

*Problem Çözme Teknikleri

Bu sunu bilgisayarbilimleri.com tarafından
hazırlanmıştır

* Problemleri çözerken bazı temel teknikler vardır. İsterseniz gelin hep birlikte bu tekniklere bakalım

*Her Zaman Bir Planınız Olsun

- *Belirsiz bir durumu yaşamak yerine her zaman bir planınız olmalıdır. Bu, en önemli kuraldır.
- *Belki oluşturduğunuz çözüm planı ilk denemelerde sonuç vermeyecek ama her seferinde sizi çözüme biraz daha yaklaştıracak ipuçları elde etmenizi sağlayacaktır.

*Problemi Tekrar İfade Edin

*Problemi tekrar ifade etmek, göremediğimiz bir ayrıntıyı görmemizi ya da problemi daha kolay çözmek adına bir ipucu yakalamamızı sağlayabilir.

* Problemi Küçük Parçalara Ayırın

* Verilen problemi adımlara ya da bölümlere ayırmak, çözümü kolaylaştırır.



*Önce Bildiklerinizden Yola Çıkın

*Programlama yaparken öncelikle bildiklerimiz ile başlamalı ve sonra yeni çözümler arayışına girmeliyiz. Problemi küçük parçalara bölerek çözebildiğiniz parçadan başlayınız.

* Problemi Basitleştirin

- * Çözmekte zorlandığınız bir problemle karşılaşırsanız problemin kapsamını daraltmayı deneyebilirsiniz.
- * Temel amacınız problemi basitçe ifade etmeye çalışmak olmalıdır.

* Benzerlikleri Arayın

* Benzerlik kavramı, çözümlmesi istenen problemle önceden çözülen problem arasındaki olası örtüşme ya da yeni çözüme ilham verme olarak tanımlanabilir.

*Deneme Yapın

*Bazen bir problemi çözenin en kolay yolu denemek ve sonuçlarını gözlemlemektir.



* Asla Vazgeçmeyin

* Asla vazgeçmemek, kişisel bir özelliktir. Kararlılık, güven ve istek kaybolduğu zaman açık düşünemezsiniz, işlemler olması gerektiğinden uzun sürer ve gittikçe zorlaşır.

*Özetle Problem Çözme Teknikleri

- * 1) Her Zaman Bir Planınız Olsun
- * 2) Problemi Tekrar İfade Edin
- * 3) Problemi Küçük Parçalara Ayırın
- * 4) Önce Bildiklerinizden Yola Çıkın
- * 5) Problemi Basitleştirin
- * 6) Benzerlikleri Arayın
- * 7) Deneme Yapın
- * 8) Asla Vazgeçmeyin

*Problem Çözme Adımları

Problem çözme sürecinde
en iyi kararı verebilmek
için izlenmesi gereken 6
adım vardır

* 1. Problemi Tanımlama:

* Problemi çözmeye başlamadan önce problemin açık, anlaşılır ve çok doğru bir şekilde tanımlanmış olması gerekir.

*2. Problemi Anlama:

*Çözümüne doğru yol almadan önce problemi çok iyi anladığınızdan emin olmanız gerekir. Problemin neler içerdiğini ve kapsamını doğru anlamalısınız.

* 3. Problemin Çözümü İçin Farklı Yol ve Yöntemler Belirleme:

* Problemin çözümü için olabildiğince farklı yol ve yöntem belirlemeli ve bu listenin, tüm olasılıkları içerdiğinden emin olmalısınız.

4. Farklı Çözüm Yolları Listesi İçerisinden En İyi Çözümü Seçme:

- * Bu adımda her bir çözümün olumlu ve olumsuz yönlerini ortaya koyup içlerinden en iyisini seçmelisiniz.

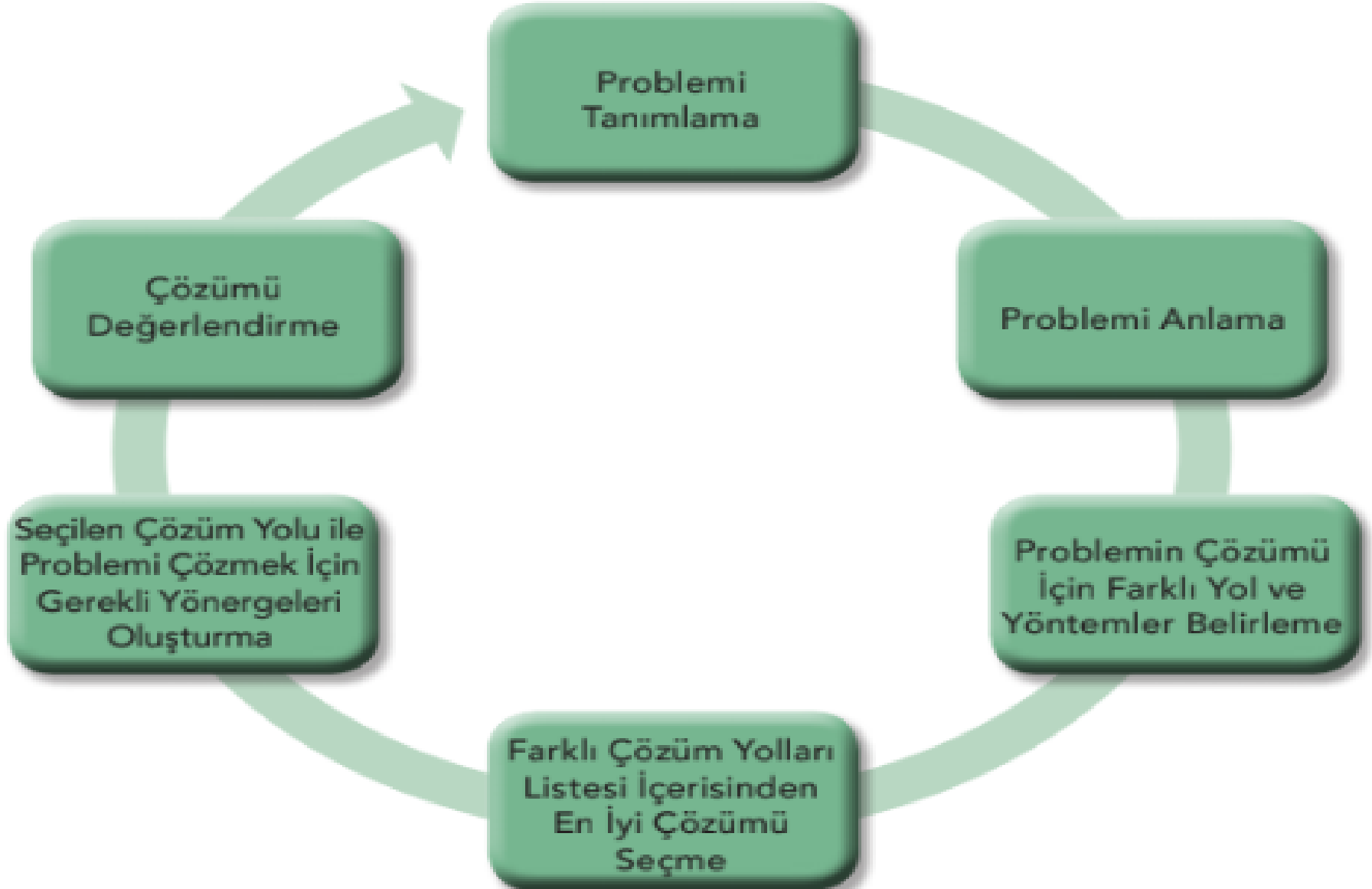
5. