HTML

**H**yper **T**ext1 **M**arkup **L**anguage, Hiper Metin İşaretleme Dili. Web sayfaları oluşturmak için kullanılan bir dildir.

|  |  |
| --- | --- |
| HTML | |
| Tasarımcı : | Tim Berners Lee |
| Dosya Uzantıları : | .html , .htm |
| MIME Türü2 : | text/html |

* HTML W3C3 ve WHATWG4 tarafından geliştirilmektedir. Dilin standartları **W3C** tarafından belirlenmektedir.

<http://www.w3.org/>

<https://whatwg.org/>

|  |
| --- |
| **NOT**  Web sayfalarınızın standartlara uygun olup olmadığını kontrol etmek için W3C’nin web sitesindeki doğrulama servisini(validation service) kullanabilirsiniz.  <https://validator.w3.org/> |

HTML Sürümleri

|  |  |
| --- | --- |
| Version | Year |
| HTML 1.0 | 1993 |
| HTML 2.0 | 1995 |
| HTML 3.2 | 1997 |
| HTML 4.0.1 | 1999 |
| XHTML 1.0 | 2001 |
| HTML 5.0 | 2014 |

* HTML sürümleri resmi olarak yayımlanmadan önce belirli aşamalardan geçmektedir. İlk olarak çalışma taslağı oluşturulmakta daha sonra aday tavsiyesi ve W3C önerisi olarak yayımlanmaktadır. Yukarıdaki tabloda resmi sürümlerin(W3C önerisi) yayımlandığı tarihler verilmiştir.
* HTML 5.0, HTML dilinin son sürümüdür ve WHATWG tarafından **HTML Living Standart** olarak adlandırılmaktadır.

Web Browser(Web Tarayıcısı)

İnternet tarayıcıları web sayfalarını görüntülemek için kullanılan programlardır.

Browserlar **Layout Engine** ve **Javascript Engine** olmak üzere iki farklı motordan oluşmaktadır.

Layout Engine : Layout Türkçe’de **yerleşim** ve **düzen** anlamlarına gelir. Layout Engine sayfa içeriğinin tarayıcı penceresinde gösterilmesini sağlar. HTML ve CSS kodlarını yorumlayarak sayfanın görüntüsünü oluşturur(Biz buna sayfanın render edilmesi diyoruz).

Javascript Engine : Javascript Engine(Javascript Motoru) ise Javascript kodlarının çalıştırılmasını sağlar.

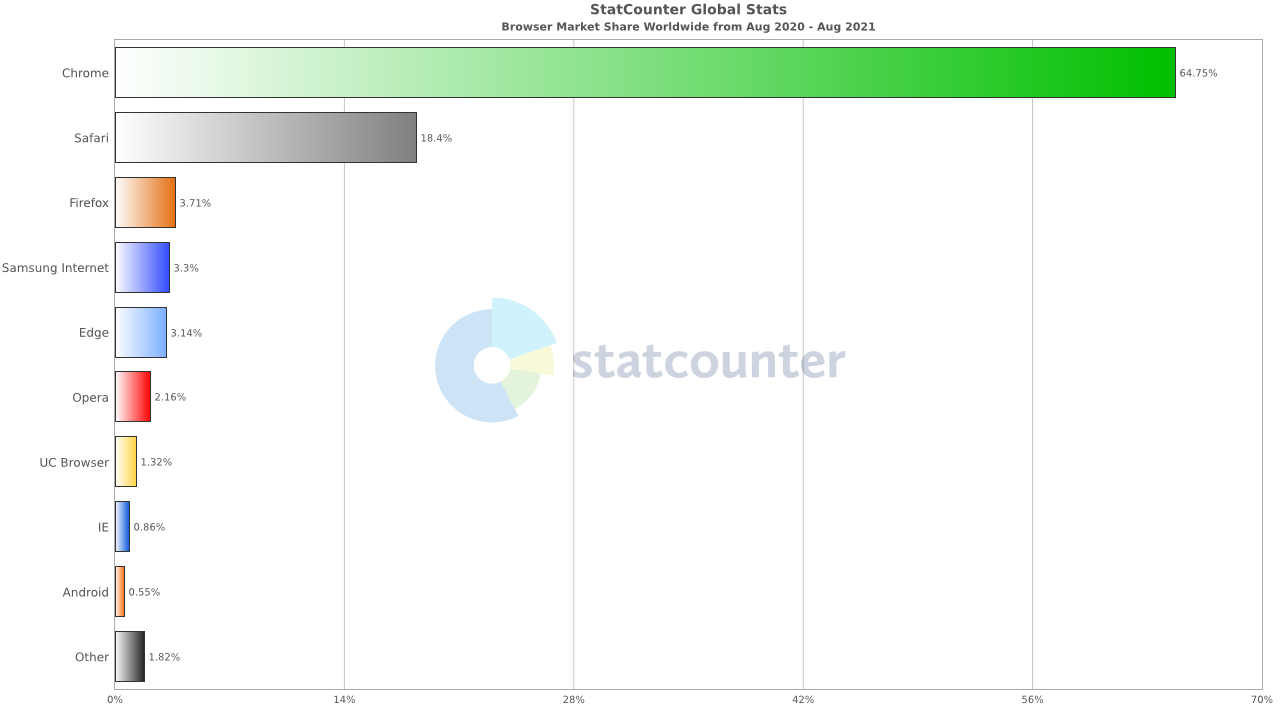
Bazı internet tarayıcılarına ait görüntüleme ve javascript motorları :

