## Flexbox Nedir?

* **Flexbox** yani esnek kutu modelini CSS' te kullanılan bir özellik olmaktan ziyade içerisinde birçok farklı hizalama özelliğini barındıran esnek kutu modeli olarak tanımlayabiliriz.
* **Flexbox CSS** ile bir web ara yüzünde öğeler arasındaki boşlukları ve hizalama şekillerini belirleyebilirsiniz. Flexbox ile öğeleri hem yatay eksende hem de dikey eksende hizalayabilir ve farklı ek hizalama seçenekleri de sunabilirsiniz.

NOT\*\* **Flexbox CSS' i** çalıştırmak için display:flex; özelliğine sahip bir kapsayıcı öğe olması gerekir. İçerisindeki öğeleri de aşağıda belirtilen flexbox CSS özellikleri ile farklı şekillerde konumlandırabilirsiniz.

## Display-flex Özelliği

Esnek kutu modeli flexboxın çalışabilmesi için kapsayıcı öğenin display:flex; ya da display:inline-flex; esnek kapsayıcı özelliklerine sahip olması gerekir.

* .container{display:flex | inline-flex;}

## Flexbox Özellikleri

Flexbox özelliklerini 2 gruba ayırabiliriz.

1. Flex container’ a verdiğimiz özellikler.
   1. Flex-direction
   2. Flex-wrap
   3. Justify-content
   4. Align-items
   5. Align-content
2. Flex container içerisindeki her bir elemana(flex-items) verdiğimiz özellikler.
   1. Align-self
   2. Order
   3. Flex-grow:0
   4. Flex-shrink:1
   5. Flex-basis:auto

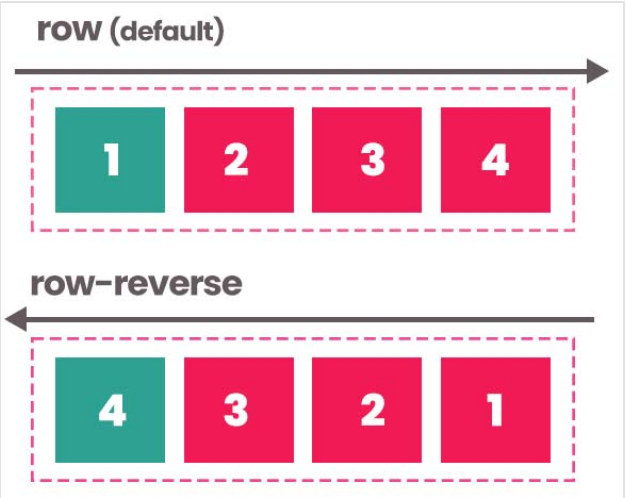
## Kapsayıcı (Container) için Flex Özellikleri

1. Flex-direction

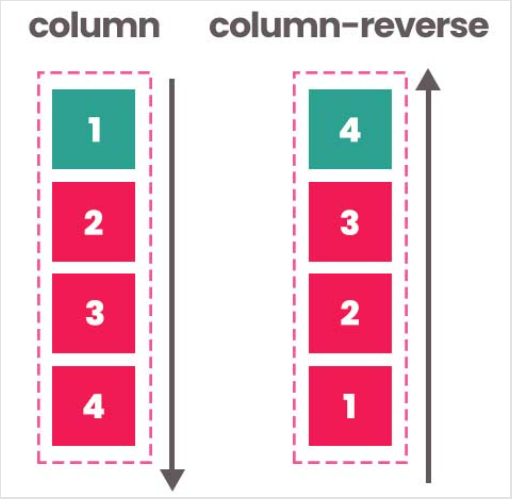
**Flex-direction** özelliği ile öğeleri satır içi yönde sıralı bir şekilde ve alt alta blok şekillerde hizalama yapılabilir.

flex-direction 4 değere sahiptir.

* + flex-direction:row;🡺 özelliği ile öğeler satır içi bloklar olarak yan yana hizalanır.
  + flex-direction:row-reverse;🡺 özelliği ile öğeler satır içi bloklar sondan başlayarak yani tersten yan yana hizalanır.



