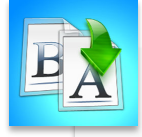


JAVASCRIPT İLE KODLAMAYA GİRİŞ



İÇİNDEKİLER

- JavaScript Nereye Yazılır?
- JavaScript Yazma, Çalıştırma ve Test Etme
- JavaScript ile Yapılabilecek İşler
- JavaScript Dilinin Genel Özellikleri



HEDEFLER

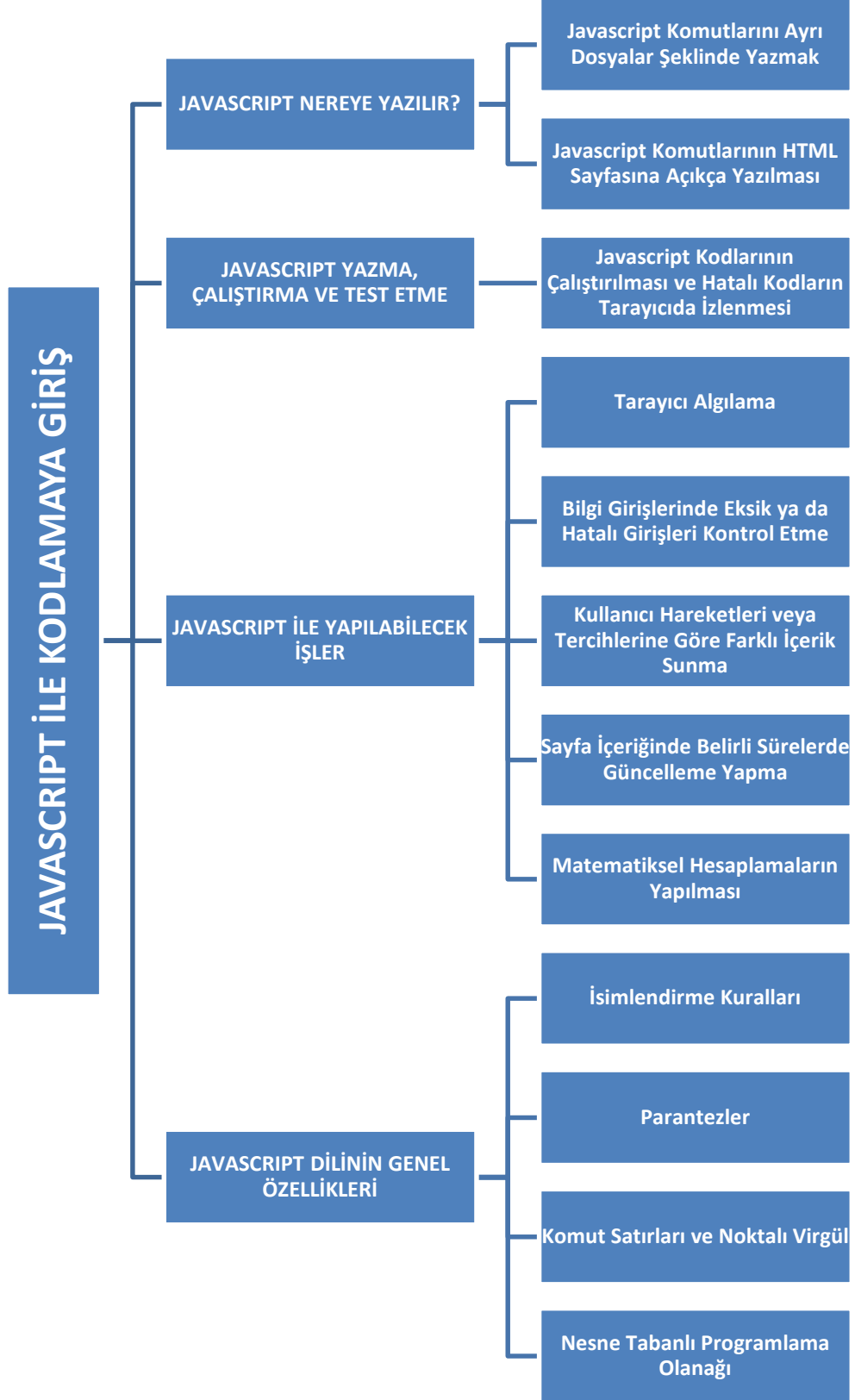
- Bu üniteyi çalıştıktan sonra;
 - JavaScript dilini HTML sayfalarına ekleyebilecek,
 - JavaScript yazmak ve çalıştırmak için gerekli programları seçebilecek,
 - JavaScript ile yapılabilecek işlemleri planlayabilecek,
 - JavaScript dilinin genel özelliklerini bileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

**İNTERNET
PROGRAMCILIĞI I**
Dr. Öğr. Üyesi Halil
ERSOY

ÜNİTE
9



GİRİŞ

İnternetin en büyük görünür parçası olan web siteleri bundan yaklaşık 40 yıl önce, 1980'li yılların başında HTML dilinin geliştirilmesiyle ortaya çıkmıştır. Çok büyük miktarda metin türündeki bilgilerin dünya çapında yer alan sunucular üzerinden kullanıcıların bilgisayarlarına aktarılması için geliştirilen HTML dili, ilk ortaya çıktığında çok büyük yenilik getirmiştir. O ana kadar bilgilerin noktadan noktaya özel programlar ile aktarılması söz konusuysen, HTML sayesinde birbirine bağlı biçimde oluşturulan web sayfaları (*hiper metin / hyper text*), web sunucuları üzerinde sürekli barındırılabilir ve istendiği anda kullanıcılar tarafından *indirilebilir* olmuştur.



Javascript dili web tarayıcıları tarafında çalıştırılan kodlar için yazılır. Java programlama dili ile ilgisi yoktur.

HTML dili ve bu dil ile geliştirilen web siteleri daha sonraları çok fazla kullanılmaya başlanmıştır. Örneğin sadece sunucuda bulunan bilgilerin kullanıcılara indirilmesi değil, aynı zamanda kullanıcıların da bu sunuculara bilgi göndermesi için web sayfaları kullanılmaya başlanmıştır. Ayrıca bu hizmetten yararlanmak isteyen kullanıcılar için çok sayıda *tarayıcı* geliştirilmiş (Safari, Internet Explorer, Chrome, Opera, Mozilla vb.) ve farklı cihazların (bilgisayar, tablet, cep telefonu vb.) bu tarayıcıları çalıştırabilir hâle gelmesi sağlanmıştır. Kullanıcı kitlesi hızla artan ve kullanım platformu hızla çeşitlenen web sayfalarının tüm kullanıcılara benzer içerik ve hizmet sunabilmesi için, bilginin sunucu ve tarayıcı üzerinde işlenebilmesi amacıyla *HTML diline ek olarak* programlama dilleri geliştirilmiştir.

Web sayfalarının kullanıcıya ulaşmasında *iki farklı aşamada çalışan* programlama dilleri ile veri veya içerik şekillendirilebilir; bunlar *sunucu taraflı* programlama dilleri ve *istemci taraflı* programlama dilleri olarak gruplandırılabilir. *Sunucu taraflı* programlama dilleri, web *sunucuları* üzerinde çalışan, web sitesinin kullanıcı tarafından talep edildiği anda derlenen ve çalışan, ardından sonuç olarak farklı HTML içeriğini önce sunucu üzerinde oluşturup sonra kullanıcıya (istemciye) gönderen dillerdir. En popüler sunucu dilleri arasında PHP, ASP.NET, ASP, CGI sayılabilir. Bu diller ile yazılan programlar, HTML sayfalarının içeriklerinin tamamının ya da bir kısmının her kullanıcı için farklı biçimde oluşmasını sağlar. Aynı zamanda bu diller ile yazılan sayfalar web sayfaları haricinde kullanıcının görmediği işlemler için de sunucu üzerinde çalışırlar (Örneğin veri tabanı bağlantıları, veri arama ve filtreleme işlemleri, güvenlik işlemleri gibi).



HTML5 standartları öncesinde, Javascript'e alternatif başka diller de mevcuttu, örneğin VBScript.

İstemci taraflı diller ise web sayfasının içinde ya da ek bir dosya olarak kullanıcıya (istemciye) indirilir ve kullanıcının *tarayıcısı tarafından çalıştırılır*. Bu dillerin amacı genellikle kullanıcının web sitesi ile etkileşimi sırasında sayfanın vereceği tepkileri tarayıcı üzerinden çabucak verebilmektir. Örneğin kullanıcının şifresini yazması istenen bir web sayfasında, istemci taraflı bir dil ile bu durum kontrol edilebilir ve şifre girilmeden ilerlenmesi sunucuyla bağlantı kurulmaya gerek duyulmadan engellenebilir. En çok kullanılan istemci taraflı dil *Javascript'tir*. Daha önceleri Visual Basic Script (VB Script) ile rekabet içinde olan Javascript, geçtiğimiz yıllarda farklı teknoloji firmaları tarafından tercih edilmiş ve *HTML5 standartları* içerisinde tek başına yer alarak standart hâle gelmiştir.

Bu ünite de Javascript diline giriş yapılacaktır. Javascript dil olarak çok daha fazla amaca hitap etse de, bu ünite de asıl amaç HTML etiketleri ve CSS kuralları ile oluşturulmuş web sayfalarına daha fazla işlevsellik katmak amacıyla Javascript komutlarının eklenmesidir. Bu nedenle kitabın daha önceki ünitelerinde anlatılan HTML ve CSS konularını bildiğiniz varsayılmıştır.

Öte taraftan Javascript diğer web teknolojileri ile birlikte çok farklı amaçlara hizmet edebilecek bir dildir. O nedenle tek bir ünite de ele alarak tamamlamak mümkün değildir. Bu nedenle bu kitapta Javascript'e 3 ünite ayrılmış, ancak bunlardan sonra da Javascript ile yazılmış hazır kütüphanelerin kullanımı ile farklı ürünlerin yaratılmasını sağlayacak ünitelere yer verilmiştir.

Bu ünitenin kapsamı aşağıdaki gibidir:

- Javascript nereye yazılır?
- Javascript hangi programlar ile yazılır ve çalıştırılır?
- Javascript ile neler yapılabilir?
- Javascript dilinin özellikleri nelerdir?

JAVASCRIPT NEREYE YAZILIR?

HTML sayfalarına Javascript komutlarını yazmanın iki yeri vardır. Bunlardan birincisi komutları ".js" uzantılı bir başka metin dosyası içerisine yazmak ve bu dosyayı HTML sayfasına dâhil etmektir. İkinci yol ise Javascript komutlarını HTML sayfasının içerisine açık biçimde yazmaktır. Bir HTML sayfasında her iki yöntemi de kullanmak söz konusu olabilir. Bu yolların komutların çalışmasında bir etkisi yoktur; ancak kodu yazan kişinin daha kolay çalışması ve aynı Javascript komutlarının farklı dosyalarda da kullanabilmesi açısından her iki yöntemin avantajları vardır. Aşağıdaki iki başlıkta bu iki yöntem anlatılmıştır.

Javascript Komutlarını Ayrı Dosyalar Şeklinde Yazmak

Aşağıdaki tablonun ilk sütununda HTML ile yazılmış basit bir web sayfasının kodları görülmektedir. Bu kodlardan `<script>` etiketi içerisinde "src" özelliği ile adı belirtilen `scriptKomutlari.js` dosyasının içeriği ise tablodaki ikinci sütunda görülmektedir.

Tablo 9.1. HTML ve Javascript Komutları

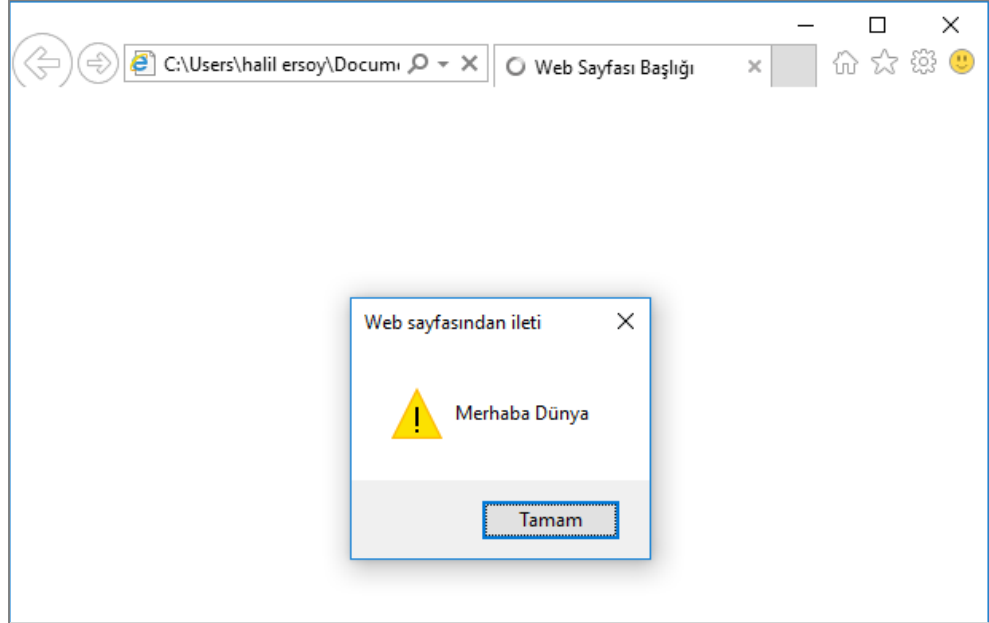
index.html	scriptKomutlari.js
<pre> <!DOCTYPE html> <html lang="tr"> <head> <meta charset="UTF-8"> <title>Web Sayfası Başlığı</title> <script src="scriptKomutlari.js"> </script> </head> <body> <h1>Sayfa Başlığı</h1> </pre>	<pre> alert("Merhaba Dünya"); </pre>

```
<p>
    Bu bir web sayfasıdır. Sayfadaki ilk paragraf
budur.</p>
</body>
```

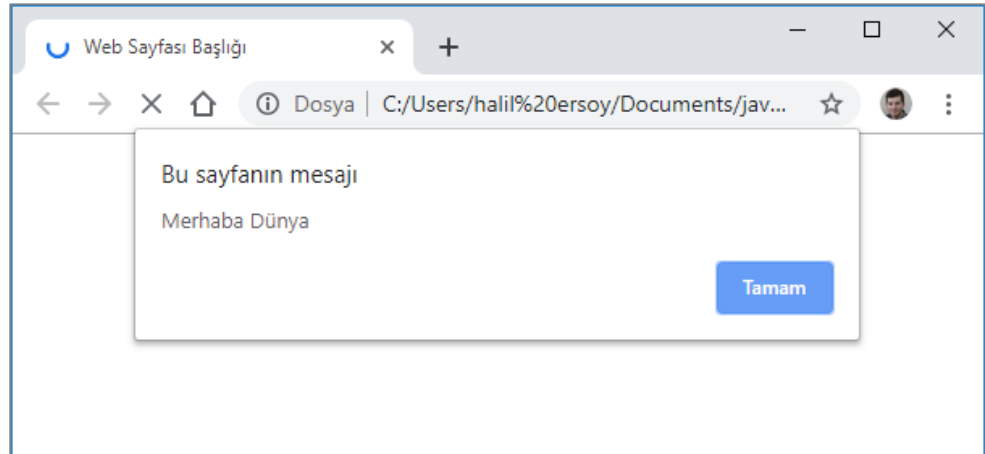


Tarayıcılar arasında çok az da olsa aynı Javascript komutlarını farklı biçimde çalıştırılabilir.

