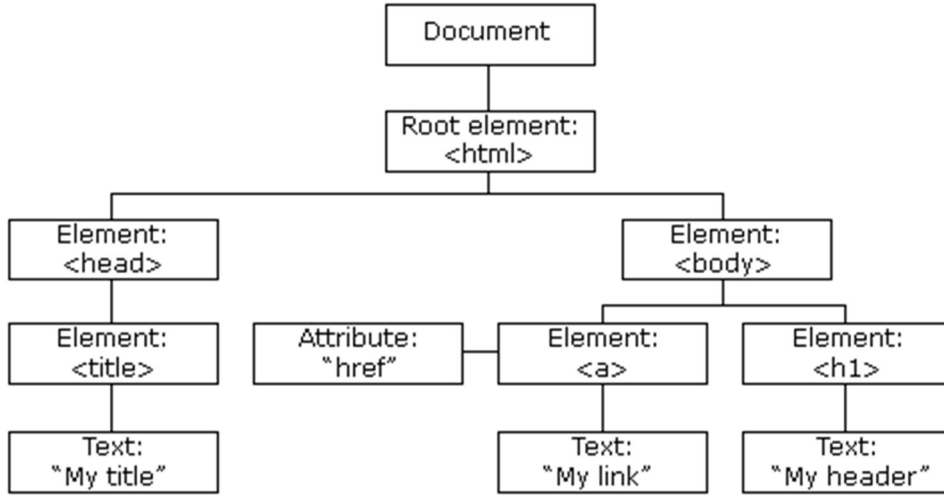


# DOM Nedir & Tarayıcılarda HTML Render İşlemi Nasıl Gerçekleşir?

DOM, bir W3C (World Wide Web Consortium) standardıdır. Programların ve komut dosyalarının bir belgenin içeriğine, yapısına ve stiline dinamik olarak erişmesine ve bunları güncellemesine olanak tanıyan, platform ve dilden bağımsız bir arabirimdir.

Bir web sayfası yüklendiğinde, tarayıcı sayfanın DOM ağacını oluşturur. Bu sayede JavaScript gibi betik diller ile sayfadaki tüm HTML özelliklerini değiştirebilir, mevcut HTML öğelerini ve nitelikliklerini kaldırabilir, yeni HTML öğeleri ve niteliklerini ekleyebilir veya sayfadaki tüm CSS stillerini değiştirebiliriz.



