

**ETİK:** Bireylerin ahlaklı ve erdemli bir hayat yaşayabilmeleri için hangi davranışların doğru, hangilerinin yanlış olduğunu araştıran bir felsefe dalıdır. Bilişim teknolojilerinin ve internetin kullanımı sırasında uyulması gereken kuralları tanımlayan ilkelere **bilişim etiği** denir. Bilişim teknolojilerinin kullanımında yaşanan sorunlar dört temel başlıkta ele alınmaktadır. Bunlar: Fikri Mülkiyet, Erişim, Gizlilik ve Doğruluk.

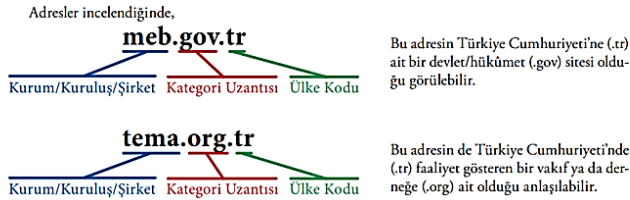
**1-Fikri Mülkiyet:** Kişinin kendi zihni ile ürettiği her türlü ürün olarak tanımlanmaktadır. Eserin (yazılım, resim, kitap, vb.) sahibi kimdir? ve kimlerin kullanımına izin verilmiştir? Sorularının cevabı fikri mülkiyet başlığının altında incelenmektedir.

Yazılımlar fikri mülkiyet açısından ikiye ayrılmaktadır. Özgür yazılımlar (Ücretsiz – Freeware) ve Ticari yazılımlar (Shareware – Paylaşım yazılımları)

**2-Erişim:** Herhangi bir arama motorunu kullanarak, istediğimiz bilgiye hızlıca erişebiliriz. Ancak bu bilgiye erişim hakkımız var mı? Bu bilgiye erişim ücretli mi?

**3-Gizlilik:** Kişiyeye ait her türlü bilgiyi saklama becerisidir. Kişisel bilgilerinize ulaşılmasını istemiyorsanız alternatif arama motorları kullanabilirsiniz ([www.duckduckgo.com](http://www.duckduckgo.com))

**4-Doğruluk:** İnternet üzerindeki bilgiler her zaman doğru ve iyi niyetli olmayabilir. Bu nedenle elde edilen bilgiler kullanılmadan önce bir dizi tedbir almak önemlidir. Bu tedbirler: \* Bilginin kaynağı belirtilmiş olmalıdır. Kaynağı belirtilmemiş bilgiye şüpheyle yaklaşılmalıdır. \* Elde edilen bilgiler en az üç farklı kaynaktan teyit edilmelidir. \* Bilgiyi aktaran internet sitesinin adresi kontrol edilmelidir.



Alan adı uzantıları birçok internet sitesi için fikir verebilir: **.com** ve **.net**→ Ticari firma, **.gov**→ Resmi kurum, **.org**→ Dernek veya Vakıf, **.edu**→ Üniversite, **.k12**→ Eğitim kurumu (anaokulu, ilkokul, lise),...

Ülke kodu: İnternet adreslerinin son kısmı ülke kodudur. Ülke kodu **.tr** olan siteler daha güvenilirdir.

**İNTERNET ETİĞİ:** Kişilik hakları, özel yaşamın gizliliği ve veri güvenliği gibi başlıklar altında incelenebilir. İnternet ortamında uyulması gereken etik kurallar: \* Bize yapılmamasından hoşlanmadığımız davranışları başkalarına yapmaktan kaçınmalıyız. \* Gerçek hayatta nasıl davranıyorsak internet ortamında da öyle davranmalıyız. \* İnternet ortamındaki kişilere saygı kuralları çerçevesinde davranmalıyız. \* İnterneti kullanırken her kültüre ve inanca saygılı olmak gerektiği unutulmamalıdır. \* İnternet ortamında insanların paylaştığı bilgilerin izinsiz kullanımından kaçınmamız gerekir.

