

DONANIM NEDİR?

- Donanım elle tutup gözle görebildiğimiz, yazılımın üzerinde çalıştığı makineler olarak tanımlanabilir.
- Bir bilgisayarın var olabilmesi için donanımlar en temel anlamda üç parçadan oluşur.
- Bunlar işlemci, hafıza ve giriş çıkış birimleridir.

Bilgisayar donanım ve yazılım olmak üzere iki bölümden oluşur. Bilgisayarın çalışabilmesi için, ikisi de gereklidir.

Donanım: Bilgisayarın gözle görülen, elle tutulan tüm elektronik parçalarına donanım denir. Örnek:Fare, klavye,anakart,sabitdisk gibi.
Donanım birimleri : Kasanın içinde bulunan donanımlara dahili donanım, dışında olan donanımlara harici donanımlar denir.

DAHİLİ DONANIMLAR

ANAKART



Tüm donanım birimleri anakarta bağlanır. Anakartın görevi işlemci ve diğer tüm donanım birimleri arasındaki iletişimi ve veri alışverişini sağlamaktır.

İŞLEMCI



İŞLEMCI bilgisayarın beyni sayılabilir. Tıpkı beyni gibi bilgisayarın işlevlerini ve birimler arasındaki veri akışını kontrol eder, komutları yerine getirir. Program komutlarını okumak ve anlamlı bir hale getirmek, bilgisayar parçalarının işleyişini yönlendirmek, bilgisayarın program ve veri akışını kontrol etmek işlemcinin görevidir. İşlemci eğer hızlı ise bilgisayarın da hızı daha iyi olacaktır.

SABİT DİSK



Sabit diskler adından da anlaşılacağı gibi esas olarak taşınamayan ve bilgisayarın içinde bulunan birimlerdir. İşletim sistemi, uygulama programları ve her türlü veriler sabit diskte saklanır. Bilgisayarın hafıza merkezidir, her şey burada saklanır. Sabit diskler çok hassastır kasanın düşürülmesi sırasında, tozdan veya elektrik kesintilerinde bozulabilir ve tamiri mümkün değildir.

BELLEK



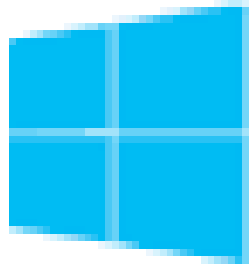
Bilgisayarın işlemleri yapmak için kullandığı belleği ifade eder.Çalışma sırasında kullanılacak programlar ve veriler bu belleğe aktarılarak işleme sokulur. B ellekteki bilgiler bilgisayar kapandığında, elektrik kesildiğinde silinir.Üzerinde işlem yapılan verilerin kalıcı olması için sabitdiske kaydetmek gerekir. Bellek kapasitesi ve hızlı fazla ise bilgisayar da hızlanır.

EKRAN KARTI

Ekran kartı görüntüyü oluşturur. Bir ucu kendisine bir ucu da ekrana bağlanan kablo ile oluşturduğu görüntüyü ekrana gönderir. Ekran kartınız iyi ise özellikle görüntünün önemli olduğu durumlarda çok rahat edersiniz. Oyunlar gibi. Eğer ekran kartı iyi değilse görüntü kalitesi düşük olacaktır.

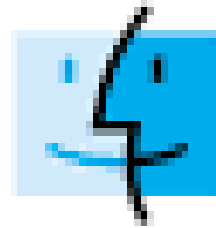
İŞLETİM SİSTEMLERİ

- İşletim sistemleri temel olarak donanım ve yazılım katmanları arasındaki fonksiyonelliği sağlayan ve bütün uygulamaların çalışacağı platformu sunan, donanımı kontrol eden, bir katmandır.
- Yani işletim sistemleri olmadan bilgisayar üzerinde program çalıştırılmaz.



Windows

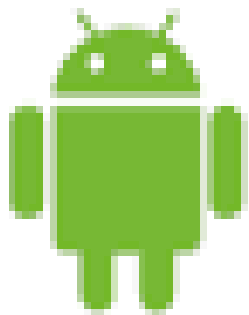
Linux



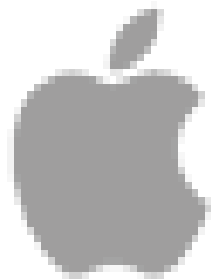
Mac OS



chromium



android



ios



UNIX®



YAZILIM NEDİR?



