

CSS Position:

CSS position -> herhangi bir nesnenin web sayfasında yani tarayıcı üzerindeki konumunu ayarlamak için kullanılır.

CSS Syntax:

position: static | relative | absolute | fixed | sticky | inherit;

Yukarıda yazımları verilen birkaç konumlandırma özellikleri mevcuttur.

Static: Varsayılan olarak ayarlanmış konumlandırma özelliğidir. Bu özellik konumlandırma işlemlerinin gerçekleştirilememesine neden olur. Yani adından da anlaşılacağı üzere elemanların sabit olarak konumlandırılmış durumudur. Statik olarak konumlandırılmış bir eleman normal sayfa akışına göre soldan sağa, yukarıdan aşağıya doğru konumlandırılır. Fakat static özelliğe sahip bir öğe için Top, right, bottom, left özellikleri geçerli değildir. Örnekte de görüleceği üzere herhangi bir özellik verilmediği sürece varsayılan özellik static olacaktır.

```
tag {  
  left: 100px;  
  top: 150px;  
}
```

Konumlandırma Özellikleri

Bu başlık static özellik ile ayarlanmıştır

Statik konumlandırmada, bir öğe sayfanın herhangi bir yerine üst, sol, sağ vb. Özelliklerle yerleştirilemez.

Bu sebeple verilen örnek için left ve top özellikleri çalışmayacaktır.

Relative: Göreceli olarak konumlandırılmış bir özelliktir. Göreceli olarak konumlandırılan elemanlar, sayfa içerisinde Top, right, bottom, left özellikleri sayesinde istenilen bölgeye taşınabilir. Bu özellik eğer sayfada herhangi bir nesne yoksa sayfa başlangıcına göre sayfada eleman var ise elemanın konumuna göre yer alma gerçekleştirir.

```
tag {  
  position: relative;  
  left: 100px;  
  top: 150px;  
}
```

Konumlandırma Özellikleri

Görelî konumlandırma ile, öğe sayfanın herhangi bir yerine yerleştirilebilir. Aşağıdaki başlık sayfanın solundan 100 piksel ve div bloğunun üstünden 150 piksel yerleştirilmiştir.

Bu başlık relative özelliği ile ayarlanmıştır

Absolute: Bir nesnenin konumlandırma özelliğinin sayfanın başlangıcı kabul edilerek ayarlama yapması özelliğidir. Sayfanın Left-Top=0 konumundan hizalama gerçekleştirmek için kullanılır. Kendisini kapsayan bir üst öğe relative konumlanmışsa o zaman öğeye göre konum almaktadır. Böyle bir eleman bulunmazsa, nesne web sayfasına göre konumlanacaktır.

```
tag {  
  position: absolute;  
  left: 100px;  
  top: 150px;  
}
```

Konumlandırma Özellikleri

Mutlak konumlandırma ile, öğe sayfanın herhangi bir yerine yerleştirilebilir. Aşağıdaki başlık sayfanın solundan 100 piksel ve sayfanın üst kısmından 150 piksel yerleştirilmiştir.

Bu başlık absolute özelliği ile ayarlanmıştır

Fixed: Sabitlenmiş bir eleman tarayıcı penceresine göre konumlandırılmıştır. Pencere scroll özelliği oluşsa bile sayfanın aşağı-yukarı hareketi nesnenin konumunu bozmayacaktır.

Sticky: Yapışkan öğe göreceli ve sabit, kaydırma konumuna bağlı arasında geçiş yapar. Oluşturulan eleman kaydırma çubuğu gelene kadar position:relative gibi davranır, kaydırma çubuğu geldikten sonra position:fixed gibi davranır. Bu özellik sayfada sabit kalan butonlar için kullanılır. Ayrıca özelliğin verimli çalışabilmesi için sticy özelliğine sahip nesneden sonra devam eden sayfada farklı nesnelere bulunmalıdır.

z-index: Nesnelerin web sayfası içerisindeki z boyutundaki konumunu belirlemek için kullanılır. Bu durum üst üste gelmiş nesnelere için ifade edilebilir.

Sticky->Örnek:

```
.d1 {  
  position: sticky;  
  top: 0;  
  background-color: dodgerblue;  
  border: 2px solid blue;  
}
```

Z-index-> Örnek

```
img {  
  width: 100%;  
  height: auto;  
  opacity: 0.4;  
}
```

