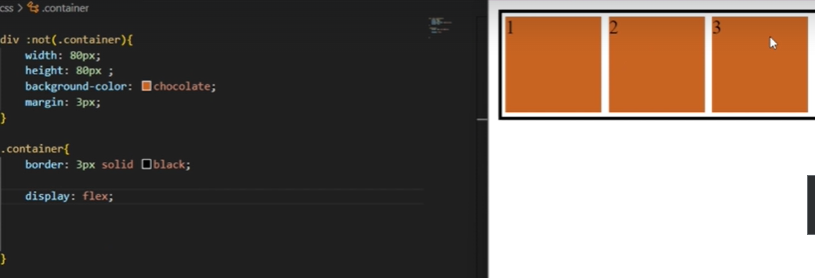


Css flex directions.

metin, yazılım, multimedya yazılımı, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

İlk başta böyleydi.



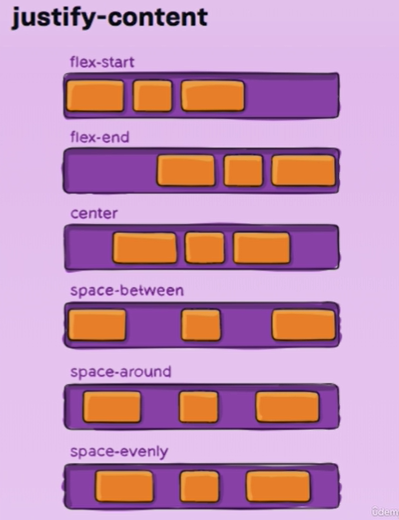
“display: flex;” dersek esnek bir yapı olur.

“display: row;” satır bazında “display: column;” ise sütun bazında hizalar.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

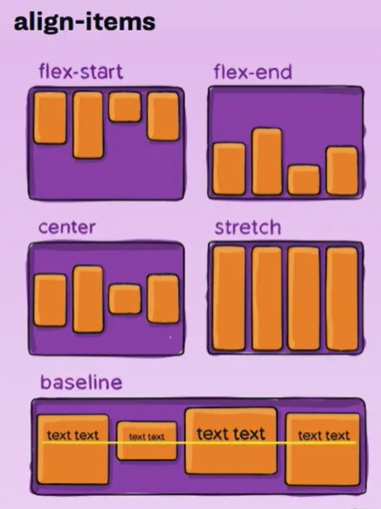
Açıklama otomatik olarak oluşturuldu

“flex-direction: column-reverse;” sütun bazındaki hizalamayı tersine çevirir.



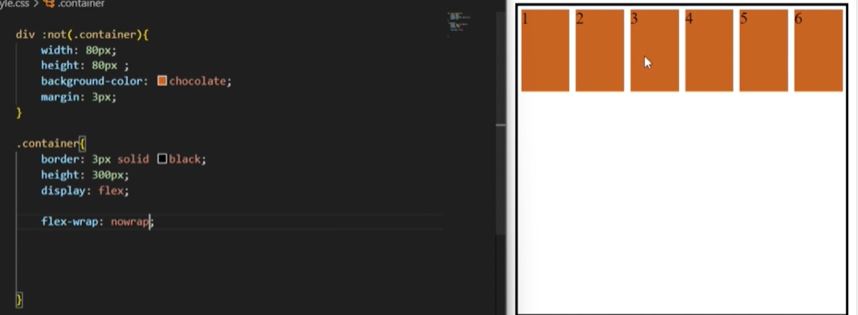
Css justify concept.

Elementleri hizalamamıza yarar.



Css align items.

Align items ise justify’nin dikeydeki versiyonu.



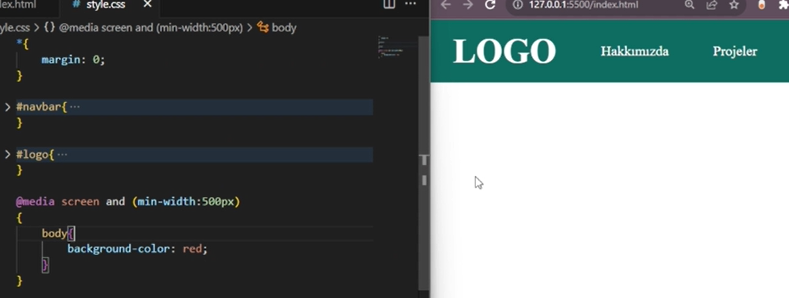
“flex-wrap: nowrap;” elementleri tek satıra hizalamaya çalışır.

metin, ekran görüntüsü, yazılım, tasarım içeren bir resim

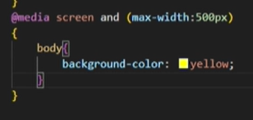
Açıklama otomatik olarak oluşturuldu

“flex-wrap: wrap;” ise elementleri, boyutlarını bozmadan hizalar. Sayfayı büyütüp küçülttükçe satırdaki nesne sayısı değişebilir.

Responsive tasarım, bütün cihazlar için web sitesinin görünümünün uyumlu olması anlamına gelir. Yani telefondan giren birisinin veya tabletten giren birisinin ekranında görüntünün bozulmaması gerekir.



Medya sorguları ile cihazların ve ekranların özelliklerini dikkate alarak belirli CSS kurallarını uygulamak için kullanılır. Mesela yukarıdaki örnekte minimum ekran boyutu 500px olan cihazlarda body’nin arkaplan rengi kırmızı olacak.



Yukarıdaki örnekte maximum ekran boyutu 500px olan cihazlarda body’nin arkaplan rengi sarı olacak.



Yukarıdaki örnekte ekran boyutu 600px den fazla olanlarda arkaplan rengi kırmızı, ekran boyutu 500px den az olanlarda arkaplan rengi sarı, bu iki ekran boyutu arasındaki ekran boyutlarında ise arkaplan rengi mor olacak. İki değer arası böyle tanımlanıyor.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Hamburger menü kullanımı için yukarıda bir örnek verilmiştir. Hamburger menü ekran boyutu küçük olan cihazlarda görüntüyü bozmamak için navigasyon öğelerini 3 çizgi altına toplamaya denir. Yukarıdaki örnekte “&#9776” ifadesi hamburger menünün 3 çizgisini temsil eder.



Yukarıdaki örnekte ekran boyutu 400px den az olan cihazlarda görüntü bozulmasın diye navigasyon öğelerini hamburger menüye alır. Aşağıda bu kodun çıktısı verilmiştir.

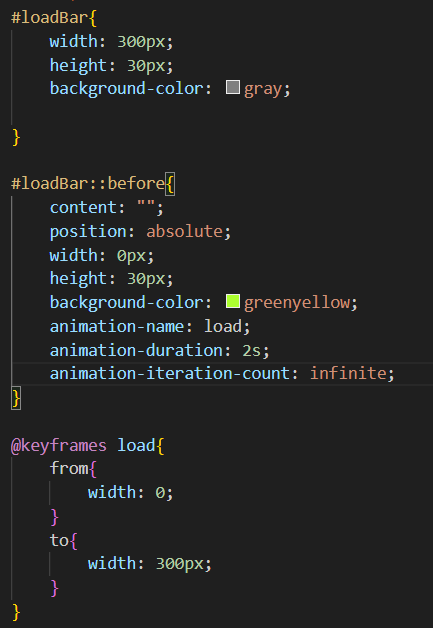


Bu resim hamburger menünün bir örneğidir.

Bu bölümde css’de animasyon nasıl yapılır onu öğreneceğiz.



Yukarıdaki kodda ekrana temsili bir Yükleniyor yazısı yazdıracağız. Ardından bir div etiketi ile yükleme animasyonumuzun bulunacağı alanı tahsis ediyoruz ve id’sini loadBar olarak atıyoruz.



İlk olarak animasyonun arkaplan rengini, yüksekliğini ve genişliğini tanımlıyoruz. Daha sonra before kelimesi ile animasyonun gözükeceği div’in öncesine bir öğe oluşturuyoruz. Bu yüklenen kısım olacak. Animasyonun üstünde gözükmesi için absolute yapıyoruz, rengini, yüksekliğini ve genişliğini ayarlıyoruz. Ardından @keyframes kullanarak bir animasyon oluşturuyoruz. Buradaki animasyonumuzun adını load olarak belirledik. Bu kod bloğunun altında animasyonun nereden başlayıp nerede biteceğini tanımlıyoruz. Ardından before kısmına tekrar dönüyoruz ve animasyon için gerekli bilgileri yazıyoruz. Hangi animasyonun oynatılacağını, animasyonun kaç saniye süreceğini ve animasyonun yenileme biçimin tanımlıyoruz. Buradaki infinite animasyonu sürekli olarak baştan başlatır.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Animasyonun nereden başlayıp nerede biteceğini bu şekilde de tanımlayabiliriz. İstersek 50% gibi ara değerler de ekleyebiliriz.



En son durumda animasyon bu şekilde gözükecektir.

“animation-delay” özelliği animasyon oynatılmadan önce ne kadar bekleyeceğini belirtir. Örneğin “animation-delay: 2s;” yaparsak animasyon oynatılmadan önce 2 saniye bekler ve bu süre bitince animasyonu oynatır.

“animation-timing-function” özelliği animasyonun nasıl olacağını belirtir. Hızlı başlayıp sona doğru yavaşlama veya yavaş başlayıp sona doğru hızlanma bunlara birer örnektir.

metin, yazı tipi, ekran görüntüsü, grafik içeren bir resim

Açıklama otomatik olarak oluşturuldu

Eğer kullanıcı yüklenme barının üstüne fare imlecini getirirse arkaplan rengi mor olacak. Buradaki “transition: .6s” özelliği ile kullanıcı fareyi hemen üstüne getirince değil de kullanıcı fareyi üstüne getirdikten 0.6 saniye sonra mora dönüşecek. Yani bir anda mor olmayacak. Yükleme barı yavaş yavaş morlaşacak ve 0.6 saniyenin sonunda rengi tam mor olacak.