## DOM Ağacı Nedir?

DOM ağacı, düğümleri HTML belgesinin içeriğini temsil eden bir ağaç türüdür. Her HTML belgesinin benzersiz bir DOM ağacı temsili vardır. Örneğin aşağıdaki kodu inceleyelim:

```
<html>

<head>

  <title>My document</title>
```

```
</head>

<body>

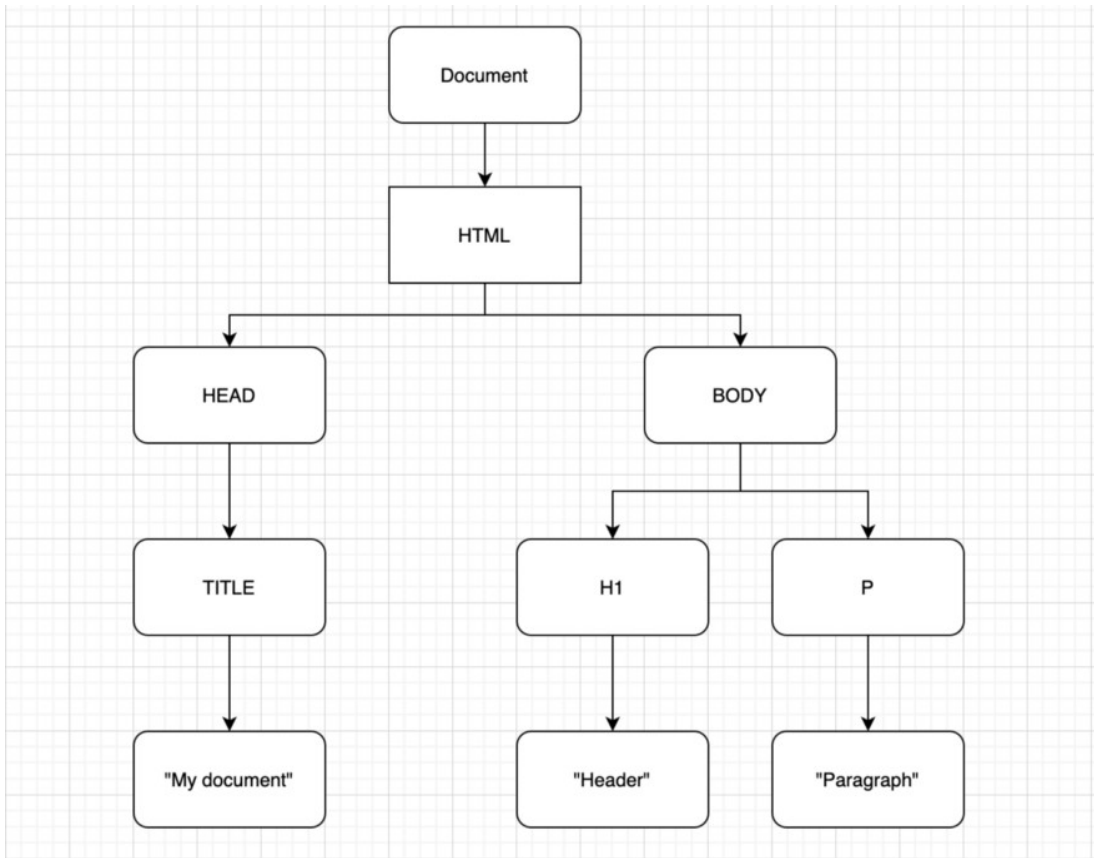
  <h1>Header</h1>

  <p>Paragraph</p>

</body>

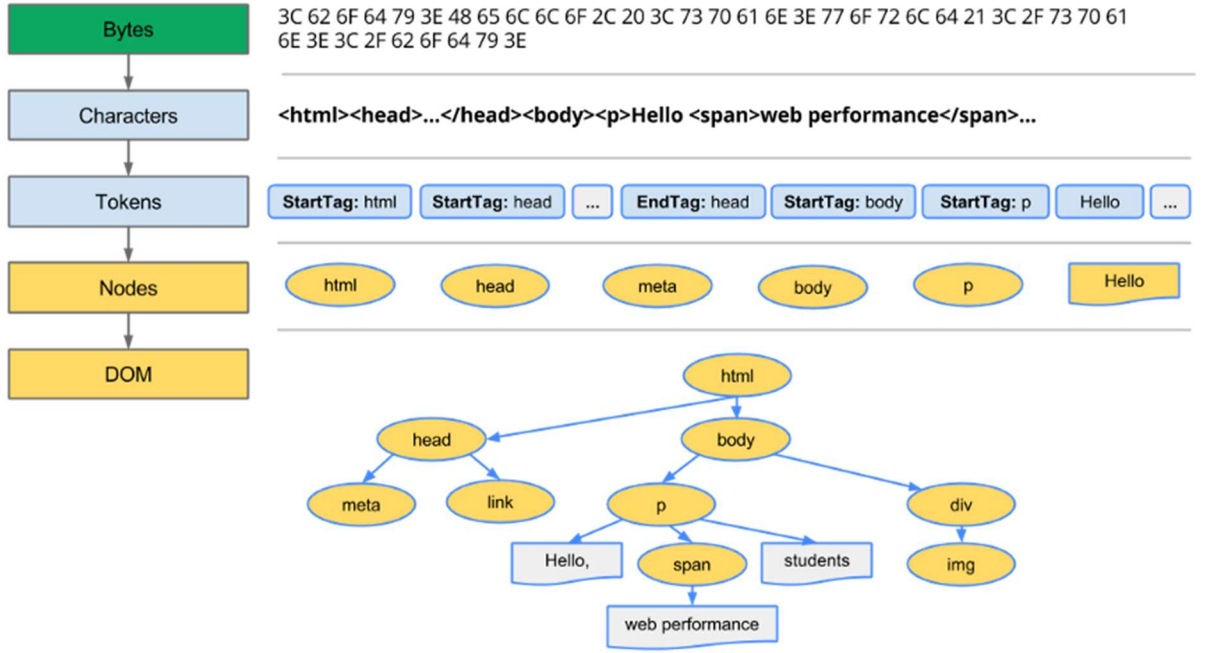
</html>
```