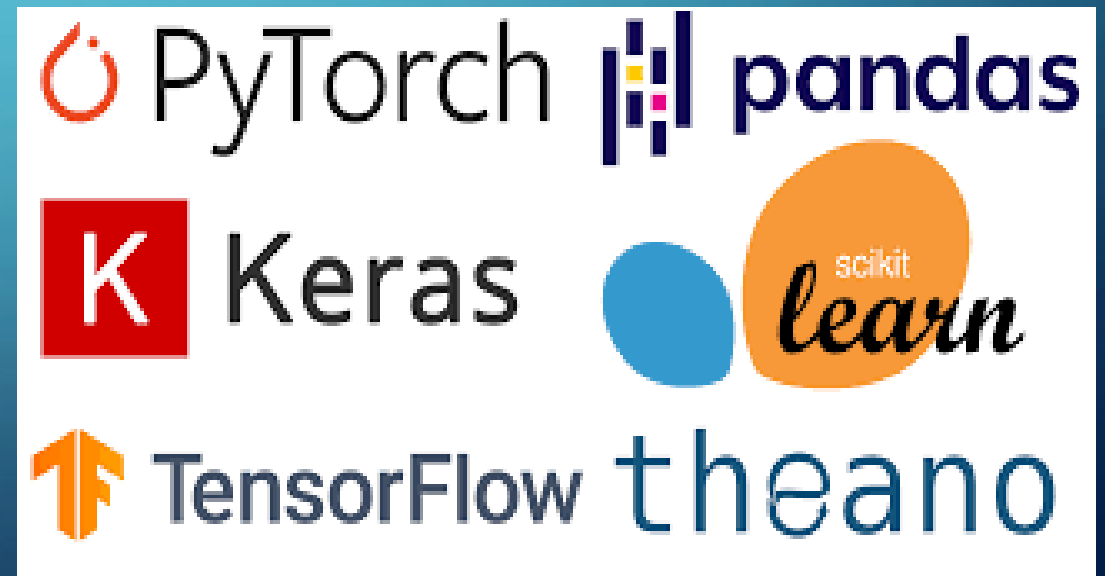
- Elektronik aygıtların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir. Bir başka deyişle, var olan bir problemi çözmek amacıyla bilgisayar dili kullanılarak oluşturulmuş anlamlı kodlar bütünüdür

YAZILIMIN ÖNEMİ VE GELECEĞİ

- Yazılım hayatımızın her yerinde olan, hepimizi ve her mesleği etkileyen bir unsurdur. Günümüzde yazılım kullanmayan bir sektörden ya da meslekten bahsetmek çok zor. Bilişim dünyası her geçen gün büyüyor ve bu dünya da yeni bir çok teknoloji hayatımıza giriyor.