Yukarıdaki “index.html” sayfasının tarayıcıda görünen hâli ise aşağıdaki Şekil 9.1. ve 9.2.’de verilmiştir. Şekil 9.1.’de tarayıcı olarak *Internet Explorer* kullanılırken, Şekil 9.2’de tarayıcı olarak *Chrome* kullanılmıştır.



Şekil 9.1. “Index.Html” Sayfasının Internet Explorer İle Görünümü



Şekil 9.2. “Index.Html” Sayfasının Chrome İle Görünümü

Öncelikle yukarıdaki iki farklı görüntünün tarayıcı kaynaklı olduğunu belirtmek gerekir. Farklı firmaların tarayıcıları çok az da olsa aynı Javascript komutlarını farklı biçimde çalıştırır ya da farklı görünümde ekrana aktarır. Ancak bu farklar bu ilk üniteye göz ardı edilecektir. Yine de üniteye anlatılan komutların doğru öğrenilebilmesi ve test edilebilmesi için tarayıcı olarak bu üniteye *Google Chrome’un* tercih edildiğini belirtmek gerekir. Öğrencilerin de birebir aynı çıktıları alabilmeleri için bu tarayıcıyı kullanmaları beklenmektedir.



Bireysel Etkinlik

- Bilgisayarınızdaki herhangi bir kelime editörü ile Tablo 9.1.'deki kodları aynı dosyalar hâlinde yazıp aynı klasöre kaydedin. Ardından bilgisayarınızdaki farklı tarayıcılar ile HTML sayfasını görüntüleyin ve varsa aradaki farklara bakın.



HTML, CSS ve Javascript ile yazılmış web sayfaları mutlaka farklı tarayıcılarda test edilmelidir.

Yukarıdaki örnekte (Tablo 9.1.) gösterilen Javascript yazma tekniğinde, “index.html” dosyası ile “scriptKomutlari.js” dosyasının bilgisayarda aynı klasörde yer alması gerekir. Eğer istenirse, “index.html” dosyasının bulunduğu klasöre “scriptler” adında bir klasör oluşturulabilir ve Javascript dosyası bu klasör altına yerleştirilebilir. Bu yol tercih edilirse, `<script>` etiketinin içindeki “src” özelliği “scriptler/scriptKomutlari.js” şeklinde değiştirilmelidir.



index.html



scriptKomutlari.js



index.html



scriptler



scriptKomutlari.js

```
<script
src="scriptKomutlari.js">

</script>
```

```
<script src="scriptler/scriptKomutlari.js">

</script>
```

Şekil 9.3. Javascript Dosyasının Yerine Göre "Src" Özelliğindeki Fark

Javascript komutlarını bu şekilde HTML sayfalarına “js” uzantılı dosyalar hâlinde çağırırken aşağıdaki prensipleri bilmek gerekir:

- Bir HTML sayfasına birden fazla Javascript dosyası eklenebilir.
- Bir `<script>` etiketiyle sadece bir Javascript dosyası eklenir. Birden fazla Javascript dosyası eklenmek istenirse, her biri için ayrı ayrı `<script>` etiketi yazılmalıdır.
- Aynı Javascript dosyası bir HTML sayfasına sadece bir kez dâhil edilir.
- Javascript dosyalarının adlarında, tıpkı HTML dosya adlarında olduğu gibi noktalama işaretleri yer alamaz. Türkçe karakter ve boşluk karakterleri ise bazı tarayıcılarda çalışsa bile önerilmez.
- Javascript dosyasının adındaki harfler büyük ya da küçük olabilir. Ancak dosya adı ile “src” özelliğindeki dosya adındaki harflerin aynı şekilde olması gerekir.
- Javascript dosyalarının içinde ayrıca `<script>...</script>` etiketleri yazılmaz.
- Javascript dosyaları HTML dosyalarına dâhil edilirken, `<script>` etiketleri genellikle `<HEAD>...</HEAD>` etiketlerinin arasına yazılır. Bu çoğu zaman uygun bir yerdir ancak nadir durumlarda `<HEAD>` etiketleri sonrasında ve `<BODY>...</BODY>` etiketlerinin içinde de yer alabilir.

Bu şekilde komutları Javascript dosyalarında tutmanın avantajları şunlardır:

- Aynı Javascript dosyası, başka HTML sayfalarına da dâhil edilebilir, böylece tek bir “js” dosyası tüm HTML sayfaları için gerekli komutları içerebilir.



Javascript dosyaları hem HTML sayfalarına hem de istenirse diğer Javascript dosyaları içine dâhil edilebilirler.

- Javascript dosyasındaki bir güncelleme, tüm HTML sayfalarına yansıtacağı için değişikliklerin daha hızlı yapılması sağlanır.
- HTML sayfalarına dâhil edilecek Javascript dosyaları farklı bir web sunucusunda yer alabilir. Örneğin aşağıdaki etikette, bir başka web sitesinin adresi ile birlikte verilen JS dosyası HTML sayfasına dâhil edilebilir.
`<script src="http://www.server.com/file.js"></script>`
- HTML dosyasında Javascript komutları yer almayacağı için sayfa daha az koddan oluşur.

Javascript Komutlarının HTML Sayfasına Açıkça Yazılması

HTML sayfalarında etiketler arasına `<script>.....</script>` etiketleri arasına Javascript komutları yazılabilir. Aşağıdaki web sayfasının kodlarını ve devamındaki çıktıyı inceleyelim.

