

# Yazılım Mühendisliğine (YM) Giriş

Ders Notu III

2024 GÜZ

# 1. Yapısal (Procedural) Programlama Dilleri

- ❑ Yapısal bir dil, istenen çıktıyı elde etmek için bir dizi ifadeyi (expression) veya komutu (command) takip eder.
- ❑ Her adım dizisine prosedür (modül) adı verilir.
- ❑ Bu dillerden birinde yazılan bir programın içinde bir veya daha fazla prosedür (fonksiyon) bulunur.

**C (1972):** Orta seviye genel amaçlı bir programlama dilidir. İşletim sistemleri, veri tabanları ve gömülü sistemler, web uygulamaları, arayüz tasarımı, oyun ve animasyon gibi uygulamalarda kullanılır.

<https://careerkarma.com/blog/who-uses-c/>

**Fortran:** Mühendislik hesaplamaları ve yüksek performanslı bilimsel hesaplamalar gerçekleştirir.  
<https://tr.wikipedia.org/wiki/Fortran>

<https://blog.rysof.com/12-companies-hiring-fortran-developers-in-2022>

**BASIC (1963):** Kullanımı kolay, üst düzey (high level) bir programlama dilidir

**ALGOL (1958) :** Algoritmik Dil anlamına gelir. Çoğunlukla bilimsel hesaplamalar için kullanılmıştır. PL/I, Simula, Pascal, C, Ada ve Scheme dilleri Algol'dan etkilenerek geliştirilmiştir.

**COBOL (1960) :** İş Uygulamaları için tasarlanmıştır.

<https://www.luxoft.com/blog/why-banks-still-rely-on-cobol-driven-mainframe-systems>

## 2. Fonksiyonel (İşlevsel) Programlama Dilleri

- ❑ İşlevsel diller, ifadelerin yürütülmesine odaklanmak yerine, programların fonksiyonları uygulayarak oluşturdukları bir programlama paradigmasıdır.
- ❑ Matematiksel işlevlerin (fonksiyonların) ve değerlendirmelerin çıktısına odaklanır.
- ❑ Yeniden kullanılabilen bir kod modülü olan her işlev, belirli bir görevi gerçekleştirir ve bir sonuç döndürür.
- ❑ Atama deyimi bulunmaz. Değişkenlerin değeri bir kere verildi mi, bir daha değişmez.
- ❑ Bir fonksiyonu çağırmak kendi sonucunu hesaplamaktan başka bir etki üretmez.

**Scala (2003)** : Yüksek düzeyli genel amaçlı bir programlama dili. Hep fonksiyonel hem de nesneye yönelik bir dildir. Java ile tam uyumlu olarak çalışır. <https://alvinalexander.com/scala/whos-using-scala-akka-play-framework/>

**Erlang (1986)** Ericsson tarafından yazılan genel amaçlı, eşzamanlı (concurrent) programlama dilidir. Telekomünikasyon sistemlerinde kullanılıyor.

<https://www.erlang-solutions.com/case-studies/>

**Elixir (2012)** *Dinamik ve fonksiyonel bir dildir. Ölçeklenebilen (sürekli genişleyebilen) ve sürdürülebilir uygulamalara uygundur.* a dynamic, functional language for building scalable and maintainable applications. <https://elixir-lang.org/cases.html>

**F# (2005)** **JavaScript** ve **.NET** dili kullanılır. Web, bulut, veri bilimi, API geliştirmek üzere kullanılır. <https://github.com/fsprojects/fsharp-companies?tab=readme-ov-file>

# 3. Nesneye Yönelik (Object Oriented) Programlama Dilleri

- ❑ Nesneye yönelik bir dil, bir programı, nitelikler (attributes) ve yöntemler (methods) olarak bilinen, veri ve program öğelerinden oluşan bir nesne grubu olarak incelenir.
- ❑ Nesnelere bir program içinde veya başka programlarda yeniden kullanılabilir.
- ❑ Kodun yeniden kullanılması ve ölçeklendirilmesi daha kolaydır.
- ❑ Karmaşık programlar için popüler bir dil tipi haline getirmiştir.

**Java** (James Gosling) 1995) : Platformdan bağımsız bir dildir. «Just In Time» derleyicisi kaynak kodundan makine kodu üretir

**Python** (1991): Veri bilimi, makine öğrenimi, sistem otomasyonu, web, API (Application Programming Interface) geliştirme ve diğer özel amaçlı uygulamalar için temel yapıdır.