Yukarıdaki kod bloğunun DOM ağacı aşağıdaki gibidir:



## Tarayıcılar HTML'yi Nasıl Render Eder?

İnternet üzerinden veriler byte paketleri olarak gönderilir. Tarayıcı ise bu veri byte'larını anladığı bir forma dönüştürmelidir. İlk olarak byte'lar HTML karakterlerine daha sonra ise Token'lara dönüştürülür. Sonraki adımda ise Token'lar düğümlere (nodes) dönüştürülür. Düğümler belirli özelliklere sahip farklı nesnelerdir. Düğümler oluşturulduktan sonra DOM ağacı oluşturulur.



DOM ağacı oluşturulurken ilk önce “document” düğümü oluşturulur. Document tüm HTML belgesini temsil eden bir düğümdür. Bir HTML etiketlerini belirten düğümlere ise “elements” denir. DOM ağacındaki herhangi bir düğümün türünü “nodeType” ile öğrenebiliriz. NodeType özelliği bir sayı döndürür. Bu sayının hangi düğüm tipini temsil ettiğini bulmak için NodeType komutunu aratıp inceleyebilirsiniz.

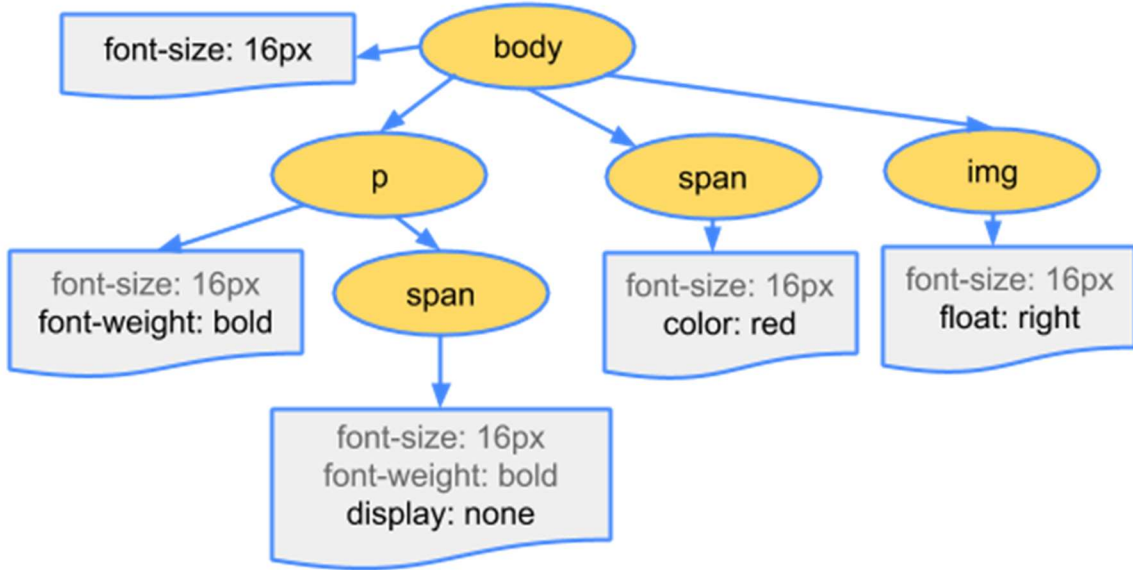
```
> document.nodeType
9
```

DOM ağacı başarıyla oluşturuldu ancak tarayıcının sayfayı oluşturabilmesi için öğelerin nasıl görüneceğine dair bilgilere ihtiyacı vardır. Bir sayfanın öğelerinin nasıl görünmesi gerektiği bilgisi CSSOM sorumluluğundadır.