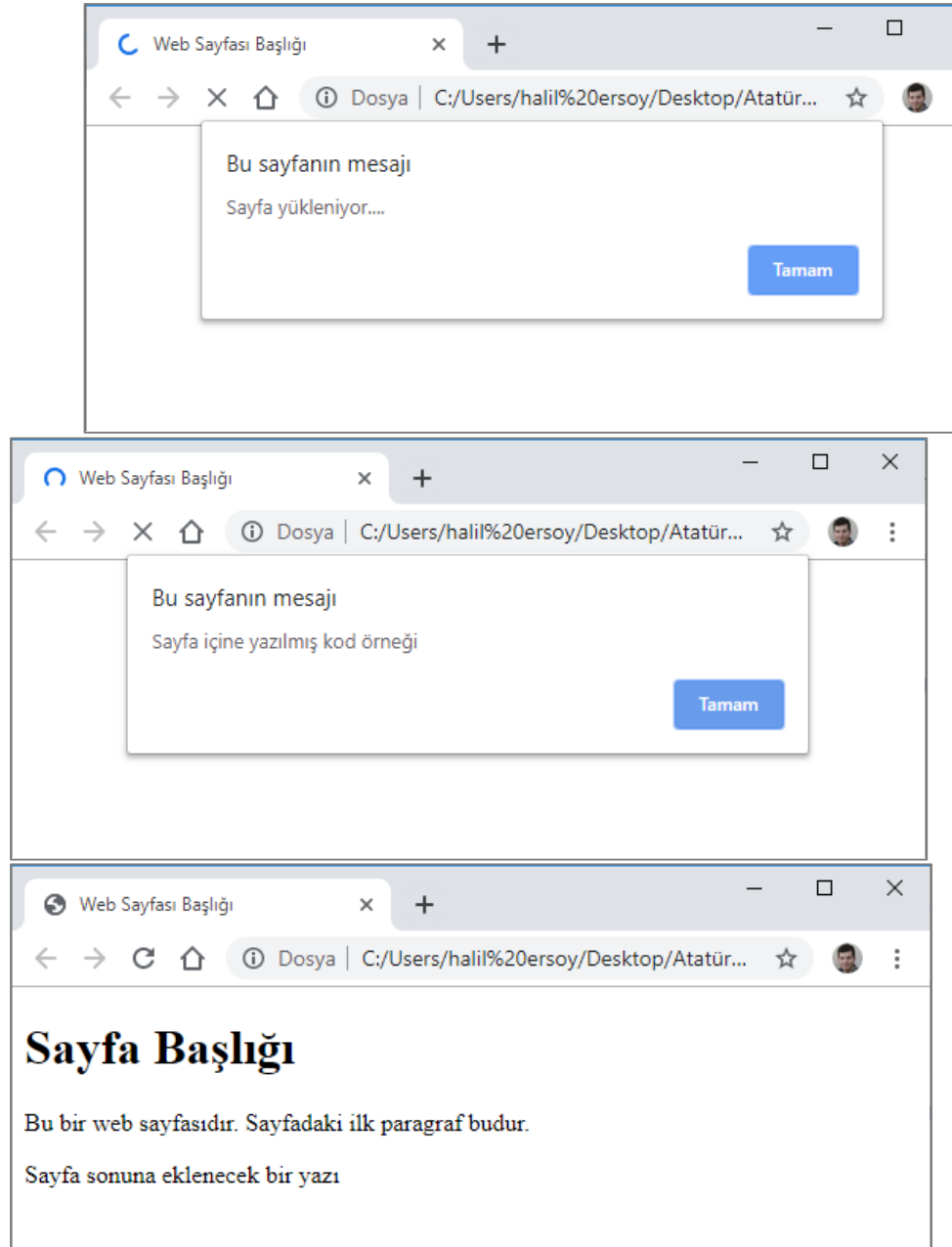
Tablo 9.2. Script Etiketleri Arasına Yazılmış Javascript Komutları

HTML ve Javascript Komutları
<pre> <!DOCTYPE html> <html lang="tr"> <head> <meta charset="UTF-8"> <title>Web Sayfası Başlığı</title> <script> alert('Sayfa yükleniyor....'); </script> </head> <body> <h1>Sayfa Başlığı</h1> <p>Bu bir web sayfasıdır. Sayfadaki ilk paragraf budur.</p> <script> alert('Sayfa içine yazılmış kod örneği'); document.write('Sayfa sonuna eklenecek bir yazı'); </script> </body> </pre>



Bir HTML sayfasında birden fazla yerde Javascript komutları yazılabilir.

Yukarıdaki HTML sayfası tarayıcıya ilk yüklendiğinde, sırası ile aşağıdaki resimlerdeki mesajlar ve içerikler ekrana gelir.



Şekil 9.4. Sayfanın Yüklenmesi Sırasında Ekrana Gelen Mesajlar

Yukarıdaki örnekte görüldüğü üzere, bir HTML sayfasına birden fazla yerde *script etiketleri* (script blokları) ile Javascript komutları yazılabilir. Komutların çalışma sırası yukarıdan aşağıyadır.

Bu yöntemle yazılan Javascript komutları sadece *yazıldıkları sayfada* çalışırlar. Bir başka HTML sayfasında aynı komutların çalışması istenirse, aynı komutların o sayfaya da yazılması gerekir.

Javascript komutlarının HTML sayfası içerisinde yazılma yönteminde sıklıkla kullanılan bir yol ise, komutları *Javascript fonksiyonları* hâlinde yazmak ve bu fonksiyonları istenilen yerden çağırarak çalıştırmaktır.

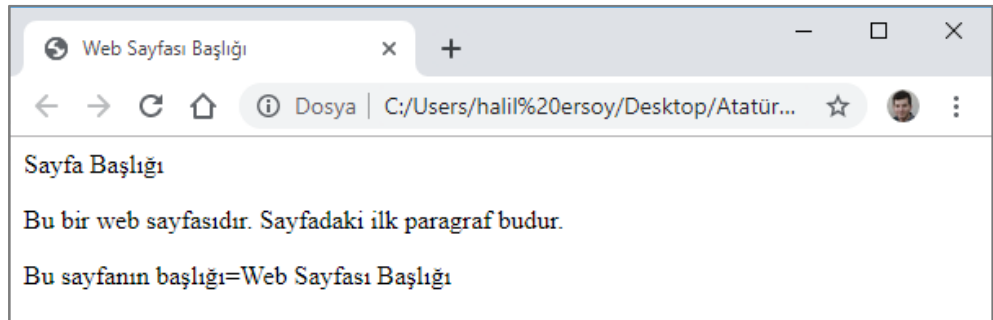
Tablo 9.3. Javascript Komutlarının Fonksiyon İçinde Yazılması ve Başka Yerden Çağırılması

HTML ve Javascript Komutları

```

<!DOCTYPE html>
<html lang="tr">
<head>
  <meta charset="UTF-8">
  <title>Web Sayfası Başlığı</title>
  <script>
    function sayfaAdresiniGoster(){
      document.write("Bu sayfanın başlığı=" + document.title);
    }
  </script>
</head>
<body>
  <h1>Sayfa Başlığı</h1>
  <p>Bu bir web sayfasıdır. Sayfadaki ilk paragraf budur.</p>
  <script>
    sayfaAdresiniGoster();
  </script>
</body>

```



Şekil 9.5. Sayfanın Çıktısı



Javascript komutları
fonksiyonlar hâlinde
yazılabilir.

Tablo 9.3.’teki kodlar ve Şekil 9.5.’teki çıktı incelendiğinde, <HEAD> etiketleri arasındaki **script** blokunda, sayfaAdresiniGoster() adından bir fonksiyon tanımlandığı görülmektedir. Bu fonksiyon, tanımlandığı yerde veya sıralamada çalışmaz, ancak daha sonra adının bir komut gibi yazıldığı zaman çalışır. <BODY> etiketleri arasındaki **script** blokunda bu fonksiyonun adı yazılarak çalışması sağlanmıştır. Fonksiyonun içindeki document.write() komutunun çıktısı olan yazı, fonksiyonun tanımlandığı yerde değil, *çağrıldığı yerde* ekrana yazılmıştır.

Fonksiyonlar konusu daha sonraki ünitelerde detaylı biçimde ele alınacaktır. Ancak bu aşamada kodları yazma yeri olarak açıklanmak istenmiştir. Fonksiyon tanımlamada kullanılacak fonksiyon ismi, fonksiyonun türü, alacağı parametreler gibi detaylar bundan sonraki ünitelerde “Fonksiyonlar” başlığı ile ele alınacaktır.



Bireysel Etkinlik

- Şekil 9.8.'deki kodları editörde yazıp sayfayı tarayıcınızda görüntüleyin.
- Sayfadaki <title> etiketindeki metni değiştirip sayfanızı kaydedin ve taraycınızı yenileyin. Aradaki farkı görün.

JAVASCRIPT YAZMA, ÇALIŞTIRMA VE TEST ETME

Javascript komutları tıpkı HTML etiketleri gibi metin tabanlı yazılır ve tarayıcı üzerinde derlenip çalıştırılır. Bu nedenle HTML komutlarını hangi program ya da editörde yazıyorsanız, aynı program veya editörde Javascript komutlarını da yazabilirsiniz. Hatta HTML ve Javascript'i basit metin editörlerinde bile (*Not Defteri®* gibi) yazabilirsiniz.

