Başlık Etiketleri Kullanımı

Efendim öncelikle HTML'in ne olduğunu ve en yaygın etiketleri öğrendiniz. Şimdi geldik yavaş yavaş biraz daha bu güzel betik dilinde ilerlemeye. Bu yazıda <head></head> etiketi arasına yazılan etiketlerden bahsedeceğiz.

Bildiğiniz üzere balık baştan kokar derler. Başlık etiketlerimiz de benzer şekilde bize sayfamızla ilgili ipuçları verir ve sayfamız için ihtiyacımız olacak dış kaynakları yüklememize yardımcı olur. Baklavayı ortasından yemeyi sevenler için biraz daha detay vermek gerekirse, hem sayfamızın düzgün ve verimli çalışması için hem de bu sayfanın tanınmasını bilinmesini hatta arama motorları tarafından fark edilmesini sağlamak için bu etiketlere ihtiyacımız var.

Öyleyse sırasıyla etiketlerimize başlayalım ve html sayfalarımızı nasıl daha bilinir ve gelişmiş hale getirebiliriz öğrenelim.

Title Etiketi

Bu etiketimiz html dökümanlarında türkçe anlamı olan "başlık" görevini üstlenir. Bu başlık birkaç farklı yerde görülebilir. Bizim en sık karşılaştığımız şekli ise browser sekmelerinin isimleridir. Örnek olarak :

<title> Kodluyoruzla Web Öğreniyorum </title>

Gibi bir başlık belirlediğimizde sekme isminde Kodluyoruzla Web Öğreniyorum yazdığını görürüz.

Bu etiket arasına yazdıklarımız aynı zamanda sayfayı favorilere eklerken de karşımıza çıkar.

Ayrıca arama motorları (Google, Yahoo, Bing ...) sayfamızın bu kısmına bakarak sitemizi listeler. Bu yüzden de oluşturduğumuz mükemmel web sayfalarının insanlara ulaşması için bu alanda yazdıklarımızın açıklayıcı, dikkat çekici ve 50-60 karakteri geçmeyecek şekilde ( Arama motorları genelde başlığın ilk 50-60 karakterini gösterir ) yazmamız gerekir.

Style ve Script Etiketleri

HTML dökümanları oluştururken en çok kullanacağımız etiketler bunlardır. Bu etiketler yalnızca <head></head> etiketleri arasında bulunmaz ancak biz burada bulunabilecek bazı özel türleri ve bu tür etiketlerin özelliklerini (attribute) inceleyeceğiz.

Style Etiketleri

Öncelikle style etiketinden başlayalım. <style></style> etiketleri arasında sayfamızı güzelleştiren, renklendiren belli özellikler tanımlayabiliyoruz. Bu kısımlarda, bir html dökümanında hangi alanın nerede ve nasıl görünmesi gerektiğini tasarlayabiliriz. Belli kuralları olan bu belirteçlere CSS diyoruz. Sayfa tasarımlarının bulunduğu bir ön döküman veya bir taslak gibi düşünebiliriz. Daha CSS ile daha detaylı bilgi için şurayı inceleyebilirsiniz : <https://www.w3schools.com/css/default.asp>

Burada dikkat etmemiz gereken bir konu var. HTML dökümanı işlenirken ve görüntülenirken sayfa sırayla işlendiği için her zaman sırasıyla en altta kalan stil belirlemeleri baskın gelecektir.

Bu etiketimizin global özelliklere(attribute) ek olarak alabildiği iki farklı özellik vardır. Bunlar media ve type. Çok yaygın olmadığı için kullanım detaylarına girmeyeceğim ama alabildikleri özelliklere şuralardan bakabilirsiniz : <https://www.w3schools.com/tags/att_style_media.asp>

Script Etiketleri

Geldik web dökümanlarının ön yüzlerini "sihirli" hale getiren etikete. Bu etiketle web sayfalarının, browser yardımıyla çalıştırabildiği kodlar yazabiliriz. Sayfamızı canlandırabilir, hareketlendirebilir her alanda değişiklik yapabiliriz. Sayfamız üzerinde hayal gücümüzle sınırlı her değişikliği yapabilmemizi sağlayan kodlar bu etiketler arasında bulunur.