**C#** (2000) : Tüm uygulamaları oluşturmak için modern, yenilikçi, açık kaynaklı programlama dilidir. Platformlar arası bir çerçeve olan .NET kullanır.

**PHP** (1995) Web geliştirmeye uygun, sunucu taraflı ve genel amaçlı bir betik (script) dilidir. Betik dil, derlemeye gerek kalmadan, komutları çevirmek için bir yorumlayıcı kullanır ve doğrudan kaynak kodu yorumlanır.

**C++** (1985): Bjarne **Stroustrup** tarafından C dilinden geliştirilmiştir. Programlara kesinlik kazandıran ve kodun yeniden kullanılmasına izin veren nesne yönelimli bir en popüler programlama dillerinden biridir. C ve C++ dillerinin sentaksları hemen hemen aynıdır.

**Ruby** (1995): Verimlilik ve sadeliğe odaklanmış, açık kaynak kodlu dinamik bir dildir. Okuması doğal ve yazması kolay bir sözdizimi vardır.

## 4. Betik Diller (Scripting languages)

❑ Programcılar, tekrarlanan görevleri (işleri) otomatikleştirmek, dinamik web içeriğini yönetmek veya daha büyük uygulamalardaki süreçleri desteklemek için komut dosyası dillerini, yani betik dillerini kullanır.

**PHP** Tüm popüler işletim sistemleri Windows, Unix, Linux, macOS PHP kullanır. MySQL veritabanıyla birlikte, Linux işletim sisteminde iyi bir performansı vardır. C/C++ diline olan benzerliği nedeniyle öğrenmesi oldukça kolaydır. <https://www.softkraft.co/companies-using-php/> İnternet için üretilmiş, sunucu taraflı, geniş kullanımlı, genel amaçlı, içerisine HTML gömülebilen betik programlama dilidir.

**Ruby:** sunucu taraflı betik (scripting) dildir. Genel amaçlıdır ve yorumlanabilir. (interpreted). Gerçek bir nesneye yönelik dildir. Hypertext Markup Language (HTML).içerisine gömülebilir (embed). Sentaksı kolaydır. Farklı veri tabanlarına bağlantısı kolaydır. <https://www.monocubed.com/blog/companies-that-use-ruby-on-rails/>

**Python:** Tüm diğer betik dillerde olduğu gibi birden fazla platformda çalışır; yani herhangi bir değişiklik yapmadan farklı işletim sistemlerinde çalışabilir.

**Perl :** Genel amaçlı, yüksek düzeyli, yorumlanabilen (interpreted) ve dinamik programlama dilidir.. Hem yapısal hem de nesne yönelik programlamayı destekler. Sözdizim olarak C diline benzerdir ve kolaydır.

**Node.js** (2009) Platform bağımsız, açık kaynaklı sunucu ortamıdır (server environmet) . Windows, Linux, Unix, macOS işletim sistemlerinde çalışır.

# 5. Mantıksal (Logic) Programlama Dilleri

- ❑ Mantıksal programlama dili, bilgisayara nasıl karar vereceği konusunda talimat vermek için bir dizi tanımı ve kuralı ifade eder.
  - ❖ Böylece bir bilgisayara neler yapması gerektiğini bildirmeye gerek olmaz.
- ❑ 1960 'lı yıllarda bilgiyi temsil etmek ve yapay zekanın incelenmesine destek amacıyla tasarlanmıştır.
- ❑ Önermeler hesabı (predicate calculus) ile bildirimsel (declarative) programlama oluşturur.
- ❑ Mantıksal programlar veri odaklıdır
- ❑ Programlar, aynı zamanda tahminler olarak da adlandırılan bir dizi mantıksal ifade kullanır.
  - ❖ Tahminler, gerçekler veya kurallar olarak sınıflandırılabilir.

Prolog (1972) Declarative

Absys (1957) Declarative

Datalog (1977 ) Declarative

Lisp (1958) Declarative

# Programlama Dillerinin Farklı bir Sınıflandırması

## A- Ön uç (front-end) Programlama Dilleri

- ❑ Ön uç programlama dilleri yazılımın kullanıcı tarafı ile ilgilidir.
- ❑ Ön uç, kullanıcının web sitesi veya uygulamada gezinirken karşılaşacağı tüm metinler, renkler, düğmeler, resimler ve gezinmeyle (navigasyon) ilgilenir.
- ❑ Grafik tasarım veya sanata ilgisi olanlar, ön uç dillerden birini öğrenmeye daha verimli olarak başlayabilir.