* + flex-direction:column;🡺 özelliği ile öğeler alt alta bloklar halinde listelenir.
  + flex-direction:column-reverse;🡺 özelliği ile öğeler sondan başlayarak tersten alt alta bloklar halinde listelenir.

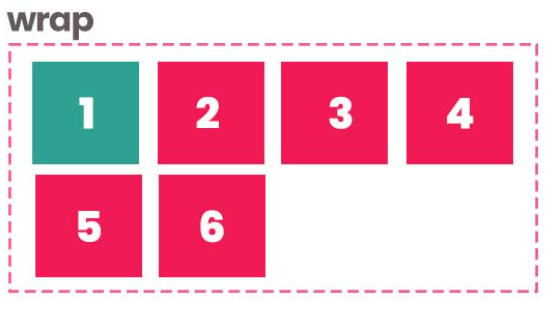


1. Flex-wrap

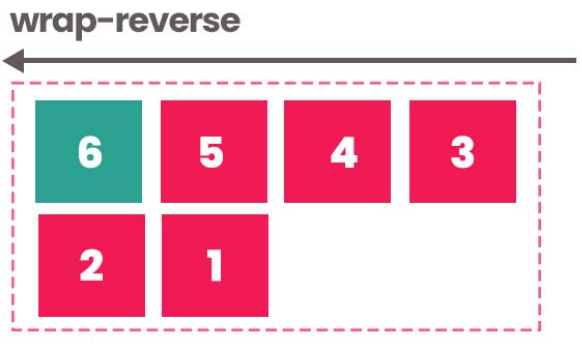
**flex-wrap** CSS özelliği öğelerin kutuya nasıl sığacağını belirtmek için kullanılır. Esnek kutu içerisinde öğelerin alt satıra inmeden taşmasına izin verebilir ya da taşmadan alt satıra geçmesini sağlayabilirsiniz.

flex-wrap 3 değere sahiptir.

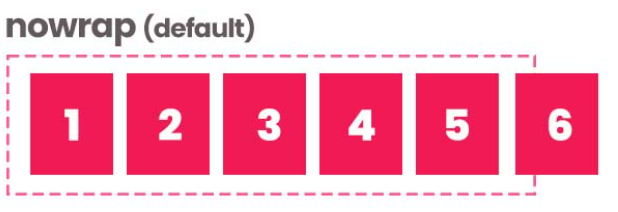
* + flex-wrap:wrap;🡺 özelliği ile öğeler flex container içerisinde taşma yapmadan bir alt satıra ya da yan bloğa geçer. flex-direction:row; fazla öğe soldan sağa doğru alt satıra geçmeye devam eder. flex-direction:column ise bloklar aşağıdan yukarı yan yana oluşmaya devam eder.



* + flex-wrap:wrap-reverse;🡺 özelliğinde örenk flex-direction:row; için bakacak olursak fazla öğe alt satıra geçtiği zaman sıralamayı terse çevirir ve row-reverse özelliği gösterir.



* + flex-direction:nowrap;🡺 özelliği ile öğeler flex kapsayıcı içerisnde flex-directionyönüne bağlı olarak genişliği ya da yüksekliği daralarak taşma yapar.



1. justify-content

**justify-content** css özelliği esnek kutu içerisindeki öğeleri yatay eksene göre hizalar.

* + Justify-content:flex-start;🡺 değeri ile öğeler esnek kapsayıcı içerisinde en baştan başlayarak dizilir.



* + Justify-content:flex-end;🡺 değeri ile öğeler esnek kapsayıcının sonuna itilir ve sonundan başlayarak dizilir.



* + Justify-content:center;🡺 değeri ile öğeler esnek kapsayıcıya ortalı olarak dizilir.



* + Justify-content:space-between;🡺 değeri ile öğe çifti arasındaki boşluklar eşit bir şekilde dizilir. Esnek kapsayıcı ile kapsayıcının en başındaki öğeler ile en sonundaki öğeler aasında boşluk kalmaz.



* + Justify-content:space-around;🡺 değeri ile öğe çifti arasındaki boşluklar eşit bir şekilde dizilir. Esnek kapsayıcı ile kapsayıcının en başındaki öğeler ile en sonundaki öğeler aasında boşluk öğeler arasındaki boşluğun yarısına eşit olur.



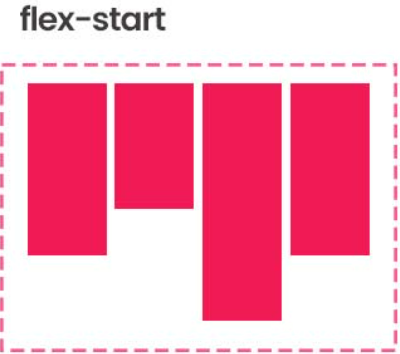
* + Justify-content:space-evenly;🡺 değeri ile öğe çifti arasındaki boşluklar eşit bir şekilde dizilir. Esnek kapsayıcı ile kapsayıcının en başındaki öğeler ile en sonundaki öğeler arasında boşluk öğeler arasındaki boşluğa eşit olur.



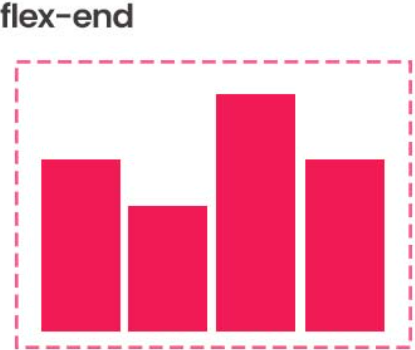
1. Align-items

**Align-items** özelliği ile nesnelerin dikeyde hizalanmasını sağlar.

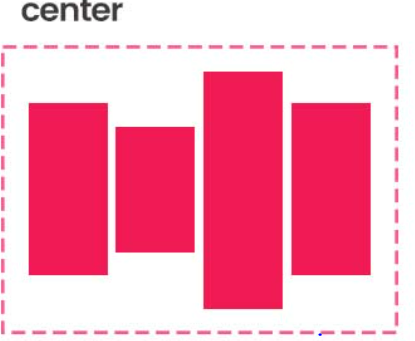
* + Align-items:flex-start/start;🡺 değerinde öğeler esnek kabın başlangıç çizgisine doğru hizalanır. flex-direction kuralına göre değişkenlik gösterir.



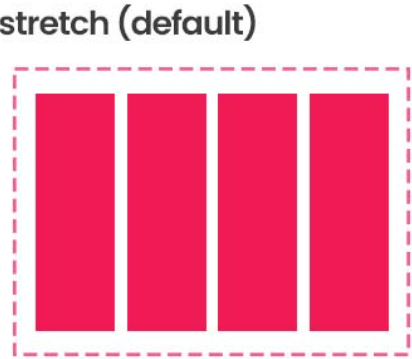
* + Align-items:flex-end/end;🡺 değerinde öğeler enek kabın en uç kenarına doğru hizalanır. flex-direction kuralına göre değişkenlik gösterir.



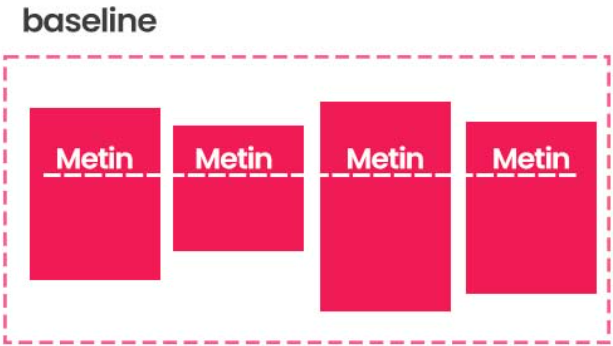
* + Align-items:center;🡺 değerinde öğeler çapraz eksene ortalı olarak hizalanır. flex-direction kuralına göre değişkenlik gösterir.



* + Align-items:strech;🡺 değerinde öğeler esnek kabı doldurmak için germe işlemi yapılarak kapsanır.