|  |  |  |
| --- | --- | --- |
| Browser | Layout Engine | Javascript Engine |
| Mozilla Firefox | Gecko | Spider Monkey |
| Google Chrome | Blink | V8 |
| Opera | Blink | V8 |
| Safari | Webkit | Nitro |
| Microsoft Edge | Blink | Chakra |

Bu motorlar tarayıcı üreticileri tarafından geliştirilmiştir ve sürekli güncellenmektedir. Aslında Layout ve Javascript Engine’in güncellenmesi demek ilgili tarayıcının yeni teknolojileri desteklemesi dolayısıyla yeni bir sürümünün çıkması demektir.

* Browser kullanım istatistiklerini gösteren harika bir web sitesi : <https://gs.statcounter.com/>

ve bu web sitesinden alınmış son bir yıla ait istatistikleri gösteren bir grafik :



* Çapraz tarayıcı testleri(Cross Browser Testing) için kullanabileceğiniz web siteleri : <https://www.browserstack.com/> , <https://crossbrowsertesting.com/>

Internet Nedir?

İnternet, dünya üzerindeki birçok bilgisayar sisteminin bağlı olduğu bir ağdır.

Bilgisayarlar arasında veri alışverişi bilgisayar ağları üzerinden gerçekleşmektedir. İnternet de bir ağdır ve dolayısıyla internetin altyapısını anlamak bilgisayar ağları hakkında bilgi sahibi olmayı gerektirmektedir. Bilgisayar ağları, bilgisayar biliminin kapsamlı bir konusudur. Bu yüzden bu başlık altında ağ topolojileri, ağ protokolleri, ağ katmanları gibi konular hakkında bilgi verilmeyecektir. Bu noktada bilinmesi gereken şey şudur : Bir ağ yaratmak istiyorsanız en az iki bilgisayarı kablo ile birbirine bağlamanız gerekmektedir. Ancak bu yeterli değildir çünkü bilgisayarlar fiziksel olarak birbirlerine bağlı olmalarına rağmen nasıl veri alışverişi yapacaklarını bilmemektedirler. Bu iş için de ağ protokolleri kullanılmaktadır. Protokol, bilgisayar ağları üzerindeki iletişim kurallarını ifade etmektedir. İki bilgisayara da aynı protokol yüklendiğinde artık aynı dili konuşurlar ve veri alışverişi mümkün hale gelir.

* İnternet standartları **ISOC** (Internet Society), **IETF** (Internet Engineering Task Force), **IAB** (Internet Architecture Board) gibi kurumlar tarafından geliştirilmektedir.
* İnternet üzerinde **TCP/IP** protokolü kullanılmaktadır.

