CSS Nedir?

CSS, açılımı “Cascading Style Sheet” olan ve dilimize Stil şablonları olarak yerleşen basit ve kullanışlı bir işaretleme dilidir. Stil Şablonu HTML ögelerine (yazı, paragraf, kenar çizgisi, resim, bağlantı...) şekil vermek amacıyla kullanılır. Diğer bir deyişle sayfamızın içeriğinin biçimlendirilme işleminin yapıldığı kısımdır. HTML/XHTML etiket dillerinin sayfa tasarımında yetersiz kalması nedeniyle “World Wide Web Consortium” (W3C, Dünya Çapında Ağ Birliği) tarafından oluşturulmuştur; içerik kısmı (HTML) ile biçimlendirme kısmını birbirinden ayırarak yüzlerce sayfayı tek bir dosya ile biçimlendirme olanağı tanımaktadır. Bu, web sayfalarımıza esneklik bize ise hız kazandırır. Tablosuz tasarımın daha da önem kazandığı günümüzde CSS kullanımı olmazsa olmazlardandır.

CSS’nin Yapısı

CSS‟nin söz dizimi aşağıda görüldüğü gibi “Seçici (selector)” ve “Bildirim (declaration)” olarak iki ana bölümden oluşur. Bildirim ise kendi içinde özellik (property) ve değer (value) olarak iki kısma ayrılmaktadır.

HTML Sayfasına CSS Nasıl Eklenir?

Önceki yazılarımızda CSS Nedir ve CSS Seçiciler kavramına değinmiştik. Bu yazıda bir stil dosyasının HTML sayfasına nasıl ekleneceğini gösterceğiz.

Tarayıcı (browser) bir stil sayfasını okuduğunda, HTML belgesini stil sayfasındaki bilgilere göre biçimlendirir. CSS’ i tanımlandığı konuma göre üç gruba ayırabiliriz. Bunlar; **Harici (External) CSS** Stilleri, **Dahili (Internal) CSS** Stilleri ve **Satır içi(Inline) CSS** stilleridir.

Harici (external) CSS Stili:

Genellikle .css uzantısı kullanan dosyalardır. Harici olarak oluşturulan bu dosyada bulunan CSS kodları sitedeki Html sayfalarının baş kısmına **<head> ..</head>** bloğu içine bağlanır.

```
<head>
<link rel="stylesheet" type="text/css" href="stil.css">
</head>
```

Bağlandığı sayfanın sayfa özelliklerini, sayfada bulunan metinler, başlıklar ve diğer nesnelerin biçimlerini ve konumlarını belirler.

Herhangi bir metin editöründe harici stil sayfası yazılabilir. Dosya herhangi bir html etiketi içermemelidir. Stil sayfası dosyası bir .css uzantısı ile kaydedilmelidir.

Dahili (internal) CSS Stili:

Tek bir sayfanın benzersiz bir tarzı varsa, dahili stil sayfası kullanılabilir. Bir HTML belgesinin **<head> </head>** bloğu içerisine **style** etiketi ile tanımlanır.
CSS kodları HTML kodları içerisinde kullanıldığı için bu stil türüne Gömülü CSS Stilleride denir.

Dahili CSS Stilinin Harici CSS Stiline göre dezavantajı: Dahili CSS Stili sadece bulunduğu HTML sayfasını etkileyecektir. Sitede aynı biçime sahip olmasını istediğiniz web sayfaları için tekrar biçimlendirme kodlarını eklemeniz gerekecektir. Bir web sitesinde onlarca hatta yüzlerce sayfa olabileceğini düşünürsek bu iş zahmetli olacaktır. Ayrıca bir biçimi örneğin sitedeki başlıkların rengini değiştirmek istediğinizde harici stil dosyasında sadece .css dosyasındaki ilgili bölümü değiştirmeniz yeterli olacakken, dahili CSS kullanırsanız her bir sayfaya girerek ilgili biçim özelliğini değiştirmeniz gerekecektir.

Dahili CSS stili kullanımı:

```
<head>
<style>
body {
   background-color: linen;
}
h1 {
   color: maroon;
   margin-left: 40px;
}
</style>
</head>
```

Satır İçi (inline) CSS Stili:

Dahili CSS stilleri gibi HTML kodları içerisinde gömülü halde kullanılır. Ancak dahili stillerde olduğu gibi sayfanın baş tarafında tanımlanmaz. Dolayısıyla genel bir biçimlendirme sağlamaz.
Sayfa içinde tek bir öğe için benzersiz bir stil uygulamak için satır içi stil kullanılabilir. Bu stil türü pek tercih edilen bir yöntem değildir. Kullanımı aşağıdaki gibidir.

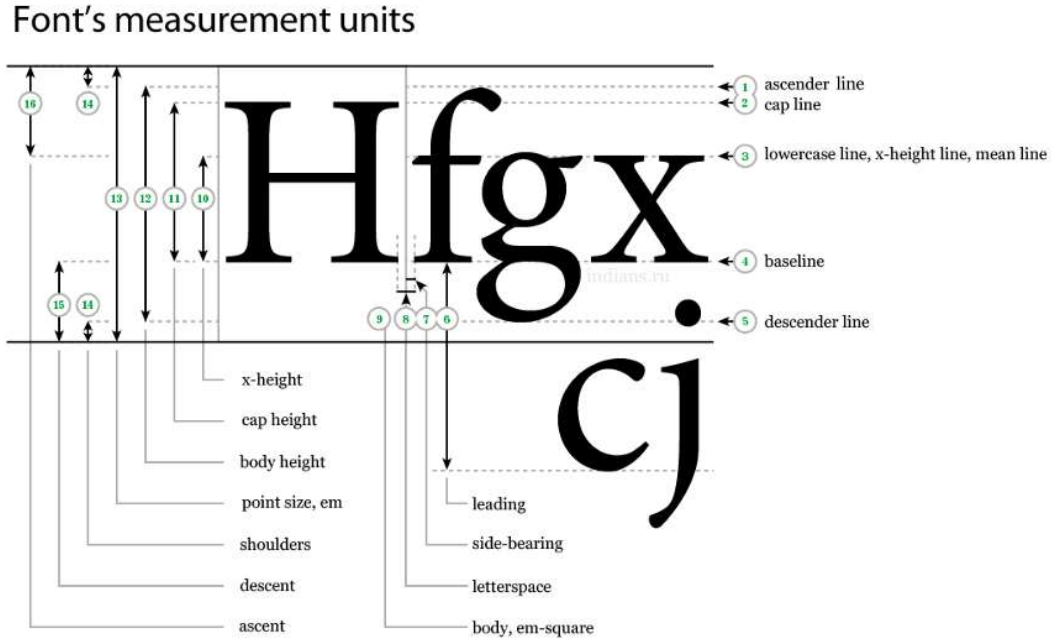
```
<h1 style="color:blue;margin-left:30px;">Örnek Başlık</h1>
```

Soru: Bir HTML öğesi için birden fazla stil belirtildiğinde hangi stil kullanılır?

Böyle bir durumda **satır içi stil** en yüksek önceliğe sahiptir ve harici ve dahili stilleri ve tarayıcı varsayılanlarını geçersiz kılacaktır. Sonra dahili, sonra harici stiller öncelik alır.

CSS' de Ölçü ve Uzunluk Birimleri

CSS içinde **uzunluk tanımlamaları** için birden fazla ölçü birimi kullanılmaktadır. Bunların bazıları matbaacılıktan, bazıları günlük hayatımızda kullandığımız ölçülerden gelirken bazıları ise bilişim teknolojilerinde özellikle **web sitelerinin** tasarlanmasında ve ölçülmesinde önemli role sahip birimlerdir.
Tüm CSS uzunluk birimleri, **pozitif** ya da **negatif** sayılar olarak ifade edilebilir, ancak bazı özellikler sadece pozitif sayıları geçerli görür (örneğin yazı tipi boyutu).



CSS de ölçü birimlerini genel olarak 2 kısıma ayırabiliriz:

1. Mutlak(Absolute) Ölçü Birimleri
2. Göreceli (Relative) Ölçü Birimleri:
    - Font Göreli Uzunluklar
    - Pencere Boyu veya Bakış Alanı Birimi (Viewport)

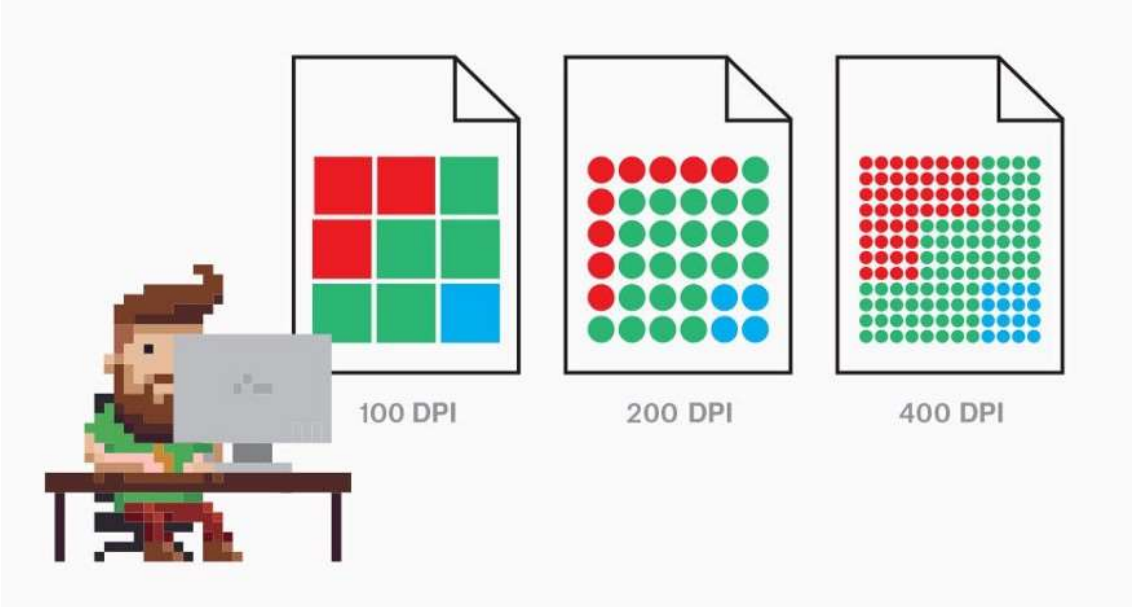
Tasarım yaparken çıkış ortamı göz önüne alınıp(ekran,baskı vb.) uygun birimlerin seçilmesi ise buradaki kilit nokta diyebiliriz. Yanlış bir kullanımda tasarımımızı efektif bir şekilde çıktıya yansıtamayabiliriz.

1) MUTLAK (ABSOLUTE) ÖLÇÜ BİRİMLERİ

Mutlak ölçü birimleri **(cm, mm, in, pt ,pc ve px*)** çıkış ortamı **baskı** olacak belgelerde kullanışlı ölçülerdir. Web tasarımı için **uygun değildir** (pixel istisnai bir durum). Belirli bir birimin aynı değerlerinin mutlak uzunlukları farklı ekranlarda farklılık gösterebilir. Bu, ekran **DPI** (inç başına nokta sayısı) farklılığından kaynaklanır. Daha yüksek çözünürlüklü ekranlar, daha küçük çözünürlüklü ekranlara kıyasla **daha yüksek bir DPI** değerine sahiptir ve bu nedenle görüntü veya metin daha **küçük** görünür.Bu sebeple, belgenin tamamını görmek için kullanıcının sürekli ekranı kaydırması gerekecektir. Bu da iyi bir kullanıcı deneyimi oluşturmaz.

- cm (santimetre)
- mm (milimetre)
- in (inç) / (1in = 96px = 2.54cm)
- pc (picas) / (1pc = 12 pt)
- pt (point) / (1pt = 1in'in 1/72'si)
- px* (piksel) / (1px = 1/96th of 1in)

Piksel (px) web tasarımında çokça tercih edilen bir birimdir. Pixel ile ilgili istisnai durumu açıklamak gerekirse, px cihazın görüntüleme biçimine göre değerlendirilir. Bu anlamda, low-dpi değerine sahip bir cihazda bir CSS pikseli (px) bir cihaz pikseline (dot) eşit diyebiliriz. Bu tarz bilgisayar ekranlarında 1 inç başına 96 nokta düşer. High-Res ekranlar ve printer gibi **yüksek çözünürlüğün** geçerli olduğu cihazlarda ise değer birden çok aygıt pikseline karşılık gelir.



2) GÖRECELİ (RELATIVE) ÖLÇÜ BİRİMLERİ

Göreceli CSS uzunluk birimleri sabit değerlere sahip değildir. Değerleri **önceden tanımlanmış** (predefined) başka bir değer veya özelliğe göredir. Göreceli birimler web tasarımında daha kullanışlıdır. Çünkü genişlik, yükseklik, yazı tipi boyutu vb. diğer bazı temel parametrelerle ilişkilendirebildiğimiz için öğeleri **düzgün şekilde** boyutlandırmayı kolaylaştırırlar.