Öte taraftan web sayfası geliştirmek ve/veya tasarlamak çoğu zaman çok fazla sayıda kod yazmayı ve sonucu görmeyi gerektirir. Bu aşamada programcılara yardımcı olacak birçok profesyonel editör kullanılmaktadır. Bu editörler sadece HTML etiketleri değil, aynı zamanda CSS ve Javascript komutlarını yazma



Javascript komutları metin tabanlı yazılır ve tarayıcı üzerinde derlenip çalıştırılır.



Örnek

- Hem HTML sayfalarını hem de Javascript komutlarını yazmak için aşağıdaki editörler kullanılabilir:
- Adobe Dreamweaver® (Ücretli)
- Microsoft Visual Studio® (Ücretli)
- Brackets® (Ücretsiz)
- Notepad++® (Ücretsiz)
- Visual Studio Code® (Ücretsiz)

aşamasında da çok fazla yardımcı olmaktadır.

Örnekteki editörler ilk akla gelenlerdir ve her geçen gün yeni editörler geliştirilmektedir. Bu ünite de anlatılan Javascript komutları için ücretsiz olan *Visual Studio Code®* kullanılmıştır.

Javascript Kodlarının Çalıştırılması ve Hatalı Kodların Tarayıcıda İzlenmesi

Yukarıdaki editörler HTML sayfalarını etiketler, CSS kuralları ve Javascript komutları ile yazma aşamasında yardımcı olurken, yazılan kodların çıktısını göstermemekte veya eksik göstermektedir.

Bu nedenle hangi editörü kullanıyor olursanız olun, HTML sayfalarının ve içerisinde Javascript komutlarının çıktısını görmek için gerçek tarayıcıları kullanmanız tavsiye edilir.



Javascript dilinde yapılan yazım yanlışları tarayıcılar tarafından algılanabilir. Ancak olası mantık hataları tarayıcı tarafından algılanmayabilir.

Günümüzde popüler olan tüm tarayıcılar (Google Chrome®, Microsoft Edge®, Microsoft Internet Explorer®, Safari®, Opera®, Firefox® vb.) Javascript'i derleyip çalıştırabilmektedir. Hatta Javascript komutlarında olası bir hatayı kendi içerisinde ayrı bir pencerede gösterebilmektedir. Bu sayede programcılar yazdıkları kodun doğru çalışmadığı zamanlarda hatanın yerini ve olası çözümleri kolaylıkla bulabilmektedir.

Bu durumu göstermek için aşağıdaki kodlarda bilerek *hata* yapılmıştır. Script bloku içinde hatalı biçimde document.vrite() komutu kullanılmıştır. Komutun doğrusu document.write() olmalıdır. Ancak hatalı biçimde yazılan bu kod ile sayfa tarayıcıda açıldığında Şekil 9.6.'daki çıktı ekrana gelecektir.

Tablo 9.4. Hatalı Javascript Komutu İçeren Sayfa Kodları

HTML ve Javascript Komutları
<pre> <!DOCTYPE html> <html lang="tr"> <head> <meta charset="UTF-8"> <title>Web Sayfası Başlığı</title> <script> function sayfaAdresiniGoster(){ document.vrite("Bu sayfanın başlığı=" + document.title); } </script> </head> <body> <h1>Sayfa Başlığı</h1> <p>Bu bir web sayfasıdır. Sayfadaki ilk paragraf budur.</p> <script> sayfaAdresiniGoster(); </script> </body> </pre>

Şekil 9.6. Hatalı Javascript Komutunun Olduğu Sayfanın Çıktısı

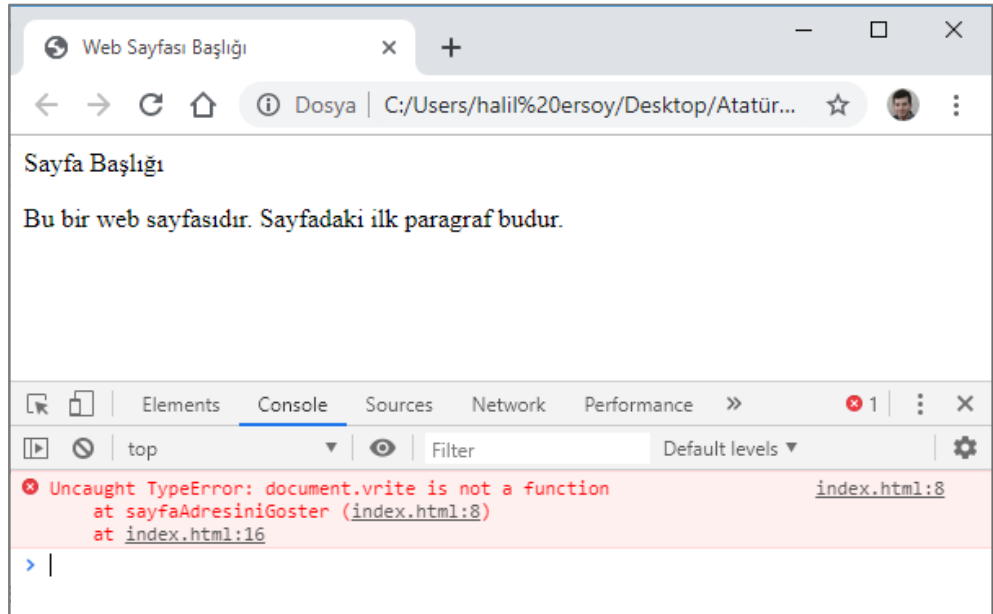


Javascript kodlarındaki yazım (sözdizim) hataları HTML etiketlerinin çalışmasını engellemez.

Javascript kodlarındaki *yazım (sözdizim) hataları* HTML etiketlerinin çalışmasını engellemez. Ancak Javascript komutlarının çalışması, hatalı yerden itibaren durur ve devamındaki Javascript komutları çalıştırılmaz. Örnekteki çıktıda Javascript komutları ile yazılması gereken yazının ekrana gelmediğini fark edebilirsiniz.

Hatanın hangi satır ve neden olduğunu anlamak için tarayıcıların “*geliştirici seçenekleri*” (*developer options*) açılmalıdır. Örnekte kullanılan Google Chrome®

için bu seçeneği klavyeden F12 tuşuna basarak açabilirsiniz. Bu seçeneği açtığınızda tarayıcının alt kısmında sayfanın kodlarıyla ilgili bilgiler alabileceğiniz pencereler açılır (Şekil 9.7.).



Şekil 9.7. Tarayıcıda Açılan Geliştirici Seçenekleri

Tarayıcıda açılan *geliştirici seçenekleri* penceresinde *Console* sekmesini seçtiğinizde, varsa *Javascript hatalarını* ve hatanın tespit edildiği *satır numarasını* görebilirsiniz. Örnekteki uyarıda index.html sayfasının 8. satırında tanımlanamaya komut (document.vrite) olduğu mesajı verilmektedir. Bu hata mesajını gören programcı, index.html dosyasındaki 8. satıra dönüp hatalı komutu düzeltmelidir.

JAVASCRIPT İLE YAPILABİLECEK İŞLER

HTML sayfalarında Javascript ile yapabileceğimiz sayısız iş vardır.

Javascript'in dil kurallarına geçmeden önce bu işlerden en çok popüler olanları kısaca aşağıdaki gibi sıralayabiliriz. Aşağıda anlatılan işlerin Javascript ile nasıl yapıldığı ise daha sonraki ünitelerde anlatılacaktır.