* + Align-items:baseline;🡺 değerinde öğeler metin taban çizgileri baz alınarak dizilim gösterir.

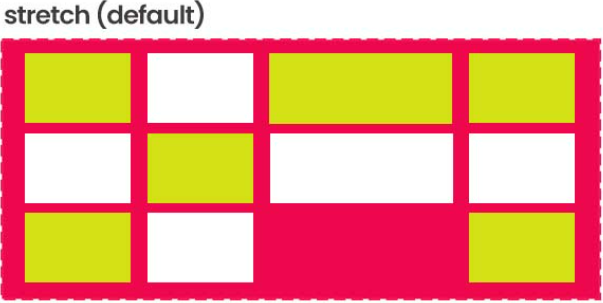


1. Align-content

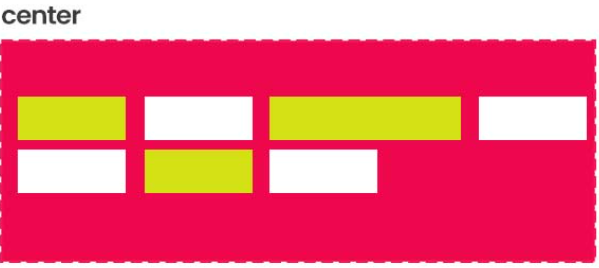
İki veya daha fazla satırdan oluşan kutuların dikeyde hizalanması için kullanlır. İki veya daha fala satır olması durumunda kulanılır.

NOT\*\* **align-content** özelliklerinin çalışabilmesi için flex-wrap değerinin *wrap* olarak ayarlanması gerekmektedir. Varsayılan *nowrap* değerinde **align-content** özelliği çalışmayacaktır.

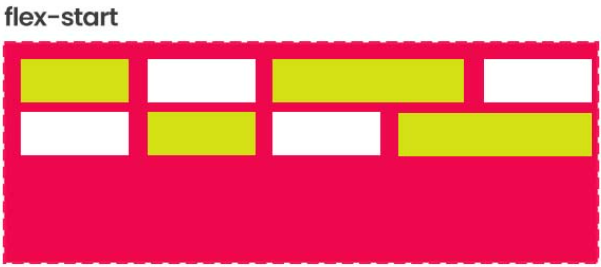
* + Align-content:stretch;🡺 align-content:stretch; varsayılan değerdir. stretch değeri auto değerdeki boyutlu öğelerin çapraz eksene göre boyutlarını sığacak şekilde width ya da height değerlerinin eşitlenerek artırılmasıdır. Boyutu auto olmayan öğeler bu tanımın dışında kalacaktır.



* + Align-content:center;🡺 align-content: center; CSS özelliği ile öğeler esnek kap içerisinde çapraz eksene ortalı bir şekilde dizilir. Bu özellik öğeleri 2 eksende de esnek kutunun içerisinde ortalamaz. Hem ana eksen hem de çapraz eksen de ortalama yapılmak isteniyorsa yani kutunun tam ortasına öğeler yerleştirilecek ise justify-content: center; Özelliği ile birlikte kullanılmalıdır. justify-content özelliği de öğeleri esnek kutu içerisinde main-axise (ana-eksene) göre hizalama yapmaktadır.



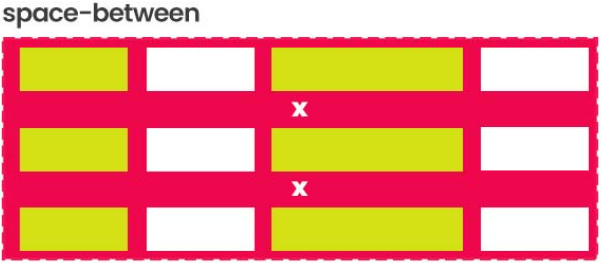
* + Align-content:flex-start;🡺 align-content: flex-start değeri ile esnek kutu içerisindeki öğeleri esnek kutunun çapraz-ekseni doğrultusunda en başa dizilir. main-axis(Ana Eksen) ve cross-axis (Çapraz Eksen) konularını incelemek için flex-direction özelliklerine göz atınız.