<script></script> etiketleri ileride öğreneceğiniz sihirli bir dünyaya açılan kapıdır. Normalde HTML dökümanlarının bir programlama dili ile yazılmadığını biliyoruz. Çünkü HTML ve CSS bir programlama dili değildir. Ancak<script></script> tagları arasına girdiğimizde işler değişmeye başlıyor. Artık bir programlama dili olan JavaScript'in dünyasına girmiş oluyoruz. HTML dökümanlarının stilleri yerleşimleri hatta bütün dökümanın kendisini Javascript yardımıyla değiştirebilir, farklı işlemler yapabilir, farklı sayfalarla veya arka planda bir veri tabanıyla haberleşebilir oradan aldığımız bilgilerle dökümanımızı güncelleyebiliriz. Ayrıca HTML dökümanımıza yeni elemanlar ekleyip çıkartabiliriz. Dökümanlarımız üzerinde hareketli animasyonlar çalıştırabilir, her türlü değişikliği yapabiliriz.

Bu yüzden bu etiketlerin bizim gücümüzü artıran sihirli bir dünyaya açıldığını söyleyebiliriz. Şimdi bunun özelliklerine bakalım.

Script Tag Özellikleri

Öncelikle bu etiketlerin bize sayfa üzerinde büyük bir güç verdiğinden bahsetmiştik. Ancak Spider Man Peter Parker'ın rahmetli amcasının da dediği gibi "with great power comes great responsibility" yani büyük güç büyük sorumluluk gerektirir. Bu yüzden bu tagın özelliklerinden bahsederken sayfanın işlenişi, browser tarafında nasıl işlendiğini ve yanlış bir kullanımın nelere sebep olabileceğini iyi anlamamız gerekiyor. Eğer script etiketini kullanırken herhangi bir özellik eklemezsek browser sırası geldiğinde doğrudan işlenir. Ve bu kısım işlenmeden sayfa yüklenmeye devam etmez. Bu noktada da *async* özelliğimiz devreye giriyor. Eğer sayfanın yüklenmeye devam ederken eşzamanlı olarak bu etiketlerle belirlediğimiz scriptlerin de yüklenmesini ve hazırlanmasını istiyorsak, yani bu kısmın asenkron çalışmasını istiyorsak etiketimize bu özelliği ekliyoruz. Herhangi bir değer girmemize gerek yok şu şekilde kullanabiliriz :

<script src="myJavascript.js" async></script>

Eğer bu etiketin sayfa yüklendikten sonra yüklenip çalıştırılmasını istiyorsak o zaman async özelliğinin yanına *defer* özelliğini de eklememiz gerekiyor. Ancak bu iki özellik de yalnızca sayfa harici kaynaktan yani bu HTML içinde yazmadığımız javascripti yüklerken kullanabileceğimiz özellikler. Buna da dikkat etmemiz gerekiyor.

Bu etiketin sihirli dünyasından bahsetmiştik. Ancak bahsetmediğimiz nokta her sihirli dünyada olduğu gibi burada da kötü büyücülerin, kötü insanların olduğudur. Bu yüzden sayfamız üzerinde bu kadar güce sahip bir alanın doğru şekilde korunması ve kullanılması hayati önem taşıyor. İşte bu amaçla da karşımıza iki özellik çıkıyor.

Bu özelliklerden birincisi crossorigin. Browserlar, istek sahteciliği gibi güvenlik sorunlarıyla aktif şekilde mücadele etmeye çalışıyor. Bu yüzden bir kaynaktaki dökümanın bir diğer kaynaktaki (farklı domain) dökümanlara erişmesinde biraz hassas davranıyorlar. Bu konuya cross origin resource sharing deniyor kısaca CORS diyebiliriz. İşte bu etiketimiz de farklı kaynaklardaki, farklı domainlerdeki scriptleri yüklememiz için bize yardımcı oluyor. Eğer bir kaynaktan(aynı domain dahil) bir şey yüklemek için belli bilgileri( Çerezlerimiz olabilir, HTTP basit giriş bilgileri olabilir) göndermemiz gerekiyorsa bu özelliğin değerini crossorigin = "use-credentials" olarak belirlemeliyiz. Eğer anonim şekilde erişmemiz gerekiyorsa yani herhangi bir bilgiye ihtiyaç yoksa crossorigin="anonymous" olarak kullanacağız.