World Wide Web(www veya Web)

**W**orld **W**ide **W**eb(Dünya Çapında Ağ) veya kısaca **Web**.

Web, internetten bağımsız bir ağ değildir. İnternetin kaynaklarını kullanır, internet ağı üzerine inşa edilmiştir ve **internet üzerinde çalışan bir servis** olarak tanımlanmaktadır. Bu noktada web ile internet arasındaki ayrımı yapmak önemlidir. Zira günümüzde bu iki terim sıklıkla karıştırılmakta ve aynı anlamda oldukları düşünülebilmekte. Web 1993 yılında HTML’in geliştirilmesi ile birlikte oluşturulmuştur. İnternetin tarihi ise webden daha eskidir.

* Web, **HTTP** (Hyper Text Transfer Protocol) protokolünü kullanır.

Hosting Nedir?

Hosting’in kelime anlamı **ev sahipliği yapmak**, **barındırmak**tır. Hosting hazırlanan web sitelerinin belirlenen alan isimlerine göre web ortamında yayımlanmasını sağlayan hizmet türüdür. Bu hizmet hosting firmaları tarafından ücret karşılığında sağlanmaktadır.

Domain Nedir?

Domain(alan adı), hatırlanması zor olan IP adresleri yerine kullanılması için internet otoriteleri tarafından geliştirilen bir isimlendirme yöntemidir.

* Bazı domain uzantıları ve anlamları (TLD - Top Level Domains) :

edu : Eğitim kurumları. (education)

gov : Devlet kuruluşları. (government)

com : Ticari alan adları için üretilmiş olmasına rağmen günümüzde her tür site için kullanılmaktadır. (commercial)

net : Günümüzde her tür site için kullanılmaktadır. (network)

org : Genellikle kar amacı gütmeyen organizasyonlar tarafından kullanılmaktadır. (organization)

* Bir domain adresinin kime ait olduğunu öğrenmek için **whois** sorgusu yapabilirsiniz. Whois hizmeti veren birçok web sitesi vardır ve işte onlardan bir tanesi : <https://whois.domaintools.com/>

HTTP Protokolü

**H**yper **T**ext **T**ransfer **P**rotocol (Hiper Metin Transfer Protokolü).

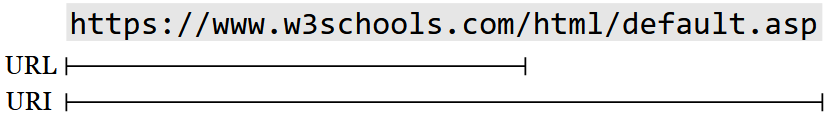
Protokolün çalışma mantığı şu şekildedir : İstemci(client) sunucudaki(server) bir porta(HTTP için genellikle 80 numaralı port) TCP bağlantısı kurarak istek gönderir. Bu portu dinleyen HTTP sunucusu, istemcinin istek mesajını bekler. İsteği aldıktan sonra sunucu istemciye istek doğrultusunda bir yanıt döndürür. Bu yanıt başlık ve gövde bölümlerinden oluşur. Başlık bölümünde durum kodu(HTTP Status Code), dönen içeriğin türü(MIME) ve karakter kodlaması(charset) gibi bazı meta bilgiler bulunmaktadır. Gövde bölümü ise sunucunun istemciye döndürdüğü veriyi(data) içermektedir.

HTTP istekleri isteğin amacına göre farklı şekillerde yapılır. Veri çekmek için **GET**, göndermek için **POST**, güncelleme işlemleri için **PUT**, silmek için **DELETE** request kullanılır.

* HTTP, **OSI**’nin uygulama katmanında(Application Layer) çalışan bir protokoldür.
* Protokol IETF tarafından geliştirilmektedir.
* Güvenli data transferi için **HTTPS** kullanılmaktadır(443 numaralı port). HTTP ile HTTPS aynı mantıkla çalışmakta HTTPS **SSL** sertifikası kullanmaktadır.

|  |
| --- |
| **NOT**  Bir web tarayıcısına bir web sitesinin adresi yazılıp enter tuşuna basıldığında ilgili sitenin sunucusuna **GET request** yapılmış olunur. |

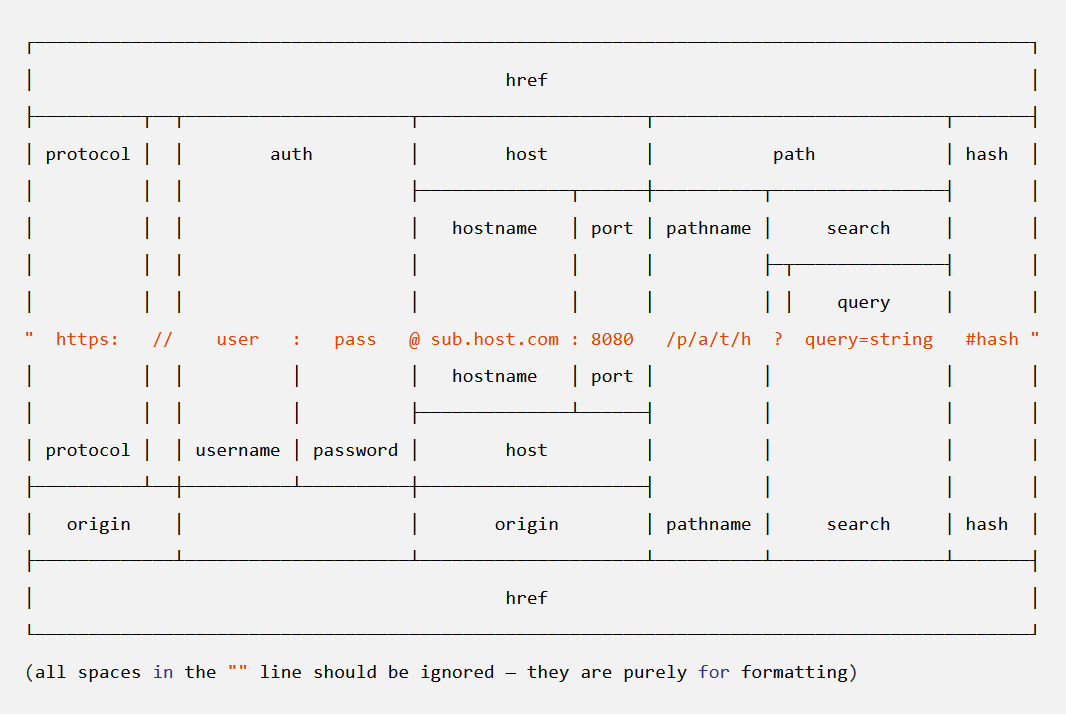
URL , URI



URI(Uniform Resource Identifier) : Bir kaynağı tanımlayan, onun yerini ifade eden karakter dizisidir.

URL(Uniform Resource Locator) : URL teknik olarak URI’nin başlangıç kısmını oluşturmasına rağmen günümüzde bu iki terim genellikle aynı anlamda kullanılmaktadır.

* Aşağıdaki görselde URL(URI) adreslerinin yapısı gösterilmektedir.



|  |
| --- |
| **NOT**  URL adresleri içerisinde domain isimleri yerine IP adresleri kullanılabilmektedir.  URL adresi içerisinde herhangi bir dizin(path) belirtilmediğinde kök dizin içerisinde olunduğu varsayılır. Dosya adı belirtilmediğinde ise sunucu tarafından varsayılan dosya response edilir. |

HTML Dosya Yolları

* Mutlak(absolute) dosya yolu : Bir dosyanın adresini(URL) ifade etmektedir. Örneğin:

https://www.example.com/images/picture.jpg

* Göreceli(relative) dosya yolu : Geçerli belgeye göre bir dosyayı işaret etmektedir. Örneğin:

/images/picture.jpg

/ : Kök dizin

./ : Mevcut dizin

../ : Bir üst dizin

HTML Elementlerinin Anatomisi

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Elementler özellik-değer çiftlerini(attribute) içerebilirler.
* Elementler yapılarına göre **Block Level Elements** ve **Inline Elements** olmak üzere iki gruba ayrılır.

Block Level Elements : %100 genişliğe sahiptirler yani bulundukları satırın tamamını kaplamaktadırlar.

Inline Elements : Genişlikleri içeriklerinin uzunluğu kadardır.

* Elementler içerik türlerine göre **Flow**, **Heading**, **Sectioning**, **Phrasing**, **Embeded**, **Metadata** ve **Interactive** olmak üzere yedi gruba ayrılır.

|  |
| --- |
| **NOT**  Blok seviyesinden elemanlar satıriçi elemanları içerebilirler ancak bunun tam tersi doğru bir kullanım şekli değildir. **Inline elementler block level elementleri içeremezler.** |

HTML Yorum Satırları(Comment Lines)

HTML elementleri içerisinde açıklama metinlerine yer verilmek istendiğinde aşağıdaki ifade kullanılmaktadır.

<!-- -->

* Javascript kodlarınızı Javascript desteği olmayan tarayıcılardan gizlemek için HTML yorum satırlarını kullanabilirsiniz.

<script>

//<!--

console.log(“Bu komut konsola bilgi yazmak için kullanıldı.”)

//-->

</script>

HTML Syntax(Sözdizim Kuralları)

* Açılan tüm etiketler(tag) kapatılmalıdır.
* Elemanların kapanış sırasına dikkat edilmelidir.
* Özellik isimleri büyük ya da küçük harflerle yazılabilmektedir.
* Benzer şekilde etiket isimleri için büyük ya da küçük harfler kullanılabilmektedir.
* Özellik değerleri çift ya da tek tırnak içinde yazılabilir. Özellik değeri boşluk içermiyorsa tırnaksız yazılabilir.
* Özelliğin değeri ve adı aynıysa sadece özellik adını yazabilirsiniz veya değer olarak iki tane çift tırnak kullanabilirsiniz. Örneğin :

<div hidden></div>

<div hidden=””></div>

<div hidden=”hidden”></div>

* Özellik değerleri içerisinde aşağıda gösterildiği gibi tırnak işareti kullanabilirsiniz.

title=”this is a ’description’”

title=’this is a “description”’

|  |
| --- |
| **NOT**  W3C etiket ve özellik isimleri için küçük harflerin kullanılmasını tavsiye etmektedir. |

DOCTYPE Tanımlaması

DOCTYPE tanımlaması yaparak tarayıcının sayfayı ilgili standarda göre yorumlamasını sağlıyoruz.

* DOCTYPE tanımı belgenin en başında yapılmalıdır.
* **HTML 5** standardına göre geliştirilen sayfalar için DOCTYPE tanımlaması aşağıdaki gibidir.

<!DOCTYPE html>

* **XHTML 1.0 TRANSITIONAL** DOCTYPE tanımlaması ise aşağıdaki gibidir.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" <http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>>

Bu şekilde bir tanımlama yaparsanız sayfalarınızda HTML 5 ile birlikte gelen yeni etiket ve özellikleri kullanamazsınız. Örneğin :

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

    <meta charset="UTF-8">

    <title>Page1</title>

</head>

<body>

    <section>Test Page</section>

</body>

</html>





section etiketi ve charset özelliği HTML 5 sürümü ile birlikte geldiği için sayfamız W3C doğrulamasından geçemedi.

HTML5 Sayfa Yapısı

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Page1</title>  </head>  <body>  </body>  </html> |

Bu etiketlerden sadece title’ın kullanılması zorunludur. html , head , body ve meta etiketleri tanımlanmasalar bile sayfalar W3C doğrulamasından geçebilmektedir.

Global Attributes

HTML5 sürümünde her element aşağıdaki özelliklere sahiptir.

id

class

style

title

lang

dir

accesskey

tabindex

contenteditable

draggable

dropzone

spellcheck

hidden

data-\*

id

Elemente benzersiz bir isim vermek için kullanılır.

* id değeri en az bir karakter içermelidir ve boşluk kullanılmamalıdır.

class

CSS sınıflarını elemente uygulamak için kullanılır.