YAZILIMIN ZAMAN İÇİNDE DERİNLEŞMESİ

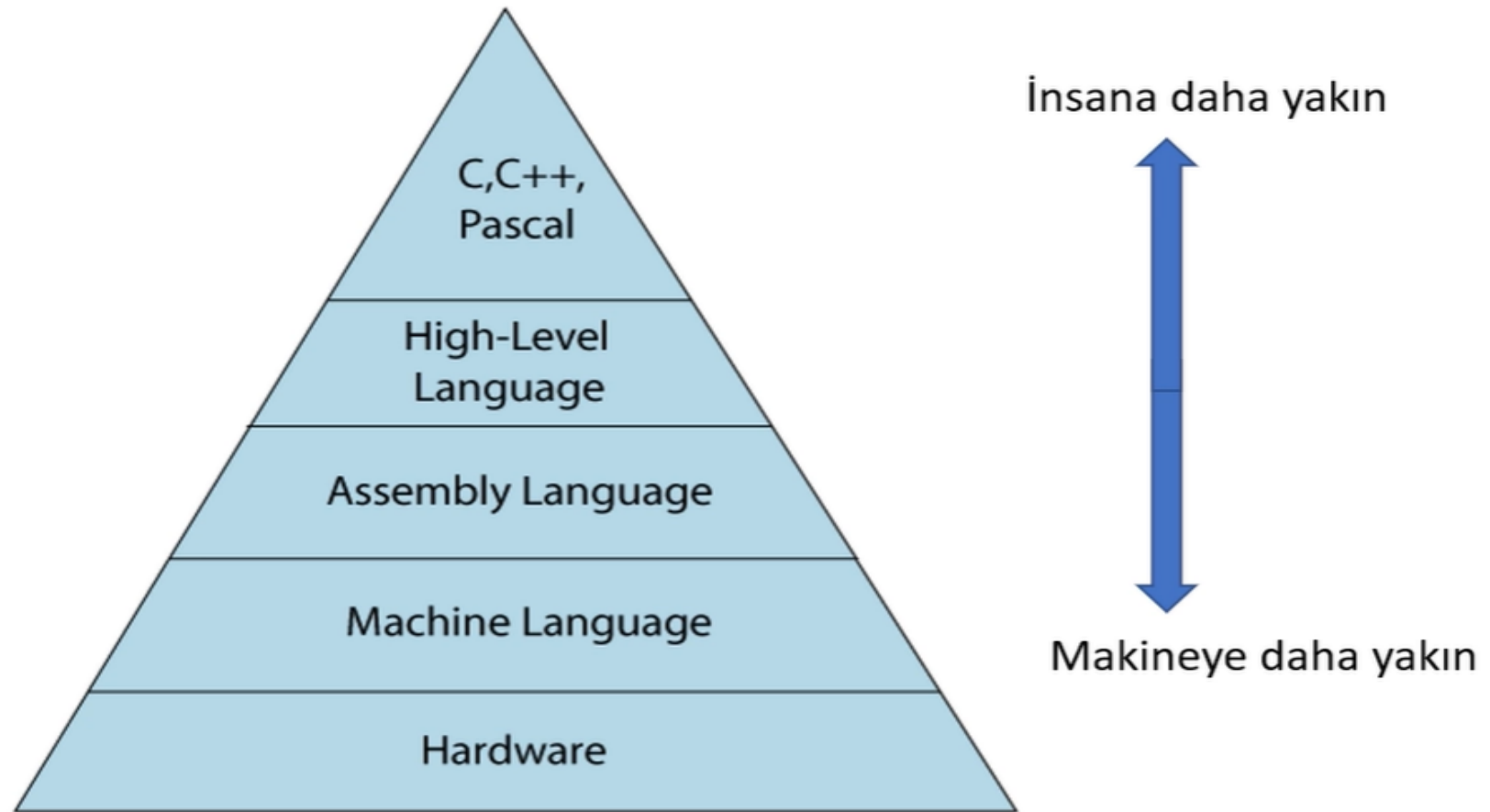
- Yazılımda ki derinleşme, uzmanlaşma da büyümektedir. Eskiden yazılım uzmanı denilen kişiler şimdiye göre daha ilkel programlar yazan insanlarken günümüzde daha detay ve daha niş sayılabilecek uzmanlık alanlarına geçiş yapmışlardır.



PROGRAMLAMA DİLİ NEDİR?

- Programlama dili, yazılımcının bir algoritmayı ifade etmek amacıyla, bir bilgisayara ne yapmasını istediğini anlatmasının yoludur.
- Programlama dilleri, yazılımcının bilgisayara hangi veri üzerinde işlem yapacağını, verinin nasıl depolanıp iletileceğini, hangi koşullarda hangi işlemlerin yapılacağını tam olarak anlatmasını sağlar.

Programlama Dillerinin Alt Grupları



ÖZETLE

- Programlama dilleri bizim anladığımız kelimeleri kullanarak, bizim hedeflediğimiz amaca yönelik vermiş olduğumuz komutları bilgisayarın anlayacağı şekle çevirir ve kullanabileceğimiz hazır kütüphaneleri veya diğer yazılımları bize sunabilecek ortamlar sunmaktır.



DERLEYİCİ(COMPİLER)

- Programcılıkta, bir programlama dilinde yazılmış olan kaynak kodunu başka bir dile (genellikle makine koduna) çeviren yazılım. Derleyiciye bunu yaptırmaktaki amaç genellikle çalışabilir bir yazılım elde etmektir. Kullanıcıların programları kullanırken kolaylık sağlamak amaçlı geliştirilmiştir

```
1 #include <iostream>
2 using namespace std;
3 int main ()
4 {
5     count << "Hello World!";
6     return 0;
7 }
8
9
10
```

Kodları şimdi
anlayabiliyorum!



YORUMLAYICI(İNTERPRETER)

- Yazılımı parça parça ele alarak çalıştırır.
- Compiler(derleyici)' den farklı olarak hata bulduğu yerde direk derleme işlemini durdurur ve hatanın nerede olduğunu söyler.

```
graph LR; A[High Level Language] --> B[Interpreter]; B --> C[Machine Language];
```

High Level
Language

Interpreter

Machine
Language

ENTEĞRE EDİLMİŞ ORTAMLAR(IDE)

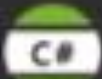
- Entegre geliştirme ortamı (IDE), programcılarının verimli şekilde yazılım kodu geliştirmesine yardımcı olan bir uygulamadır.
- Yazılım düzenleme, oluşturma, test etme ve paketleme gibi özellikleri kullanımı kolay bir uygulamada birleştirilerek geliştiricilerin üretkenliğini artırır.




FileEditViewGotoHelp

EXTENSIONS


@popular




C# 1.2.2 356K ★★★★★
C# for Visual Studio Code (p...
Microsoft Install




Python 0... 211K ★★★★★
Linting, Debugging (multi-t...
Don Jayamanne Install




Debugger for Chrome 148
Debug your JavaScript code...
Microsoft JS Diagno... Install



C/C++ 0.7... 143K ★★★★★
Complete C/C++ language ...
Microsoft Install



Go 0.6.39 99K ★★★★★
Rich Go language support f...
lukehoban Install



ESLint 0.10... 88K ★★★★★
Integrates ESLint into VS Co...
Dirk Baumer Install

app.ts

www.ts

package.json

README.md

```
1 import app from './app';
2 import debugModule = require('debug');
3 import http = require('http');
4
5 const debug = debugModule('node-express-typescript:server');
6
7 // Get port from environment and store in Express.
8 const port = normalizePort(process.env.PORT || '3000');
9 app.set('port', port);
10
11 // create
12 const server = http.createServer();
13 server.listen(port);
14 server.on('error', (err) => {
15   if (err.syscall !== 'listen') {
16     throw err;
17   }
18   const bind = typeof port === 'string'
19     ? ` ${port}` : ` ${port} `;
20   function normalizePort(val: any): number|string|boolean {
21     let port = parseInt(val, 10);
22
23     if (Number.isNaN(port)) {
24       port = 3000;
25     }
26
27     if (port < 0) {
28       return false;
29     }
30     if (port > 4899) {
31       return false;
32     }
33     return port;
34   }
35
36   if (port === false) {
37     console.log(`Invalid port: ${port}, defaulting to 3000`);
38     return 3000;
39   }
40
41   return port;
42 }
43
44 const server = http.createServer(app);
45 server.listen(port);
46
47 export { server };
```