Bir diğer özelliğimiz ise integrity özelliğidir. Integrity türkçeye bütünlük, doğruluk, dürüstlük şeklinde çevirilebilir. Webin gelişmesiyle birlikte bir HTML sayfasına yüklenen kaynaklar çoğaldı. Özellikle tekrar eden ihtiyaçlar için zaman geçtikçe en verimli çözümler üretilmeye ve kullanılmaya başlandı. Bu çözümlerin kullanılması yaygınlaştıkça belli riskler de ortaya çıkmaya başladı. Örnek olarak HTML sayfamıza ekleyeceğimiz bir dış script bir güvenlik açığıyla karşı karşıya kaldığında o scripti kullanan bütün sayfalar aynı anda etkilenmiş olacak. Bunun bir örneği yakın zamanda birçok firmayı etkileyen Gigya servisinde olmuştu. Bu servisi kullanan sitelerin sayfalarına ekledikleri harici bir gigya servisi bir saldırıya maruz kalmıştı ve o servisi kullanan binlerce sitede bu açıktan etkilenmişti. İşte CDN dediğimiz bu gibi hazır scriptleri eklediğimiz durumlarda bir doğrulama yöntemine ihtiyaç duyuyoruz. Yani browser bir şekilde, sayfamızı etkileyecek kodların bizim istediğimiz eklediğimiz kodlar olduğunu doğrulaması gerekiyor. Bu noktada da integrity özelliği devreye giriyor. Sayfamızda kullanacağımız hazır kodların bir imzasını bu özelliğe değer olarak ekliyoruz. Bu imza doğrudan kodun kendisinden oluşturulur ve belli bir karakter uzunluğundadır. Ayrıca kodda bir harf bile değişecek olsa imza tutmayacaktır. Bu sayede eğer kodda zararlı/zararsız herhangi bir değişiklik olursa browser imzalar uyuşmayacağı için kodları sayfamıza yüklemeyecektir.

**Örnek olarak şu şekilde kullanabiliriz :**

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js" integrity="sha384-0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxss1yVqOtnepnHVP9aJ7xS" crossorigin="anonymous"></script>

Bir diğer etiketimiz de refreferrerpolicy. Bu etiket de scripti yükleyeceğimiz zaman, alacağımız kaynağa atacağımız verileri eklemek için kullınılır. Detaylı kullanımına buradan bakabilirsiniz :<https://www.w3schools.com/tags/att_script_referrepolicy.asp> Bu da crossorigin gibi kaynak paylaşımı maksadıyla kullanılan özelliklerdendir.

HTML sayfamızı oluştururken sayfa içerisindeki kod ne kadar uzun olursa okunması, yazılması ve incelenmesi o kadar zor olur. Bu yüzden kodları farklı sayfalara bölüp kullanmak hem daha kullanışlı hem de daha verimli olur. İşte bu amaçla farklı sayfalardaki scriptleri yükleyebilmek için de script etiketini kullanabiliriz. Bu amaçla script etiketinin *src* özelliğini kullanırız. Bu özellikle hem kendi dosya sistemimizde hem de internet üzerinde herhangi bir adreste bulunan kodları kendi sayfamıza ekleyebiliriz.

Örnek olarak kendi dosya sistemimizde, html dökümanımızla aynı dizinde bulunan bir script dosyasını çağırmak için:

<script src="myJavascript.js"></script>

Veya bir web sayfasındaki başka bir scripti çağırmak için :

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js" integrity="sha384-0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxss1yVqOtnepnHVP9aJ7xS" crossorigin="anonymous"></script>

Son olarak da bu etiketle kullanabileceğimiz type özelliğimiz, yüklenecek dosyanın içeriğinin, browser tarafından nasıl yorumlanması gerektiğini belirtir. Örnek olarak javascript dosyası yüklemek için yüklenecek kodların javascript kodunu belirtmek için şöyle yazabiliriz :

<script type="application/javascript">

document.getElementById("someTestDiv").innerHTML = "This code runs as js";

</script>

Ya da Ecmascript için :

<script type="application/ecmascript">

document.getElementById("someTestDiv").innerHTML = "This code runs as js";

</script>

Link Etiketi

Script etiketini anlatırken bir HTML dökümanında yüklenecek kodların bölünmesinin bir çok kolaylık sağlayacağından bahsetmiştik. İşte <link></link> etiketi de script etiketinin src özelliği ile kullanılması gibi link etiketini de farklı kaynaklardan farklı dosyaları HTML dökümanımıza dahil etmek için kullanabiliriz. Bu etiket dökümanımızda bulunmayan dış kaynaklarla dökümanımız arasındaki ilişki kurmak için kullanılır.

Bu etikette de crossorigin özelliği mevcuttur.