### Örnek olarak

- ❖ Web uygulamalarında HTML, CSS, JavaScript ve React dilleri verilebilir.
- ❖ Mobil uygulamalarda Dart, Swift, Kotlin dilleri verilebilir.

# Ön-Yüz Programlama Dilleri

## Top 10 Front-End Web Development Languages





# Programlama Dillerinin Farklı Sınıflandırması

## B- Arka uç (back-end) Programlama Dilleri

- ❑ Arka uç dilleri, web uygulamalarında sunucu tarafının depolanması ve işlemlerinin yürütümüyle (processing) ilgilenir.
- ❑ Yazılımda, kullanıcının doğrudan ilişkili olmadığı, ancak arka plan deneyimini destekleyen kısımdır.
  - ❖ Arka plan deneyimine veri mimarisi, komut dosyası oluşturulması ve uygulamalar ile temel veri tabanları arasındaki iletişim dahildir
- ❑ Mühendislik deneyimi olanların arka uç geliştirmede daha verimli olacakları beklenir.
  - Örnek olarak,
    - ❖ web uygulamalarında JavaScript (Node. JS), PHP, Java, Python, Ruby, C# dilleri verilebilir.

# En iyi Arka-Yüz Programlama Dilleri (2024)

## Types of Backend Programming Languages



# Mobil Uygulamalar için Arka-Yüz Dilleri



# Bir başka Dil Sınıflandırması:

## Yüksek Düzeyli (High-level) ve Düşük Düzeyli (Low-level) Diller

- ❑ Yüksek ve düşük seviyeli programlama dillerinin birbirinden en önemli farkı, dilin bir programcı tarafından mı yoksa bir bilgisayar tarafından mı yazıldığına kolaylıkla anlaşılabilmesidir.
- ❑ *Yüksek düzeyli diller*, insan diline çok daha yakın programlama dilleridir.
  - ❖ Yüksek seviyeli diller bellek (memory) açısından fazla verimli olmayabilir; ancak kullanıcı dostudur.
  - ❖ Yüksek düzeyli programlama dillerinin yazılması, anlaşılması, bakımı (sürekliliği) ve hata ayıklaması daha kolaydır.
  - ❖ Günümüzde kullanılan en popüler programlama dilleri üst düzey dillerdir.
- ❑ *Düşük seviyeli diller*, makine dostudur
  - ❖ Bu diller bellek (memory) kullanımı açısından oldukça verimlidir, ancak bir çevirici (translator) yardımı olmadan anlaşılması çok zordur.
  - ❖ Konuşma dilini kullanmadıkları için kullanıcı dostu değildir ve kodlama amaçlı kullanılmaz.
  - ❖ Düşük düzeyli dillere örnek olarak makine kodu ve assembly dilleri verilebilir.

# Farklı bir Sınıflandırma:

## Yorumlanabilen (Interpreted) ve Derlenebilen (Compiled) Diller

□ Yorumlanan ve derlenen dillerin her ikisi de üst düzey kodu (kullanıcı tarafından anlaşılabilir kodu) bilgisayar (makine) tarafından okunabilir koda nasıl dönüştürdükleri ile ilgili terimlerdir.

□ **Yorumlanmış dillerde**, yorumlayıcı çalışma (run time) sırasında kodu satır satır okur, çalıştırır ve makine koduna dönüştürür.

❖ Kod çalışma esnasında makine diline dönüştürüldüğü için performans nispeten düşüktür.

❖ Yorumlanabilme özelliğine diller daha esnek ve platformdan bağımsız hale gelir.

✓ Örnekleri : Python, JavaScript, PHP, Ruby, ..... dilleri olarak verilebilir.

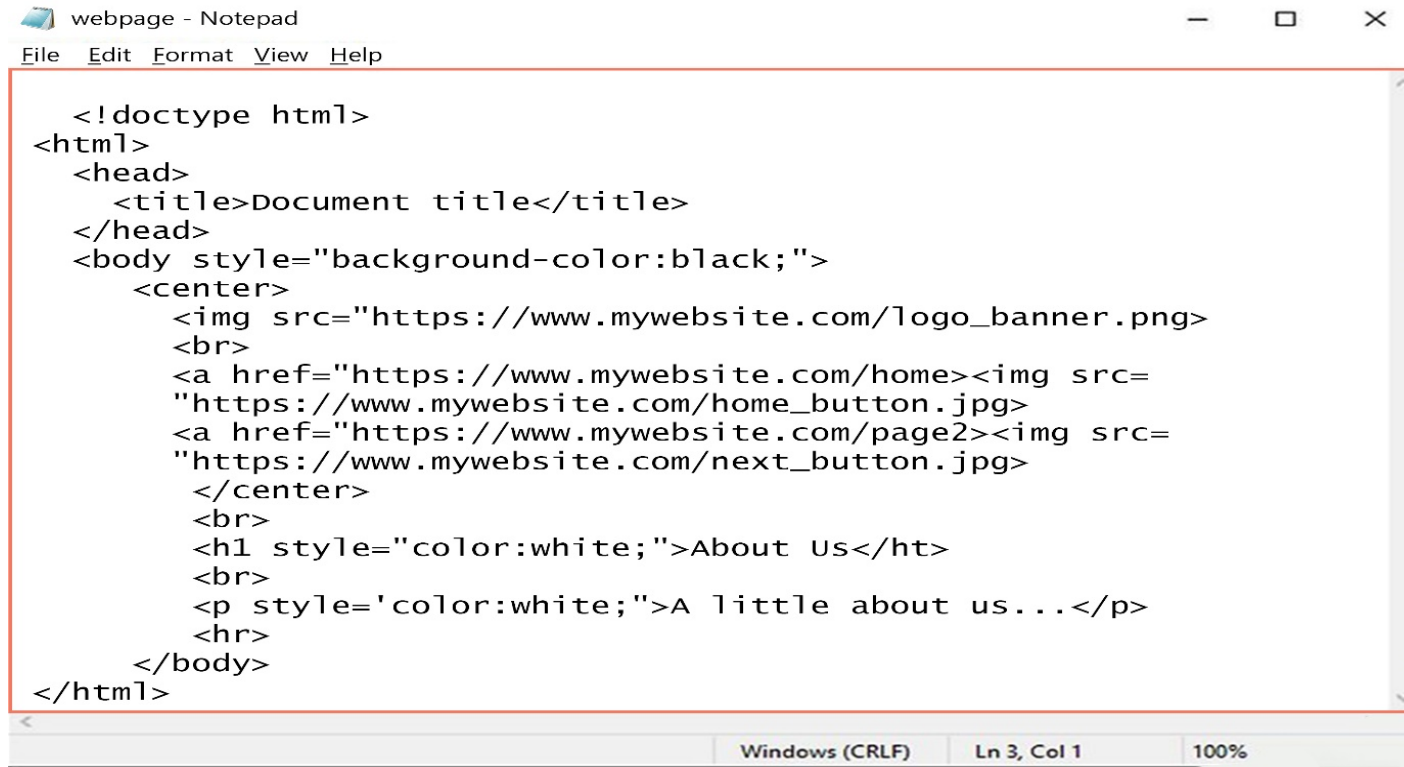
□ **Derlenmiş dillerde**, tüm programın makine koduna dönüştürüldüğü bir süreçten geçilir.

❖ Bu süreç, programın yürütülmesini hızlandırır; ancak herhangi bir değişiklik yapıldığında *programın yeniden derlenmesi* gerekir.

❖ Örnekleri C, C++, C#, Rust, Erlang ... dilleri olarak verilebilir.

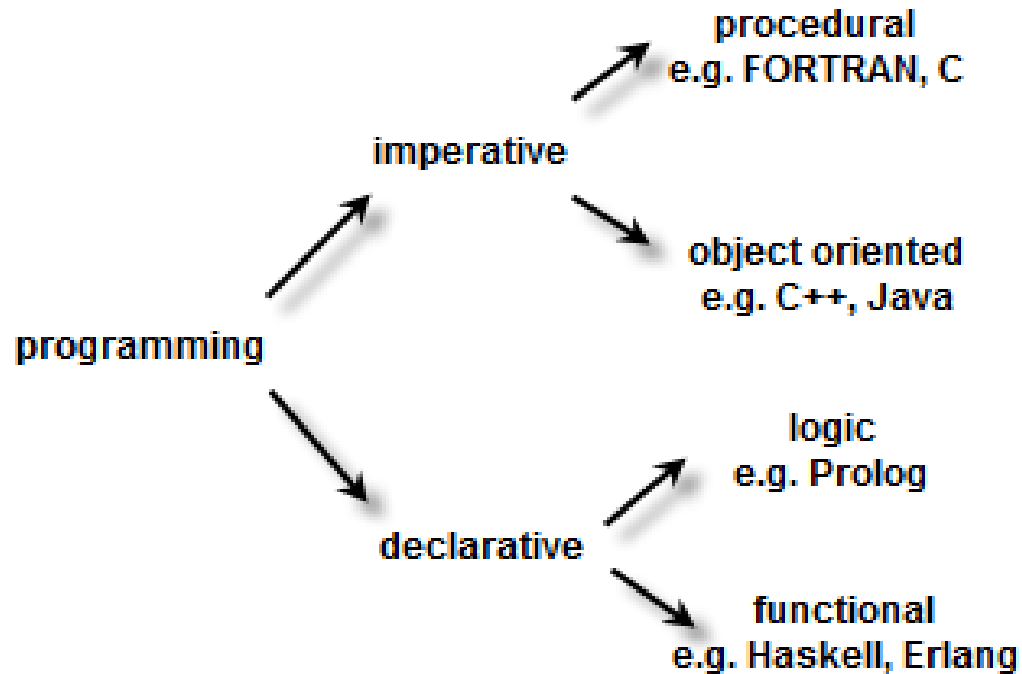
# Markup Language – İşaretleme Dili

- ❑ Hypertext Markup Language (HTML) gibi biçimlendirme (işaretleme) dillerinin programlama dilleri olarak sınıflandırıldığı yaygındır.
- ❑ İşaretleme dilleri programlama dilleriyle aynı kategoride kabul edilemez.
  - ❖ Bu diller, bir sayfa veya belgedeki içeriğin formatını ve yapısını kontrol eden sembollerden oluşan metin kodlama sistemleridir.

A screenshot of a Notepad window titled "webpage - Notepad". The window contains HTML code for a webpage. The code includes a document type declaration, a head section with a title, and a body section with a black background. The body content is centered and includes a banner image, two buttons with href attributes, a white heading "About Us", and a white paragraph "A little about us...".

```
<!doctype html>
<html>
  <head>
    <title>Document title</title>
  </head>
  <body style="background-color:black;">
    <center>
      
      <br>
      <a href="https://www.mywebsite.com/home">
      <a href="https://www.mywebsite.com/page2">
    </center>
    <br>
    <h1 style="color:white;">About Us</ht>
    <br>
    <p style='color:white;'>A little about us...</p>
    <hr>
  </body>
</html>
```

# Programlama Stilleri / Paradigmaları



## ❑ Imperative (Zorunlu stil) Programlama (What to do and how to do something)

- ❖ Programın nasıl çalıştığını açıklamaya odaklanır.
- ❖ Bir programın ne yaptığını ve ayrıca bunu nasıl yaptığını söyler.
- ❖ Her kod satırı (statement) hesaplama durumunu (veri alanını) değiştirir.
  - ✓ C, C++, Java, PHP, Python, Ruby

## ❑ Declarative Programming: (what you do and not how to do it)

- ❖ Programın sonuca nasıl ulaştığına ilişkin hiçbir ayrıntı belirtmeden programın neyi başarması gerektiğine odaklanır
  - ✓ Programın mantığı tanımlanır, fakat ayrıntılı kontrol akışı içermez.
- ❖ Programın ne yaptığını söyler ama nasıl yapacağını söylemez.

# Prolog Örneği

```
cat(tom).                % tom is a cat
mouse(jerry).           % jerry is a mouse

animal(X) :- cat(X).    % each cat is an animal
animal(X) :- mouse(X). % each mouse is an animal

big(X) :- cat(X).       % each cat is big
small(X) :- mouse(X).  % each mouse is small

eat(X,Y) :- mouse(X), cheese(Y). % each mouse eats each cheese
eat(X,Y) :- big(X), small(Y).    % each big being eats each small being
```

```
?- animal(X)
```

```
X= tom
```

```
?- cat(tom)
```

```
Yes
```



# Imperative (Zorunlu, Gerekli ) ve Declarative (Bildirimsel ) Stil Programlama

## ❑ Imperatif /Zorunlu Stil Diller

- ❖ Yapısal (Procedural) ve Nesneye Yönelik Programlama (Object-Oriented Programming)
  - ✓ Örneğin; C, C++, C#, PHP, Java , Assembly gibi diller

## ❑ Bildirimsel Stil Diller

- ❖ L ojik (Logical = programlama, fonksiyonel (Functional) programlama, düzgün ifadeler (regular expressions), veri tabanı sorgulama dilleri
  - ✓ Prolog ,Javascript (multiparadigmatic) , Scala, Lisp, SQL (Structured Query Language)
- ❖ Örnek bir SQL sorgusu aşağıda verilmiştir.

```
SELECT * FROM Books  
WHERE Category='Programming'  
AND language='english';
```