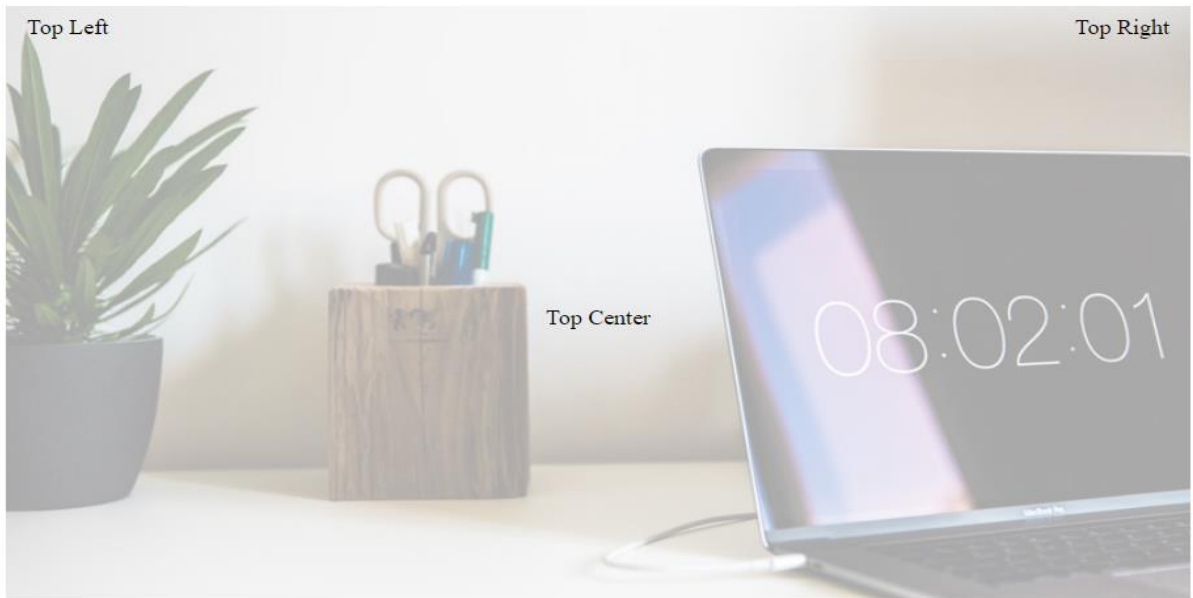
```
.topleft {  
  position: absolute;  
  top: 8px;  
  left: 16px;  
  font-size: 18px;  
}
```

```
.topright {  
  position: absolute;  
  top: 8px;  
  right: 16px;  
  font-size: 18px;  
}
```

```
.topcenter {  
  position: absolute;  
  left: 0;  
  top: 50%;  
  width: 100%;  
  font-size: 18px;  
  text-align: center;  
}
```

Resim Yazısı

Resmin köşelerinde bulunan text'ler z-index ve position özelliği ile eklenmiştir.



CSS Navigation Bar:

1. **Adım:** Bir liste düzeni oluşturmak: oluşan liste düzenine göre çıktı yandaki şekilde gerçekleşecektir.

```
<ul>
  <li><a href="index.html">Anasayfa</a></li>
  <li><a href="haber.html">Haberler</a></li>
  <li><a href="iletisim.html">İletişim</a></li>
  <li><a href="hakkında.html">Hakkında</a></li>
</ul>
```

- [Anasayfa](#)
- [Haberler](#)
- [İletişim](#)
- [Hakkında](#)

2. **Adım:** Listenin özelliklerini CSS. Style üzerinden ayarlamak. Bu adımlar sayesinde 1. Adımda gerçekleştirilen liste için link tiplerinin madde gösterimleri ortadan kaldırılacak.

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
```

- [Anasayfa](#)
- [Haberler](#)
- [İletişim](#)
- [Hakkında](#)

3. **Adım:** Liste elemanlarının background özelliği sayesinde renk verilecek ve display özelliği sayesinde alt alta sıralı düzenleme gerçekleştirilecek.

```
li a {
  display: block;
  width: 60px;
}
```

- [Anasayfa](#)
- [Haberler](#)
- [İletişim](#)
- [Hakkında](#)

Without
display:block

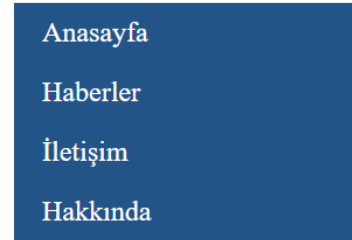
- [Anasayfa](#)
- [Haberler](#)
- [İletişim](#)
- [Hakkında](#)

With
display:block

4. **Adım:** Yatay Buton Bar-> Vertical Navigation Bar:

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}
li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}
li a:hover {
  background-color: #555;
  color: white;
}
```

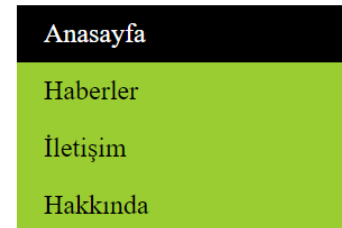
Vertical Navigation Bar



5. **Adım:** Oluşturduğumuz bu bar üzerinde bulunduğumuz sayfa linki "active" özelliği verme.

```
.active {
  background-color: #4CAF50;
  color: white;
}
```

Vertical Navigation Bar



Yukarıda verdiğimiz hover özelliği sayesinde butonlar üzerinde gezinti gerçekleştirilirken renklerin değişimleri gözlemlenebilir.

Şimdi uygulamanız: Fixed özelliğine sahip Responsive tasarımda bir Sayfa tasarlamak:

