Bilgisayar Teknolojileri Bölümü Programlamanın Temelleri



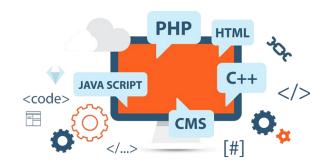
Öğr. Gör. Halil ARSLAN

- Program
- Programlama Dili
- Seviyeler (Düşük, Yüksek)
- Programlama
- Algoritma
- Kaynak Kod (Açık Kaynak)
- Derleme
- Yorumlama
- Debug

- IDE
- GUI

- Program: Bilgisayara bir hesaplamayı nasıl yapacağını söyleyen bir dizi yönergeye denir.
- Belirli bir işlemi/işlemleri gerçekleştirebilmek için verilen komut ya da komutlar bütünüdür.
- **Programlama Dili:** İstenilen hesaplamaları yapmak için, elde edilen veriyi saklamak, girdi/çıktı aygıtlarına veri gönderme/alma gibi işlemleri yapmak için kullanılan dildir.
- Programlama dillerinin belirli yazım kuralları (Syntax) vardır.



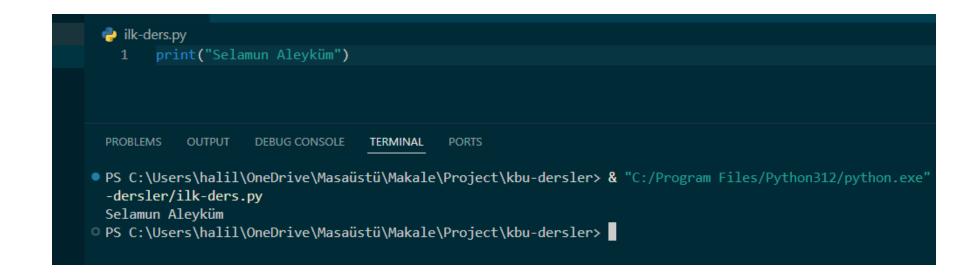




- Seviye: Bir programlama dilinin insan algılamasına olan yakınlığının bir ölçüsüdür.
- Programlama dilleri seviyelerine göre;
- Çok yüksek,
- Yüksek,
- Orta,
- Alçak,
- Makine
- şeklinde sınıflandırılmaktadır.

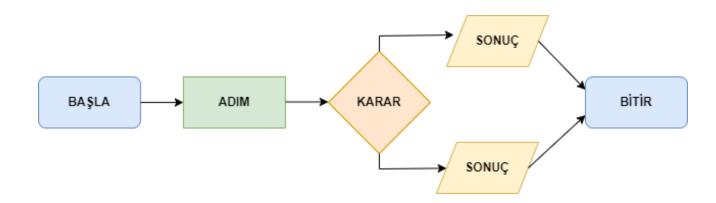


- Programlama: Bilgisayarların donanıma nasıl davranacağını anlatan, bilgisayara yön veren komutlar, kelimeler, aritmetik işlemlerdir.
- Yazılan kaynak kod genellikle bir derleyici ve yorumlayıcı yardımıyla belirli bir sistemde çalıştırılabilir hale getirilir.

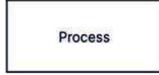


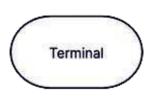
- Algoritma: Belirli bir sorunu çözmek veya bir amaca ulaşmak için tasarlanan, açıkça tanımlanmış ve sıralı adımlar dizisidir. Matematikte ve bilgisayar biliminde kullanılan algoritma, bir başlangıç durumundan başlayıp sonlu sayıda işlemle sonuca ulaşan prosedür ya da yol haritasıdır.
- Örnek olarak, bilgisayarların açılması, otomatik sileceklerin çalışması veya günlük hayatta yapılan yemek tariflerindeki aşamalar algoritma örnekleridir. Algoritma kavramı, ismini 8. yüzyılda yaşamış matematikçi El-Harezmi (Ebu Abdullah Muhammed Bin Musa el-Harezmi) 'den alır ve bilgisayar programlarının temelini oluşturur.

• Akış diyagramı (flowchart), bir süreci, algoritmayı veya sistemi grafiksel olarak gösteren, adımların ve işlemlerin birbirine bağlandığı bir şemadır. İşlem adımları belirli sembollerle temsil edilir ve işlemlerin nasıl ilerlediği oklarla gösterilir. Akış diyagramları karmaşık süreçlerin daha kolay anlaşılması, incelenmesi, planlanması ve anlatılması için kullanılır.

