

PATH (Eriřim Yolu) nedir, nasıl kullanılır?

HTML de web sayfalarımızı düzenlerken sık sık başka dosyaları kullanma ihtiyacı duyarız. (Resim, css dosyası, js dosyası vs). Bu dosyaları sayfamıza eklemek için dosyanın bulunduđu yeri (adresini) tam olarak tanımlayamadığımızda ise sorun yaşarız.

Bazen de bir veb sayfasını bilgisayarımıza indirip açmaya çalıştığımızda karşımıza beklenmedik bir sorun çıkar ve sayfadaki fotoğraflar görünmediğı gibi CSS dosyası görevini yapamadığı için içeriğı düzenlenmemiş bir görüntüyle karşılaşırız. Bazılarımız arkadaşlarımızın github'a attığı çalışmalarını indirip bilgisayarımızda açmak istediğinde böyle bir akıbete maruz kalmıştır sanırım.

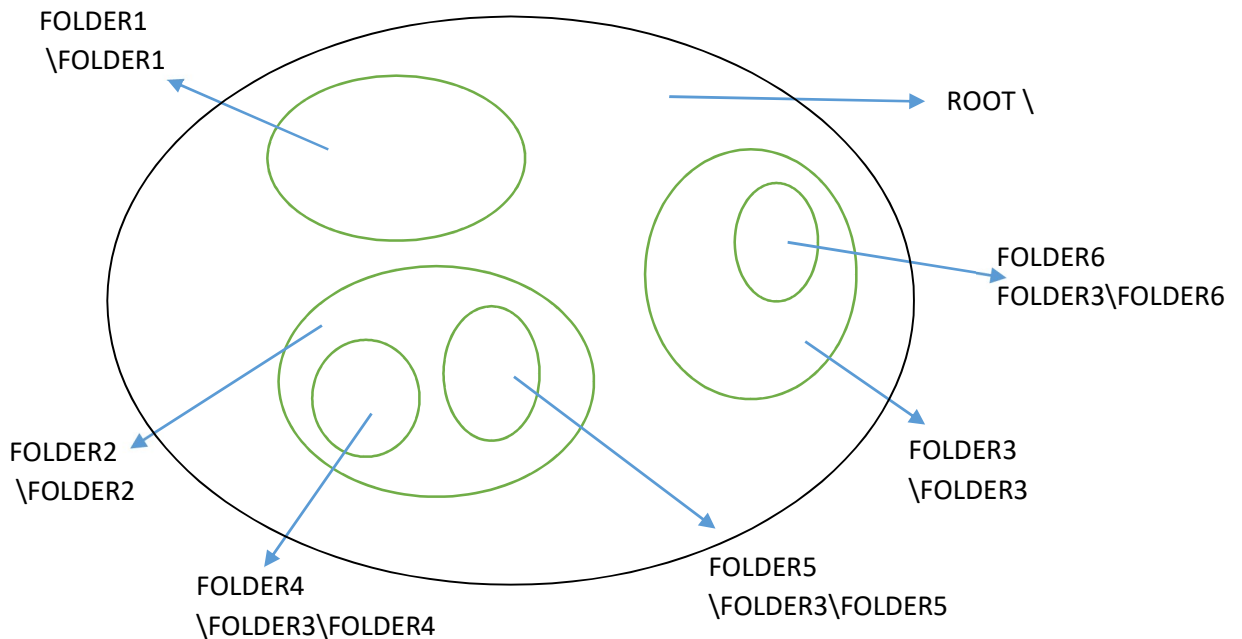
İşte burada PATH adını verdiğimiz kavram devreye giriyor. PATH (eriřim yolu) nedir, nasıl belirlenir?

Path (Eriřim Yolu), her işletim sisteminin kendine özgü olarak kullandığı, işletim sisteminden hedef dosyaya kadar varan yol haritasıdır. Path sayesinde bilgisayar içerisindeki dosyaların tümünün eşsiz olması sağlandığı gibi aynı zamanda dosyaların tam yerlerinin de bilinmesi mümkün olur.

Path tanımlaması işletim sistemlerine özgü olup farklılık göstermekle birlikte temelde aynı (benzer) metod ve yöntemleri kullanır. Biz de konuyu daha iyi anlayabilmek için referans noktası olarak Windows-DOS'taki PATH tanımlama formatını ele alacağız ve buradan yola çıkarak HTML'deki PATH kullanımını anlamaya çalışacağız.

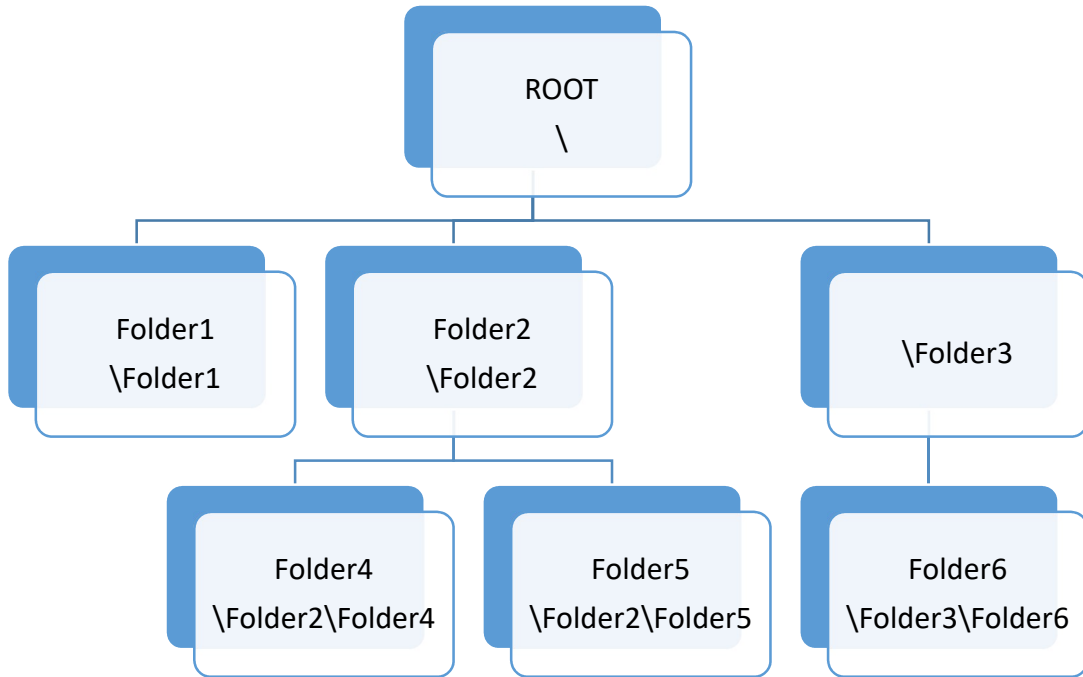
Dos işletim sisteminde diskler harflerle ifade edilir ve bunun bir disk olduğunu göstermek için ":" kullanılır. Örneğın C:, D:, E: gibi. Buradaki C: harddiskimizi, D: Cd sürücümüzü, E: de flash diskimizi tanımlar. (Diski tanımlayan harfler değışebilir. Yani bazen birden fazla diskiniz olduğunda C: ve D: disklerinizi tanımlar, CD Rom'unuzu tanımlamak için de F:, G: gibi başka harfler kullanılır.)

Disk Haritası



Her diskin ROOT adını verdiğimiz bir kök dizini vardır ve “\” ile ifade edilir. Yani disk içerisindeki tüm yol tanımlamalarının başlangıç noktası “\” tır. Kök dizini bütün dosya ve klasörlerinizi içerisinde barındıran büyük bir dolap veya klasör olarak farz edebilirsiniz. Bu durumda “C:\” yazdığımızda C: diskinin kök dizinini, “\” yazdığımızda ise o an üzerinde işlem yaptığımız diskin kök dizinini tarif etmiş oluruz. Yani yol tarifi vereceğimiz dosya HTML dosyamız ile aynı disk üzerinde bulunuyorsa “C:” ifadesini kullanmamıza gerek kalmaz, yol tarifine doğrudan “\” ile başlayabiliriz.

Kök dizin ya da kök klasör içerisine klasörler, o klasörler içerisine başka klasörler, onların içerisine de başka klasörler oluşturur ve dosyalarımızı bu klasörler içerisinde saklarız. Bu klasörler DOS işletim sisteminde “dizin (directory)”, Windows işletim sisteminde ise “klasör (folder)” olarak adlandırılır. Tüm bu birbiri içerisine geçmiş klasörler hiyerarşik bir yapı oluştururlar. Bu yapıyı “dizin ağacı (tree)” olarak adlandırıyoruz.