Siber zorbalığa maruz kalınırsa: hesabı engelleyin, sosyal medya platformundaki "Bildir/Şikayet Et" bağlantısını kullanarak şikayet ediniz. Devam etmesi durumunda ekran görüntülerini ve mesajları kaydedip hukuki yollara başvurunuz.

**BİLGİ GÜVENLİĞİ:** Kişisel ya da kurumsal düzeyde bizim için büyük önem taşıyan her türlü bilgiyi tehditlere karşı korumak için alınan tedbirlere **Bilgi Güvenliği** denir. Bilgi güvenliği; Gizlilik, Bütünlük ve Erişilebilirlik olmak üzere üç temel öğeden oluşur. Bu üç öğeden birinin bile zarar görmesi durumunda güvenlik zafiyeti oluşabilir. **Gizlilik:** Bilginin yetkisiz kişilerin eline geçmemesi için korunmasıdır. **Bütünlük:** Bilginin yetkisiz kişiler tarafından değiştirilme ve silinmeye karşı korunmasıdır. **Erişilebilirlik:** Bilginin yetkili kişilerce ihtiyaç duyulduğunda ulaşılabilir ve kullanıma hazır durumda olmasıdır.

Bilişim sistemlerine karşı yapılan kötü niyetli davranışlara siber saldırı veya atak denilmektedir.

**Siber** ya da **siber uzay**; temeli bilişim teknolojilerine dayanan, tüm cihaz ve sistemleri kapsayan yapıya verilen genel addır.

**Siber Savunma:** Siber ortamda yaşanabilecek suç, saldırı, terörizm, savaş gibi tüm kötü niyetli hareketlere karşı alınacak tedbirler bütünüdür.

**Siber Suç:** Bilişim teknolojileri kullanılarak gerçekleştirilen her tür yasa dışı işlemdir.

**Siber Saldırı:** Hedef seçilen şahıs, şirket, kurum, örgüt gibi yapıların bilgi sistemlerine veya iletişim altyapılarına yapılan planlı ve koordineli saldırıdır.

**Siber Savaş:** Farklı bir ülkenin bilgi sistemlerine veya iletişim altyapılarına yapılan planlı ve koordineli saldırılardır.

**Siber Zorbalık:** Bilgi ve iletişim teknolojilerini kullanarak bir birey ya da gruba, özel ya da tüzel bir kişiliğe karşı yapılan teknik ya da ilişkisel tarzda zarar verme davranışlarının tümüdür.

**PAROLA:** Bir hizmete erişebilmek için gerekli olan, kullanıcıya özel karakter dizisidir. Güçlü bir parolanın belirlenmesi için aşağıdaki kurallar uygulanmalıdır: • Parola, büyük/küçük harfler ile noktalama işaretleri ve özel karakterler içermelidir. • Parola, -aksi belirtilmedikçe- en az sekiz karakter uzunluğunda olmalıdır. • Parola, başkaları tarafından tahmin edilebilecek ardışık harfler ya da sayılar içermemelidir. • Her parola için bir kullanım ömrü belirleyerek belirli aralıklar ile yeni parola oluşturulması gerekir.

**VİRÜSLER:** Bulaştıkları bilgisayar sisteminde çalışarak; sisteme ya da programlara zarar vermek amacıyla oluşturulan yazılımlardır. Bilgisayarın yavaşlamasını, programların bozulmasını, dosyaların silinmesini ya da şifrelenmesini, kişisel bilgilerin başka kişilerin eline geçmesini sağlayabilirler. Çeşitleri: **Bilgisayar Solucanları**; kendi kendine çoğalan ve çalışabilen, bulaşmak için ağ bağlantılarındaki açıkları kullanan, kötü niyetli programlardır. **Truva Atları**: Truva atları çalıştırıldıklarında bilgisayarda bir açık kapı meydana getirirler. Bilgisayarın kontrolünü başka kişilere verebilirler. Kişisel bilgilerinizi yöneticisine iletebilir ve başka virüslerin, açık kapıdan faydalanarak sisteme bulaşmasını sağlayabilirler. **Casus Yazılımlar**: İnternet'ten indirilerek bilgisayara bulaşan ve gerçekte başka bir amaç ile kullanılsa bile arka planda kullanıcıya ait bilgileri de elde etmeye çalışan programlardır. Bunlar, sürekli reklam amaçlı pencerelerin açılması ya da İnternet tarayıcıya yeni araçların eklenmesine neden olabilir.