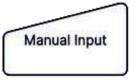


- Process (Süreç): Dikdörtgen şekil. Bir işlem adımını, hesaplama veya veri manipülasyonunu temsil eder. Process[İşlem Adı]
- Terminal (Başlangıç/Bitiş): Oval/yuvarlak şekil. Akışın başlangıç veya bitiş noktasını gösterir. Start([Başlangıç]), End([Bitiş])
- Decision (Karar): Eşkenar dörtgen. Evet/hayır veya doğru/yanlış gibi karar noktalarını temsil eder. Decision {Soru?}
- Manual Input (Manuel Giriş): Üstten eğik dikdörtgen. Kullanıcının manuel veri girişi yapması gereken noktaları gösterir. ManualInput[/Kullanıcı Girişi/]

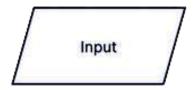


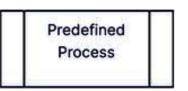


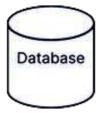




- Input (Girdi): Paralelkenar. Sistem veya süreçe veri girişini temsil eder.
   Input[/Veri Girişi/]
- Predefined Process (Önceden Tanımlı Süreç): Yan çizgili dikdörtgen. Başka yerde tanımlanmış alt süreci gösterir. SubProcess[[Alt Süreç Adı]]
- Database (Veritabanı): Silindir şekil. Veri depolama alanını temsil eder.
   DB[(Veritabanı)]
- Manual Operation (Manuel İşlem): Trapez şekil. İnsan müdahalesi gereken manuel işlemleri gösterir. ManualOp[Manuel İşlem]

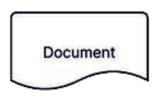


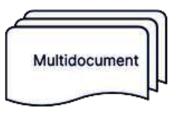




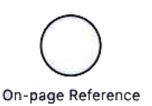


- Document (Doküman): Alttan dalgalı dikdörtgen. Bir belge veya raporu temsil eder. Doc[Belge Adı]
- Multidocument (Çoklu Doküman): Üst üste binmiş dalgalı dikdörtgenler. Birden fazla belgeyi temsil eder. MultiDoc[[Çoklu Belge]]
- Delay (Gecikme): Yarım oval. Süreçte bekleme veya gecikme olduğunu gösterir.
   Delay(Bekleme Süresi)
- On-page Reference (Sayfa İçi Referans): Daire. Aynı sayfa içinde akışın başka bir noktaya atlamasını gösterir. RefPoint((A))