Bu etikette ilişki kurmak istediğimiz dış kaynağı href özelliği ile veriyoruz. Bu özelliğin açılımı Hypertext REFerence şeklindedir. Örnek olarak bir CSS sayfasını HTML dökümanımız ile ilişkilendirmek için şu kodu kullanabiliriz :

<link rel="stylesheet" href="styles.css">

href özelliğini hem dış kaynaklardan hem de dökümanımızın bulunduğu dosyadan ilişkilendirme yapmak için kullanabiliriz.

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">

Bu etiketin önemli özelliklerinden birisi de *rel* özelliğidir. Bu özellik dış kaynaktaki dosyayı kendi HTML dökümanımıza ne şekilde ilişkilendirmek istediğimizi belirttiğimiz kısımdır. En çok kullanılan değerleri "stylesheet" ve "icon" dur. Stylesheet eklemek istediğimiz dosyanın bir stil dosyası olduğunu söyler. Böylece browser oradaki komutları HTML'imizi şekillendirmede ve değiştirmede kullanır.

Yukarıda HTML dökümanımızın stil ve script dosyalarını ayrı ayrı yazmanın faydalarından bahsetmiştik. Script etiketi ile bu javascript dosyalarını ekleyebiliyorduk ancak CSS yani stil dosyalarını eklemenin yöntemini öğrenmemiştik. İşte bu link etiketi ve rel özelliği yardımıyla istediğimiz stil dosyasını da <head></head> etiketlerinin arasında ekleyerek web sayfamızı daha renkli ve eğlenceli hale getirebiliriz.

<title></title> etiketi yardımıyla sekmelerde görünecek, HTML dökümanımızın başlık ismini belirlemiştik. Ancak şu an bu yazının olduğu sekmeye bakarsanız başlığın yanında bir de küçük bir logo/ikon göreceksiniz. İşte bu ikonlar eklemek için de rel = "icon" kullanıyoruz. Bulunduğumuz dizinde **.ico** uzantılı bir **favicon.ico** dosyamız varsa bu dosyayı sekmelerde ismin yanında ikon olarak göstermek üzere şöyle bir kod kullanabiliriz :

<link rel="icon" href="/favicon.ico" type="image/x-icon">

Son olarak link etiketi de type özelliği kullanır. Bu özellikle de ilişkilendirdiğimiz dosyanın tipini vermiş oluyoruz. Yaygın kullanılan değerleri stil dosyaları için type = "text/css" şeklinde, ikonlar için de type="image/x-icon" şeklindedir.

Link etiketi global özellikleri de destekler. Diğer detaylı özellikleri için de buraya bakabilirsiniz : <https://www.w3schools.com/tags/tag_link.asp>

Meta Etiketi

Tanıtacağımız son etiket ise meta etiketi. Meta veriler bilgisayar bilimleri dünyasında genellikle verinin verisi anlamında kullanılır. Yani bir veriyle ilgili bilgiler meta bilgiler olarak tanımlanır. İşte HTML dökümanımızla ilgili verilerin olduğu etiketler de meta etiketleridir. Burada vereceğimiz bilgiler sitemizi arama motorlarına, sosyal medyaya ve diğer sitelere tanıtmak ve dökümanımızla ilgili bilgiler vermek için kullanılacak veriler olacak.

Sadece 4 farklı özelliği olan <meta></meta> etiketi dökümanımızla ilgili birçok bilgiyi barındıran farklı kullanım şekillerine sahip. Bu kullanımlar sayfamızla ilgili önemli bilgiler içerdiği için ayrı ayrı inceleyeceğiz.

Dünyada farklı farklı diller ve farklı alfabeler mevcut. Örnek olarak latin alfabesi, arap alfabesi, çin alfabesi, elf alfabesi vs vs. Ancak hepsinde farklı karakterler olduğu için browserin bu karakterli görüntüleyebilmek için doğru şekilde çözümlemesi gerekir. İşte HTML dökümanımızdaki bu karakterlerin çözümlenme biçimlerini belirttiğimiz <meta> etiketi özelliğimiz *charset* özelliğidir. Bizim latin alfabesi için verilen charset kodu UTF-8 dir.