**Zararlı Programlara Karşı Alınacak Tedbirler:** • Bilgisayara anti virüs ve İnternet güvenlik programları kurularak bu programların sürekli güncel tutulmaları sağlanmalıdır. • Tanınmayan/güvenilmeyen e-postalar ve ekleri kesinlikle açılmamalıdır. • Ekinde şüpheli bir dosya olan e-postalar

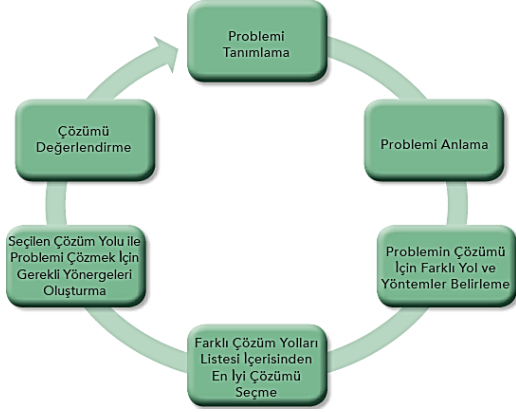
açılmamalıdır. Örneğin resim.jpg.exe isimli dosya bir resim dosyası gibi görünse de uzantısı exe olduğu için uygulama dosyasıdır. • Zararlı içerik barındıran ya da tanınmayan web sitelerinden uzak durulmalıdır. • Lisanssız ya da kırılmış programlar kullanılmamalıdır. • Güvenilmeyen İnternet kaynaklarından dosya indirilmemelidir.

**PROBLEM ÇÖZME:** Problemleri formüle edebilme, farklı ve yaratıcı çözüm yolları önerebilme, çözümü kesin ve doğru biçimde ifade edebilme becerisidir.

**Bilgi işlemsel düşünme:** Bilgisayar biliminin kavramlarından yararlanarak problem çözme, sistem tasarlama ve insan davranışlarını anlama olarak tanımlanabilir.

**Programlama:** Bilgisayarın donanıma nasıl davranacağını anlatan, bilgisayara yön veren komutlar ve işlemler bütünüdür.

**Program:** yapılacak bir işlemi ya da hesaplamayı gerçekleştirmek için birbirini izleyen komut ya da yönergelerden oluşan yapıdır. Programlama, karmaşık bir süreçtir ve programcılar programlamada **hata (bug)** yapabilirler. Programlama hatalarını bulma ve düzeltme işlemine **hata ayıklama (debugging)** denilir. Problem çözme sürecinde en iyi kararı verebilmek için izlenmesi gereken 6 adım vardır.



**Algoritma Nedir?** : Bir problemin çözümünde izlenecek yol anlamına gelir ve problemin çözümünün adımlar halinde yazılmasıyla oluşturulur. Genellikle matematikte ve programlamada bir işi yapmak için tanımlanan, belli bir başlangıcı ve sonu olan, açıkça belirlenmiş basamaklardır.

## VERİ TÜRLERİ

**1-Sayısal veriler (Nümerik):** Tüm sayı tiplerini içerir. Sayısal veri, hesaplama işlemlerinde kullanılabilen tek veri türüdür. Pozitif ya da negatif tam sayılar ve reel (ondalıklı)sayılar kullanılabilir. Banka hesap numarası ya da posta kodu gibi sayısal; fakat hesaplama için kullanılmayan veriler de vardır. Bu tür veriler sayısal olarak tanımlanmaz. Örnek: ücret=1400.75 gün=30

**2-Alfa numerik/Karakter Veri:** Karakter veri seti; tüm sayılar ("0".."9"), harfler ("a".."z", "A".."Z") ve özel karakterleri ("#", "&", "\*", ..) kapsar. Bu veri setinden oluşturulan değer, tırnak içinde belirtilir. Örnek: adres="Çınar sokak Sevgi apt. No 18/3"