+0 CSSImportRule

+0 CSSSupportsRule

export

exports

import

importScripts

+0 MessagePort

normalizePort

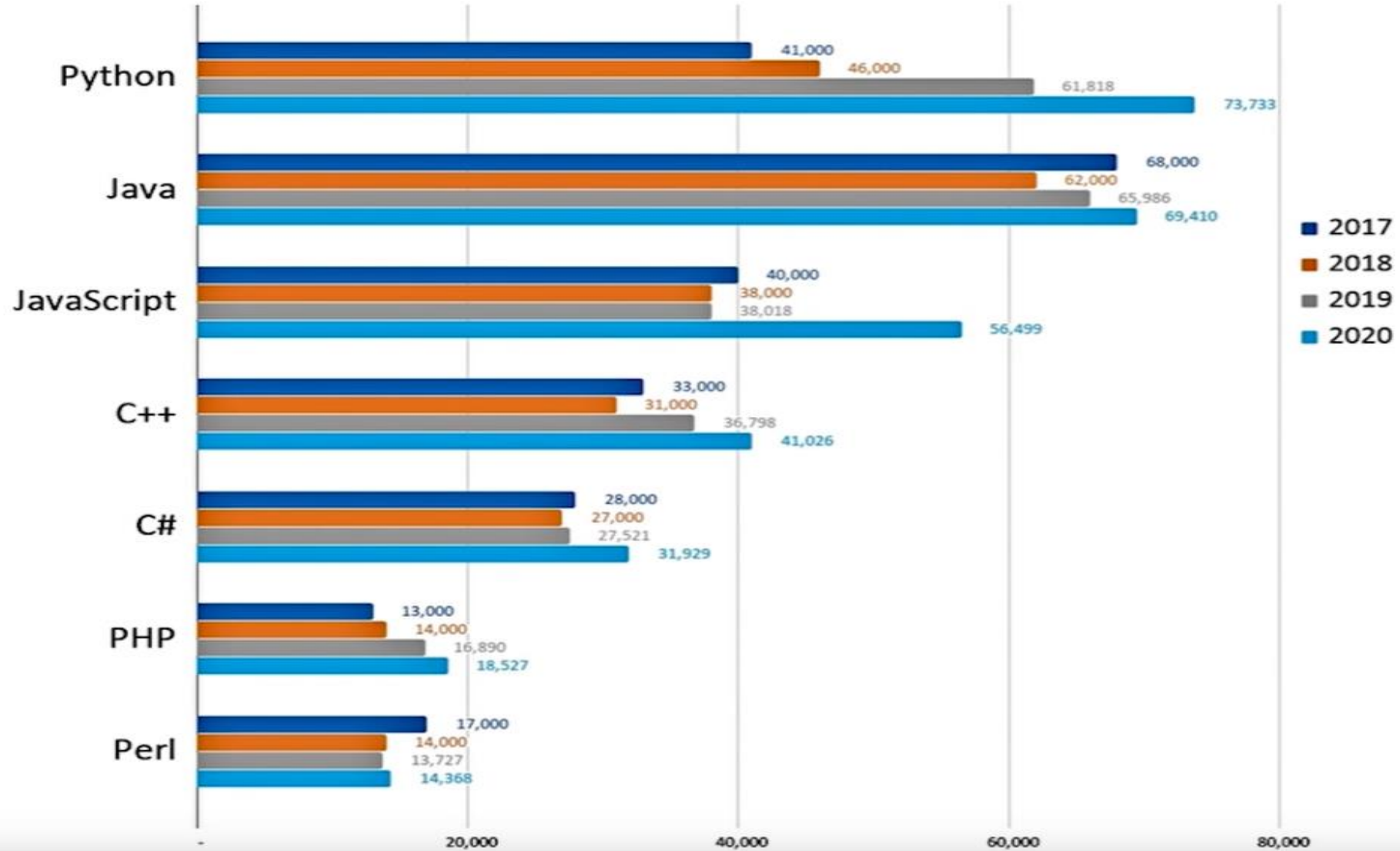
port const port: number | string | boolean

Ln 9, Col 21 Spaces: 2 UTF-8 LF TypeScript

POPÜLER PROGRAMLAMA DİLLERİ

- Programlama dilleri belirli bir amaca yöneliktir.
- Geçmişteki programlama(c,c++) dilleri derinlik ve özellik itibariyle oldukça sıkı ve sonuca yönelik dillerdir.
- Günümüzdeki diller(Python,go) ise daha esnektir.

Kullanım popülarlığıne göre diller;










HANGİ DİL İLE BAŞLANMALI VE NASIL İLERLENMELİ

- İlgilendiğiniz yazılım alanına en uygun dili seçerek başlamanız önerilir.
- Başlangıçta hangi programlama dilinin seçildiğinden daha da önemli olan unsur bir programlama dilinin nasıl öğrenileceğini öğrenmektir.



Which Programming Language to Learn Based on Your Career Goals

| Front-end web development | Back-end web development | Mobile development |
|--|--|---|
|  JavaScript |  JavaScript |  Swift |
|  Elm |  Scala |  Java |
|  TypeScript |  Python |  Objective C |
| |  Go |  JavaScript |
| |  Ruby | |
| Game development | Desktop applications | Systems programming |
|  Unity |  Scala |  Go |
|  TypeScript |  Go |  Rust |
| |  Python | |

NASIL ÖĞRENMELİYİZ?

