

Yeni Başlayanlar İçin

Gökhan Halimoğlu

JAVASCRIPT

El Kitabı

JAVASCRIPT EL KİTABI

Gökhan Halimoğlu

sanalkurs.net

JavaScript El Kitabı

(*JavaScript Handbook*)

Gökhan Halimoğlu

Sanalkurs Yayınları

Programlama Serisi

Her hakkı saklıdır. Copyright © 2009

Teknik Hazırlık:
Gökhan Halimoğlu

Baskı:
Sanalkurs.net

I. Baskı

www.gokhanhalimoglu.com
www.sanalkurs.net/ghalimoglu

İÇİNDEKİLER

Önsöz.....	2
JavaScript nedir?	3
JavaScript Kod Yazılım	4
Document.write	5
Alert (Uyarı mesajları)	6
Function (Fonksiyonlar)	7
Prompt (Değer girişi).....	9
Confirm (Onay)	11
if/else komut grubu.....	13
SORU	15
Döngüler-for/next.....	16
While Döngüsü.....	18
SORU	20
Nedir Bu Form Elemanları?	21
Nesneler.....	24
Window (Pencere)	25
Location.....	27
Olaylar	28
Zaman fonksiyonları.....	32
KAYNAK DOSYALAR.....	34

Önsöz

Bu e-kitap ile Javascript uygulamaları geliştirmek ve orta seviye Javascript kodlama bilgisine sahip olmak mümkün.

Konulara ayrılmış olarak, seviye seviye dersler mevcut. Sırası ile gittiğiniz, kodları uyguladığınız sürece siz de Javascript uygulamaları geliştirebilirsiniz.

Bu kitap bir “JavaScript El Kitabı” olarak değerlendirilebilir. Diğer kitaplardaki sunumdan değişik bir tarz işlemeye çalıştım. Hatırlamak istediğiniz, öğrenmek istediğiniz, işinize yarayacak tüm Javascript kodlamalarını derleyerek, aşama aşama içindekiler kısmıyla birlikte sizlere aktarmaya çalıştım.

Umarım yararlı olur.

JavaScript nedir?

C dilinin tarayıcılara uyarlanmış hali olarak adlandırabiliriz. Web sayfalarında dinamik içerik sağlamak ya da kullanıcıyla iletişim kurmak için kullanılmaktadır. Ad benzerliğine karşın '[Java](#)' ile ilişkili değildir. Yaygın yanlışlardan biri JavaScript'in *basitleştirilmiş Java* olarak tanımlanmasıdır. Başlangıçta yalnız Netscape tarayıcıları üstünde çalışan JavaScript artık tüm yaygın ağ tarayıcıları tarafından desteklenmekte. (Vikipedi'den)

JavaScript Kod Yazalım

Bir Web sayfasında JavaScript ile kod yazabilmek için ilk önce tarayıcıımıza kullanacağımız script dilini tanıtmamız gerekir.

```
<script language="JavaScript">
```

İçerik

```
</script>
```

Kodlarımızın HTML değil JavaScript olduğunu gösterir. Kodlarımızı yazmak için gerekli tag'leri hazırladıktan sonra Javascript ile kodlama yapabiliriz.

Document.write

JavaScript'te Sayafaya yazı yazdırmak için **document.write** kullanılır. Aşağıdaki gibi kullanılır.

```
document.write("Merhabalar")
```

Örnek sayfamıza yazdırmak için, temel komutlarımız ile birlikte yazalım.

```
<script language="javascript">  
document.write("Merhabalar")  
</script>
```

İlk Javascript kodlamamıza başlamış olduk.

Alert (Uyarı mesajları)

JavaScript'te uyarı mesajları yazmak için “alert” kullanılır. Aşağıdaki gibi kullanılır.

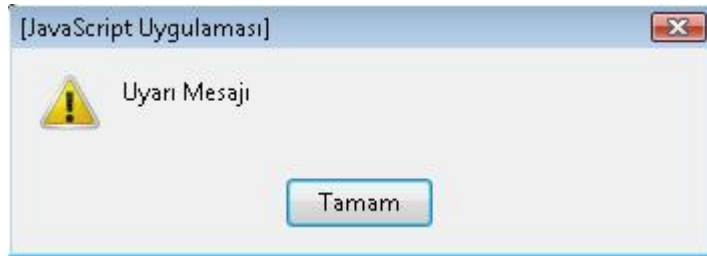
```
Alert(“mesajınızı buraya yazın”)
```

```
<script language=“javascript”>
```

```
Alert(“Uyarı mesajı”)
```

```
</script>
```

Gelecek uyarıya “Tamam” butonuna basarak programın bir alt satırdan çalışmasına devam etmesini sağlarız.



Function (Fonksiyonlar)

Javascript fonksiyonları çağırıldıkları zaman işlemeye başlarlar. Aşağıda bir fonksiyon tanımlamayı öğrenelim.

Fonksiyonlar;

```
{  
  
}
```

Tag'leri arasına yazılırlar.

Örnek bir fonksiyon belirleyelim ve “alert” ile uyarı mesaj’ı gönderelim ziyaretçilerimize.

```
function mesaj()  
{  
  alert(“merhaba”)  
}
```

Örnekler;

```
<script language=”javascript”>  
function mesaj()  
{  
  alert(“merhaba”)  
}  
</script>
```

```
{  
alert("merhaba")  
}
```

İçerikler parantezlerin içerisine yazılır.

```
<script language="javascript">  
function mesaj()  
{  
document.write("merhabalar")  
}  
</script>
```

Hazırlamış olduğumuz bu fonksiyonlarımızı daha sonra istediğimizde çağırabiliriz. Fonksiyonlarımızı çağırma işlemine ilerleyen konularda bahsedilecektir.

Prompt (Değer girişi)

Prompt kullanıcıdan bilgi girişleri için kullanılır. Ziyaretçilerinizden bir değer, sayı, karakter v.b. istenmesi durumlarında “prompt” kullanırız.

Nasıl kullanırız?

```
var deger=prompt(“değer giriniz..”,”lütfen sadece rakam”)
```

Kullanıcılarımızdan girilmesini istenen bilgileri kullanacağımız için bir değişkene atarız. Örneğin yukarıda “Deger” ismiyle bir değişken belirledik ve kullanıcıdan girileni

Deger=kullanıcının bilgisi olarak eşitledik.

```
<script language=”javascript”>  
var deger=prompt(“değer giriniz..”,”lütfen sadece rakam”)  
</script>
```

Yukarıdaki girilen bilgiyi, öğrendiğimiz yöntemler ile sayfamıza yazdıralım.

```
<script language=”javascript”>  
var deger=prompt(“değer giriniz..”,”lütfen sadece rakam”)  
Document.write(deger)  
</script>
```

Bunu bir fonksiyon içerisine yerleştirelim ve sayfamız açılır açılmaz çağıralım.

```
<script language="javascript">
function mesaj()
{
var deger=prompt("değer giriniz..","lütfen sadece rakam")
document.write(deger)
}
</script>
```

Sayfamız açıldığında istediğimiz fonksiyonu yüklemek için ;

```
<body onload="mesaj()"></body>
```

“onload” komutu ile sayfamız açıldığı anda fonksiyonumuz işlem yapar.

Şimdi sıra size, kodlarımız son olarak şu hali aldı ve sayfamız açıldığında bizden bilgi girişi istenilecek. Kullanıcıdan alınan bilgiyi sayfaya yazdıracaktır.

```
<script language="javascript">
function mesaj()
{
var a=prompt("Kelime veya cümle giriniz","lütfen sadece kelime giriniz..")
document.write(a)
}
</script>
<body onload="mesaj()">
</body>
```

Confirm (Onay)

Confirm Kullanıcı ile iletişime geçer ve ona, "Devam edilsin mi ?" sorusunu sorar. Örneğin bir link verdiniz ve ziyaretçinin siteyi terk edeceği bilgisini veriyorsunuz sonuç olarak evet/hayır butonları ile ziyaretçinin cevabını işleme dönüştürebiliriz.

```
confirm("Devam edilsin mi ?")
```

Confirm'de sonuç, evet ise şunu yap değilse şunu yap gibi şartlar belirtilebilir. Ancak biz "if" yapısını öğrenmediğimiz için tek bir örnek ile konumuzu kapatıyoruz.

Örnek;

Örnek, kullanıcıdan istenen "Sayfa'dan çıkılsın mı?" örneğidir. Kullanıcı onaylarsa, sayfa kapatılır. Onaylamaz ise kullanıcıya mesaj gönderilip iptal edildiği belirtilir.

```
function kapat()
{
    if (confirm("Sayfamızdan çıkmak istediğinizden emin misiniz?"))
    {
        parent.close();
    }
    else
    {
        alert("İptal edildi."); {
    }
}
```

Yukarıdaki kodları sayfanıza kopya edebilir sayfanızda deneyebilirsiniz. İleriki konularda **if-else** yapısından bahsedeceğiz, ancak o zaman bu örneği anlayabilirsiniz. Şimdilik sadece örneği inceleyip deneyebilirsiniz.

Örnek;

```
<a href="http://ghalimoglu.com" onclick="return  
window.confirm('Gökhan Halimoğlu'nun sayfanıza gitmek  
istediğinizden emin misiniz?');">Ana Sayfam</a>
```

if/else komut grubu

If/then/else olarak bilinen komut grubu, JavaScript'te **if/else** olarak kullanılmaktadır. Bu komut grubu iki değeri karşılaştırır ve sonuca göre farklı işlemler yapmanızı sağlar.

İngilizce de “**if=eğer**” “**else=değilse**” anlamında kullanılmaktadır.

If/else kullanımı öğrenmeden önce, karşılaştırma operatörlerini öğrenelim.

Denk	=
Denk değil	!=
Küçük	<
Büyük	>
Küçük eşit	<=
Büyük eşit	>=

Ayrıca “**++**” Tırnak içerisindeki **++** operatörü birer artırmayı sağlar.

If (değer==değer)

Karşılaştırma operatörleri kullanılarak, parantez içerisinde belirtilirler. Aşağıdaki örnekten incelemeye devam edelim

```
function mesaj()
{
var a=prompt("Kelime veya cümle giriniz","lütfen sadece kelime giriniz..")
if(a=="gökhan")
alert("merhaba Gökhan.")
else
alert("Gökhan yok mu?")
}
```


Yukarıdaki örnekte, bir fonksiyon hazırladık ve fonksiyon içerisinde kullanıcıdan isim girmesini istedik.

Kullanıcı tarafından girilen bilgiyi, değişkenimiz ile karşılaştırıyoruz

“a”.

```
İf (a=="gökhan") ise;  
Alert("merhaba Gökhan")  
değilse,  
alert("Lütfen sonra tekrar deneyin")
```

şeklinde bir mesaj göndermiş oluruz.

Komutumuzu *.html olarak kaydedebilmeniz için aşağıda derlenmiş hali bulunmaktadır.

```
<html>  
<head></head>  
<script language="javascript">  
function mesaj()  
{  
var a=prompt("Kelime veya cümle giriniz","lütfen sadece kelime  
giriniz..")  
if(a=="gökhan")  
alert("merhaba Gökhan.")  
else  
alert("Gökhan yok mu?")  
}  
  
</script>  
<body onload="mesaj()">  
  
</body>  
</html>
```

SORU

Şimdi sizlerden aynı mantık ile sayfanız için girilen rakamın 5'ten **büyük ya da küçük** olduğunu söyleyebilecek bir program yazmanızı istiyorum.

Döngüler-for/next

Döngüleri bir öbeği birden çok tekrarlamak için kullanırız. Örneğin bir ismi alt alta binlerce defa yazdırabiliriz.

Genel olarak kullanım biçimi şu şekildedir.

```
for(başlangıç; koşul; artma sayısı)  
{  
Döngüdeki işlemler...  
}
```

Örnek;

```
<html>  
<body>  
<script language="JavaScript">  
  for(a=1;a<10;a++)  
  {  
    document.write(a+"<br>")  
  }  
  alert("döngü tamamlandı!")  
  
</script>  
</body>  
</html>
```

```
for(a=1;a<10;a++)
```

Döngü koşulları

<pre>{ document.write(a+"
") }</pre>
Döngüdeki işlemler

Yukarıdaki örnekte, alt alt' **a=1** 'den başlayarak **a<10** olasıya kadar,

1
2
3
4
5
.
.
.
9

Şeklinde 9'a kadar devam edecektir. Ve döngü tamamlandıktan sonra "**döngü tamamlandı!**" şeklinde bir uyarı gönderecektir.

UYARI:

alert("döngü tamamlandı!") ifadesini, döngü içerisine almış olsaydık; sayılar her ekrana yazılışında uyarı iletilisiyle karşılaşılıyor olacaktık. Bunla karşılaşmamak için "**}**" tag dışarısına alert vererek döngünün bittiğini kullanıcıya bildirebiliriz.

While Döngüsü

Bu döngüde döngüdeki işlemlerin kaç kere yapılması gibi şeyler söz konusu değil. **While** döngüsünde durum gerçekleştiği sürece döngü devam eder.

Kullanımı;

```
While(a<11)
{
İçerik...
}
```

Yukarıdaki gibi, döngümüzde “**a<11**” şartı devam edene kadar döngü devam eder.

Örnek;

Aşağıdaki örneğimizde verdiğimiz koşul “10” olduğu için 1’den 10’a kadar olan sayıların kareleri toplamını verecek olan bir örnektir.

```
<html>
<body>
<script language="javascript">
var a=0
var c=0
while (a<10)
{
a++
b=(a*a)
c=c+b
}
document.write(c)
</script>
</body>
</html>
```

var a=0 var c=0

Değişkenler

{ a++ b=(a*a) c=c+b }

İçerik

SORU

Kullanıcı tarafından girilen rakam ve kelimemiz olsun. Bu bilgiler doğrultusunda girilen rakam kadar ekrana kelimeyi alt alta yazdıran bir program yapmanızı istiyorum.

Nedir Bu Form Elemanları?

Form elemanları kullanıcıdan bilgi girişlerini kontrol edebildiğimiz buton, text, radyo düğmeleri v.b. elemanlardır.

Javascript uygulamalarında form elemanları üzerinde fazla durmayacağım, genelde ASP ve PHP gibi diller içerisinde form bilgilerini almak için kullanırız. Burada içerisine girilen bilgiyi kayıt edemeyeceğimiz için ben aktarımlar, devretme, işleme gönderme tarzı örnekler sunup yolumuza devam etmeyi düşünüyorum.

Bu örneğimizde bir kullanıcı giriş (login) sistemi geliştirelim. Başlayalım isterseniz ilk olarak form elemanlarından (text) in nasıl aktarılabilceği üzerinde durmak istiyorum.

Bir form oluşturalım: Bunu için sadece “parola” istense yeterli olacaktır sanırım.

```
<form name="goko">
<tr><td><input type="password" name="gok">
<input type="button" value="Giriş" onclick="oku()"></td></tr>
</form>
```

Formumuzu oluşturduk. Burada **dikkat etmemiz gereken form’a ve oluşturduğumuz form elemanlarına (name=’’) isimlendirme verilmesidir.**

Şimdi formumuz hazır olduğuna göre, yapmamız gereken bilgileri içerisine girilen bilgileri kullanabilmek.

Aşağıda yazacağımı adınız gibi bilmeniz gerekiyor:

```
document.FORM_ADI.PASS_ADI.value==” “
```


Yukarıdaki tablo içerisinde sorguya göre, kendi form elemanlarımızın adını düzenleyelim.

```
document.goko.gok.value
```

Artık kullanıcı giriş sistemi (login) için gerekli algoritmayı alalım.

İlk olarak text'e girilen bilgiyi kontrol etsin, doğruysa "işlem" değilse "işlem" işte bu kadar basit. **if/else** diyerek kodlamamıza başlayalım o zaman.

Ben "oku()" adından bir fonksiyon hazırlıyorum sizlerde istediğiniz isimle bir fonksiyon açın ve if/else kodlarımızı yazın.

```
<script>
function oku()
{
if (document.goko.gok.value=="123")
document.write("Süper!")
else
document.write("Başarısız oldunuz.")
}
</script>
```

Function oku()
Oku adında bir fonksiyon açtık.

if (document.goko.gok.value=="123")

Eğer;text'in içine girilen bilgi "123" ise
 Documen.write("süper" değilse
 Document.write("başarısız oldunuz")
 Yaz demiş olduk.

Giriş(login) sistemimizin son halini aşağıdaki kodlarda bulabilirsiniz.

```
<html>
<head>
</head>
<script>
function oku()
{
if (document.goko.gok.value=="123")
document.write("süper!")
else
document.write("başarısız oldunuz!")
}
</script>

<body><center>
<table border="0" bordercolor="black">
<center><br><br>

<tr><td></td></tr>
<form name="goko">
<tr><td><input type="password" name="gok">
<input type="button" value="Giriş" onclick="oku()"></td></tr>
</form>
</table>
</body>
</html>
```

Nesneler

Javascript içerisinde kendi hazırlayacağımız nesneler dışında belirli kalıplaşmış nesneler vardır. Bunlar kodlamalar içerisinde çok işimize yarayacaklardır. Bunlardan önemli olan birkaçından bahsetmek istiyorum.

