

1.Ders : C Dilini Tanıyalım

C ile Programlamayı Öğrenelim

C güçlü bir programlama dilidir. Hızlı, taşınabilir ve tüm platformlara uygundur.

Programlamada yeniyseniz, C programlama yolculuğunuza başlamak için iyi bir seçimdir.

Bu site, C programlama dilini öğrenmeye nasıl başlayacağınız, neden öğrenmeniz gerektiği ve nasıl öğrenebileceğiniz konusunda kapsamlı bir rehberdir.

İÇİNDEKİLER TABLOSU

[C Programlama İçerik Listesi \(Index\)](#)

[C Programlama Nedir?](#)

[C Öğrenme Sebepleri](#)

[İlk C Programınız](#)

[Hatırlanacak şeyler](#)

C Programlama Nedir?

C ile programlamaya başlamadan önce, önce dili tanımaya başlayalım.

C, Windows ve iOS gibi İşletim sistemlerinden 3D filmler oluşturmak için kullanılan yazılıma kadar geniş bir uygulama

yelpazesi için kullanılan genel amaçlı bir programlama dilidir.

C ile programlama oldukça verimlidir. 40 yaşından büyük olmasına rağmen popüler olmasının ana nedeni budur.

C programları taşınabilir. Bir sistemde yazılmış kaynak kodu, herhangi bir değişiklik yapmadan başka bir işletim sisteminde çalışır.

Belirtildiği gibi, programlamayı öğrenmeye başlamak için iyi bir dildir. C programlamayı biliyorsanız, yalnızca programınızın nasıl çalıştığını anlamayacak, aynı zamanda bir bilgisayarın nasıl çalıştığına dair zihinsel bir resim oluşturabileceksiniz.

[C Programlama Dilinin Tarihçesi](#)

[C Programlama Dilinin Özellikleri](#)

C öğrenirseniz ne kazanacaksınız?

1 – Bir bilgisayarın nasıl çalıştığını anlayacaksınız.

C'yi biliyorsanız, yalnızca programınızın nasıl çalıştığını bilmezsiniz, aynı zamanda bir bilgisayarın nasıl çalıştığını hakkında zihinsel bir model oluşturabilirsiniz. Python ve Java'dan farklı olarak C'nin sağladığı özgürlüğü de

göreceksiniz.

C, daha önce asla mümkün olmadığını düşündüğünüz programları yazmanıza izin verir veya en azından bir bütün olarak bilgisayar mimarisi ve programlaması hakkında daha geniş bir anlayışa sahip olacaksınız.

2 – C, diğer programlama dillerine benzer bir dildir.

Java, Python, JavaScript gibi hemen hemen tüm üst seviye programlama dilleri, C dili ile birbirlerine yakınlardır. Ayrıca, programlamadaki ortak fikirleri ifade etmek için de iyi bir dildir. Konuştuğunuz kişinin C bilmemesi farketmez, yine de fikrinizi anlayabilecekleri bir şekilde aktarabilirsiniz.

3 – Milyonlarca insanı etkileyen açık kaynaklı projeler üzerinde çalışma fırsatı.

İlk başta, C'nin önemli bir dil olduğu gerçeğini göz ardı edebilirsiniz. Bir mobil uygulama geliştirmeniz gerekiyorsa, Java (Android için), Swift ve Objective C (iOS için) gerekir. Ve web uygulaması oluşturmak için C #, PHP, ASP.net, Ruby, Python gibi onlarca dil vardır. O zaman, C programlama neden?

Python uygulamaları geniş bir yelpazede yapmak için kullanılır. Python'a katkıda bulunmak istiyorsanız, milyonlarca Python programcısının kullandığı *Python yorumlayıcı* üzerinde çalışmak için C programlamayı bilmeniz gerekir. Bu sadece bir örnek. Bugün kullandığınız çok sayıda yazılım C tarafından desteklenmektedir

C programlamanın kullanıldığı daha büyük açık kaynaklı projelerden bazıları Linux Çekirdeği, Python Yorumlayıcı, SQLite Veritabanı'dır.