* + Align-content:flex-end;🡺 align-content: flex-end değeri ile esnek kutu içerisindeki öğeleri esnek kutunun çapraz-ekseni doğrultusunda en sona dizilir.



* + Align-content:space-between;🡺 align-content: space-between değeri ile esnek kutu içerisindeki ilk hat çapraz eksene göre en başta son hat ise çapraz eksene göre en sonda dizilir. Diğer hatlarda onları ortalayacak şekilde dizilirler. align-content:space-between değerinde her hat arası boşluk eşit olacak şekilde öğeler dizilir. Her hat içerisindeki öğelerin dizilimini de ayarlamak istiyorsanız align-items özelliğini kullanabilirsiniz.



* + Align-content:space-around;🡺 align-content: space-around değeri ile çoklu hata sahip flexbox yapısı içerisinde her hat arası boşuk eşit olacak şekilde esnek kaba çapraz eksende ortalı sıralanır, en üstteki ve en sondaki hattın esnek kapsayıcı arasındaki boşluk hatlar arasındaki boşluğun **yarısına eşittir.**



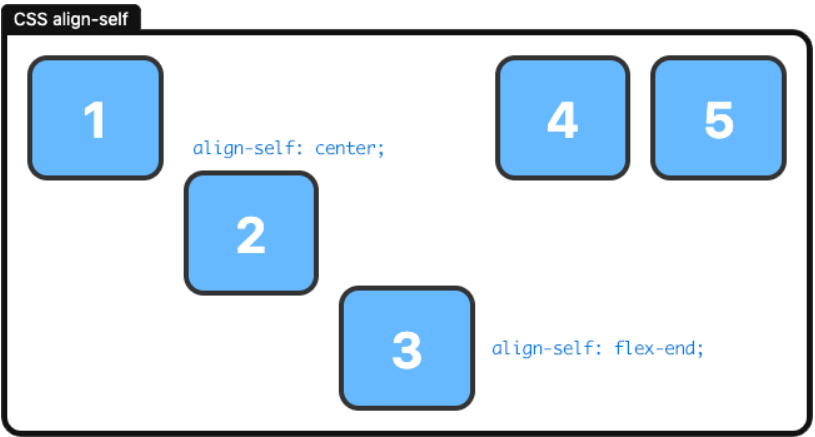
* + Align-content:space-evenly;🡺 align-content: space-evenly değeri ile çoklu hata sahip flexbox yapısı içerisinde her hat arası boşuk eşit olacak şekilde esnek kaba çapraz eksende ortalı sıralanır, en üstteki ve en sondaki hattın esnek kapsayıcı arasındaki boşluk hatlar arasındaki **boşluğa eşittir**.



## Öğeler (items) için Flex Özellikleri

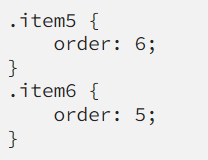
1. Align-self

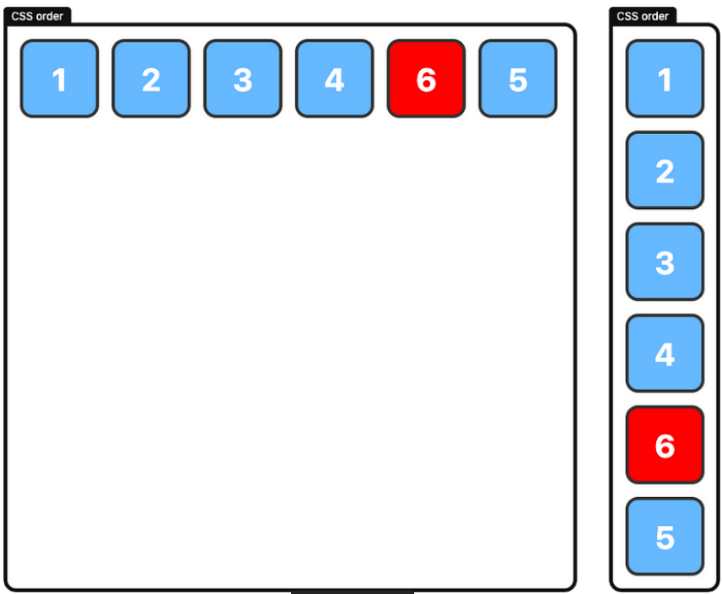
align-content ile aynı işlemleri yapar. Dikey eksende öğelerin nasıl hizalanacağı belirlenir. Tek farkı kapsayıcı yerine bunu öğelere uygulamasıdır.