- Bir dilin gelişimi için, pratik ve uygulama çok önemlidir.
- Teorik bilgileri derslerden, kitaplardan öğrenebiliriz ancak daha kapsamlı işlemler yapabilmek için sürekli pratik ve uygulama çok önemlidir.
- Ayrıca başka yazılımcıların kodlarını okuyarak, inceleyerek de kod yazma kabiliyetimizi geliştirebiliriz.



Algoexpert

VS.



JAHackerrank

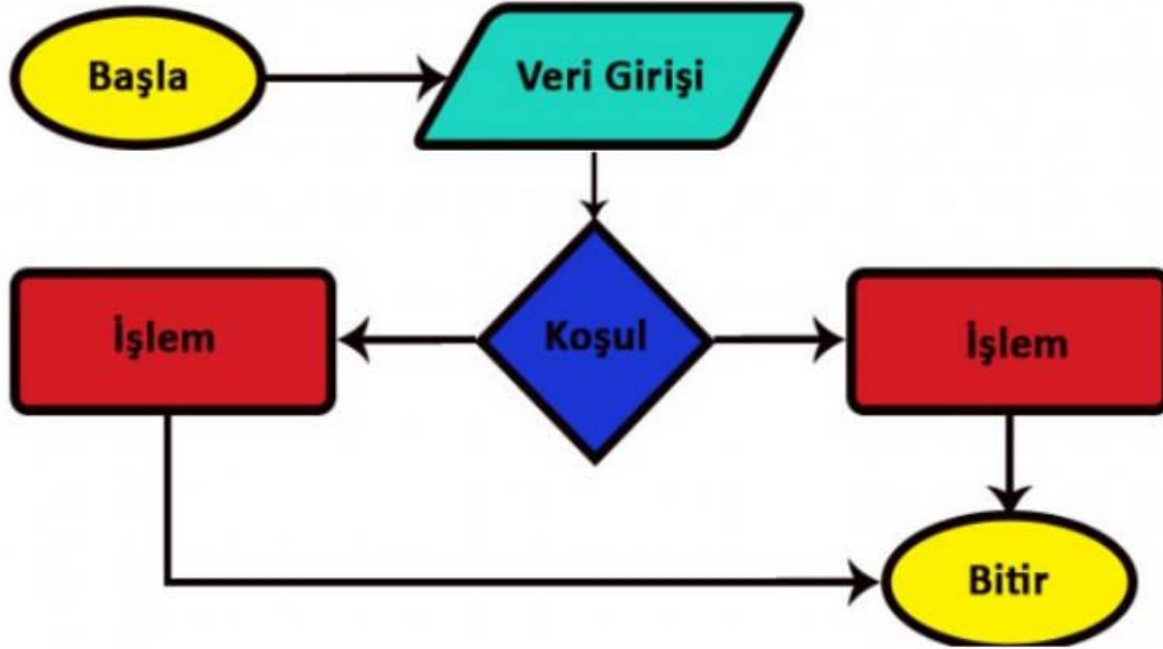
VS.



Leet Code

ALGORİTMA NEDİR?

- Bir sorunu çözebilmek için gerekli olan sıralı ve mantıksal adımların tümüne Algoritma denir.
- Bir algoritmada
 - Her adım mutlaka belirleyici olmalıdır. Hiçbir şey şansa bağlı olmamalıdır.
 - Algoritmanın sonu olmalıdır.
 - Algoritmalar tüm olasılıkları göz önüne alarak yazılmalıdır.



