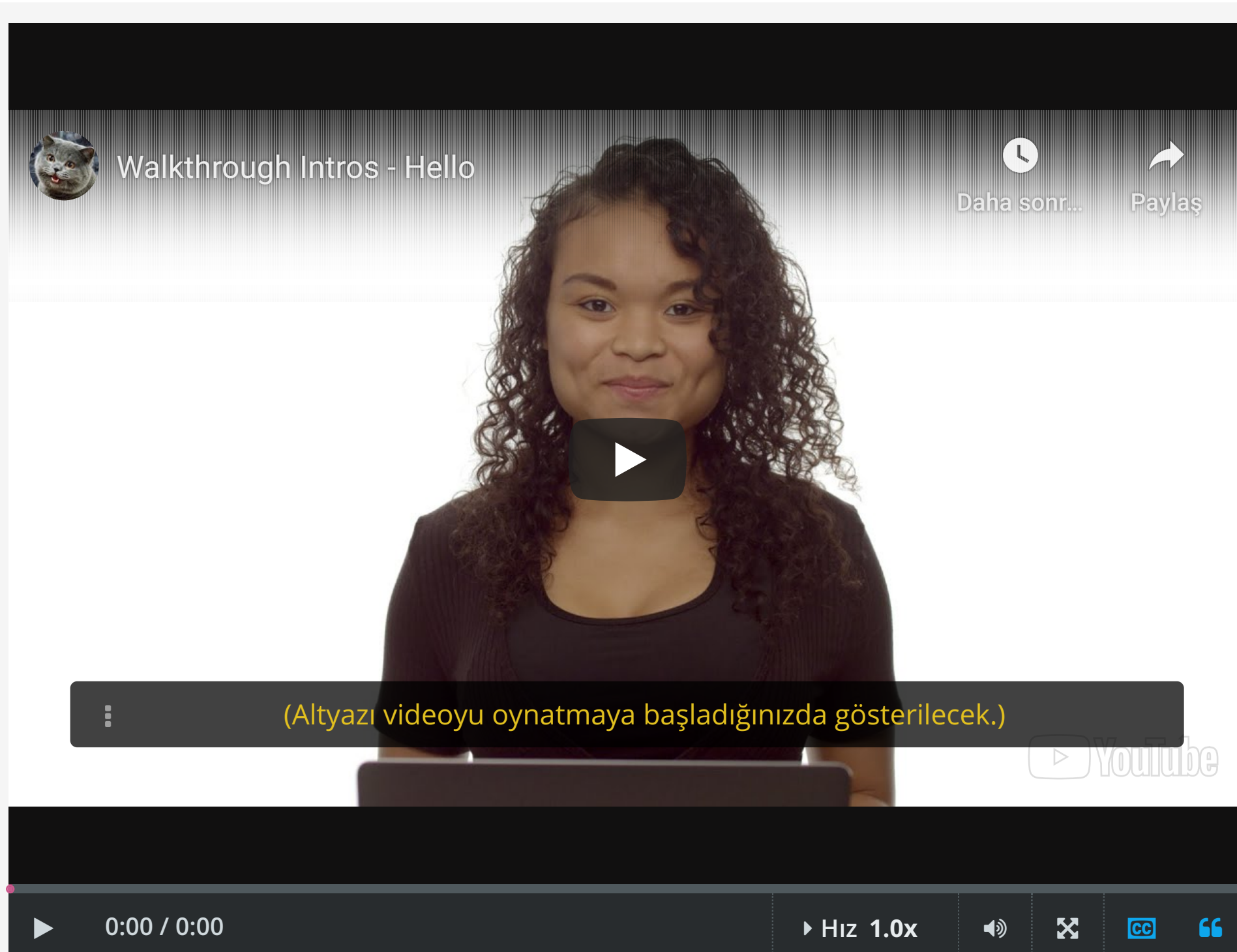




"Hello" Görevi Tanımı

[Bu sayfaya yer imi koy](#)

Video



Altyazının başlangıcı. Sona atla.

Merhaba dünya. Birlikte ilk C programımızı yazacağız.

Merhaba dünya, Dünya'daki en ünlü programlama yapıtaşlarından birisidir.

Bilgisayar bilimi yolculuğuna başlarken yapılan bir çeşit ritüeldir.

Bu problemde en basit C programlarının nasıl inşa edilebileceğini göreceğiz

İlk başta her bir kod satırını anlayamayabilirsiniz ve bu sorun değil.