- frame, self, _top
- location
- submit
- reset
- location

Bunlardan bazılarıdır.

Window (Pencere)

Pencere (Window) üzerinde sadece nesneler değil olaylardan da bahsetmek istiyorum.

Öncelikle yeni bir pencere açalım.

```
Window.open
```

Window.open("yeni pencere aç") olarak kullanacağız ancak bu sadece istediğimiz pencereyi açacaktır. Şimdi bu pencere üzerinde istediğimiz özellikleri sunmayı öğrenelim.

İlk olarak pencere komutlarımızdan bahsedelim sonra üzerinde uygulamaya geçebiliriz.

```
window.name ; Başlığınız  
toolbar ; toolbar'ın gösterilme özelliği (yes/no )  
status; statusbar'ın gösterilme özelliği (yes/no )  
menubar; menubar'ın gösterilme özelliği (yes/no )  
scrollbars; scrollbar gösterilme özelliği (yes/no )  
resizable; açılacak olan ekranın boyutunun değiştirilebilirliği (yes/no)  
width; genişlik  
height; yükseklik
```

Şimdi yeni bir sayfa açalım ve istediğimiz özellikleri aktaralım.

```
window.open("http://ghalimoglu.com", "göko", "toolbar=no,  
status=no, menubar=yes,scrollbars=yes,resizable=no,width=760,  
height=460")
```

Şimdi penceremizi kapatalım;

```
Window.close();
```

Komutumuz yukarıdaki gibidir. Fonksiyon olarak hazırlayabilir gövdede veyahut butonda çalıştırabiliriz.

Veya;

```
<a href="javascript:window.close();">Pencereyi  
kapat!</a>
```

Şeklinde link olarak da sayfamızı kapattırabiliriz.

Window.close yerine kullanabileceğimiz iki seçeneğimiz daha var:

- parent.close
- self.close

Diğer iki seçenekte aynı kullanım şekillerinde de kullanılabilir.

Location

Ne işe yarıyor bu location?

Yönlendirme işlemlerinde kullanıyoruz location'ı.

Nasıl kullanıldığına gelince, örnek bir pencere yönlendirmesi yazalım; kullanımı gayet basittir.

```
window.location.href="url"
```

Veya istenilen sayfa için;

```
window.location.href=ders.html
```

şeklinde olabilir.

“location” nesnesini, örneğin az önce yapmış olduğumu login (kullanıcı giriş sistemi)’inde kullanabiliriz.

Sayfaya yazı yazdırma yerine, kullanıcıya giriş yaptırdıktan sonra istenilen sayfaya yönlendirmesini söyleyebiliriz.

Aynı kodlama ile şimdi sizlerin kullanıcı giriş sistemimizi geliştirmenizi istiyorum.

Girilen şifre doğruysa web sitenizin istenilen sayfasına yönlendirilsin, yanlış ise “yalnız girdiniz” şekilde bir uyarı iletisi göndermesini istiyorum.

Olaylar

Onload:

Onload; sayfamızın açıldığı anda hangi işlemi yapacağını belirttiğimiz olay biçimidir.

Daha açıklayıcı olmak gerekirse, yazmış olduğunuz bir fonksiyonu sayfa açıldığında anda çalıştırmak istiyorsanız onload olayını kullanabilirsiniz.

Örnek;

Bir fonksiyon hazırlayalım ve bunu sayfamız ilk açıldığında “onload” ile çağıralım.

```
function mesaj()  
{  
  alert(“sitemize hoş geldiniz”)  
}
```

Yukarıdaki fonksiyonumuzda, kullanıcıya Uyarı halinde “hoş geldiniz” mesajı sunmuş olacağız. Bunu “onload” olayı ile sayfamız ilk açıldığında gerçekleştirelim.

```
<html>  
<body onload=”mesaj()”>  
</body>  
</html>
```

Html tagleri arasına yerleştirdik ve şimdi `<script></>` taglerimizle bütünleştirerek sayfamızın tamamını oluşturalım.