Yazı Tipi Göreli(Bağıl) Uzunluklar

Yazı tipi göreli birimleri, önceden tanımlanmış yazı tipi boyutu veya yazı tipi ailesi değeri ile ilgilidir ve şunları içerir:

- **em**

Bu ölçü biriminde varsayılan font ailesinde *(font-family)* yer alan ***M* harfinin genişliği** ölçü için referans olarak alınır. **em** birimi, temel elementin (base element) veya üst elementin (parent element) yazı tipi boyutuna eşit bir değer alır. Örneğin, üst elementin yazı tipi boyutu 20px ise, 1em tüm alt elementler için 20px (20×1) olarak hesaplanır. Değerin bir tam sayı olmasına gerek yoktur. Yani 0,5em şeklinde de bir tanım yapmamız mümkündür. Bu durumda alacağımız birim karşılığı 10px olacaktır.
Unutulmaması gereken önemli nokta ise **em** değerinin **hemen üst etiketin** değerini almasıdır. Bu, iç içe geçmiş elemanlarda istenmeyen sonuçlara yol açabilir.
Örneğin iç içe geçmiş 3 elementimiz var. İlk eleman (root)20px font boyutunda, iç içe iki eleman ise fontu 2em olarak ayarlanmış. Kök öğenin içine yerleştirilen öğenin yazı boyutu 40 px (20 x 2) olurken iç öğenin içine yerleştirilmiş öğenin yazı boyutu 80 px (40 x 2) olur.

- **rem (root em)**

rem birimi **em** biriminden farklı olarak her zaman kök elementin (HTML belgelerinde **html** öğesidir.) değerini alır ve art arda sıralı **em** gibi kullanılmaz. rem birimi özellikle **duyarlı tasarım** sürecinde oldukça faydalıdır. rem ile HTML elementinin yazı tipini değiştirerek tüm sayfanın ölçeklendirilmesi sağlanabilir.

- **ex (x-height)**

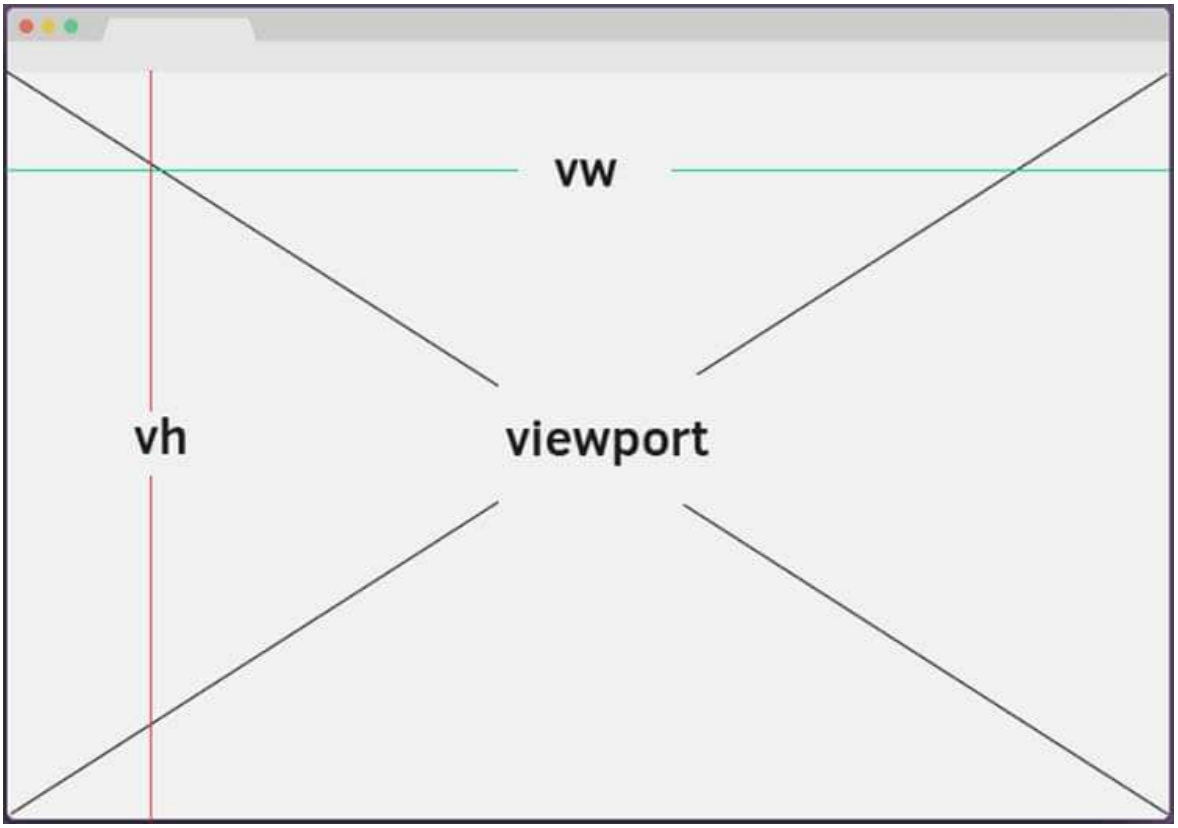
ex birimi **em** ile aynı mantıkta çalışır ,"geçerli yazı tipinin x yüksekliği" veya bir "**em**' in yarısı" olarak tanımlanır. Belirli bir fontun x yüksekliği ise, o fontun küçük harf x'inin yüksekliğidir. Yazı tipi ailesi değiştikçe değişir. Nadiren kullanılan bir birimdir.