**3-Mantıksal veri:** Veri setinde yalnızca iki kelime barındırır: Doğru (true) ve Yanlış(false). Bu veri evet ya da hayır şeklindeki karar verme süreçlerinde kullanılır. Örneğin elde edilen değer, beklenen değer mi, geçti mi, arabası var mı, öğrenci lise mezunu mu gibi sonucu kesin doğru ya da yanlış olan durumlarda mantıksal veri tanımlaması yapılır.

| Veri                             | Veri Türü                         | Açıklama   |
|----------------------------------|-----------------------------------|--|
| Ürün satış bedeli: 49,99, 101,50 | Sayısal: Reel                     | Bir ürünün satış bedeli hesaplama işlemlerinde kullanılır.   |
| T.C. Kimlik No: 10654876542      | Karakter dizisi                   | Kimlik bilgileri hesaplama amaçlı kullanılmaz.   |
| Ağırlık: 67                      | Sayısal: Tam sayı                 | Kilo cinsinden tam sayı olabilir ve hesaplamalarda kullanılır.   |
| Şirket İsmi: ABC Firması         | Karakter dizisi                   | Tamamen karakterlerden oluşur.   |
| Kredi Onayı: Var, Yok            | Mantıksal                         | Bu durumda onay ya vardır "Doğru" ya da yoktur "Yanlış".   |
| Posta Kodu: 06110, 34217         | Karakter dizisi                   | Posta kodları işlem yapmak için kullanılmaz.   |
| Tarih: 21042017                  | Karakter dizisi, Sayısal Tam sayı | İşlem yapmak için tam sayı biçiminde tanımlanabilir; aksi takdirde dizi olarak tanımlanması daha uygundur. |
| IBAN: TR0600006543000012         | Karakter dizisi                   | Para transferi için bankaya verilen kodlar hesaplama amaçlı kullanılmaz.                                   |

## BİLGİSAYAR VERİYİ NASIL SAKLAR?

Bilgisayar veriyi hafızada (RAM bellek) saklar. Her bir değişken için hafızada belirli bir alan ayrılır ve bu alan her seferinde tek bir değer saklayabilir. Kullanıcı, var olan değer yerine yeni bir değer atadığında eski değer silinir. Hafızada bu konumlar geçicidir. Programın çalışması bittiğinde ya da bilgisayar kapatıldığında bu veriler silinir.

Verilerin daha sonra tekrar kullanılması gerekiyorsa sabit disk gibi kalıcı bir konuma kaydedilmeleri gerekir. Bu şekilde kaydedilen verilere "**dosya**" adı verilir. Temel anlamda program dosyaları ve veri dosyaları olmak üzere iki dosya türü vardır.

**SABİTLER:** "Sabit" olarak tanımlanan veriler problemin çözüm süreci boyunca asla değişmeyen değerlerdir. Sabit değerler sayısal, karakter ya da özel semboller olabilir. Örneğin, pi değeri değişmeyen bir değer olacağı için sabit olarak tanımlanmalıdır. (Pi=3.14)

**DEĞİŞKENLER:** Her programlama dilinde program içerisinde üretilen değerleri geçici olarak saklamak amacı ile değişkenler kullanılır. Matematikte **X** ne ise bilgisayarda değişken "O" dur. Hangi değeri atarsanız (x=5), o değeri temsil eder. (y=2x+1 ise, y=11 olur) Değişkenin ismini programcı belirler. Değişken olarak tanımlanan verilerin değeri, program çalıştığı sürece değişebilir.

Programcılar çözüm sürecinde ihtiyaç duyulan her bir değişkene **farklı** bir isim vermelidir.

Değişken, farklı veri türlerinde olabilir ancak ismi, içerdiği değer ile tutarlı olmalıdır. Böylece program satırları daha anlaşılır olur ve değişken ismi, ne için kullanıldığını da belirtir.

