

# FLEXBOX

## Bootstrap Flexbox Nedir?

**Flexbox**, (ya da **Flexible Box Layout**), CSS ile esnek ve dinamik düzenler oluşturmak için kullanılan bir düzenleme modelidir. **Bootstrap**, 4. sürümünden itibaren **Flexbox** sistemini varsayılan olarak kullanmaya başlamıştır. Bu, sayfa elemanlarının yatayda veya dikeyde esnek bir şekilde hizalanmasını ve yerleşmesini sağlar. Flexbox, geleneksel CSS düzenleme yöntemlerine göre çok daha güçlü ve esnektir, çünkü öğeleri esnek bir şekilde düzenler ve farklı ekran boyutlarına göre uyum sağlar.

Bootstrap'ta **Flexbox**, elemanları hizalamak, sıralamak, büyütme veya küçültme için çeşitli yardımcı sınıflar (utility classes) sunar. Bu sayede sayfa düzenlerinde, özellikle de duyarlı (responsive) tasarımlarda büyük kolaylık sağlar.

## Flexbox Temel Kavramlar

### 1. Flex Container (Esnek Kapsayıcı)

Flexbox ile çalışmaya başlamak için bir **flex container** (esnek kapsayıcı) oluşturmanız gerekir. Flex container, içinde bulunan öğelere **flex item** (esnek öğe) özelliklerini uygulayacaktır.

- **.d-flex**: Bir öğeyi flex container yapar.
- **.d-inline-flex**: Flex container yapar ama öğenin satır içi (inline) olmasını sağlar.

```
<div class="d-flex">
```

```
<!-- Flex öğeler burada olacak --> </div>
```

## 2. Flex Items (Esnek Öğeler)

Flex container içerisine eklediğiniz her öğe, otomatik olarak **flex item** haline gelir. Flex item'lar, kapsayıcıları içinde esnek bir şekilde sıralanabilir ve hizalanabilir.

### Flexbox ile Kullanılabilen Temel Özellikler

#### 1. Hizalama (Aligning)

Flexbox ile öğelerinizi yatayda ve dikeyde kolayca hizalayabilirsiniz. Bu, özellikle **ortalamak** ve öğeleri **sayfa boyunca dağıtmak** için çok kullanışlıdır.

- **.justify-content-\***: Yatayda hizalama (soldan sağa doğru).
- **.align-items-\***: Dikeyde hizalama (yukarıdan aşağıya doğru).
- **.align-self-\***: Belirli bir öğeyi dikeyde hizalamak için kullanılır (flex item).

### Bootstrap Flexbox Nedir?

**Flexbox**, (ya da **Flexible Box Layout**), CSS ile esnek ve dinamik düzenler oluşturmak için kullanılan bir düzenleme modelidir. **Bootstrap**, 4. sürümünden itibaren **Flexbox** sistemini varsayılan olarak kullanmaya başlamıştır. Bu, sayfa elemanlarının yatayda veya dikeyde esnek bir şekilde hizalanmasını ve yerleşmesini sağlar. Flexbox, geleneksel CSS düzenleme yöntemlerine göre çok daha güçlü ve esnektir, çünkü öğeleri esnek bir şekilde düzenler ve farklı ekran boyutlarına göre uyum sağlar.

Bootstrap'ta **Flexbox**, elemanları hizalamak, sıralamak, büyötmek veya küçöltmek için çeşitli yardımcı sınıflar (utility classes) sunar. Bu sayede sayfa düzenlerinde, özellikle de duyarlı (responsive) tasarımlarda büyük kolaylık sağlar.

## Flexbox Temel Kavramlar

### 1. Flex Container (Esnek Kapsayıcı)

Flexbox ile çalışmaya başlamak için bir **flex container** (esnek kapsayıcı) oluşturmanız gerekir. Flex container, içinde bulunan öğelere **flex item** (esnek öğe) özelliklerini uygulayacaktır.

- **.d-flex:** Bir öğeyi flex container yapar.
- **.d-inline-flex:** Flex container yapar ama öğenin satır içi (inline) olmasını sağlar.

#### Örnek:

```
html
Kodu kopyala
<div class="d-flex">
  <!-- Flex öğeler burada olacak -->
</div>
```

### 2. Flex Items (Esnek Öğeler)

Flex container içerisine eklediğiniz her öğe, otomatik olarak **flex item** haline gelir. Flex item'lar, kapsayıcıları içinde esnek bir şekilde sıralanabilir ve hizalanabilir.

## Flexbox ile Kullanılabilen Temel Özellikler

### 1. Hizalama (Aligning)

Flexbox ile öğelerinizi yatayda ve dikeyde kolayca hizalayabilirsiniz. Bu, özellikle **ortalamak** ve öğeleri **sayfa boyunca dağıtmak** için çok kullanışlıdır.

- **.justify-content-\***: Yatayda hizalama (soldan sağa doğru).
- **.align-items-\***: Dikeyde hizalama (yukarıdan aşağıya doğru).

- **.align-self-\***: Belirli bir öğeyi dikeyde hizalamak için kullanılır (flex item).

## Yatay Hizalama (Main Axis)

Flexbox'ta, **yatay hizalama** "main axis" (ana eksen) üzerinde yapılır. Bu eksen, varsayılan olarak **soldan sağa** doğrudur. Yatayda hizalama yapmak için `justify-content` sınıfını kullanabilirsiniz.