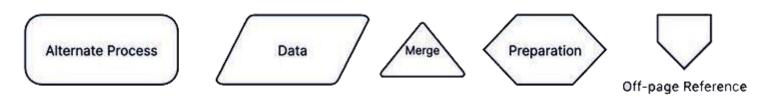


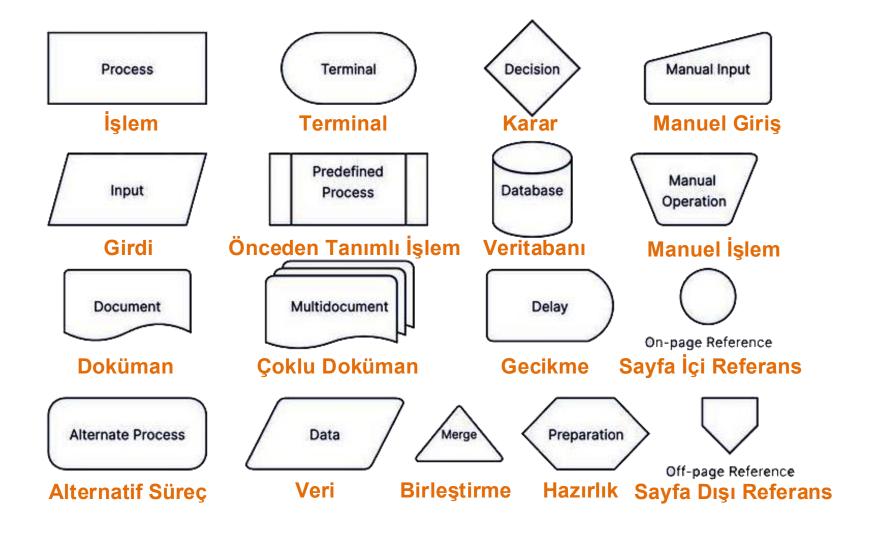






- Alternate Process (Alternatif Süreç): Yuvarlak köşeli dikdörtgen. Alternatif bir işlem yolunu gösterir. AltProcess[Alternatif Süreç]
- Data (Veri): Paralelkenar (Input'a benzer). Veri işleme veya depolamayı temsil eder.
   Data[/Veri İşleme/]
- Merge (Birleştirme): Ters üçgen. Birden fazla akışın birleştiğini gösterir.
  Merge{{Birleştirme}}
- Preparation (Hazırlık): Altıgen. Süreçten önce yapılması gereken hazırlık işlemlerini temsil eder. Prep[Hazırlık İşlemi]
- Off-page Reference (Sayfa Dışı Referans): Aşağı bakan beşgen. Akışın başka bir sayfaya devam ettiğini gösterir. OffPage[Sayfa 2'ye devam]





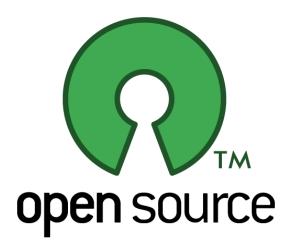


#### https://mermaid.live

#### Temel Kavramlar?

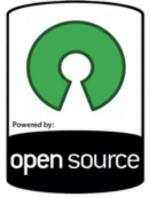
flowchart TD Merge1{{Siparis} Bilgilerini<br/ Mermaid Start([BA\$LANGIC]) >Birlestir}} Akış Diyagramı GetData[/Müşteri Bilgilerini Getir/] DB2[(Veritabanina<br/>Kaydet)] Decision1{Giris Yap veya<br/>>Kayıt 0l?} OffPage[Kargo Süreci<br/>Sayfa 2] Kodu ManualInput1[/Kullanici Bilgilerini Gir/] Doc4[Onay E-postas1] MultiDoc1 --> ManualInput2 Process1[Bilgileri Doğrula] End3([BİTİŞ]) ManualInput2 --> PreProc2 DB1[(Veritabanina<br/>Kaydet)] PreProc2 --> Delay2 PreProc1[[Kimlik Doğrulama<br/>br/>Güvenlik Modülü]] Start --> GetData Delay2 --> Decision4 RefA((A)) GetData --> Decision1 Decision4 --> |Hayır | Doc3 Input1[/Ürün Kataloğunu Yükle/] Decision1 --> Kayıt | ManualInput1 Doc3 --> End2 ManualInput1 --> Process1 ManualOp1[Ürün Seçimi Yap] Decision4 --> Evet | Merge1 Process2[Stok Kontrolü] Process1 --> DB1 Merge1 --> DB2 Decision2{Stok Var m1?} DB1 --> RefA DB2 --> OffPage AltProc1[Bekleme Listesine Ekle<br/>Alternatif] Decision1 -->|Giris| PreProc1 OffPage --> Doc4 Delay1(Bildirim Ayarla<br/>>Bekleme) PreProc1 --> RefA  $Doc4 \longrightarrow End3$ Doc1[Stok Giris<br/>br/>Bildirimi] RefA --> Input1 style Start fill:#90EE90 End1([BİTİ\$]) Input1 --> ManualOp1 style End1 fill:#FFB6C1 Process3[Sepete Ekle] ManualOp1 --> Process2 style End2 fill:#FFB6C1 Decision3{Alisverise<br/>br/>Devam Et?} Process2 --> Decision2 style End3 fill:#90EE90 Prep1[Sepet Özeti<br/>
>Hazırla] Decision2 --> Hayır | AltProc1 style Decision1 fill:#FFE4B5 Process4[Fiyat Hesapla] AltProc1 --> Delav1 style Decision2 fill:#FFE4B5 Delay1 --> Doc1 Doc2[Fatura Olustur] style Decision3 fill:#FFE4B5 MultiDoc1[[Kargo Etiketleri<br/>
Coklu Doküman]] Doc1 --> End1 style Decision4 fill:#FFE4B5 ManualInput2[/Ödeme Bilgilerini Gir/] Decision2 --> Evet | Process3 style DB1 fill:#87CEEB PreProc2[[Ödeme İşlemi<br/>br/>Banka API]] Process3 --> Decision3 style DB2 fill:#87CEEB Delay2(Sistem Onavi<br/>>Bekle) Decision3 --> | Evet | RefA style RefA fill:#DDA0DD Decision4{Odeme<br/>Basarılı mı?} Decision3 --> Hayır | Prep1 style Merge1 fill:#F0E68C Doc3[Hata Mesajı] Prep1 --> Process4 style PreProc1 fill:#F5DEB3 End2([BİTİS]) Process4 --> Doc2 style PreProc2 fill:#F5DEB3 Doc2 --> MultiDoc1