Tarayıcı Algılama

Web sayfalarını ziyaret eden kullanıcılar çok farklı tarayıcı ve hatta cihaz ile bu ziyaretlerini yapmaktadırlar. Farklı cihaz ve tarayıcılar kimi web sitelerini farklı biçimde algılamakta, yazılmış olan kimi CSS ve Javascript komutlarını farklı biçimde çalıştırmaktadır. Farklılık web sitesi geliştiriciler için can sıkıcıdır çünkü aynı web sayfasının farklı tarayıcılar için farklı versiyonlarının yazılması gerekmektedir. Farklı yazılan web sayfalarının hangi kullanıcıya hizmet edeceği ise kullanıcının tarayıcısının adının, versiyonunun ve tarayıcının çalıştığı cihaz hakkında ön bilgi edinilmesini gerektirir.

Bir HTML sayfasının başında yazılacak Javascript komutları ile tarayıcının adı ve farklı özellikleri öğrenilebilir; ardından kullanıcıya kendi tarayıcısı ile uyumlu içerik veya sayfalar gösterilebilir.



Javascript ile içinde bulunduğunuz bir sayfadaki herhangi bir form alanına girilen eksik bilgi tespit edilebilir.

Bilgi Girişlerinde Eksik ya da Hatalı Girişleri Kontrol Etme

Günümüzde birçok web sitesi, sadece bilgi sunmak değil aynı zamanda kullanıcıdan bilgi alarak farklı hizmetler sunmaktadır. Kullanıcının girmesi gereken bilgilerin eksik ya da hatalı olması, web sitesinin hizmeti verememesine ve daha önemlisi olası bir hatadan dolayı web sunucusunun sıkıntı yaşamasına neden olabilir. Örneğin web sayfalarındaki formlarda kullanıcıların girmesi gereken kimi bilgilerin sunucuya gönderilmeden önce biçim olarak kontrol edilmesi ve olası eksikliklerde bu bilginin sunucuya hiç gönderilmemesi gerekir.

Javascript ile içinde bulunduğunuz bir sayfadaki herhangi bir form alanına girilen eksik bilgi tespit edilebilir ve kullanıcıya derhâl uyarı mesajı verilebilir. Örneğin e-posta adresinin girilmesi gereken bir kutuda “@” sembolünün olmaması Javascript kodları ile denetlenip, kullanıcıya “Yanlış ya da eksik bir e-posta adresi girdiniz” uyarısı verilebilir.

Kullanıcı Hareketleri veya Tercihlerine Göre Farklı İçerik Sunma

Çoğu web sitesi, kullanıcıların sayfalar içerisinde hareketlerini (tıklamalar, fare ile üzerinden geçme, listeden bir seçeneğin seçilmesi, takvimde belirli bir gün seçilmesi vb.) algılamak ve bu harekete cevap olarak farklı içerikler göstermek ister. Örneğin seyahat bileti arayan bir yolcunun hareket noktasını seçtikten sonra olası varış noktalarının listelenmesi kullanıcı için büyük kolaylık olacaktır.

Javascript ile kullanıcıların farklı içerik öğeleri ile etkileşimi (tıklama, seçme, iptal etme vb.) algılanabilir ve sonrasındaki uygun içerik, sayfaya yansıtılabilir.

Sayfa İçeriğinde Belirli Sürelerde Güncelleme Yapma

Anlık değişimlerin fazla olduğu konularda, web siteleri, kullanıcılarına sürekli yeni bilgiyi sunmak isterler. Örneğin döviz kurlarını içerik olarak sunan web sitesi, kullanıcı web sayfasına bağlandığı andan itibaren her 5 saniyede bir döviz kurlarını güncel hâliyle göstermek isteyebilir.

Kullanıcının herhangi bir hareketine bağlı olmayan bu güncelleme işlemi, Javascript ile hazırlanmış ve belirli periyotlarda tekrar çalışan fonksiyonlar ile yapılabilir.

Matematiksel Hesaplamaların Yapılması

Kullanıcılar kendi girecekleri bazı rakamlar ile önceden yazılmış formüller sayesinde hesaplamalar yapmak isteyebilirler. Örneğin üniversite tercihlerinde kullanılacak bir web sayfasında, kullanıcının sayfaya gireceği doğru ve yanlış cevap sayıları ile netleri hesaplanabilir ve bu netlere göre puan hesaplanabilir.

Javascript ile kullanıcıların sayısal veya sözel veri girmesi ve bu verilerin belirli formüllerde hesaplanarak sonuçların ekran yazılması sağlanabilir.

JAVASCRIPT DİLİNİN GENEL ÖZELLİKLERİ



Yapılabilecek bu işlerde Javascript haricinde yollar da kullanılabilir, ancak Javascript hem hızlı hem de sunucuyu yormayan bir çözümdür.

Javascript diğer programlama dilleri gibi kendisine özgü özellikleri olan bir dildir. Dili oluşturan farklı öğelerin sözdizimi ve özellikleri bu ve ilerleyen ünitelerde ele alınacaktır. Ancak Javascript dilinin genel özellikleri aşağıdaki alt başlıklarda belirtilmiştir.

İsimlendirme Kuralları

Javascript'te programcı tarafından yaratılacak bir değişken, fonksiyon ya da nesnenin ismi aşağıdaki kurallara uymak zorundadır. Aksi takdirde kodlar hata verecektir.



En çok yapılan kodlama hatlarının başında, bir değişken ya da fonksiyonun tanımlandıktan sonra farklı şekilde (büyük/küçük harf) kullanılmaya çalışılmasıdır.

- Değişkenler, fonksiyon isimlendirmeleri ve komutlarda *büyük ve küçük harf ayrımı vardır*. Örneğin alert() komutu doğru, Alert() ifadesi ise geçersizdir.
- Değişken, fonksiyon ve diğer isimlendirmelerde İngiliz alfabesindeki *harfler* (A-Z / a-z), *rakamlar* ve *alt tire* (_) ve *dolar* (\$) sembolleri kullanılabilir. Bunların dışında sembol kullanılmasına izin verilmez.
- Değişken, fonksiyon ve diğer isimlendirmelerde *ilk karakter rakam olamaz*, mutlaka harf veya alt tire karakteri olmalıdır.
- \$ sembolünün *ilk karakter* olarak kullanılması önerilmez, bu Javascript kütüphaneleri ile birlikte kullanıldığında çakışmalara neden olabilir.
- Javascript *diline has komut ve ifadeler*, isimlendirmede kullanılamazlar.
- Kullanılan isimdeki karakter sayısı *sınırsızdır*.
- *Aynı isimde* iki değişken, fonksiyon ya da nesne aynı web sayfası tarafından aynı anda tanımlanamaz. Ancak fonksiyonların kendi içindeki lokal değişkenler, bir başka fonksiyon içinde yine aynı isimle lokal olarak tanımlanabilirler.

Aşağıdaki tabloda geçerli ve geçersiz isimlendirme örnekleri mevcuttur.

Tablo 9.5. Geçerli ve Geçersiz İsimlendirme Örnekleri

Geçerli isimlendirmeler	Geçersiz isimlendirmeler
<ul style="list-style-type: none"> • isim • ogrenci • Nokta1 • s1 • ses_seviyesi • GENISLIK • A255xl • _hataKodu 	<ul style="list-style-type: none"> • dosya ismi • öğrenci • 1Nokta • s-1 • ses.seviyesi • GENISLIK? • 255Axl • _hata%Kodu • 5000 • function • if

Parantezler

Javascript'teki bazı komutların yapısı gereği, ilgili bazı ifadelerin veya komutların gruplanması gerekebilir. Bu grupları oluşturmak için *normal parantezler* () ya da *küme parantezleri* { } kullanılır. Hangi komutta hangi parantezin kullanılacağı ileride anlatılacaktır, ancak önemli olan açılmış olan bir *parantezin mutlaka kapatılması* gerektiğidir. Aşağıdaki örnekteki her türlü parantezin mutlaka kapatılmış olduğunu görebilirsiniz.