Endişeye gerek yok, yakında anlayacaksınız.

O halde hadi başlayalım ve Merhaba Dünya diyelim.

Öncelikle CS50 Sandbox'ta [bu sayfayı](#) açın ve aşağıdakileri takip edin:

Dosyaları Listelemek

Hello, world! (Merhaba dünya!)

Yukarıda verdiğimiz linki açtığınızda, yazı editöründe (text editor) sağda, hello.c adındaki bir dosyada C ile yazdığımız ilk program duruyor.

Klasör ikonuna tıklayın, ve göreceksiniz ki hello.c şu an için bulunan tek dosya. O ikonun altındaki /root/sandbox'tan bahsetmek gerekirse, hello.c, sandbox adlı bir klasörde ("directory" olarak da bilinir) bulunmaktadır, bunun kendisi de root adında başka bir klasörün içindedir. Bunların hepsini gizlemek için klasör ikonuna tekrar tıklayın.

Sonra, aşağıdaki *terminal penceresi*'nde, *komut istemi (prompt)* olarak da bilinen dolar işaretinin (\$)hemen sağına, aşağıdakini tam olarak (küçük harflerle) yazın, ve Enter tuşuna basın.

```
ls
```

Sonuç olarak sadece hello.c görmelisiniz. Bunun nedeni, sizin aynı klasördeki dosyaları listemiş olmanızdır. Ancak bu sefer, o klasör simgesi ile temsil edilen grafik kullanıcı arayüzü (graphical user interface - GUI) kullanmak yerine, sadece klavyenizdeki tuşları kullanarak, komut-satırı arayüzü (command-line interface - CLI) kullanınız. Bilhassa, "liste" (list) için bir kısayol olan ls komutunu çalıştırdınız (execute ettiniz). (Bu o kadar çok kullanılan bir komut ki, bunu yazarlar adını sadece ls olarak tanımlayarak birkaç tuş basımını azaltmak istemişler.)

Buradan itibaren, bir komutu execute etmek (çalıştırmak) onu terminal penceresine yazmak ve daha sonra enter tuşuna basmak anlamına gelecektir. Komutlar büyük küçük harfe duyarlıdır (case-sensitive), o nedenle büyük harf yerine küçük harf veya tam tersi şekilde yazmayın.

Programları Derlemek (Compiling)

Şimdi, bir programı çalıştırmadan önce, hatırlayın, onu bir compiler (derleyici) ile derlememiz (clang gibi), yani *source code* 'dan (kaynak koddan) *machine code* 'a (makine koduna, sıfır ve birlere) çevirmemiz gerekmektedir. Aşağıdaki komutu tam da bunun için çalıştırın:

```
clang hello.c
```

ve sonra ad bunu tekrar çalıştırın.

```
ls
```

Bu sefer sadece hello.c değil aynı zamanda a.out da listelenmiş olarak görmelisiniz (aynısını grafik olarak da o klasör ikonuna tıklayarak görebilirsiniz). Bunun sebebi, clang 'ın hello.c içindeki kaynak kodu a.out içindeki aynı zamanda daha sonra bahsedeceğimiz "assembler output" yerine geçen, makine koduna çevirmesidir.

Şimdi programı aşağıdaki komutu execute ederek çalıştırın:

```
./a.out
```

Hello, world, indeed! (Gerçekten de Hello World!)

Programları İsimlendirmek

Şimdi, a.out bir program için en kullanıcı dostu isim değil. Haydi hello.c 'yı tekrar derleyelim, ama bu sefer makine kodunu daha uygun bir şekilde hello adındaki bir dosyaya kaydederek. Aşağıdaki komutu girin:

```
clang -o hello hello.c
```

Aradaki boşlukları atlamamaya dikkat edin! Daha sonra aşağıdaki komutu girin:

```
ls
```

Şimdi sadece hello.c (ve daha önceki a.out) değil aynı zamanda da hello 'yu da listelenmiş olarak görmelisiniz. Bunun sebebi, -o 'nun bir komut satırı değişkeni (command line argument) - bazen *flag* ya da *switch* olarak da bilinir - olması, yani clang 'e çıktı olarak (output'un o'sundan geliyor) hello adında bir dosya oluşturtmasını söylüyor Aşağıdaki komutu girerek yeni adlandırdığınız dosyayı deneyin:

```
./hello
```

Tekrardan Hello World!

