**CSS DİSPLAY KULLANIMI**

**Display** özelliği bir HTML elemanı için kullanılan kutu türünü belirler. Web sayfamızda elemanların ne tür konumlandırılacağı ve sayfamızda görünüp görünmeyeceğini belirler.

**Display** elamanın hangi eleman gibi davranacağını belirtir.

Web sayfalarımızı oluşturan HTML etiketleri sayfa yerleşimlerine göre **inline** elemanlar ve **blok** elemanlar olmak üzere iki gruba ayrılır.

**İnline (satır) Elemanlar**

İnline aynı hizada, yan yana görüntülenir yeni bir satır başlatmazlar. Yükseklik ve genişlik gibi özellikler verilmez. Bu elemanlar kendisini oluşturan içerik kadar alan kaplarlar ve satır sonunu zorlamaz yani bir alt satıra geçmezler.

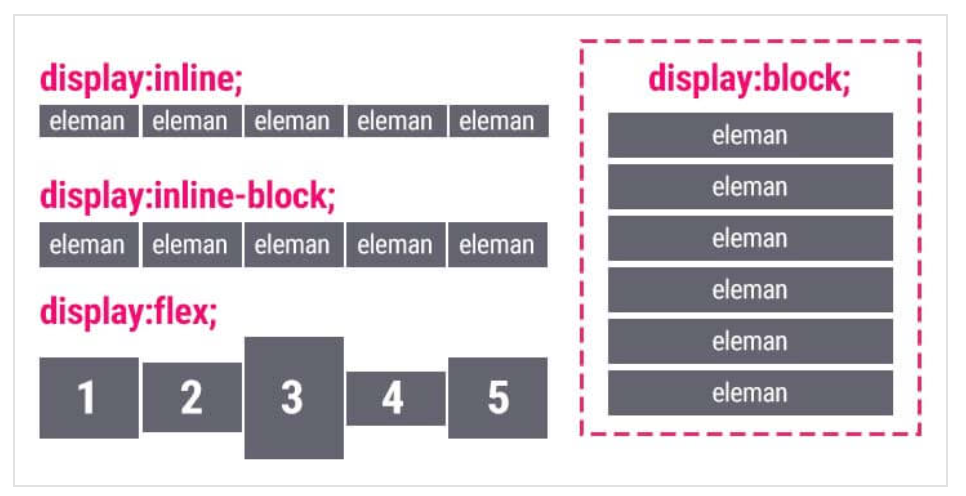
* span
* a
* b
* input
* img
* label
* strong
* small

**Blok Elemanlar**

Öncesi ve sonrası satır sonu olan bir elementtir. Bulundukları yerin mevcut tam genişliğine sahip olurlar. Blok seviyesi elemanların yüksekliği, genişliği, dolgusu, kenar boşlukları vb. özellikleri ayarlanabilir.

* h1
* p
* li
* div
* ul
* table

Display özelliği ile inline elemanları bir blok elemanı haline dönüştürülebilir. Bir blok elemanı da inline elemanı haline dönüştürülebilir.



* **display:inline;** özelliği tanımlanan bir öge height ve width değeri tanımlanmayacaktır. İnline-level düzeyinde olan bu ögeler tarayıcıda yan yana sıralanacaktır.
* **display:block;** özelliği bir ögenin block-level olmasını sağlar. Block-level ögeler h1, p, div gibi her hangi başka bir dışarıdan CSS etkisi olmadığı taktirde alt alta sıralanacaktır.
* **display:flex;** özelliği flexbox/esnek kutu modelini kullanmanızı sağlar. Flexbox özelliği ile öğeleri istediğiniz şekilde hizalayabilir ve konumlandırabilirsiniz.
* **display:none;** özelliği HTML ögesini gizlemek ve yok saymak için kullanılır. Atanan bir öge içeriği ile birlikte tarayıcıdan yok sayılır ve web sayfasında gösterilmez.
* **display:inline-block;** özelliği sayesinde ögeleri **display:inline;** da olduğu gibi satır içi yan yana dizebilirsiniz. İnlineBlock-Level bir CSS özelliği olduğu için width ve height değerleri tanımlayabilirsiniz.