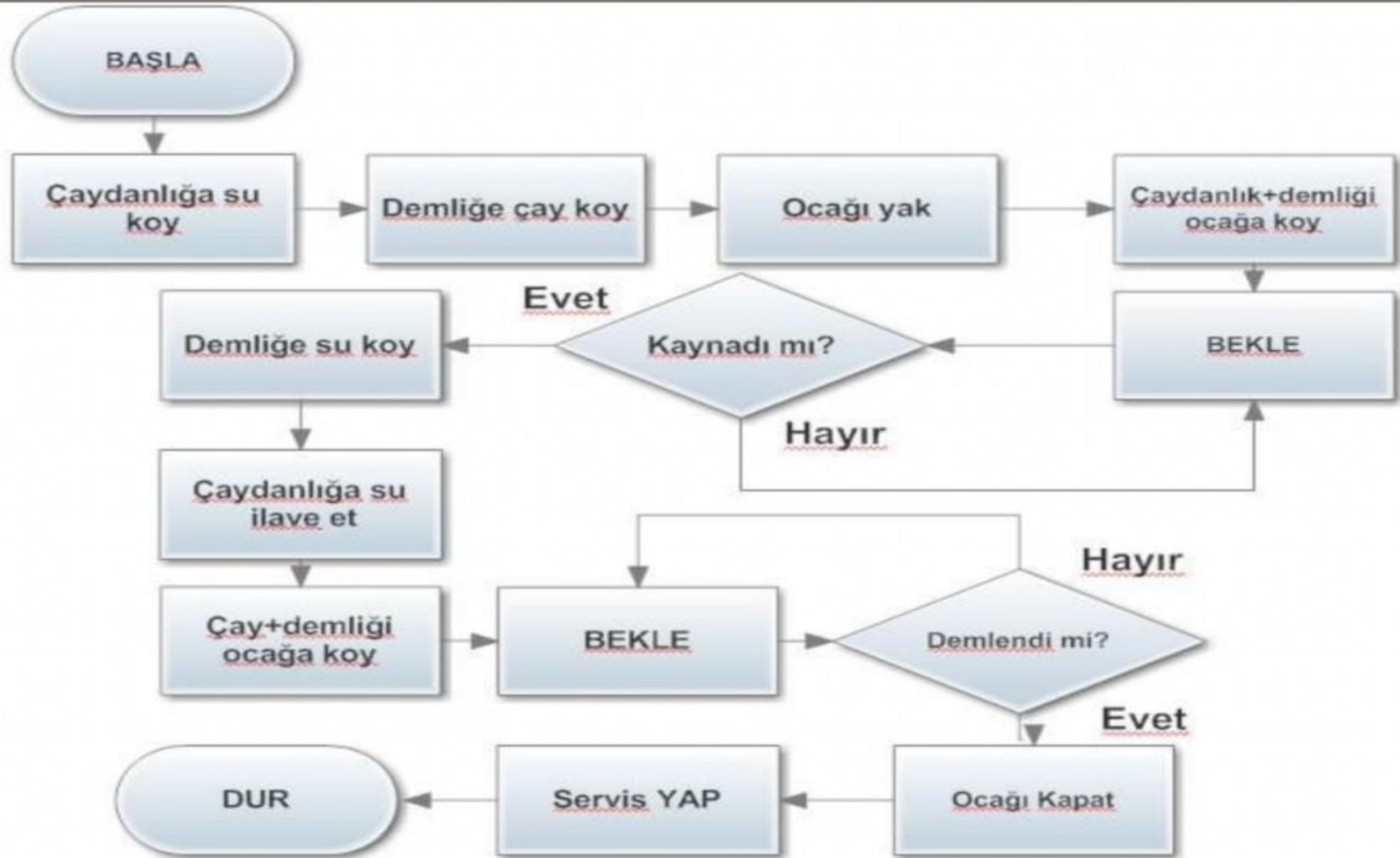
Algoritma Nedir ?

AKIŞ DİYAGRAMI NEDİR?

- Akış şeması algoritmaları ve işlemleri birbirine oklarla bağlı değişik tiplerdeki kutular içerisinde gösteren yaygın bir şema tipidir. Akış şemaları çeşitli alanlardaki işlem ve uygulamaların yönetilmesi, belgelendirilmesi, tasarlanması ve çözümlenmesinde kullanılır.