Büyük açık kaynak kodlu bir proje için yaygın olarak kullanılan bir diğer dil C ++ 'dır. C ve C ++ 'ı tanıyorsanız, yüz milyonlarca insanı etkileyen büyük açık kaynaklı projelere katkıda bulunabilirsiniz.

4 – Daha iyi programlar yazacaksın.

Dürüst olmak gerekirse, bu ifade her zaman doğru olmayabilir. Bununla birlikte, bilgisayarın nasıl çalıştığını ve belleği yönettiğini bilmek, diğer programlama dillerinde nasıl etkili kod yazılacağı konusunda fikir verir.

5 – Diğer programlama dillerini çok daha kolay öğreneceksiniz.

Pek çok popüler programlama dili, C'ye dayanmaktadır. C'yi biliyorsanız, C ++ 'ı çok daha rahat öğrenmeye başlarsınız.

C # ve Java gibi diller C ve C ++ ile ilgilidir. Ayrıca, JavaScript ve PHP'nin sözdizimi C'ye benzer.

C ve C ++ programlamayı biliyorsanız, başka bir dile geçme probleminiz olmaz.

İşletim sisteminizde C programlamayı derleyin ve çalıştırın

C programlamayı çalıştırmak için kullanabileceğiniz çok sayıda derleyici ve metin editörü vardır. Bu derleyiciler ve metin editörleri sistemden sisteme farklılık gösterebilir.

Bilgisayarınızda C programlamayı çalıştırmanın en kolay yolunu bu bölümde (Windows, Mac OS X veya Linux) bulacaksınız.

[Linux'ta C ile programlamaya başlamak](#)

[Mac OS X'te C ile Programlamaya Başlayın](#)

[Windows'ta C ile Programlamaya Başlayın \(XP, 7, 8 ve 10\)](#)

İlk C Programınız

Bu program genellikle yeni başlayanlara programlama dilini tanıtmak için kullanılır. Öyleyse başlayalım.

```
#include
int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

“Merhaba, Dünya!” Programı nasıl çalışır?

Programınıza **stdio.h** başlık dosyasını ekleyin.

C programlama küçüktür ve kendi başına fazla bir şey yapamaz. Programı çalıştırmak için gerekli kütüphaneleri kullanmanız gerekir. *Stdio.h bir başlık dosyasıdır* ve C derleyici o dosyanın konumunu bilir. Dosyayı kullanmak için, *#include önişlemcisini* kullanarak programınıza eklemeniz gerekir.

Neden bu programda stdio.h dosyasına ihtiyacınız var?

Mesela bu programda, tırnak işareti içindeki metni görüntüleyen **printf ()** işlevini kullandık. **Printf ()**, **stdio.h** dosyasında tanımlandığı için, **stdio.h** dosyasını eklemeniz gerekir.

Main () Fonksiyonu (main() function)

C programlamada, kod yürütme **main ()** fonksiyonunun başlangıcından başlar.

Kıvrımlı ayraçların içindeki kod {}, **main ()** fonksiyonunun gövdesidir. **Main ()** fonksiyonu her C programında zorunludur.