* Değer olarak birden fazla sınıf adı verilebilir. Bunun için sınıf adları arasına boşluk konulmalıdır.
* Sınıflar içerisinde elementin aynı özelliklerine farklı değerler atanmışsa son yazılan sınıf içindeki kodlar geçerli olacaktır.
* Sınıf isimlerinde harf, rakam, \_ ve – kullanılabilir. (rakam ile başlamamalıdır)

style

Elemente CSS stilleri tanımlamak için kullanılır.

title

Elemente açıklama eklemek için kullanılır. Bu açıklama fare ilgili elemanın üzerindeyken ipucu metni olarak gözükür.

lang

Elementin içerdiği metnin dilini ayarlamak için kullanılır. Örnek :

<div lang=”en-US”></div> <!-- en(lisan kodu)-US(ülke kodu) -->

Aşağıdaki gibi sadece lisan kodu da yazılabilir.

<div lang=”en”></div>

* Lisan ve ülke kodlarının bir listesini <EK> klasöründe bulabilirsiniz.

dir=”rtl – ltr”

Elementin içerdiği metnin yönünü ayarlamak için kullanılır. Pratikte içeriği sola ya da sağa hizalamaktadır.

rtl : Sağdan sola. (right to left)

ltr : Soldan sağa. (left to right)

accesskey

Elemana klavye kısayolu tanımlamak için kullanılır.

* Kullanılan işletim sistemine göre tuş kombinasyonu değişmektedir. Detaylı bilgi için <https://www.w3schools.com/tags/att_global_accesskey.asp> bu adresten yardım alabilirsiniz.

tabindex=”{number}”

Elemanın tab sırasını ayarlamak için kullanılır.

* Sayı değeri küçük olana öncelik verilir.

hidden

Elemanı gizlemek için kullanılır.

* Gizlenen elemanlar sayfadaki yerlerini korumazlar.

[5]data-\*

Bu özellik ile HTML elementleri üzerinde veriler saklanabilmektedir.

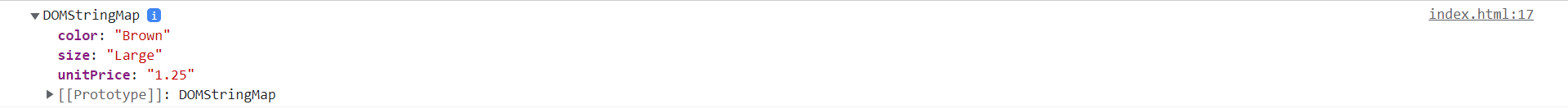
Example

<div id="egg" data-color="Brown" data-size="Large" data-unit-price="1.25">Brown Egg - Large</div>

Veriler DOMStringMap adında bir obje içerisinde tutulmaktadır. Bu objeye erişmek için elementin dataset özelliği kullanılır.

var element = document.getElementById("egg");

console.log(element.dataset);



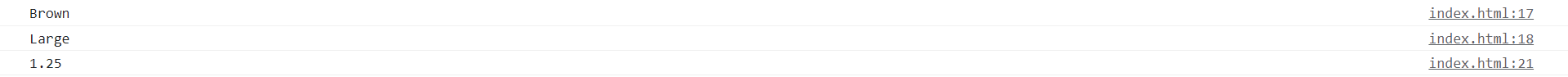
var element = document.getElementById("egg");

console.log(element.dataset.color); *//Brown*

console.log(element.dataset.size); *//Large*

*/\* Nitelik adında - sembolü ile birden fazla kelime kullanılmışsa camelCase sözdizimi ile okuma yazma yapılmaktadır \*/*

console.log(element.dataset.unitPrice); *//1.25*



Değer değişikliği yapmak için aşağıdaki sözdizimi kullanın.

Example

var element = document.getElementById("egg");

element.dataset.color = "White";

console.log(element.dataset.color); *//White*

Okuma-güncelleme işlemleri için getAttribute() ve setAttribute() metodları kullanılabilmektedir.

Example

var element = document.getElementById("egg");

element.setAttribute("data-size", "Medium");

console.log(element.getAttribute("data-size")); *//Medium*

* data-\* niteliklerini **CSS selector** olarak kullanabilirsiniz.

[data-color="Brown"]{

*/\* buraya CSS kodları yazılacak \*/*

}