Çay Demleme Algoritması:

- 1- Başla
- 2- Çaydanlığa su doldur
- 3- Ocağı yak
- 4- Çaydanlığı ocağa koy
- 5- Bir süre suyun kaynamasını bekle
- 6- Su kaynadı mı? Kaynamadıysa 5. adıma git Kaynadıysa 7. adıma git
- 7- Çayı demle
- 8- Çaydanlığa su ilave et



ARİTMETİK OPERATÖRLER

- Toplama $a+b$
- Çıkarma $a-b$
- Çarpma $a*b$
- Bölme a/b
- Üs alma a^b
- Mod alma $a\%b$

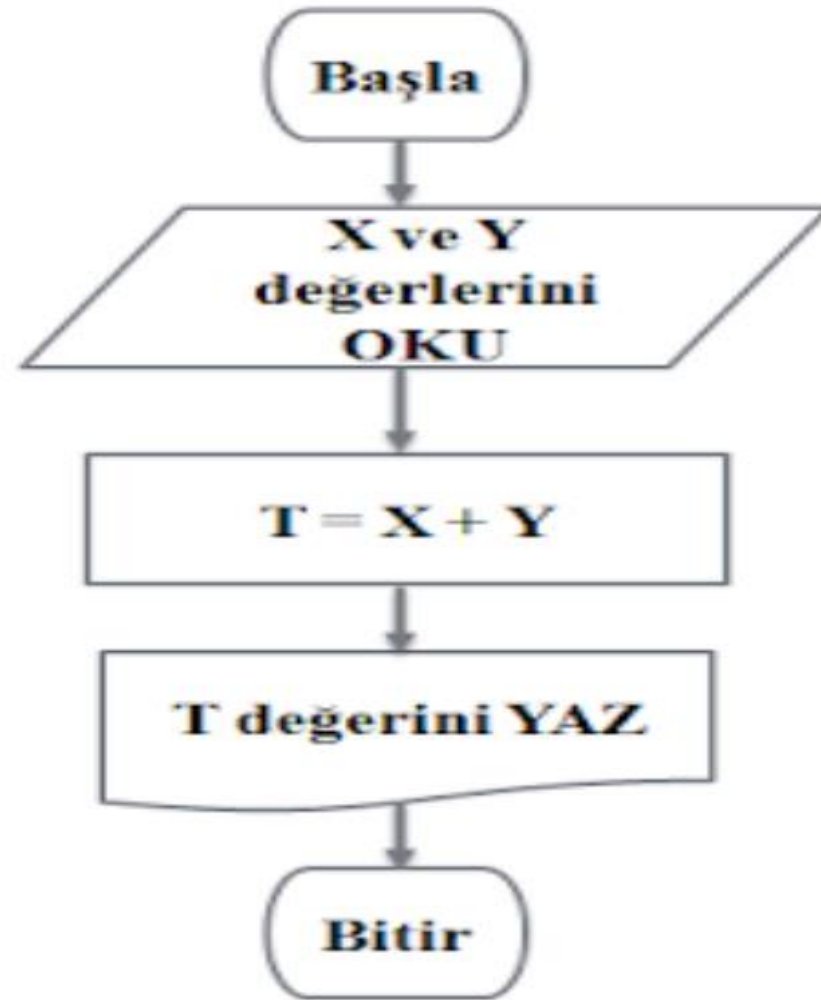
İŞLEM ÖNCELİĞİ

- Matematiksel işlemler içeren kümemizin içindeki matematiksel işaretlerin bir işlem önceliği vardır.
- İşlem Önceliği Sırası:
 - 1. Parantezler ()
 - 2. Üs Alma a^b
 - 3. Çarpma ve bölme $a*b$ veya a/b
 - 4- Toplama ve çıkarma $a+b$ veya $a-b$

İKİ SAYININ TOPLAMI ALGORİTMASI

- 1- Başla
- 2- Birinci sayıyı gir(x)
- 3- İkinci sayıyı gir(y)
- 4- İki sayıyı topla($x+y$)
- 5- Sayıların toplam(t) değerini yaz
- 6- Bitir

Akış Şeması



İKİ SAYIDAN BÜYÜK OLANI BULAN ALGORİTMA

- 1- Başla
- 2- Birinci sayıyı giriniz (X)
- 3- İkinci sayıyı giriniz (Y)
- 4- Eğer $X > Y$ ise 1. sayı büyüktür yaz
- 5- Eğer $Y > X$ ise ekrana 2. sayı büyüktür yaz
- 6- Değilse ekrana sayılar birbirine eşittir yaz
- 7- Bitir