```
int main() {  
}
```

Printf () Fonksiyonu (printf() function)

Printf (), formatlanmış çıktıyı ekrana gönderen bir kütüphane işlevidir (tırnak işareti içindeki dizeyi görüntüler). İfadenin sonunda noktalı virgül olmalıdır dikkat edin.

Merhaba Dünya! ekranda.

Bunun çalışması için **stdio.h** dosyasını programınıza dahil etmeniz gerektiğini **unutmayın**.

Return ifadesi (The return statement)

Return ifadesi 0 döndürür; main () fonksiyonu içindeki programı sonlandırır. Bu ifade zorunlu değildir. Ancak, onu kullanmak iyi bir programlama alışkanlığı olarak kabul edilir.

Alınacak önemli notlar

- Tüm C programları main () fonksiyonu ile başlar ve zorunludur.
- Programda gerekli olan başlık dosyasını kullanabilirsiniz. Örneğin: Karekökü hesaplamak için **sqrt ()**. Bir sayının gücünü bulmak için **pow ()**. Bunlar için programınıza **math.h** başlık dosyasını dahil etmeniz gerekir.
- C büyük / küçük harf duyarlıdır; büyük harf ve küçük harf kullanımı farklı anlamlara gelir.
- Program, main () fonksiyonu içindeki **return** ifadesiyle karşılaştığında **sona erer**. Ancak, ana fonksiyon içindeki return ifadesi zorunlu değildir.
- Bir C programındaki ifade bir noktalı virgülle bitmelidir.

C ile programlama öğrenme aşamasındaki çalışmalarınız için ufak bir kaç öneri

- Bir günde C programlamayı öğrenemezsiniz. C programlamayı daha hızlı öğrenmeye çalışırsanız, önemli kavramları kavramakta zorluk yaşayabilirsiniz.
- Eğer sadece sentaksı (**syntax**) öğrenir ve takılı kalırsan, o an için C kodun işe yarayabilir, fakat C programlamayı doğru şekilde öğrenmemiş olursun.
- Sadece örnek koda bakmayın, sisteminizde çalıştırın
İnternet C programlamasını anlamanıza yardımcı olacak düzinelerce örneğe sahiptir. Ancak, sisteminizde çalıştırmadan bir roman gibi okuyorsanız, yanlış yapıyorsunuz demektir.
- C programlamanın yeni bir özelliğini öğrenmek istiyorsanız, bu özellik ile ilgili kod yazmayı deneyin. Yapamıyorsanız, kodu kopyalayın ve sisteminizde çalıştırın. Ardından anlamaya çalışın. Anladıktan sonra, kodu değiştirin, farklı yazın ve yeniden çalıştırın.
- Bir hata ayıklama aracı kullanmayı öğrenin. (Debugging Tool)
Bir hata ayıklama aracı veya hata ayıklayıcı, programcılarının bir programı herhangi bir noktada durdurmalarına olanak veren ve hataları algılayan ve düzelten yardımcı olan bir yazılımdır (IDE'de zaten

mevcuttur).

- Programınızda hatalar varsa, hatayı bulmak için başınızı kaşımak yerine, programı herhangi bir noktada durdurmak ve hatayı tespit etmek için değişkenlerin değerini bulmak için hata ayıklayıcıyı kullanabilirsiniz.
- Bir hata ayıklayıcının nasıl kullanılacağını bilmek, her programcının öğrenmesi gereken önemli bir beceridir.
- C topluluklarına katılın.
Basit C programları yazmaya başladığınızda çevrimiçi topluluklara ve forumlara katılın. Diğer programcılara yardım edebilir veya takılı kaldığınızda yardım isteyebilirsiniz.
- [C Programcıları için özel Google forumu](#) – C programlamaya yeni başlayanlar için bir mücevher.
- [Codechef](#) – Pratik sorular, geniş bir programcı topluluğu.
- [StackOverflow](#) – En Popüler Programlama Soru & Cevap web sitesi

Burada belirtilen programlama uygulamalarının bazıları tüm programlama dillerinde geçerlidir, bazıları ise yalnızca C programlama için geçerlidir.

- **Satır başına bir ifade kullanın.** Aşağıdaki kodda yanlış olan ne?

```
int count; float squareRoot = 10.0; printf("Square root = %f", squareRoot);
```

Aslında, kod tamamen geçerli. Ancak, bu daha iyi olmaz mıydı:

```
int count;  
float squareRoot = 10.0;  
printf("Square root = %f", squareRoot);
```

Buradaki amaç, programcı arkadaşlarınızın anlayabileceği bir kod yazmaktır.

▪ Adlandırmalarda kurallı ve Tutarlı olun!

Değişkenlere ve fonksiyonlara uygun bir ad verin ve bunda tutarlı olun.