- **.justify-content-start**: Öğeleri sola hizalar.
- **.justify-content-center**: Öğeleri ortalar.
- **.justify-content-end**: Öğeleri sağa hizalar.
- **.justify-content-between**: Öğeleri, ilk ve son öğe arasına eşit boşluk bırakacak şekilde dağıtır.
- **.justify-content-around**: Öğeleri, aralarındaki boşlukları eşit olacak şekilde dağıtır.

### Örnek:

```
<div class="d-flex justify-content-center">
```

```
<div>Öğ 1</div>
```

```
<div>Öğ 2</div>
```

Bu örnekte, iki öğe yatayda **ortalanmış** olur.

## Dikey Hizalama (Cross Axis)

Dikey hizalama "cross axis" (çapraz eksen) üzerinde yapılır. Varsayılan olarak bu eksen **yukarıdan aşağıya** doğrudur. Dikey hizalama yapmak için `align-items` sınıfını kullanabilirsiniz.

- **.align-items-start**: Öğeleri üstte hizalar.
- **.align-items-center**: Öğeleri dikeyde ortalar.
- **.align-items-end**: Öğeleri altta hizalar.
- **.align-items-baseline**: Öğeleri, metinlerin alt çizgisine hizalar.
- **.align-items-stretch**: Öğeleri, kapsayıcı yüksekliğine yayar.

```
<div class="d-flex align-items-center" style="height: 100px;">

  <div>Öğ 1</div>

  <div>Öğ 2</div>

</div>
```

Bu örnekte, öğeler dikeyde **ortalama** yapılır.

## 2. Flex Direction (Esnek Yön)

**flex-direction** özelliği, öğelerin hangi ekseninde sıralanacağını belirler. Bootstrap'ta, bu özellik **.flex-row** ve **.flex-column** sınıflarıyla uygulanır.

- **.flex-row:** Öğeleri **yatay** (soldan sağa) olarak sıralar.
- **.flex-column:** Öğeleri **dikey** (yukarıdan aşağıya) olarak sıralar.
- **.flex-row-reverse:** Öğeleri **ters yönde** yatay olarak sıralar (sağdan sola).
- **.flex-column-reverse:** Öğeleri **ters yönde** dikey olarak sıralar (aşağıdan yukarıya).

### Örnek:

```
<div class="d-flex flex-column">

  <div>Öğ 1</div>

  <div>Öğ 2</div>

</div>
```

Bu örnekte, öğeler **dikey olarak sıralanır**.

## 3. Flex Wrap (Sarılma)

Flexbox ile öğelerinizin, container'ın boyutlarına göre **sarılmasını** sağlayabilirsiniz. Bu özellik, öğelerin fazla yer kaplamasını engeller.

- **.flex-wrap:** Öğelerin sarılmasına olanak tanır.
- **.flex-nowrap:** Öğelerin sarılmasını engeller (varsayılan).
- **.flex-wrap-reverse:** Sarılma yönünü tersine çevirir.

```
<div class="d-flex flex-wrap">
  <div>Öğ 1</div>
```

```
<div>Öğ 2</div>
<div>Öğ 3</div>
<div>Öğ 4</div>
</div>
```

Bu örnekte, öğeler yanyana sığmazsa **sarılır**.

#### 4. Order (Sıra)

Flexbox ile öğelerin sırasını değiştirebilirsiniz. Bootstrap'ta, öğelerin sırasını değiştirmek için **order** sınıflarını kullanabilirsiniz.

- **.order-1, .order-2, .order-3,...**: Öğelere sırasıyla farklı numaralar atar.

```
<div class="d-flex">
  <div class="order-3">Öğ 1</div>
  <div class="order-1">Öğ 2</div>
  <div class="order-2">Öğ 3</div>
</div>
```

Bu örnekte, öğeler **sırasıyla değiştirilir**.

#### 5. Align Self (Özelleştirilmiş Hizalama)

Flex item'lar üzerinde daha özel hizalamalar yapabilirsiniz. Bu, **align-self** özelliği ile yapılır. Her bir öğe, kapsayıcıya göre kendi hizalamalarını alabilir.

- **.align-self-start**: Öğeyi üstte hizalar.
- **.align-self-center**: Öğeyi dikeyde ortalar.
- **.align-self-end**: Öğeyi altta hizalar.
- **.align-self-stretch**: Öğeyi dikeyde gerer (yükseklikle).

```
<div class="d-flex" style="height: 100px;">
  <div class="align-self-center">Öğ 1</div>
  <div>Öğ 2</div>
</div>
```

Bu örnekte, **ilk öğe dikeyde ortalır**, diğer öğe ise normal hizalamada kalır.

## Özet

Bootstrap'ta **Flexbox** kullanımı, özellikle **esnek, duyarlı ve dinamik** düzenler oluşturmak için çok faydalıdır. **Yatay ve dikey hizalama, sıra değiştirme, sarılma (wrap)** gibi özelliklerle elemanlarınızı rahatça düzenleyebilir, ekran boyutlarına göre uyumlu ve esnek tasarımlar oluşturabilirsiniz. Bootstrap, bu işlemleri kolaylaştırmak için çeşitli **yardımcı sınıflar (utility classes)** sunar ve böylece CSS yazmaya gerek kalmadan düzeninizi oluşturabilirsiniz.