Değişkenlere isim verirken ve bunları kullanırken dikkat edilmesi gereken kurallar şunlardır:

1. Değişkenlere isim verirken boşluk kullanmayınız. (sayı1=9)
2. Değişkenlere isim verirken bir karakter ile başlayınız. (s1,ad,..)
3. Matematiksel semboller kullanmamaya dikkat ediniz. (+,-,/,...)

| Yanlış   | Doğru  |
|----------|--------|
| 1 sayı   | sayı1  |
| Okul No. | okulNo |
| Soru?    | soru   |

**Fonksiyonlar:** Fonksiyonlar, belirli işlemleri yürüten ve sonuçları döndüren bir işlem kümesidir. Bilgisayar programlama dillerinde bir veya birden çok eylemi gerçekleştirmek için yazılan ve ana programa parametre ve fonksiyon dönüş değeri ile bağlı olan alt

programdır. Fonksiyonlar programla dillerinin olmazsa olmazlarından. Sayfalarca kod yazacağınza fonksiyonları kullanarak program satırlarınızı çok daha az satırlı hale getirebilirsiniz. Programın daha kolay geliştirilmesinisağlarlar. Her programlama dili, içerisinde kendine özgü fonksiyonlar barındırır. Bu fonksiyonların kütüphaneleri, programlama dili ve bilgisayara göre değişiklik gösterir.

**int(7.8) Sonuç: 7** **abs(-56) Sonuç: 56**  
**Parametre Fonksiyon** **Parametre Fonksiyon**

Tablo 1: Fonksiyon türleri ve örnekler

| Fonksiyon                         | Tanım  | Örnek                        | Sonuç      |
|-----------------------------------|--|------------------------------|------------|
| <b>Matematiksel Fonksiyonlar</b>  |  |                              |            |
| Sqrt (N)                          | N değerinin karekökünü döndürür.                               | Sqrt(16)                     | 4          |
| Abs (N)                           | N değerinin mutlak değerini döndürür.                          | Abs(-6)                      | 6          |
| Integer (N)                       | N değerine en yakın ya da eşit tam sayıyı döndürür.            | Integer(6.7689)              | 6          |
| Random                            | 0 ile 1 arasında rastgele bir sayı döndürür.                   | Random                       | 0.6783456  |
| <b>Dizi Fonksiyonlar</b>          |  |                              |            |
| Mid (S, n1, n2)                   | Dizinin n1 pozisyonundan başlayan n2 kadar karakteri döndürür. | Mid(S, 3, 3)<br>S= "Yasemin" | "sem"      |
| Left (S, n)                       | Dizinin sol tarafındaki n kadar karakteri döndürür.            | Left(S, 3)<br>S= "Yasemin"   | "yas"      |
| Right (S, n)                      | Dizinin sağ tarafındaki n kadar karakteri döndürür.            | Right(S, 4)<br>S= "Yasemin"  | "emin"     |
| Length (S)                        | Dizideki karakter sayısını döndürür.                           | Length(S)<br>S= "Yasemin"    | 7          |
| <b>Dönüştürme Fonksiyonları</b>   |  |                              |            |
| Value (S)                         | Dizi olarak tanımlanan değişkeni sayısal değere çevirir.       | Value("65.21")               | +65.21     |
| String (N)                        | Sayısal değeri dizi değerine çevirir.                          | String(+65.21)               | "65.21"    |
| <b>İstatistiksel Fonksiyonlar</b> |  |                              |            |
| Average (list)                    | Birkaç sayı için ortalama değeri döndürür.                     | Average(12, 24, 6)           | 14         |
| Sum (list)                        | Birkaç sayının toplam değerini döndürür.                       | Sum(3, 5, 8)                 | 16         |
| <b>Yardımcı Fonksiyonlar</b>      |  |                              |            |
| Date                              | Sistemin andaki tarih değerini döndürür.                       | Date                         | 04/23/2017 |
| Time                              | Sistemin şu andaki zaman değerini döndürür.                    | Time                         | 20.57.36   |

**Operatörler:** Programlama dillerinde tek başlarına herhangi bir anlamı olmayan ancak programın işleyişine katkıda bulunan karakter ya da karakter topluluklarına operatör denir. Operatörlerin etki ettikleri sabit ya da değişkenlere "operand" denir. Örnek: 32/3→32 ve 3 operand, "/" ise operatördür. 6<=8 → 6 ve 8 operand, "<=" ise operatördür.