```
int a, b;
```

Burada, a ve b iki değişkendir ve ne yaptıkları hakkında hiçbir fikrim yok. Bunun yerine şöyle bir isim seçebilirsiniz:

```
int counter, power;
```

Ayrıca, adlandırma sırasında bir düzen izleyin. Örneğin:

```
int base_number, powerNumber;
```

Her ikisinde de: kelimeleri ayırmak için _ kullanmak ve ilk kelimedenden sonra büyük harf kullanmak popülerdir. Ancak, ikisini de bir programda kullanmayın; birini seçin ve onunla tutarlı olun.

▪ Yorumları Kullanma

Yorumlar, derleyicinin yoksaydığı, kodun bir parçasıdır.

Programınızda neleri başarmaya çalıştığınızı açıklamak için programınızdaki yorumları kullanabilirsiniz. Bu, programcı dostunuzun kodu anlamasına yardımcı olur.

C programındaki yorumları // kullanarak kullanabilirsiniz. Örneğin:

```
// My first C program

#include
int main()
{
    printf("Hello, World!\n");    // displays Hello, World! on
the screen
    return 0;
}
```

“Kodunuzu yorumlamak banyonuzu temizlemek gibidir – asla yapmak istemezsiniz, ancak gerçekten sizin ve misafirleriniz için daha keyifli bir deneyim yaratır.”

Ryan Campbell

ve son olarak

C, eğer bir **acemi iseniz**, programlama öğrenmeye başlamak için iyi bir dildir. Ve eğer sıkı bir profesyonel programcıysanız, her zaman öğrenin; kesinlikle programlama temellerini geliştirmenize yardımcı olacaktır.

Dersler, daha önce C programlama bilgisine sahip olmayan yeni başlayanlar için tasarlanmıştır. Her ders detaylı bir şekilde örneklerle ve ayrıntılı açıklamalarla yazılmıştır.

C Programlama Eğitimi İçerik Listesi (C Tutorial Index)

1.Bölüm : GİRİŞ (INTRODUCTION)

Anahtar Kelimeler ve Tanımlayıcılar (Keywords & Identifier)

Değişkenler ve Sabitler (Variables & Constants)

C Veri Tipleri (C Data Types)

C Giriş / Çıkış (C Input/Output)

C Operatörleri (C Operators)

Temel örnekler (Basic Examples)

2.Bölüm : AKIŞ KONTROLÜ (FLOW CONTROL)

if ... else deyimi (if...else Statement)

C For Döngüsü (C for Loop)

C While Döngüsü (C while Loop)

Break ve Continue deyimleri (break and continue)

switch deyimi (switch Statement)

Karar Yapıları Örnekleri (Decision Examples)

3.Bölüm : FONKSİYONLAR (FUNCTIONS)

C Fonksiyonlar Giriş (Functions Introduction)

Kullanıcı tanımlı Fonksiyonlar (User-defined Function)

Fonksiyon Türleri (Function Types)

C Yineleme (Recursion in C)

C Depolama Sınıfı (C Storage Class)

C Fonksiyon örnekleri (Function Examples)

4.Bölüm : DİZİLER (ARRAYS)

C Dizileri Giriş (C Arrays Introduction)

Çok Boyutlu Dizi (Multidimensional Array)

Diziler ve Fonksiyonlar (Arrays & Functions)

Dizi Örnekleri (Array Examples)

5.Bölüm : İŞARETÇİLER (POINTERS)

C İşaretçiler (C Pointers)

İşaretçiler ve Diziler (Pointers & Arrays)

İşaretçiler ve Fonksiyonlar (Pointers & Functions)

Bellek yönetimi (Memory Management)

İşaretçi Örnekleri (Pointer Examples)

6.Bölüm : Yapı ve dosya (STRUCTURE & FILE)

C yapısı (C Structure)

Yapı ve İşaretçiler (Structure & Pointers)

Yapı ve İşlevler (Structure & Functions)

C Unions (C Unions)

Yapı örnekleri (Structure Examples)

Dosya İşleme (Files Handling)

7.Bölüm : EK KAYNAKLAR (ADDITIONAL RESOURCES)

C örnekleri (C Examples)

C Kütüphane Fonksiyonları (C Library Function)