Tablo 9.6. Normal Ve Küme Parantezleri Açıldıktan Sonra Kapatılmalıdır.

JavaScript Komutları
<pre><script> var birSayi=11; if((birSayi % 2) == 0) { alert(birSayi+" sayisi çift sayıdır."); } else { alert(birSayi+" sayisi tek sayıdır."); } </script></pre>

Eğer açılan bir parantez (normal ya da küme parantezi) daha sonra kapatılmaz ise bu *yazım hatası* olarak karşımıza çıkar ve kodlar o noktadan sonra çalıştırılmaz.

Komut Satırları ve Noktalı Virgül

Popüler birçok dile benzer biçimde Javascript'te komutlar *noktalı virgül* karakteri ile biter, ardından bir başka komut başlar. Bu özellik farklı komut ve ifadelerde detaylıca gösterilecektir, ancak aşağıdaki örnekte, her satırın sonunda olan noktalı virgül (;) karakterinin, o komutu bitiren sembol olduğu anlamına geldiği fark edilmelidir. Bu sembol sayesinde bir komutun bittiği anlaşılacağı için, eğer istenirse komutlar bu sembol ile birlikte *yan yana* da yazılabilir. Aşağıdaki iki komut bloku aynı şekilde geçerlidir.

Tablo 9.7. Komut Sonlarındaki Noktalı Virgül Karakteri

JavaScript Komutları
<pre><script> var a=5; var b=4; document.write("toplam= " + (a + b)); </script></pre>
<pre><script> var a=5; var b=4; document.write("toplam= " + (a + b)); </script></pre>



Yazılan kodlarda düzgün hizalama ve isimlendirme, olası hataların yapılmasını engeller.

Benzer biçimde bir komut noktalı virgül ile bittikten sonra, bir sonraki komut başlamadan önce, araya istenildiği kadar **boşluk** bırakılabilir. Aşağıdaki komut bloku bu anlamda geçerlidir.

Tablo 9.7. Gereksiz Boşlukların Yer Aldığı Komutlar

Javascript Komutları
<pre><script> var a=5; var b=4; document.write("toplam= " + (a + b)); </script></pre>

Öte taraftan yukarıdaki 3 alternatiften **ilkini** kullanmanız önerilir, çünkü komutların programcı tarafından **anlaşılır** biçimde yazılması profesyonel hayatta çok önemlidir. Yazılan komutların aradan zaman geçtikten sonra tekrar gözden geçirilmesi gerektiğinde ya da farklı bir programcı tarafından düzenlenmesi gerektiğinde, hizalamalar ve düzgün yazılmış kodlar anlamayı kolaylaştıracaktır.

Nesne Tabanlı Programlama Olanağı

Javascript, nesne tabanlı programlama dilidir. Hem yeni sınıflar ve nesneler yaratmaya, hem de HTML etiketleri ile oluşturulmuş HTML belgesinin nesnelere ulaşarak kodlama yapmayı sağlar.

Öte taraftan bu kitaptaki Javascript diline ayrılmış 3 ünite, nesne tabanlı programlama kapsamında sınıf ve nesne oluşturma ele alınmamıştır. Ancak document.title.ToString() komutunda olduğu gibi, Javascript ile var olan HTML sayfasındaki nesnelerin kullanımı anlatılacaktır.

Nesne tabanlı programlama, yazılan kodların tasarımını kolaylaştırır. Aynı zamanda kodların farklı programcılar tarafından bireysel ihtiyaçlar doğrultusunda tekrar tekrar kullanılmasını sağlar. Son olarak kodların güncellenmesinin kolaylaşmasını ve gereksiz detayların gizlenerek daha hızlı algoritma oluşturulmasını sağlar.



Nesne tabanlı programlama yapabilmek için bu konuda tek başına bir ders alınması tavsiye edilir.



Özet

•HTML sayfalarının içeriği, sayfayı ziyaret eden kişinin talepleri, kullanıcının tarayıcı veya cihazı ile bağlantılı olarak değişik biçimlerde görüntülenebilir. Bunu saptamak üzere HTML içeriklerine müdahale edebilen programlama dilleri geliştirilmiştir. Bu diller sunucu taraflı ve istemci taraflı programlama dilleri olarak ikiye ayrılabilir. Sunucu taraflı diller HTML içeriklerini sunucu üzerinde düzenleyerek sayfa yüklenmeden önce sunucu üzerinde HTML sayfalarını hazırlar. Ardından bu kodlar kullanıcının web tarayıcısına (istemci) gönderilir. Tarayıcıya ulaşan (indirilen) kodlar, web sayfası olarak gösterilmek üzere son kez derlenir. Bu derlenme aşamasında HTML, Javascript ve CSS kodları çalıştırılarak kullanıcının ekranına web sayfası oluşturulur.

•Javascript, istemci taraflı bir dildir. Tarayıcılar tarafından çalıştırılır. HTML etiketleri ile oluşturulmuş web sayfasının içeriğini veya görünümünü ekranda görüntülenmeden önce veya sonra değiştirebilir.

•Javascript Nereye Yazılır?

- Javascript kodları HTML sayfaları içerisinde iki yolla yazılabilir.
- Bunlardan ilki HTML sayfası dışında ayrı bir Javascript kod dosyası oluşturup (örneğin kodlar.js) bu dosyayı HTML sayfası içerisinde çağırma yapar. Bu yöntemde çağırma genellikle HTML sayfasındaki <HEAD>...</HEAD> etiketleri arasında yapılır. Ancak bu zorunlu değildir, sayfadaki başka bir noktada da çağrı yapılabilir. Çağırma komutu HTML etiketlerinden biri olan <script>...</script> ile yapılır. Örneğin <script src="kodlar.js"></script> komutudur, içinde bulunduğu HTML sayfası ile aynı klasörde bulunan "kodlar.js" dosyasını HTML sayfasına dâhil eder.
- İkinci yöntem ise Javascript komutlarını HTML sayfaları içerisine açıkça yazmaktır. Bu yöntemde yine <script>...</script> etiketleri kullanılır. Bu defa bu etiketlerin arasına Javascript komutları açıkça yazılır. Örneğin <script>document.write("Merhaba dünya");</script> satırı, HTML sayfasında nerede kullanılırsa oraya "Merhaba Dünya" ifadesini yazar.

•Javascript Yazma, Çalıştırma ve Test Etme

- Javascript tıpkı HTML sayfaları gibi metin tabanlı bir editörde yazılabilir. Çoğu geliştirici HTML sayfalarını kodlamak üzere kullandığı web editörü ile Javascript kodlamaktadır.
- Öte taraftan herhangi bir metin editöründe (Not Defteri® gibi) bu dosyalar açılıp yazılabilir.
- Javascript kodlarını çalıştırmak için kodları dâhil ettiğiniz HTML sayfasını tarayıcınızda açmalısınız. Bu aşamada tavsiyemiz, popüler olan tüm tarayıcılarda sayfanızı açmanızdır. Bunun sebebi, Javascript'in çok nadir de olsa farklı tarayıcılarda farklı şekilde çalıştırılmasıdır. Hatta günümüzde cep telefonu ve tablet kullanımının artmasıyla denemelerinize bu cihazları da eklemenizi tavsiye ederiz.
- Yazılan Javascript komutlarında yazım (sözdizim) hataları veya mantık hataları olabilir. Çoğu tarayıcı yazım hatalarını algılar ve hatanı olduğu satırdan itibaren Javascript komutlarını çalıştırmayı durdurur. Bu durumda web sayfasının HTML komutları yine de çalışır. Çoğu kullanıcı bunun farkına varmaz, ancak sayfadaki beklenmeyen içerik veya hareketleri fark edebilir. Geliştirici olarak yapmamız gereken tarayıcılardaki "geliştirici seçeneklerini" aktif hale getirip, Javascript hatalarını görmek ve editörümüze dönüp düzeltmeler yapmaktır.
- Eğer kullandığınız editör Javascript diline destek veriyorsa, kod yazma aşamasında yanlış yazımları fark ederek uyarı verir. Tavsiyemiz bu özellikte bir editör kullanmanızdır.