Operatörler; matematiksel, mantıksal ve ilişkisel operatörler olarak sınıflandırılabilir. İlişkin örnekler Tablo 2'de yer almaktadır.

Tablo 2: Operatör türleri ve örnekler

| Operatör            | Bilgisayar Sembolü | Örnek         | Sonuç |
|---------------------|--------------------|---------------|-------|
| <b>Matematiksel</b> |                    |               |       |
| Toplama             | +                  | 6.7 + 2.1     | 8.8   |
| Çıkarma             | -                  | 5.6-3.4       | 2.2   |
| Çarpma              | *                  | 3.0*4.0       | 12.0  |
| Bölme               | /                  | 40.0/8.0      | 5     |
| Modül Alma          | MOD                | 9 MOD 3       | 3     |
| <b>İlişkisel**</b>  |                    |               |       |
| Eşit                | ==                 | 6 == 8        | False |
| Küçüktür            | <                  | 6 < 8         | True  |
| Büyüktür            | >                  | 6 > 8         | False |
| Küçük ya da eşittir | <=                 | 6 <= 8        | True  |
| Büyük ya da eşittir | >=                 | 6 >= 8        | False |
| Eşit değildir       | <>                 | 6 <> 8        | True  |
| <b>Mantıksal</b>    |                    |               |       |
| Değil               | NOT                | NOT True      | False |
| Ve                  | AND                | True AND True | True  |
| Veya                | OR                 | True OR False | True  |

\*\*İlişkisel operatörlerle yapılan işlemlerin sonucunda ortaya mantıksal değerler çıkar. Yanlış (False) çıkar.

\*\*Python dilinde eşit değildir için "!="i kullanılır.

Örnek:  
 >>>a=5%2  
 >>>a  
 1

\*\*Python dilinde mod almak için "%" işareti kullanılır.

## İşlem Önceliği

Matematiksel, mantıksal ve ilişkisel operatörlerin bir hiyerarşisi yani öncelikleri vardır. En içteki araçtan () en dışına doğru işlem yapılmalı, araç içerisinde ise işlem önceliklerine dikkat

Tablo 3: İşlem önceliği

| İşlem Sırası  | Veri Türü                    | Sonuç Değeri Türü |
|---|------------------------------|-------------------|
| () hiyerarşiyi sıralar, araç içerisindeki işlemler en içten en dışa doğru yapılmalıdır. |                              |                   |
| <b>Fonksiyonlar</b>   |                              |                   |
| <b>Matematiksel Operatörler</b>   |                              |                   |
| Kuvveti (Üs)  | Sayısal                      | Sayısal           |
| \, MOD  | Sayısal                      | Sayısal           |
| *,/   | Sayısal                      | Sayısal           |
| +, -  | Sayısal                      | Sayısal           |
| <b>İlişkisel Operatörler</b>  |                              |                   |
| =, <, >, <=, >=, <>   | Sayısal, dizi ya da karakter | Mantıksal         |
| <b>İlişkisel Operatörler</b>  |                              |                   |
| NOT   | Mantıksal                    | Mantıksal         |
| AND   | Mantıksal                    | Mantıksal         |
| OR  | Mantıksal                    | Mantıksal         |

edilmelidir.

Örnek değişkenlere ilişkin işlem sonuçları için aşağıdaki tabloyu inceleyiniz.

| Değişken             | İşlem          | Çıktı |
|----------------------|----------------|-------|
| x=10<br>y=15<br>z=20 | x+y-z          | 5     |
|                      | x-y*z          | -290  |
|                      | z mod y        | 5     |
|                      | (x-y)*5        | -25   |
|                      | x<y            | True  |
|                      | x<y AND x<z    | True  |
|                      | NOT (x>0)      | False |
|                      | x<>y OR z >=20 | True  |

## Söz Dizimi

Bilgisayarın işletim sistemi, dili ve uygulamalarına ilişkin kurallara "**söz dizimi**" denir. Bir hata oluşursa buna "**yazım hatası**"; hatayı bulup düzenleme işlemine ise "**hata ayıklama**" denir.