## CSSOM Nedir?

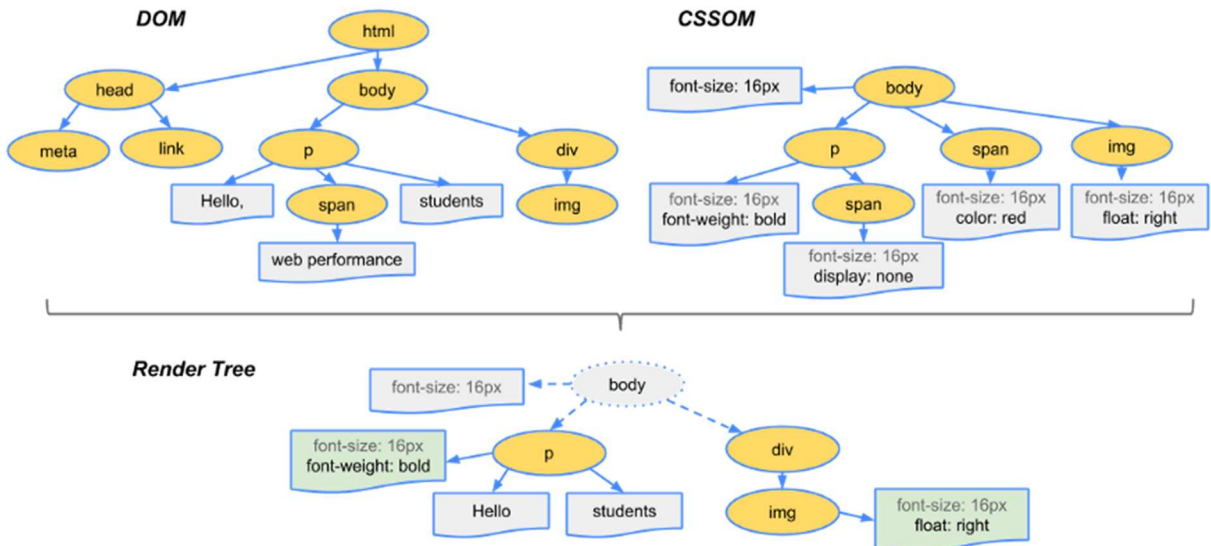
DOM ağacı oluşturulurken <head> içinde yer alan CSS bağlantısına bir istek gönderilir ve bu isteğin sonucunda CSS stilleri döner. HTML etiketlerinde olduğu gibi CSS bilgileri de byte’larla gelir ve belli aşamalardan geçerek CSS Nesne Modeli(CSSOM) oluşturulur.

CSS byte'ları karakterlere, daha sonra token'lara ve düğümlere dönüştürülür; son olarak da CSS Nesne Modeli veya kısaca CSSOM olarak bilinen bir ağaç yapısı oluşturulur.



## Render Tree Nedir?

DOM ve CSSOM ağaç yapıları iki bağımsız yapıdır. DOM, sayfanın HTML ögesinin ilişkileri hakkındaki tüm bilgileri içerirken, CSSOM, öğelerin nasıl şekillendirileceğine ilişkin bilgileri içerir.



Render Tree, sayfadaki tüm görünür DOM içeriğiyle ilgili bilgileri ve farklı düğümler için gerekli tüm CSSOM bilgilerini içerir. Bir öge CSS tarafından gizlenmişse (örneğin, display: none), düğümün Render Tree’de temsil edilmeyeceği anlamına gelir.

Gizli öge (örneğin, display: none), DOM’de bulunur, ancak Render Tree içinde bulunmaz. Bunun nedeni, Render Tree’nin hem DOM’dan hem de CSSOM’dan gelen bilgileri birleştirmesidir.

Tarayıcı Render Tree oluştururken öncelikle DOM ağacının kökünden başlayarak görünür her bir düğümü işler. (script, meta gibi etiketler Render Tree içinde yer almazlar ayrıca CSS ile gizlenen düğümler de Render Tree içinde yer almazlar)

## **Layout ve Paint Adımları**

Render Tree ile birlikte ekranda gösterilecek olan tüm içeriğe ve stil bilgilerine sahibiz ancak ekranda henüz bir görüntü oluşmadı. İlk olarak, tarayıcının sayfadaki her nesnenin tam boyutunu ve konumunu hesaplaması gerekir.

Her bir nesnenin kesin boyutunu ve konumunu belirlemek için tarayıcı, Render Tree’nin kökünden başlar ve sayfadaki her bir nesnenin hesaplamasını yapar. Bu hesaplama sonucunda her bir ögenin kesin konumu ve boyutu belirlenmiş olur. Bundan sonraki aşamada ise tarayıcı düğümleri tek tek ekrana boyar.