Sanırım konu biraz anlaşıldı. Dosyalarımızın erişim yolunu (PATH) yazarken bu hiyerarşiyi kullanırız. Örneğin Folder6 klasörü içerisinde bulunan “style.css” dosyamızın yolu bu durumda;

```
\\Folder3\\Folder6\\style.css
```

şeklinde.

Peki bu durum HTML’de web sayfası düzenlerken nasıldır?

Aslında arada çok fark bulunmuyor. Yapacağımız tek şey “\” karakterini “/” ile değiştirmekten ibaret. Yani bu dosyayı web sayfamıza taşıırken erişim yolunu (Path) şu şekilde yazıyoruz.

```
/Folder3/Folder6/style.css
```

Şimdi devam edelim. Tabii ki bu yol tarifi yalnızca sizin bilgisayarınıza ve o anki duruma özgü. Dolayısıyla düzenlediğiniz web sayfasını veya çalışmanızın içerisinde bulunduğu klasörü başka bir bilgisayara ya da başka bir klasöre taşıdığınızda yine

çalışmayacak, dosyalar yazdığınız adreste bulunamayacaktır. Çünkü oradan kaldırıp başka adrese taşıdınız. Doğal olarak ta erişim yolu (path) değişti.

Peki o halde ne yapacağız? Bu engeli aşmanın bir yolu var mı?

Evet var. Yol tarifimize kök dizinden (root) değil de bulunduğumuz klasörden başlamak. Öyle bir yol tarifi ki, içerisinde çalıştığımız klasörü hangi bilgisayara ya da hangi disk veya klasöre taşırsak taşıyalım, HTML sayfamız içerisinde tanımladığımız yollar istediğimiz dosyaları bulacak ve sistem tıkır tıkır çalışacak.

Klasör yapımızdan ve bu klasörler arasındaki hiyerarşiden bahsetmiştik. Bir klasör veya dosya adını herhangi bir yol tanımlamadan kullandığımızda, bilgisayarımız o dosya veya klasörü geçerli dizin (klasör) içerisinde arar. Yani “index.html” isimli bir dosyamız var ve bu dosya içerisine biz

```

```

tagı ile bir resim yerleştirdik. (Dikkat ederseniz osya adını herhangi bir yol tanımlaması yapmadan kullandım. Böyle bir durumda bilgisayar, “logo.png” adındaki resim dosyamı geçerli klasör, yani “index.html” dosyamın bulunduğu klasör içerisinde arayacaktır. Bulursa görüntüleyecek, bulamazsa doğal olarak görüntüleyemeyecek. Ama bizim dosyamız, “index.html” dosyamızın, yani web sayfamızın bulunduğu klasör içerisinde oluşturulmuş “img” klasörü içerisinde. Ve bulamadı. O halde yol tarifimizi

```

```

olarak yapıyoruz. Dikkat ettiyseniz dosya yolu yazarlar başlangıçta “/” kullanmadım. Çünkü başlangıçta kullanılan “/”, kök dizini sembolize eder ve aramanın kök dizinden başlayacağını belirtir. Halbuki ben bilgisayarın dosyamı aramaya kök dizinden (root) değil, içerisinde bulunduğum klasörden başlamasını istiyorum. Bu nedenle de yol tanımıma “/” ile değil de, doğrudan “logo.png” dosyamın içerisinde bulunduğu “img” klasörünün adını yazarak başladım. Peki “index.html” dosyama CSS adında başka bir klasörde bulunan “style.css” dosyasını eklemek isteseydim? Tabii ki

```
<style type="text/css" rel="CSS/style.css"></style>
```

Şeklinde yazmam gerekecekti. Yol tarifinde kullanılan bir diğer yöntemimiz ise “..”. “../” ile başlayan bir yol tarifi, bilgisayarımızın aramaya geçerli dizin/klasörden bir önceki klasörden başlamasını sağlar. Bu işlemi geriye gitmek istediğiniz klasör sayısı kadar çoğaltabiliriz.

```
../style.css
```

```
../../CSS/style.css
```

Peki “/” ile başlayan dosya yolumuz VSCode altında çalışıyor da, index.html dosyamızı doğrudan üzerine tıklayarak açtığımızda çalışmıyor? Konu fazla uzadı için bu konuyu başka bir makalede ele alacağım.

Görüşmek üzere...