

# PYTHON İLE PROGRAMLAMAMANIN TEMELLERİ

```
for object to mirror_  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
selection at p  
_ob.select= 1  
_ob.select=1  
context.scene.objects.active  
["Selected" + str(modifier  
_ob.select = 0
```

```
print( please select  
-- OPERATOR CLASSES --
```

```
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"
```

## 1.1. Yazılım Geliştirme Süreci

Bir bilgisayar programı, bilgisayar sistemindeki elektrik sinyallerinin akışını yöneten bir dizi komuttur. Bu sinyaller, bilgisayarın hafızasını etkileyerek ekran, klavye, fare ve hatta diğer bilgisayarlar ile etkileşim sağlar. Bu şekilde insanlar da karmaşık problemleri çözmek, oyun oynamak gibi değişik işlemleri gerçekleştirebilir.

Yazılımlar, gerçek bilgisayar sistemlerinde kullanılmak üzere yazılır. Yazılım mühendisleri belirli sistemler üzerinde çalışacak şekilde programları üretirler. Bu sistemler, donanım alt yapıları ve kullandıkları işletim sistemi ile tanımlanır. Yazılım geliştiriciler; derleyici (compiler), hata ayıklayıcı (debugger) ve yanaylaç (profiler) gibi somut araçları kullanırlar.

Yazılım geliştirme süreci şu şekilde işler;

- Programcı programlama dili kullanarak kodları oluşturur,
- Yazılan kod bütünü, hata ayıklayıcı (debugger) kullanılarak hatalara karşı denetlenir,
- Hataları giderilmiş kodlar, derleyici (compiler) kullanılarak bilgisayarın yorumlayabileceği elektriksel sinyallere dönüştürülür.

### 1.1.1 Yazılım

Yazılım, bilgisayarın donanımını anlamlı hale getiren, bilgisayarları kullanıcıların amaçları doğrultusunda kullanmasını sağlayan kod, komut ve programlardır.

Bilgisayarın sabit diskine yüklenmiş olan programlar elektromanyetik bir örüntü oluştururlar. Elektronik sembollerden oluşan bu örüntünün, program çalışmadan önce hafızaya transfer edilmesi gerekir. Bu elektronik semboller **ikilik sayı** sistemini kullanan, sıfır ve bir değerlerinden oluşan bir dizidir. Bu ikilik kodlara **Makine Dili** denir.

Örnek: Bir ikilik program şu şekilde görünür: 10001011011000010001000001001110

İşlemci olarak adlandırılan bilgisayar donanımı, bu sinyalleri çözümleyerek yapılmak istenen işlemi gerçekleştirir.

## 1.1.2. Yazılım Geliştirme Ortamları

Yazılımlar, ikilik dizileri daha anlaşılabilir kılan **kelime** ve **sembolleri** kullanır. Böylece bilgisayarların dilini öğrenmek ve karmaşık problemleri çözen programlar yazmak kolaylaşır. Geçtiğimiz yaklaşık 60 yıllık süreçten günümüze kadar FORTRAN, COBOL, C, C++, Java ve C# gibi pek çok üst düzey dil geliştirilmiştir. Bu tür programlama dilleri ile uygulama geliştiren programcılar, donanım ya da **makine dili** gibi konulardaki detaylar ile ilgilenmeden çok etkili yazılımlar geliştirebilir.

Geleneksel olarak yeni bir dilde yazılan ilk program "Merhaba, Dünya!" adı verilen programdır.

Python dili ile yazılmış ilk örneğe bakalım: `print ("Merhaba, Dünya!")`

Bu print komutunun bir örneği olup ekranda Merhaba, Dünya! yazar. Tırnak işaretleri programda bir değer başlangıcını ve sonucunu gösterir ve ekranda gözükmez.