## Örnek Algoritma ve Akış Şemaları

Bir asansörün çağrı gelen kata nasıl gittiğini gösteren algoritma:

1. Başla
2. Çağrı yoksa 2. adıma git
3. Gidilecek katı oku
4. Eğer Gidilecek kat==Bulunulan kata, o zaman 7. adıma git
5. Eğer Gidilecek kat>bulunulan kattan, o zaman bir kat yukarı çık ve 5. adıma git
6. Eğer Bulunulan kat>Gidilecek kat bir kat aşağı in ve 6. adıma git
7. Kapıyı aç
8. Bitir

Önceden belirlenmiş iki sayının toplamını ekrana yazdıran algoritma

1. Başla
2. a=5

3. b=9
4. c=a+b
5. Ekranaya yaz c
6. Bitir

Klavyeden girilen iki sayıyı toplayıp ekrana yazdıran algoritma

1. Başla
2. Birinci sayıyı giriniz, sayi1
3. İkinci sayıyı giriniz, sayi2
4. toplam=sayi1+sayi2
5. Ekranaya yaz toplam
6. Bitir

Kenar uzunlukları girilen dikdörtgenin alanını hesaplayan algoritma

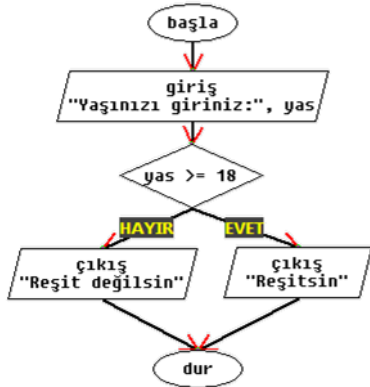
1. Başla
2. Kısa kenarı gir, a
3. Uzun kenarı gir, b
4. Alan=a\*b
5. Ekranaya yaz, Alan
6. Bitir

Girilen 3 sayının ortalamasını hesaplayıp ekrana yazdıran algoritma

1. Başla
2. Birinci sayıyı giriniz, sayi1
3. İkinci sayıyı giriniz, sayi2
4. Üçüncü sayıyı giriniz, sayi3
5. ortalama=(sayi1+sayi2+sayi3)/3
6. Ekranaya yaz, ortalama
7. Bitir

Yaşı girilen kişinin reşit olup olmadığı ekrana yazan algoritma ve akış şeması:

1. Başla
2. Yaşınızı giriniz, yas
3. Eğer yas>=18 o zaman ekrana yaz "Reşitsiniz" değilse ekrana yaz "Reşit değilsiniz"
4. Bitir



Girilen sıcaklık derecesine göre suyun halini ekrana yazdıran algoritma. (Su 0 ve altında katı, 0> ve 100< ise sıvı 100>= ise gazdır)

1. Başla
2. Sıcaklık giriniz, sıcaklik
3. Eğer sıcaklik<=0 ise ekrana yaz "Katı"

4. Eğer sıcaklik>0 ve sıcaklik<100 ise ekrana yaz "Sıvı"
5. Eğer sıcaklik>=100 ise ekrana yaz "Gaz"
6. Bitir.

İsmi girilen kişiyi "Merhaba" diye selamlayan algoritma:

1. Başla
2. İsminizi giriniz, isim
3. Ekranaya yaz "Merhaba "+isim
4. Bitir.

Girilen iki adet sayıdan büyük olanını ekrana yazan algoritma:

1. Başla
2. Birinci sayıyı giriniz, sayi1
3. İkinci sayıyı giriniz, sayi2
4. Eğer sayi1> sayi2 ise Ekranaya yaz **sayi1** değilse Ekranaya yaz **sayi2**
5. Bitir

Kenar uzunlukları girilen dikdörtgenin alanını hesaplayan algoritma (kenar uzunluğu olarak 0 veya daha küçük bir sayı girilemez):

