

Modern Web Geliştirmeye Giriş - HTML'in Temelleri

1. Web'in Yapı Taşı HTML

Bir web sitesini ziyaret ettiğinizde tarayıcınızın görüntülediği her şeyin temelinde **HTML** (**HyperText Markup Language - Üst Metin İşaretleme Dili**) bulunur. HTML, bir programlama dili değil, bir **işaretleme dilidir**. Görevi, bir web sayfasının içeriğini yapılandırmak ve bu içerin in ne anlamına geldiğini (örneğin, bu bir başlık, bu bir paragraf, bu bir resim) tarayıcıya bildirmektir.

Modern web geliştirme üç temel teknolojinin ayrılmaz bir bütünüdür:

- **HTML (Yapı)**: Web sayfasının iskeletini oluşturur. İçeriği anlamsal olarak organize eder.
- **CSS (Sunum)**: Sayfanın görsel tasarımını, renklerini, yazı tiplerini ve yerleşimini kontrol eder. HTML iskeletini giydirir.
- **JavaScript (Davranış)**: Sayfaya etkileşim, dinamik fonksiyonlar ve karmaşık özellikler ekler. Eve elektrik ve su tesisatı dösemek gibi, onu "yaşanabilir" kılar.

Bu derste, bu üçlünün temel taşı olan HTML'ye odaklanacağız.

2. Bir HTML Belgesinin Anatomisi

Her HTML belgesi, tarayıcının sayfayı doğru yorumlayabilmesi için belirli bir standart yapıya sahiptir. İşte en temel HTML5 sayfası:

```
HTML

<!DOCTYPE html>
<html lang="tr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sayfa Başlığı</title>
</head>
<body>
    <h1>Ana Başlık</h1>
    <p>Bu bir paragraftır.</p>
</body>
</html>
```

Bu yapıyı inceleyelim:

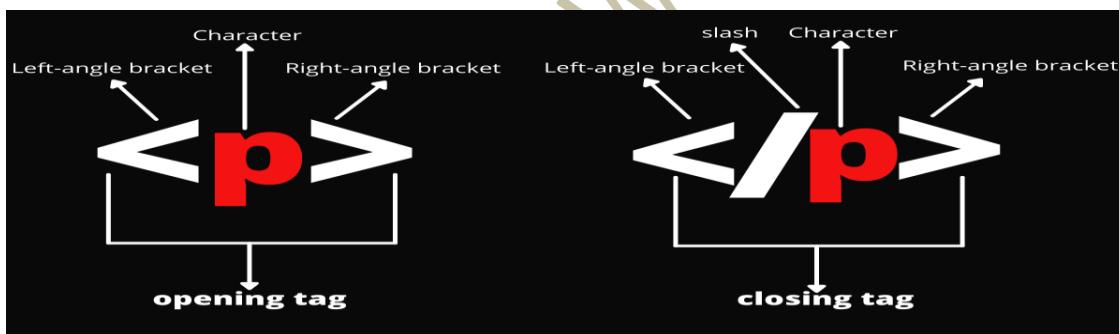
- **<!DOCTYPE html>**: Bu bir etiket değildir. Tarayıcıya bu belgenin modern bir HTML5 belgesi olduğunu bildiren bir talimattır.
- **<html lang="tr">**: Tüm HTML içeriğini kapsayan kök öğedir. lang="tr" niteliği, sayfa dilinin Türkçe olduğunu belirterek erişilebilirlik ve arama motorları için önemli bir bilgi sağlar.

- <head>: Sayfa hakkında tarayıcıya ve arama motorlarına yönelik "meta" bilgileri içerir. Bu bölümdeki içerik doğrudan sayfada görüntülenmez.
 - <meta charset="UTF-8">: Belgenin karakter kodlamasını belirtir. UTF-8, Türkçe karakterler de dahil olmak üzere evrensel karakter setini destekler ve modern web için standarttır.
 - <meta name="viewport" ...>: Sayfanın farklı ekran boyutlarında (mobil, tablet, masaüstü) düzgün görüntülenmesini sağlayan, duyarlı (responsive) tasarım için kritik bir meta etiketidir.
 - <title>: Tarayıcı sekmesinde görünen ve arama sonuçlarında kullanılan sayfa başlığını tanımlar.
- <body>: Başlıklar, paragraflar, resimler, listeler gibi kullanıcının ekranda gördüğü **tüm içeriğin** yer aldığı bölümdür.

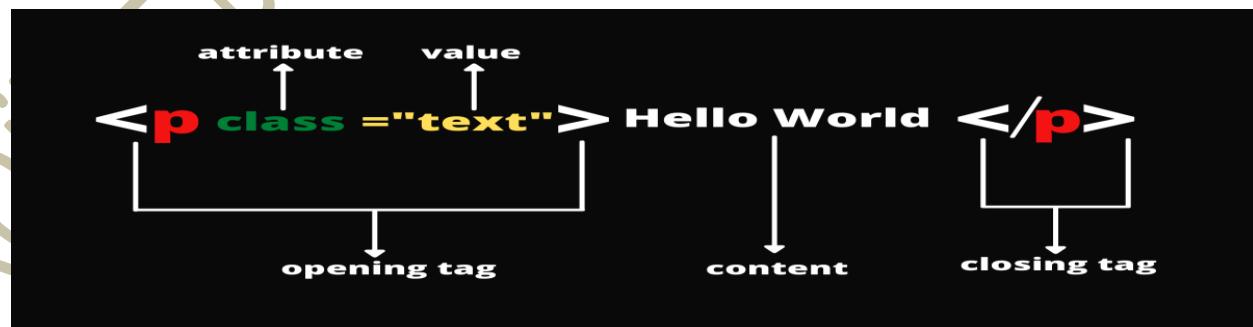
3. HTML'in Yapı Taşları: Öğeler, Etiketler ve Nitelikler

HTML belgeleri **öğeler (elements)** kullanılarak oluşturulur. Bir öğe genellikle üç bölümden oluşur:

Aşağıdaki resim bir HTML etiketinin anatomisini açıklamaktadır:



1. Açılmış Etiketi (Opening Tag): Öğenin nerede başladığını belirtir (ör. `<p>`).
2. İçerik (Content): Öğenin kapsadığı metin veya diğer öğeler.
3. Kapanış Etiketi (Closing Tag): Öğenin nerede bittiğini belirtir (ör. `</p>`).



Örnek: <p>Bu metin bir paragraf ögesinin içeriğidir.</p>

- **Etiket (Tag):** < ve > karakterleri arasında yazılan ve öğeyi tanımlayan anahtar kelimelerdir (<p>, <h1>, <body> gibi).
- **Öğe (Element):** Açılmış etiketi, içerik ve kapanış etiketinin tamamına verilen isimdir.

Bazı öğeler içerik barındırmaz ve kapanış etiketine sahip değildir. Bunlara **boş öğeler (void elements)** denir.

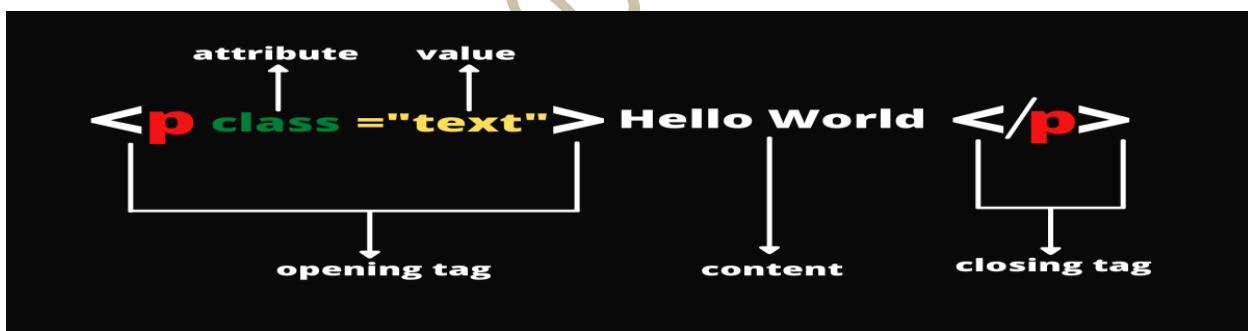
Örnek:

- : Bir resim ekler.
-
 : Bir satır sonu ekler.
- <hr> : Tematik bir ara (yatay çizgi) ekler.

Nitelikler (Attributes)

Nitelikler, HTML öğelerine ek bilgi sağlamak için kullanılır ve **her zaman açılış etiketinin içine** yazılırlar. isim="değer" formatında kullanılırlar.

- Google'a git: <a> (anchor) ögesi bir bağlantı oluşturur. href niteliği, bağlantının hedefini belirtir.
- : ögesi bir resim gömer. src niteliği resmin kaynak yolunu, alt niteliği ise resim yüklenemediğinde veya ekran okuyucular için alternatif bir metin sağlar. [alt niteliği, web erişilebilirliği için zorunludur.](#)



4. Semantik HTML: "Ne" Olduğunu Söyle, "Nasıl" Göründüğünü Değil

HTML'in ilk yıllarda, etiketler genellikle içeriğin nasıl görüneceğini tanımlamak için kullanılıyordu (kalın, <i> italik gibi). Ancak modern web geliştirmenin temel prensibi endişelerin **ayrılığıdır (separation of concerns)**. Bu ilkeye göre:

- HTML, sadece yapıyı ve anlamı tanımlamalıdır.
- CSS, görünümü ve stili tanımlamalıdır.

HTML5 ile birlikte, içeriğin *anlamını* tanımlayan **semantik etiketler** standart haline gelmiştir. Sadece <div> gibi anlamsız kapsayıcılar kullanmak yerine, sayfanın bölgelerini anlamlarına göre

etiketlemek en iyi pratiktir.

Başlıca Semantik HTML5 Öğeleri:

- <header>: Sayfanın veya bir bölümün başlığını, logosunu, navigasyonunu içeren tanıtıcı alanı.
- <nav>: Ana navigasyon menüsü, site içi bağlantıları barındıran bölüm.
- <main>: Sayfanın ana ve benzersiz içeriğinin bulunduğu alan. Bir sayfada yalnızca bir tane olmalıdır.
- <article>: Kendi başına bir bütün olan ve başka bir yerde de kullanılabilen içerik (ör. blog yazısı, haber makalesi).
- <section>: Belgenin tematik olarak gruplandırılmış bir bölümü (ör. "Hakkımızda", "Hizmetlerimiz").
- <aside>: Ana içerikle dolaylı olarak ilgili olan içerik (ör. kenar çubuğu, ilgili linkler).
- <footer>: Sayfanın veya bir bölümün alt bilgisi (telif hakkı, iletişim bilgileri vb.).

Neden Semantik HTML Kullanmalıyız?

1. **Erişilebilirlik (Accessibility):** Ekran okuyucu gibi yardımcı teknolojiler, sayfayı daha iyi anlar ve görme engelli kullanıcılarla daha anlamlı bir deneyim sunar.
2. **SEO (Arama Motoru Optimizasyonu):** Google gibi arama motorları, içeriğinizin ne hakkında olduğunu daha iyi anlar ve sitenizi daha doğru bir şekilde sıralar.
3. **Bakım Kolaylığı:** Kod, hem kendiniz hem de diğer geliştiriciler için daha okunaklı ve anlaşılır hale gelir.

5. HTML'in Evrimi: Sürümelerden Yaşayan Standarda

HTML, web'in başlangıcından bu yana sürekli bir gelişim içindedir.

- **HTML 1.0 - 4.01 (1992-1999):** Web'in temelleri atıldı. Bu sürümler, bugünkü standartlara göre oldukça basitti ve sunum ile yapıyı sıkça bir arada barındırıyordu (, <center> gibi artık kullanılmayan etiketler içeriyordu).
- **XHTML:** HTML'i daha katı XML kurallarına dayandırma girişimi idi ancak geniş çapta benimsenmedi.
- **HTML5 (2014):** Web'de devrim yarattı. Multimedya (<audio>, <video>), grafik (<canvas>, <svg>) ve en önemlisi yukarıda bahsedilen semantik öğeleri standart hale getirdi. Eklentilere olan bağımlılığı azalttı.

Günümüzün Standarı: HTML Living Standard

Artık "HTML 5.1, 5.2" gibi numaralandırılmış sürümlerden bahsetmek teknik olarak doğru değildir. Web'in hızlı gelişimine ayak uydurmak için HTML standarı, **WHATWG (Web Hypertext Application Technology Working Group)** tarafından sürdürulen bir "**"Yaşayan Standart (Living Standard)"**" haline gelmiştir. Bu, standardin sürekli olarak güncellendiği ve tarayıcıların bu yaşayan standarı uyguladığı anlamına gelir. Kısacası, bugün kullandığımız HTML, her zaman en güncel versiyondur.

6. Modern Geliştirme Araçları ve Ekosistemi

HTML yazmak için teorik olarak Notepad gibi basit bir metin düzenleyici yeterlidir. Ancak modern geliştiriciler, verimliliği artıran profesyonel araçlar kullanır:

- **Kod Editörleri ve IDE'ler:**
 - **Visual Studio Code (VS Code):** Günümüzün endüstri standartı. Ücretsiz, güçlü, eklentilerle genişletilebilir.
 - **WebStorm:** JetBrains tarafından geliştirilen, özellikle büyük JavaScript projeleri için güçlü, ücretli bir IDE.
 - **Sublime Text:** Hızlı ve hafif bir kod editörü.
- **Tarayıcı Geliştirici Araçları (Browser DevTools):**
 - Her modern tarayıcıda (Chrome, Firefox, Edge) bulunan bu araçlar (genellikle F12 tuşu ile açılır), bir web sayfasının HTML yapısını canlı olarak incelemenize, CSS kurallarını test etmenize ve JavaScript hatalarını ayıklamanıza olanak tanır. Her web geliştiricisinin en temel aracıdır.
- **Sürüm Kontrol Sistemleri:**
 - **Git:** Proje kodunuzdaki değişiklikleri takip etmenizi sağlayan sistemdir. **GitHub** ve **GitLab** gibi platformlar, Git tabanlı projeleri barındırmanız ve başkalarıyla iş birliği yapmanıza olanak tanır.