Python dili ile yazılmış aşağıdaki örneğe bakalım:

Toplam = 0

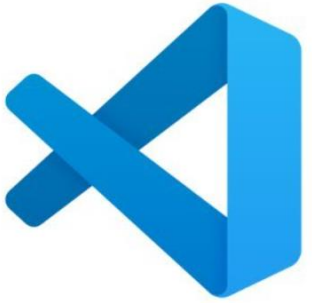
DersSaati = 3

Hafta = 14

Toplam = DersSaati \* Hafta

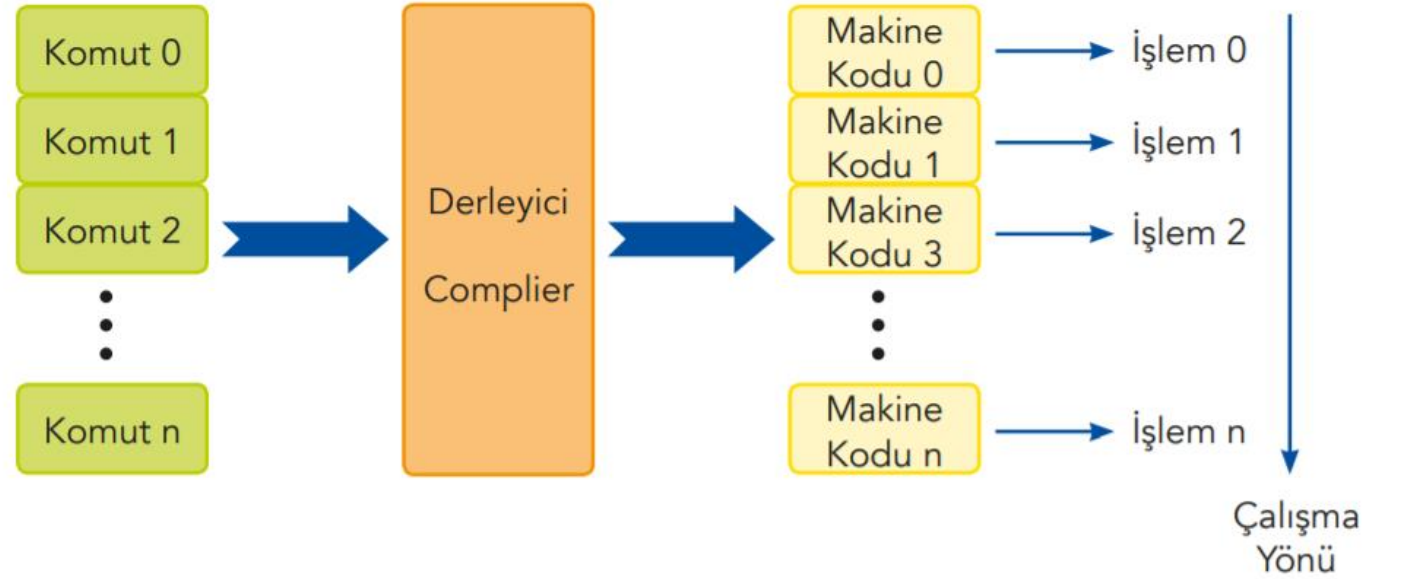
### 1.1.3. Editörler

Bir editör, programcının **kaynak kodu** yazmasını ve dosyaya kaydetmesini sağlar. Çoğu editör, renklendirme desteği sunarak dilin özelliklerini ortaya çıkarır ve programcının üretkenliğinin artmasını destekler. Dili oluşturan parçaların kurallara uygun bir şekilde düzenlenmesi söz dizimi (syntax) olarak ifade edilir. Geliştirme araçlarının yazılanları tam olarak doğru anlaması için, kullanılan kelime ve sembollerin kurallara uygun biçimde dizilmesi önemlidir. Yalnızca doğru biçimde ifade edilen programlar **makine koduna** dönüştürülmek üzere kabul edilir. Bu nedenle bazı editörler yazım yanlışları konusunda renkleri ya da farklı vurgulamaları kullanarak yazım hataları konusunda programcıyı uyarır.