İşleri Kolaylaştırmak

Hatırlayın, clang'ı çalışma prosedürünü make 'ın bizim için nasıl yapılacağını bulmasına izin vererek otomatik hale getirebiliyorduk, ve bu bizi birkaç fazla tuşa basmaktan kurtarıyordu. Aşağıdaki komutu girin ve bu programı son bir kez derleyin:

```
make hello
```

Fark etmişsinizdir ki aslında make , clang komutunu daha da çok komut satırı değişkenleri ile çalıştırıyor, değil mi? Bunun hakkında daha fazlası bir başka sefere!

Şimdi programın kendisini son bir kez aşağıdaki komutu girerek çalıştırın:

```
./hello
```

Oh be!

Şimdi sizin ödeviniz:

Ödevinizi, bir sonraki ekranda çıkan kutuya kodunuzu yazarak göndereceksiniz. Ödevinizi bize göndermeden, aşağıdaki ödev tanımını okuduğunuzdan emin olun. Sonra da gönderirken dikkat etmeniz gerekenler şunlar:

- Bu ödevden bir puan alacaksınız ve ilerleyişinize işlenecek.
- Ödevi sistemde bize göndermeden önce, mutlaka **CS50 Lab'de** test etmenizi öneririz.
- Eğer sonucunuz yanlışsa 0 puan, doğruysa 1 puan alacaksınız. Doğru yapana kadar birkaç kez deneme şansınız var.
- Eğer sonucunuz yanlışsa, aşağıda "See Output" kısmından neler olduğunu inceleyebilirsiniz.
- Cevabınız doğru olduğunda, mutlaka Gönder tuşuna basıp sonucu bize göndermeyi unutmayın!**
- İnternet bağlantınız ve problemlere bağlı olarak, sonuçların otomatik değerlendirilmesi bazen uzun sürebilir. Bu durumda 5-10 dakika beklemeniz gerekebilir. Eğer çalışmıyorsa sayfanızı yenileyebilirsiniz.

Ödev: Kullanıcıdan Girdi Alma

Yukarıda program, nasıl derleyip çalıştırırsanız çalıştırın ancak hello, world yazdırıyor (print ediyor). Sizin ödeviniz aynı ders videosunda yaptığımız gibi bunu azıcık kişiselleştirmek olacak. **Bu programı öyle bir değiştirin ki kullanıcıdan öncelikle ismini girmesini istesin (prompt). Daha sonra da hello, bilmem-kim, yazdırsın - burada bilmem-kim, kullanıcının gerçek ismi olacak.**

Öncesindeki gibi, programınızı aşağıdaki komut ile derlediğinizden emin olun:

```
make hello
```

Ve yine programınızı aşağıdaki komut ile birkaç kez farklı girdiler ile çalıştırın:

```
./hello
```

Ekibimizin Çözümü

Ekibimizin bu probleme getirdiği çözümü denemek için [bu sandbox](#) içinde, karşınıza çıkan boş ekranda aşağıdaki komutu çalıştırın:

```
./hello
```

Ödeviniz için ipuçları:

Kullanıcılara isimlerini nasıl soracağınızı hatırlamıyor musunuz?

Hatırlayın ki get_string'! şu şekilde kullanabilirsiniz, onun *return value* olarak geri döndürdüğü değeri string tipinde name isminde bir değişkende saklayarak.

```
string name = get_string("What is your name?\n");
```

Bir stringin formatını nasıl düzenleyeceğinizi hatırlamıyor musunuz?

Kullanıcının adını bir karşılama mesajı ile nasıl birleştireceğinizi (concatenate) bilmiyor hatırlamıyor musunuz? Unutmayın ki printf 'i sadece yazdırmak için değil aynı zamanda bir stringi düzenlemek için de kullanabilirsiniz (printf içindeki f bu nedenle format kelimesinden gelir) tıpkı aşağıdaki gibi, name bir string olarak kullanılarak:

```
printf("hello, %s\n", name);
```

Undeclared identifier (belirtilmemiş tanımlayıcı) kullanımı?

Aşağıdaki hatayı, diğer hataların en üstünde mi görüyorsunuz?

```
error: use of undeclared identifier 'string'; did you mean 'stdin'?
```

Hatırlayın, get_string kullanmak için, dosyanın en üstüne cs50.h (get_string 'ın içinde tanımlandığı) eklemeniz gerekmektedir, aşağıdaki gibi:

```
#include <cs50.h>
```