1. Order

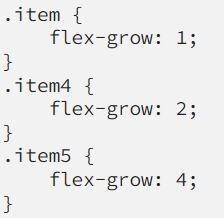
Öğelerin sıralanışını değiştirmek için kullanılır. Burada dikkat edilmesi gereken, varsayılan olarak 0 olduğu için tüm öğeler, eğer sonuncu öğeyi bir öncesine almak isterseniz, bir önceki öğenin de order değerini belirtmeniz gerekir.





1. Flex-grow

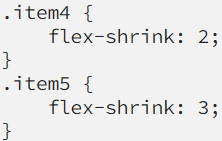
Bir öğeyi diğerlerine göre daha büyük göstermek için bu özelliği kullanıyoruz. Varsayılan değeri 0’dır. Eğer öğelerden birisinin değerini 2 yaparsanız diğerlerine göre daha büyük olacaktır ancak kalanı daralacaktır. İlk olarak tüm öğelerin değerini 1 yapıp, daha sonra bir öğenin değerini 2 yaparsanız, sonuç olarak diğer öğelere göre 2 kat daha büyük bir öğeye sahip olacaksınız.





1. Flex-shrink

Bu aslında flex-grow özelliğinin tam aksine, küçültme işlemi için kullanılır. Varsayılan olarak değeri 1’dir. Eğer bir öğeye değer olarak 2 verirseniz, diğerlerine göre küçülmesi gerektiğinde daha önce küçülecektir. Değeri artırırsanız örneğin 4. öğeye 2, 5. öğeye 3 değeri verirseniz, küçülme esnasında her ikiside küçülecektir ancak 5. öğe 4. öğeye göre daha hızlı küçülecektir. Zaten bu öğelerde artık küçülecek yer kalmadığında diğer öğeler küçülmeye başlar. Esnek yapılarda bildiğiniz gibi kapsayıcıya göre genişlik ve yükseklik ayarlanmaktadır. Bu gibi durumlarda büyümeye ve küçülmeye müdahale için bu özellik işe yarayacaktır.

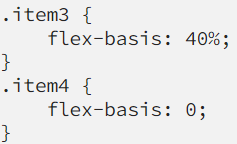


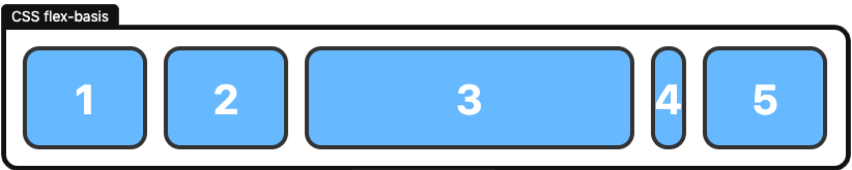


1. Flex-basis

Öğelerin eğer alanı yetiyorsa olması gereken genişliği belirlemek için kullanılır. Değer olarak tüm uzunluk birimleri (px, pt, em vb.) kullanılabilir. Varsayılan olarak değeri auto’dur. Eğer değer olarak 0 belirlenirse, doğrudan sadece içindeki değer kadar genişliği olur.

Örneğin 3. öğeye %40, 4. öğeye ise 0 değerlerini verdiğimizde aşağıdaki gibi bir sonuç elde ederiz.





1. Flex

Sırasıyla flex-grow, flex-shrink ve flex-basis özelliklerinin kısa kullanımıdır. İlk değeri zorunlu olup (shrink ve basis) değerlerini belirtmek zorunlu değildir. Varsayılan olarak 0 1 auto değerine sahiptir. Ve en sık kullanımı da şöyledir;