- **ch (character)**

ch birimi ex birimine benzer ancak *küçük harf x* karakterinin yüksekliği yerine *0 (sıfır)* karakterinin **yüksekliğini** alır. Yine font ailesine bağlı olarak değişkenlik gösterir.

Pencere Boyu veya Bakış Alanı Birimi (Viewport)

Viewport, tarayıcı ekranın kullanıcı tarafından görünen kısmını ifade eder. HTML5, web tasarımcılarının < *meta* > etiketi ile görüntüleme alanı üzerinden kontrol sahibi olmamızı sağlayan bir yöntem geliştirmesi sayesinde bu yeni ölçü birimini CSS'e kazandırmış oldu.



Tasarımcılar için büyük kolaylık sağlayan bu ölçü birimi, artık javascript ile yapılan ekrana sığdırma gibi bazı işlemleri otomatik olarak yapmamıza olanak tanımaktadır.
İlk başta bakış alanını belirleyip sonra sayfa boyutunu değiştirdiğinizde buna göre yeni bir ölçeklendirme yapılır. Kullanıcı tarayıcı boyutunu her değiştiğinde hesaplamalar tekrar tekrar yapılır.
Viewport olarak kullanabileceğiniz ölçü birimleri *vh, vw, vmin* ve *vmax* 'dir.

- **vw (viewport width)**

vw , görüntü alanı genişliği birimidir. 1vw görünüm alanının 1/100'üne eşittir. Örneğin pencerenin genişliği 1000px ise 1vw 10px'e karşılık olacaktır. **vw** temel olarak yüzde ifadesine benzese de üst elementlerden (parent) veya üst elementlerin genişliğinden bağımsız işler.

- **vh (viewport height)**

vh, görüntü alanı yüksekliği birimidir ve **vw** ile benzer kurallarda işler, yani 1vh görünüm alanının 1/100'üne eşittir. Örneğin pencerenin yüksekliği 1000px ise 1vh 10px olacaktır.

vmin (viewport minimum)

vmin görüntü alanı yüksekliği (vh) ve görüntü alanı genişliği (vw) değerlerine bakar ve minimum olan değer üzerinden 1/100 olarak oranlar. Örneğin, 1900x1080px bir ekranda min değer 1080px olacaktır. Dolayısıyla 1vmin 10.8px olarak değerlendirilir.

- **vmax (viewport maximum)**

vmax tıpkı **vmin** gibi görüntü alanının yükseklik ve genişlik değerini inceler ancak bu sefer max olan değer üzerinden işlem gerçekleştirir. Yani, yukarıdaki örnek üzerinden ilerleyecek olursak 1900x1080px bir ekranda 1vmax karşılığı 19px olacaktır.

- **viewport meta etiketinin kullanımı**

<meta name="viewport" content="width=değer, height=değer, initial-scale=değer, minimum-scale=değer, maximum-scale=değer, user-scalable=değer, target-desityDpi=değer" />

- **% (percentages)**

Başka bir uzunluğa bağlı olarak verilen yüzde orantı birimdir. Örneğin sayfamızda 500px genişliğinde bir alan olduğunu düşünürsek ve içerisine eşit boylarda 4 kutu oluşturacaksak kutuların genişliklerine "width: 25%;" değeri veririz. Böylece kutuların mutlak boyutunu 125px değerine eşitleyebiliriz. Yüzde ölçülendirme birimi **responsive tasarım** anlayışının en temel ölçü birimidir.

Tarayıcı desteği

Son olarak bu ölçü birimlerinin tarayıcı desteği ise şu şekilde;

Length Unit					
em, ex, %, px, cm, mm, in, pt, pc	1.0	3.0	1.0	1.0	3.5
ch	27.0	9.0	1.0	7.0	20.0
rem	4.0	9.0	3.6	4.1	11.6
vh, vw	20.0	9.0	19.0	6.0	20.0
vmin	20.0	9.0*	19.0	6.0	20.0
vmax	26.0	Not supported	19.0	7.0	20.0