#### • 1.1.4. Derleyiciler

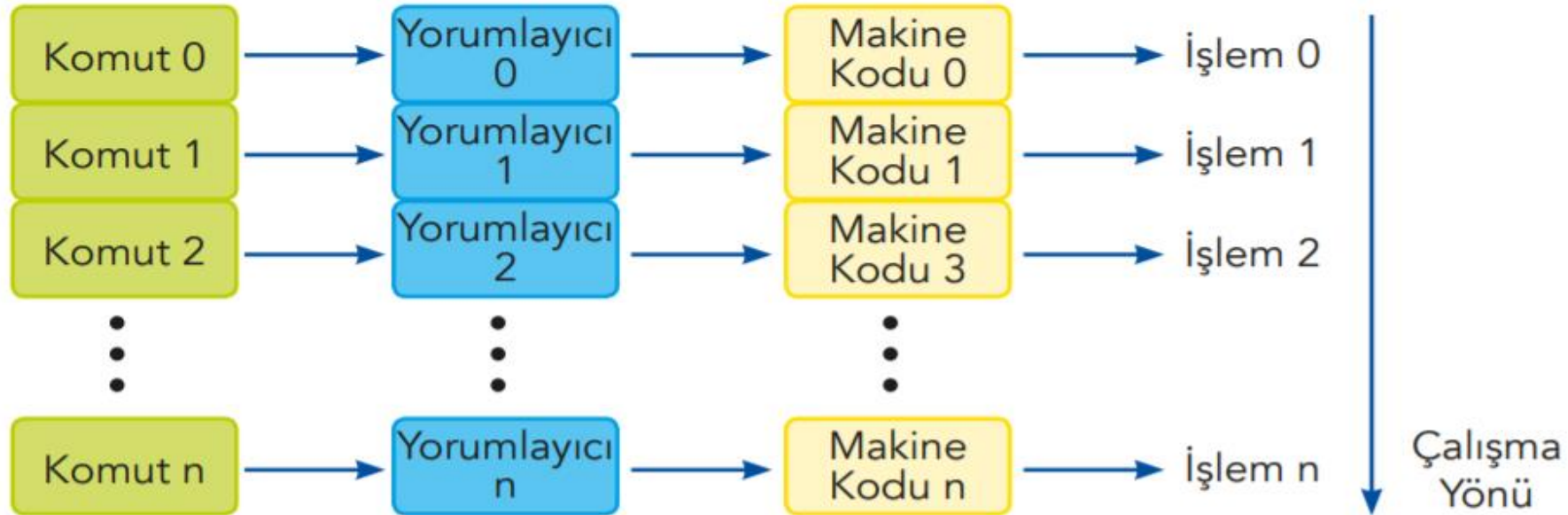
Derleyiciler, kaynak kodları hedef koda dönüştürür. Derleyiciler, kaynak kod içeriğini dönüştürerek hedef kod içeren bir dosya oluşturur. Derlenerek oluşturulan popüler dillere örnek olarak C, C++, Java ve C# verilebilir. Derleyiciler kaynak kodları bir bütün olarak derlerler.



### 1.1.5. Yorumlayıcılar

Yorumlayıcılar da derleyiciler gibi üst düzey kaynak kodu hedef koda (genellikle makine kodu) çevirir ancak derleyicilerden farklı çalışır. Derleyiciler herhangi farklı bir dönüşüm gerekmeden defalarca çalışılabilir bir program kodu üretirken, yorumlayıcılar kullanıcı kaynak kodu her çalıştırdığında **satır satır** makine diline çevirir.

Genellikle derlenen programlar yorumlanan programlara göre daha hızlı çalışır **çünkü derleme işlemi yalnızca bir kez yapılır**. Diğer yandan yorumlanan programlar, herhangi bir platformda tekrar derlenmelerine gerek kalmadan uygun bir yorumlayıcı ile hemen çalıştırılabilir. Python, yorumlanan bir dildir.



### **1.1.6. Hata Ayıklayıcılar**

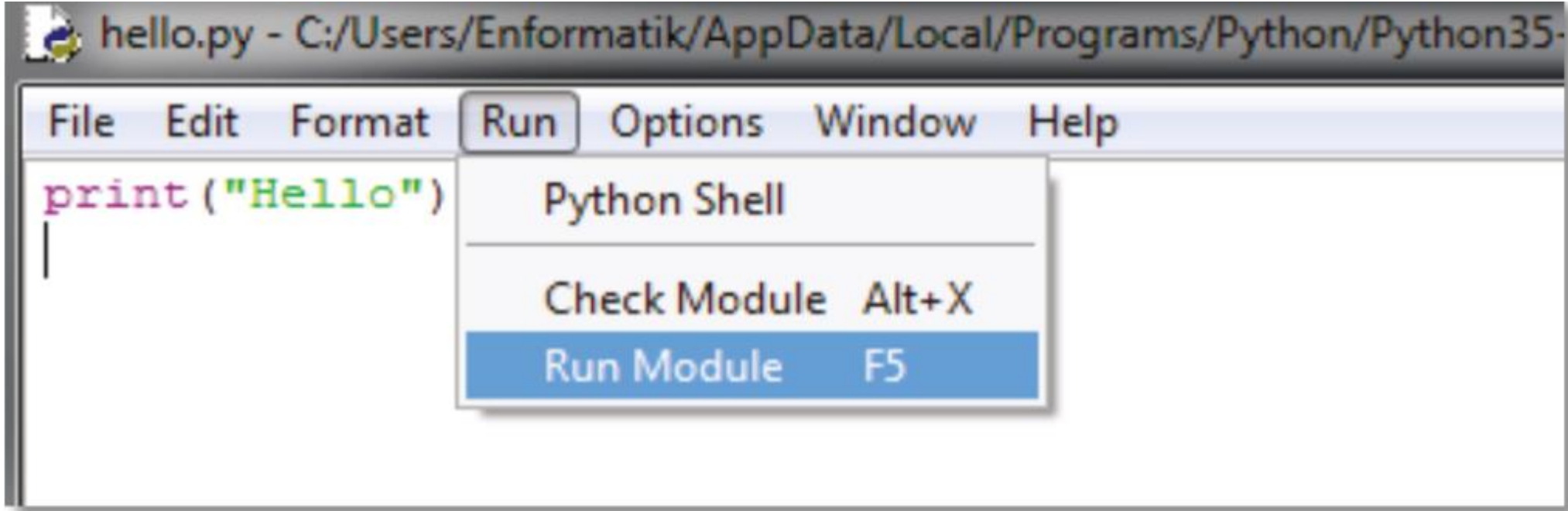
Hata ayıklayıcılar, programcının bir programdaki olası hataları bulmasına ve düzeltmesine olanak sağlayarak programın doğru çalışması için yardımcı olur. Hata ayıklayıcı programlar ile programın hangi satırında hata olduğu belirlenir. Programcı, değişkenlerin değerlerine bakarak neyin yanlış gittiğini anlayabilir.



### 1.1.8. Bütünleştirilmiş Geliştirme Ortamları

Çoğu yazılım, geliştirici bütünleştirilmiş geliştirme ortamlarını (integrated development environment-IDE) kullanır. Bu ortamlar, editörleri, hata ayıklayıcıları ve diğer programlama yardımcılarını kapsar. Aşağıda en sık kullanılan Python IDE ortamları listelenmiştir.

- **IDLE:** Python.org web sitesinde yer alan ücretsiz program geliştirme ortamıdır. Bu site içerisinde downloads menü başlığı altında Python için son sürümleri bulmak mümkündür. Ayrıca, birçok işletim sistemi için gerekli dosyalara erişim bağlantıları da bu sayfada yer almaktadır.



## 1.2. Neden Python?

Python, öğrenmesi kolay, tamamen özgür ve ücretsiz bir programlama dilidir. Nesnelere dayalı bir dil olup okunabilirliği yüksektir. Python'un dili başka programlama dilleri ile kıyaslandığında, bunun daha az kod ile işlemleri yapmasının mümkün olduğu görülecektir. Python, bütün işletim sistemleri ile uyum içerisinde çalışmaktadır. Programlama yapısı içerisinde birçok kütüphaneyi barındırmaktadır. Bu kaynaklarla daha az kod yazmak mümkündür. Python ile masaüstünde çalışan uygulamalar geliştirilebileceği gibi, web üzerinde çalışan uygulamalar geliştirmek hatta Raspberry-Pi gibi donanımları da programlamak mümkündür.

*C#* dilinde ekrana "Merhaba Dünya" yazan kod : `Console.WriteLine("Merhaba Dünya");`

Java dilinde ekrana "Merhaba Dünya" yazan kod : `System.out.println("Merhaba Dünya");`

Python dilinde "Merhaba Dünya" yazan kod : `print("Merhaba Dünya")`