**CSS POSİTİON ÖZELLİĞİ**

CSS position özelliği HTML elemanlarının sayfa içerisindeki konumunu belirlememize yardımcı olan bir özelliktir. Position özelliğinin yanında CSS’in top, bottom, left ve right özellikleri de konumlandırmak için yardımcı olarak kullanılır.

## **CSS Position Çeşitleri**

* position: static;
* position: relative;
* position absolute;
* position: fixed;
* position: sticky;

### Position Static

### Static değeri tüm HTML elemanlarının varsayılan position değeridir. Bir [HTML](https://juniortoexpert.com/tr/web-tasarim/html/) elemanına position değeri verilmezse her zaman static değeri geçerlidir. Position static değerinde top, bottom, right, left ve z-index CSS özellikleri etki göstermez.

### Position Relative

HTML elemanına relative değeri verildiğinde kendi konumunu korur ve top, bottom, left ile right değerleri için sıfıra sıfır noktasını kendi konumu olarak belirler. Sayfa üzerinde top, bottom, left ve right değerleri değiştirildiğinde relative özelliği verilmiş elemanın kapladığı boşluk değişmez static gibi davranır.

### Position Absolute

Absolute değeri, HTML elemanının sayfa akışı üzerindeki konumunu bozar. Position relative değerinde olduğu gibi kendi kapladığı boşluk static gibi davranıp konumunu korumaz, sayfa akışı kayar. Absolute değeri almış eleman kendisine en yakın olan relative değerine sahip bir üst eleman içerisinde konumlanır. Eğer üstünde relative değerine sahip bir konteyner yoksa, absolute değeri almış eleman sayfanın sol üst köşesinde konumlanır. Top, bottom, left ve right değerleri elemanın sayfa içerisinde konumlanmasına yardımcı olur.

### Position Fixed

Fixed değeri absolute değeri ile benzer. Fixed değeri de HTML elemanının sayfa akışı üzerindeki konumunu bozar. Fixed değeri ile HTML elemanı sayfa üzerinde istenilen bir noktaya sabitlenebilir. Top, bottom, left ve right değerleri fixed değeri verilmiş elemanın sayfa içerisinde konumlanmasına yardımcı olur.

### Position Sticky

Sticky değeri, relative ile fixed değerlerinin karışımı olarak düşünülebilir. Sticky pozisyonu verilmiş eleman belirli bir eşiği geçene kadar belirtilmiş noktada sabit kalır. Bu belirli nokta ise sticky pozisyonu verilmiş elemanın bir üst konteynır elemanıdır.

## **Justify-Content**

Flex özelliğine sahip olan bir kapsayıcının yatay eksende nasıl hizalanacağını ayarlar. Objeler başta mı, ortada mı, sonda mı, aralıklı mı dizileceğine karar vermemizi sağlar. Bu kararı vermek için 5 parametre kullanır:

* **flex-start:** Nesneler kapsayıcının başından başlayarak dizilirler.
* **space-around:** Nesneler arasında eşit boşluklar bırakılır.
* **space-between:** İlk nesne başta, son nesne sonda olacak şekilde nesneler eşit olarak kapsayıcının içerisinde dizilirler.
* **center:** Nesneler tam ortalı olacak şekilde hizalanır.
* **flex-end:** Son nesne en sağa yaslanacak şekilde hizalanır.

## **Align-İtems**

Bu özellik ise **justify-content**‘in yatay eksende yaptığı şeyleri dikey eksende yapmaya yarar. Yine aynı şekilde hizalama işlemini dikey eksende nasıl yapmamız gerektiğine karar vermemiz gerekmektedir. Bu özellik de yine 5 parametre kullanır ve bunlardan 3 tanesi justify-content özelliği ile aynıdır.

* **flex-start:** Nesneler, kapsayıcının en üstünden başlayarak dizilirler.
* **flex-end:** Nesneler, kapsayıcının en altından başlayarak, yukarı doğru dizilirler.
* **center:** Nesneler ortalı bir şekilde dikey olarak sıralanırlar.
* **baseline:** İlk satırdaki yazı tipi göz önüne alınarak hizalanmaya başlanılır.
* **stretch:** Tüm nesneler kapsayıcı yüksekliğine eşit olacak şekilde dizilirler.