<meta charset="UTF-8">

Bir diğer önemli özelliğimiz ise http-equiv'dir. Browserlar farklı sunuculara istek atarlarken belli bilgileri karşı tarafa gönderirler. İşte bu isteklerin arasında isteğin detaylarıyla ve yöntemiyle ilgili bilgilerin olduğu header'lar bulunur. Biz de dökümanımızda o dökümana ulaşan birisinin browser'inde header alanında bir bilgi tutmak istiyorsak bu meta etiketi özelliğini kullanabiliriz. Örnek olarak charset ile belirttiğimiz özellik HTML5 ile gelmiştir. Daha önceki versiyonlarda ise şu şekilde bir kullanım vardır :

<meta http-equiv="Content-type" content="text/html" charset="UTF-8">

Ayrıca refresh başlığını(header) bu meta yardımıyla belirleyerek sayfamızın belli sürede bir yenilenmesini veya belli bir süre sonra başka bir sayfaya yönlendirilmesini sağlayabiliriz.

<meta http-equiv="refresh" content="10;URL=kodluyoruz.html">

Yukarıdaki kodda sayfa yüklendikten 10 saniye sonra URL ile verdiğimiz değer olan kodluyoruz.html'ye yönlendirilecek.

Burada kullandığımız diğer etiket de *content* etiketidir. Bu da meta olarak verdiğimiz bilgilerin içeriğini tanımlamamızı sağlar.

Son özelliğimiz de *name* özelliğimizdir. Bu da meta bilgi olarak vereceğimiz bilginin tanımlayıcısıdır diyebiliriz. Örnek olarak sayfamızda en çok geçen harfin ne olduğunu belirteceğimiz bir meta bilgisi yazmak isteyelim:

<meta name="enCokGecenHarf" content="a">

Bu şekilde istediğimiz meta bilgiyi sayfamızın başlık etiketleri arasında kullanabiliriz.

Buraya kadar meta etiketini öğrendik peki sosyal medyalar ve arama motorları sitemizle ilgili bilgileri nasıl bu meta verilerden alıyor? Bu sorunun cevabı da yaygın kullanılan meta etiketleri kalıplarında bulunuyor. Örnek olarak github'da sağ tıklayıp sayfa kaynağını görüntüle dediğimizde karşımıza bir çok meta etiketi çıkıyor :

<meta property="og:url" content="https://github.com">

<meta property="og:site\_name" content="GitHub">

<meta property="og:title" content="Build software better, together">

<meta property="og:description" content="GitHub is where people build software. More than 50 million people use GitHub to discover, fork, and contribute to over 100 million projects.">

<meta property="og:image" content="https://github.githubassets.com/images/modules/open\_graph/github-logo.png">

<meta property="og:image:type" content="image/png">

<meta property="og:image:width" content="1200">

<meta property="og:image:height" content="1200">

<meta property="og:image" content="https://github.githubassets.com/images/modules/open\_graph/github-mark.png">

<meta property="og:image:type" content="image/png">

<meta property="og:image:width" content="1200">

<meta property="og:image:height" content="620">

<meta property="og:image" content="https://github.githubassets.com/images/modules/open\_graph/github-octocat.png">

<meta property="og:image:type" content="image/png">

<meta property="og:image:width" content="1200">

<meta property="og:image:height" content="620">

<meta property="twitter:site" content="github">

<meta property="twitter:creator" content="github">

<meta property="twitter:card" content="summary\_large\_image">

<meta property="twitter:title" content="GitHub">

<meta property="twitter:description" content="GitHub is where people build software. More than 50 million people use GitHub to discover, fork, and contribute to over 100 million projects.">

<meta property="twitter:image:src" content="https://github.githubassets.com/images/modules/open\_graph/github-logo.png">

<meta property="twitter:image:width" content="1200">

<meta property="twitter:image:height" content="1200">

İşte burada kullanılan meta taglar, facebook ve twitter paylaşımlarımızda bir url paylaşmak istediğimizde karşımıza çıkıyor. Sosyal medyada bir url paylaşmak istediğimizde url'yi kopyaladıktan sonra bir kart görürüz. Bu kartta bir başlık bir resim ve kısa bir açıklama bulunur. İşte o bilgiler bu meta verilerden alınır.

Arama motorları bizi meta verilerinde verilen "keywords" lere göre sıralamaya alır :

<meta name="keywords" content="Kodluyoruz,programlama,web">

Ayrıca arama motorlarında sayfa başlığının altındaki açıklamalarda da yine bu meta verilerine öncelik verilir :

<meta name="description" content="Kodluyoruzla web öğrenmeye hazır mısınız?">

Bazı arama sonuçlarında yazar adı görürsünüz. Bu kısım da yine meta etiketinin marifetidir:

<meta name="author" content="Kodluyoruz">

Meta etiketiyle söyleyeceğimiz son şey de viewport konusu. Akıllı telefonlarla birlikte geliştiriciler artık farklı cihazlara, farklı cihaz ekranlarında düzgün görünen kodlar yazmaya çalışıyor. Her ekranın genişliği boyutları farklı olduğu için tek bir ekran türüne göre tasarım ve kodlama yapmak da bu sorunu çözmüyor. Bu yüzden farklı cihazlarda da iyi görünen siteler yapmak için temel görünen alanı, bu alanın genişliğini vs tanımlamamız gerekiyor. İşte viewport burada yardımımıza koşuyor.

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Burada genişliğin cihaz genişliğinde olduğunu ve zoom oranının 1.0 olacağını tarayıcıya belirmiş oluyoruz. Böylece mobilde masaüstü görünümü gibi bir görünüm değil olması gerektiği gibi düzgün bir görüntü elde ediyoruz. Detaylı bilgi için şurayı inceleyebilirsiniz : <https://fatihhayrioglu.com/meta-viewport-etiketi/>

Şimdiye kadar başlık etiketlerini, başlık etiketlerinin özelliklerini ve genel kullanım alanlarını öğrenmiş olduk.

**Şimdi sıra kısa bir alıştırma yapalım. Lütfen soruları cavaplarken kendinize biraz zaman tanıyın ve yukarıdaki bilgilere göre hareket edin.**

**Bu bilgiler :**

→ Web sitesinde en çok kullanılan baskın renk nedir? (dominantColor) → Sayfanın başlığı nedir? (title) → Sayfanın açıklaması/konusu ne üzerinedir? (description)

Bu verilerin sayfadan kolayca alınabilmesi için hangi etiketler hangi özelliklerle nasıl kullanılmalı?

**Cevaplar :**

1. CDN (Content Delivery Network) yardımıyla bootstrap stil ve js dosyalarını sayfamıza eklemek istiyoruz. Bunun için google'da bootstrap cdn diye arattığımızda karşımıza yukarı öğrendiğimiz link ve script etiketleriyle bootstrapi nasıl ekleyebileceğimiz çıkıyor. link etiketimizde integrity özelliğimizle imza kontrolü yaptığımıza crossorigin özelliğimizi kullandığımıza ve ilişkilendirme(rel/relation) türünün stylesheet olduğuna dikkat edin. Script etiketimizde de aynı özellikler bulunmakta :

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>

1. Bu tip sayfayla ilgili veriler için <meta>etiketini kullandığımızdan bahsetmiştik. Ayrıca başlık için de <title> etiketini kullandığımızı söylemiştik. Şimdi bu etiketlerin doğru kullanımlarına bakalım :

<title>Kodluyoruz HTML</title>

<meta name="dominantColor" content="red">

<meta name="description" content="Kodluyoruz ile webe giris">

Kaynaklar

* <https://clutch.co/seo-firms/resources/meta-tags-that-improve-seo>
* <https://www.geeksforgeeks.org/most-commonly-used-tags-in-html/>
* <https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML>
* [ttps://www.w3schools.com/tags/tag\_script.asp](javascript:void(0))
* <https://www.w3schools.com/tags/tag_head.asp>
* <https://www.w3docs.com/learn-html/html-head-tag.html>
* <https://help.ahrefs.com/en/articles/2433739-what-is-meta-refresh-redirect-and-why-is-it-considered-a-critical-issue>
* <https://en.wikipedia.org/wiki/Meta_refresh>
* <http://www.yunusbassahan.com/blog/genel/arama-motorlari-ve-sosyal-medya-servisleri-icin-meta-tag-ler.html>
* <https://stackoverflow.com/questions/6535405/what-is-the-attribute-property-ogtitle-inside-meta-tag>
* <https://www.w3schools.com/tags/tag_meta.asp>
* <https://gist.github.com/lancejpollard/1978404>
* <https://www.w3schools.com/tags/tag_link.asp>
* <https://www.w3schools.com/tags/att_link_rel.asp>
* <https://www.w3schools.com/tags/att_link_type.asp>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Link_types>
* <https://www.w3schools.com/tags/att_meta_charset.asp>
* <https://www.w3schools.com/tags/att_meta_name.asp>
* <https://stackoverflow.com/questions/2359866/html-set-image-on-browser-tab>
* <https://stackoverflow.com/questions/34429024/what-is-the-purpose-of-the-integrity-attribute-in-html>
* <https://www.w3schools.com/tags/att_meta_http_equiv.asp>