- Açık Kaynak (Open Source): Bir bilgisayar yazılımının makine diline dönüştürülüp kullanımından önceki, programcılar tarafından okunur, anlaşılır, yeni amaçlara uygun değiştirilebilir halinin gizli tutulmayıp açık, yani okunabilir halde kamuyla paylaşılıyor olmasına verilen isimdir.
- Kaynak kodları isteyen herkese açık olan yazılımdır.
- Kullanıcıların değiştirme özgürlüğü sağlanır.



• Özgür Yazılım (Free Software): Kullanıcıların çalıştırma, kopyalama, dağıtma, inceleme, değiştirme ve geliştirme özgürlükleri tanıyan yazılım türüdür.









- Derleyici (Compiler): Kaynak dili yüksek seviyeli bir dil, amaç dili alçak seviyeli (makine) bir dil olan çevirici programlara denir.
- Kaynak koddan, donanıma veya işletim sistemine uygun başka bir çalışan program oluşturmaktadır.
- Kaynak Kod -> Derleyici -> Çalışabilir Program
- Yorumlayıcı (Interpreter): Kaynak kodların satır satır ele alınarak doğrudan çalıştırılmasıdır.
- Kaynak Kod -> Yorumlayıcı -> Çalışan Kodlar

- Debug: Mantıksal hataları giderebilmek ve yazılımdaki hataları (bug) bulabilmek için yapılan işlemin adıdır.
- Genellikle yazılan programın adım adım ve denetim altında çalıştırılmasıdır.



• IDE (Integrated Development Environment): Tümleşik Geliştirme Ortamı, bilgisayar programcılarının hızlı ve rahat bir şekilde program geliştirebilmesi amacıyla içerisinde birçok araç barındıran, programlama diline göre sözdizimi renklendirmesi yapabilen kod yazım editörüdür.



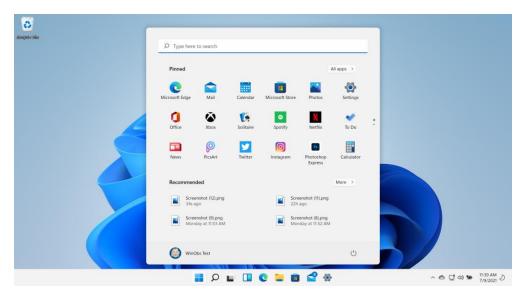








• GUI (Grafical User Interface): Grafiksal Kullanıcı Arayüzü, Bilgisayarlarda işletilen komutlar ve bunların çıktıları yerine simgeler, pencereler, düğmeler ve panellerin tümünü ifade etmek için kullanılan genel addır.









# \* Programlama Dillerinin Sınıflandırılması ve Değerlemesi

- Genel özelliklerine göre:
- Temel, Veriye Yönelik, Nesneye Yönelik
- Seviyelerine göre:
- Çok yüksek, yüksek, orta, alçak, makine dilleri
- Uygulama alanlarına göre:
- Genel amaçlı, sistem programlama, bilimsel mühendislik, veritabanı, yapay zeka



- Programlama dillerinin değerlemesi:
- İfade Gücü (Expressivity)
- Veri Türleri ve Yapıları (Data Types and Structures)
- Giriş/Çıkış Kolaylığı (Input/Output Facilities)
- Taşınabilirlik (Portability)
- Alt Programlama Yeteneği (Modularity)
- Verimlilik (Efficiency), Genellik (Generality)
- Okunabilirlik (Readability), Öğrenme Kolaylığı (Pedagogy)
- Esneklik (Flexibility)
- Yapısallık (Support for Structural Programming)
- Nesne Yönelimlik (Object Orientation)