Özet(devamı)

- Kodlarda yapılan mantık hataları ise web sayfasında herhangi bir hata uyarısı neden olmaz ve geliştirici seçenekleri ile de ortaya çıkmaz. Bu hatalar çıktının farklı olması ya da kodların bir süre sonra çalışmaması ile ortaya çıkar. Bu durumda geliştirici tekra rkodlarına dönüp düzeltmeler yapmalıdır. Mantık hatalarından korunmanın en iyi yolu yazılan kodları farklı girdiler veya farklı ortamlarda çalıştırarak olası aksilikleri önleyecek kodları ilave etmektir. Bu aşamada tarayıcıların "geliştirici seçenekleri" yine kodların çalışmasını izlemek için (hata olmasa bile) kullanılabilir.
- Javascript ile Yapılabilecek İşler**
- Javascript ile yapılabilecek işler çok fazladır. Bu kitapta bunlardan en çok tercih edilenleri anlatmaya çalışılmıştır. Detayları daha sonra anlatılacak olsa da geliştiricilere fikir vermek amacıyla aşağıdaki işlerin Javascript ile yapılabilecek işler olduğu bilinmelidir.
- Tarayıcı algılama, web sitesi sahiplerinin farklı tarayıcı kullanan kullanıcılara hizmet sunmasında son derce gerekli bir işlemdir. Farklı tarayıcılara göre farklı içerik, CSS veya Javascript komutları gerekebilir. Bunu algılamak için yine Javascript komutları kullanılır.
- Bilgi girişlerinde eksik ya da hatalı girişleri kontrol etme, bir sonraki adımda hata olabilecek durumları engellemek için yapılması gereken işlemlerdendir. Örneğin eposta adresi girilmesi gereken bir kutunun boş bırakılması ya da eposta adresi olmayan bir ifadenin girişmiş olması Javascript ile kontrol edilebilir ve tespit edildiğinde daha fazla işlem yapmadan kullanıcı uyarılabilir.
- Kullanıcı hareketleri veya tercihlerine göre farklı içerik sunma, sayfalarımıza etkileşim katmanın bir yoludur. Örneğin kullanıcının bir resim ya da metin üzerine fare ile gelmesi sonucunda o resim ya da metnin daha büyük görünmesi gibi etkileşimler Javascript ile yapılabilir.
- Sayfa içeriğinde belirli sürelerde güncelleme yapma günümüzde çok fazla yapılmaktadır. Anlık değişimleri çok olduğu web sitelerinde, her 10 saniyede ya da her 10 dakikada bir içeriğin belirli bir kısmının (tamamı da olabilir) yenilenmesi gerekebilir. Belirli periyotlarda yapılacak bu güncelleme Javascript ve sunucu taraflı dillerin işbirliği ile yapılabilir.
- Matematiksel hesaplamaların tarayıcı ekranında yapılması, kullanıcılar için basit hesaplamalarda farklı yazılımlara gerek duymadan sonucu hesaplamasını sağlar. Javascript ile hesap makinesi gibi oluşturulmuş ekranlar üzerinden işlem yapılabilir.
- Javascript Dilinin Genel Özellikleri**
- Genel sözdizim özellikleri açısından Javascript C/C++ diline benzer. Bu kapsamda isimlendirme, komut yazım stili ve nesne tabanlı programlama desteği açısından bu dillere benzerlik gösterir.

DEĞERLENDİRME SORULARI

1. Aşağıdakilerden hangisi istemci taraflı bir dildir?
 - a) PHP
 - b) ASP.NET
 - c) CSS
 - d) CGI
 - e) ASP
2. Javascript dili ile yazılmış kodlar aşağıdaki programların hangisi ile çalıştırılabilir?
 - a) Not Defteri®
 - b) Adobe Dreamweaver®
 - c) Visual Studio Code®
 - d) Internet Explorer®
 - e) NotePAD++®
3. Aşağıdakilerden hangisi bir Javascript dosyasını HTML sayfasının içerisine doğru şekilde dâhil eden komuttur?
 - a) `<script "kodlar.js"></script>`
 - b) `<script file="kodlar.js"></script>`
 - c) `<script>"kodlar.js"</script>`
 - d) `<script>kodlar.js</script>`
 - e) `<script src="kodlar.js"></script>`
4. Aşağıdakilerden hangisi bir Javascript dosyasını HTML sayfasına dâhil ederek kullanmanın avantajlarından biri değildir?
 - a) Birden fazla HTML sayfasında aynı komutları kullanabilmek
 - b) Komutları sunucu tarafında çalıştırabilmek
 - c) Komutları düzeltmede zaman kazanmak
 - d) Başka sunucular üzerindeki dosyaları dâhil edebilmek
 - e) HTML kodlarında satır sayısını azaltmak
5. Javascript komutları HTML kodları içerisinde nereye yazılmalıdır?
 - a) HEAD etiketleri arasına
 - b) BODY etiketleri arasına
 - c) P etiketleri arasına
 - d) HTML etiketleri başlamadan önce
 - e) SCRIPT etiketleri arasına

6. Javascript komutlarındaki yazım hatalarını tespit etmek için tarayıcıların hangi özelliğini aktif hâle getirmek gerekir?
 - a) CSS desteği
 - b) Tarayıcı geçmişi
 - c) Sık kullanılanlar özelliği
 - d) Geliştirici seçenekleri
 - e) Tam ekran desteği
7. Javascript komutlarında eğer bir yazım hatası bulunursa, sayfa tarayıcıda çalıştırılmak istendiğinde aşağıdakilerden hangisi gerçekleşir?
 - a) Hatalı ve sonraki Javascript komutları çalıştırılmaz.
 - b) Web sayfası görüntülenmez.
 - c) Tarayıcı Javascript komutlarının tümünü çalıştırmaz.
 - d) Tarayıcı ekranına Javascript hatası ile ilgili uyarı mesajı gelir.
 - e) Tarayıcı hatalı kodu düzeltici adım atabiliyorsa düzeltir ve çalıştırır.
8. Aşağıdakilerden hangisi Javascript ile yapılabilecek işlerden biri değildir?
 - a) Matematiksel işlem yapmak
 - b) İstenmeyen tarayıcı algılandığında bilgisayarda farklı tarayıcı açmak
 - c) Her dakika ekrandaki hava durumu bilgisini güncellemek
 - d) Eksik girilen telefon numarasını tespit etmek
 - e) Bir resmi, tıklandığında büyütme
9. Aşağıdakilerden hangisi Javascript'te tanımlanabilecek geçerli bir isimdir?
 - a) RESIM BOYU
 - b) Resim-boyu
 - c) resimBoy
 - d) 40Resim Boyu
 - e) !ResimBoy
10. Javascript komutlarının sonuna hangi karakter konur?
 - a) ;
 - b) .
 - c) !
 - d) /
 - e) \

Cevap Anahtarı

1.c, 2.d, 3.e, 4.b, 5.e, 6.d, 7.a, 8.b, 9.c, 10.a

YARARLANILAN KAYNAKLAR

Javascript Tutorial. 6 Temmuz 2019 tarihinde <https://www.w3schools.com/js/> adresinden erişildi.

Javascript Dersleri. 7 Temmuz 2019 tarihinde <http://javascript.sitesi.web.tr/> adresinden erişildi.