Örnek onload uygulamamız;

Web sitemize giriş yapıldığı anda, hoş geldiniz mesajı ile ziyaretçilerimizi karşılayacağız.

Örnek uygulama;

```
<html>
<script language="javascript">
function mesaj()
{
alert ("Sitemize hoşgeldiniz.")
}
</script>
<body onload="mesaj()">
</body>
</html>
```

OnMouseOver/Out:

Belirlediğimiz cisim alan içerisinde Mouse ile üzerine gelindiğinde istediğimiz işlemi yapabileceğimiz bir komut over/out olarak kullanıldığında haber başlıkları/resim slaytları v.s. hazırlanabilir.

Nasıl kullanıldığını bir göz atalım.

```
onMouseOver="işlem"
```


İşlem yazan yer üzerinde, fonksiyonlar, komutlar kullanabiliriz. Bu aşamaya gelmişken birde /out/ olayından bahsedip bir duyuru sistemi örneği yapmayı öğrenelim.

OnMouseOut=""işlem""

Şimdi bununla ilgili üzerin gelindiğinde Uyarı mesaj'ı veren bir link örneği göstermek istiyorum. Daha sonra over/out örneğinin bir arada bulunduğu haber başlığı şeklinde bir örnek ile devam edeceğim.

```
<html>
<head>
<script language="JavaScript">
    function goko()
    {
        alert("Merhabalar");
    }
</script>
</head>
<body>
<a href="" onmouseover="goko()">ghalimoglu.com</a>
</body>
</html>
```

Bu örneğimizde sayfaya link ekledik ve üzerine gelindiğinde bir uyarı mesaj'ı çıkmasını göstermiş olduk. Herkes kodları yazıp deneyebilir.

Duyuru Örneği;

```
<html>
<body>
<marquee onmouseover="this.stop()"
onmouseout="this.start()">sanalkurs.net</marquee>
</body>
</html>
```

```
onmouseover="this.stop()"
```

Mouse üzerindeyken
("dur!")

```
onmouseout="this.start()">
```

Mouse üzerinden
çekildikten sonra ("devam
et!")

Aynı yöntem ile, sağdan-sola doğru kayan resimler, linkler şeklinde de olabilir.

Yönlendirmeleri

```
direction=" "
```

Komutu ile istediğiniz alana yapabilirsiniz.

Örneğin **down** yazarsanız aşağı yönde gidecektir.

Zaman fonksiyonları

Zaman fonksiyonlarında ekranımıza, hangi gün, hangi ay gibi bilgileri aktarabileceğiz.

Bunlardan önce ilk kodumuz,

```
new date()
```

Bu kod ile değişkenin zaman değeri taşıdığını belirteceğiz.

```
getDate() Ayın kaçı olduğunu gösterir.  
getMonth() Hangi ayda olduğumuzu gösterir. Ocak için 0, Şubat için  
1, Mart için 2, ...,  
getHours() Saat değerini ifade eder. 24'lük sistem kullanılır.  
getMinutes() Dakika değerini ifade eder.  
getSeconds() Saniye değerini ifade eder.
```

Şimdi, değişkenimizi belirleyelim ve scriptimizi yazalım, ekranda bir çıktı alalım.

```
<html>
<body>

<h3>Bulunduğumuz yıl:</h3>
<script language="JavaScript">
var zaman = new Date()

document.write(zaman.getFullYear())
</script>

</body>
</html>
```

Yukarıdaki script'te sadece yıl için gösterdim. Siz diğerleri için de uygulayabilirsiniz.

KAYNAK DOSYALAR

Hazırlamış olduğumuz dosyaları aşağıdakiler linklerden birinden indirebilirsiniz.

<http://rapidshare.de/files/46346652/oernekler.rar.html>

<http://vip-file.com/download/9e2a63929038/--rnekler.rar.html>

<http://www.2shared.com/file/5176317/a0fe5a52/rnekler.html>

http://s1.dosya.tc/_rnekler.rar.html

<http://www.uploadyeri.com/download.php?file=d9d0b5bd4593370f90bb2ec8331ee6b0>

Sonunu getirdiyseniz kendiniz için atmış olduğunuz adımı başarı ile bitirmişsiniz demektir.

Umuyorum herkese az da olsa bir şeyler katabilmişimdir.