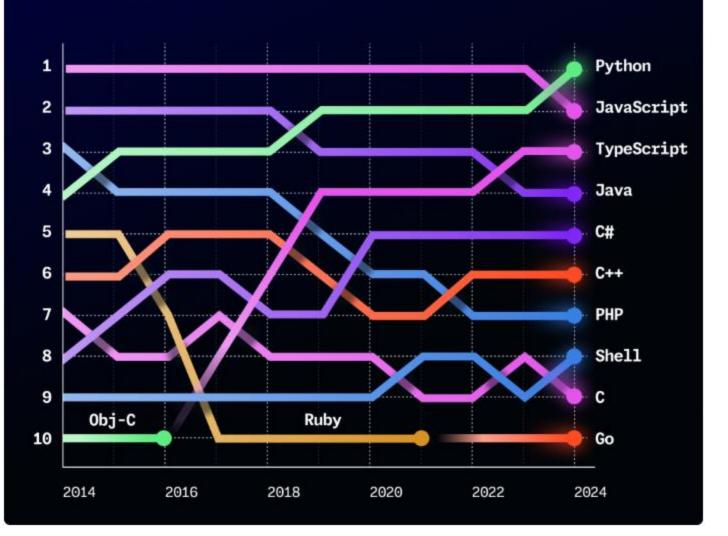


- Python: Yüksek seviyeli, çok paradigmalı ve dinamik tipli bir programlama dilidir. Donanım detaylarından tamamen soyutlanmış olması ve insan diline yakın sözdizimi sayesinde hem öğrenilmesi hem de okunması son derece kolaydır. Nesne yönelimli, prosedürel ve fonksiyonel programlama paradigmalarını bir arada destekleyerek geliştiricilere büyük esneklik sunar.
- Python'un en belirgin özelliği yüksek ifade gücüdür. Az miktarda kod ile karmaşık işlemler gerçekleştirilebilir ve zengin standart kütüphanesi sayesinde pek çok işlevsellik hazır olarak sunulur. Liste comprehensions, decorators ve generators gibi gelişmiş dil özellikleri, kod yazımını hem daha kısa hem de daha elegant hale getirir. Duck typing ve dinamik tip sisteminin sağladığı esneklik, hızlı prototipleme ve geliştirme süreçlerini kolaylaştırır.

- Python: Dilin yorumlamalı doğası, derleme süreci gerektirmeden anında test ve çalıştırma imkanı sağlar. Python kodu önce bytecode'a dönüştürülür ve Python sanal makinesinde çalıştırılır. Bu yapı platformdan bağımsızlık sağlarken, C veya C++ gibi derlenmiş dillere kıyasla çalışma zamanında performans kaybına yol açabilir. Ancak geliştirme hızındaki artış ve kod okunabilirliğindeki üstünlük, birçok uygulama için bu dezavantajı telafi eder.
- Python'un kod okunabilirliği konusundaki yaklaşımı benzersizdir. Girintileme ile kod bloklarının belirlenmesi, kodu görsel olarak daha anlaşılır kılar ve "Pythonic" kod yazma felsefesi, temiz ve bakımı kolay yazılımlar üretilmesini teşvik eder. Geniş ekosistemi ve topluluk desteği ile yapay zeka, makine öğrenmesi, veri bilimi, web geliştirme ve otomasyon gibi pek çok alanda endüstri standardı haline gelmiştir.

# Top programming languages on GitHub

RANKED BY COUNT OF DISTINCT USERS CONTRIBUTING TO PROJECTS OF EACH LANGUAGE.





#### Aggregated Survey Results on Top 10 Programming Languages

TIOBE	GitHub	PYPL	Stack Overflow	IEEE
Python	Python	Python	Python	Python
C++	JavaScript	Java	JavaScript	Java
Java	TypeScript	JavaScript	SQL	JavaScript
С	Java	C/C++	HTML\CSS	C++
C#	C#	C#	TypeScript	TypeScript
JavaScript	C++	R	Rust	SQL
Visual Basic	PHP	PHP	Go	C#
Go	Shell	TypeScript	Bash\Shell	Go
Fortran	С	Swift	C#	С
Delphi\Object Pascal	Go	Rust	C++	HTML