1. Başla
2. Kısa kenarı gir, a
3. Eğer a <= 0 ise Adım 2'ye git
4. Uzun kenarı gir, b
5. Eğer b <= 0 ise Adım 4'e git
6. Alan=a\*b
7. Ekranaya yaz, Alan
8. Bitir

ALGORİTMA ÖRNEKLERİ:

1. Kullanıcının İsmi Alarak Merhaba (kullanıcı ismi) Yazdıran Python Örneği
2. Girilen 2 Sayıyı Toplayan Python Örneği
3. Girilen 2 Sayının Ortalamasını Bulan Python Örneği
4. Girilen Vize ve Final Notu Ortalaması Hesaplayan Python Örneği
5. Girilen 3 Yazılı Notunun Ortalamasını Bulan Python Örneği
6. Yazılı Ortalaması Girilen Öğrencinin Sınıf Geçme Durumunu (GEÇTİ – KALDI) Gösteren Python Örneği
7. Girilen Sayının Tek mi Çift mi Olduğunu Bulan Python Örneği.
8. Yaşı Girilen Kişinin Ehliyet Alıp Alamayacağını Gösteren Python Örneği
9. Kenarları Girilen Dikdörtgenin Alanı ve Çevresini Bulan Python Örneği
10. yarıçapı girilen dairenin alanını hesaplayan Python örneği:
11. Vize ve Final notu girilen öğrencinin geçip geçmediğini hesaplayan algoritma ve akış şeması (vizenin%40,finalin %60'ı hesaplanır. Final en az 60 olmak zorundadır)
12. Girilen sayı 0 dan büyük ise "pozitif" küçük ise "negatif" sifıra eşit ise "sıfır" mesajını verdiren algoritma ve akış şemasını tasarlayınız.

13. Kullanıcının girdiği 3 sayıdan büyük olanını yazdıran algoritma ve akış şemasını tasarlayınız.

## İlişkisel Operatörler

| İfade  | Anlamı  |
|--------|---|
| $x=y$  | Eğer x ve y birbirine eşitse (matematiksel olarak) doğrudur, değilse yanlıştır. |
| $x<y$  | Eğer x, y'den küçükse doğrudur; değilse yanlıştır.                              |
| $x<=y$ | Eğer x, y'den küçük ya da eşitse doğrudur; değilse yanlıştır.                   |
| $x>y$  | Eğer x, y'den büyükse doğrudur; değilse yanlıştır.                              |
| $x>=y$ | Eğer x, y'den büyük ya da eşitse doğrudur; değilse yanlıştır.                   |
| $x!=y$ | Eğer x, y'den farklı ise (büyük ya da küçük) doğrudur; değilse yanlıştır.       |

### Örnekler:

1.  **$a**b = a$  üssü  $b$  şeklinde dönüşür.**

**$2**3 = 8$  gibi...**

|                                |                            |          |                                |   |
|--------------------------------|----------------------------|----------|--------------------------------|---|
| a=3<br>b=4<br>a!=b and not b<a | x = 3<br>y = 3<br>(x**y)%5 | 3**2+5*2 | a = 3**2<br>b = 2**3<br>a != b | a = 3**2<br>b = 2**3<br>a == b or not a<b |
|--------------------------------|----------------------------|----------|--------------------------------|---|

pow(x,y) fonksiyonu,  $x^y$  şeklinde sonuç döndürür.

sqrt(x) fonksiyonu,  $\sqrt{x}$  şeklinde sonuç döndürür.

Abs(x) şeklinde kendi tanımlamış olduğumuz fonksiyon, mutlak değer olacak şekilde sonuç döndürür.

Buna göre **pow(sqrt(4),Abs(-4)-4)** işleminin sonucu nedir?

1. Aşağıdakilerden hangisi sayısal veri türünde tanımlanabilir?  
A) TC Kimlik No B) Banka Hesap No C) Not Ortalaması  
D) Posta Kodu E) Doğum Tarihi

2. Web sitesi uzantılarını, anlamlarını ve örneklerini içeren 5 örnek veriniz..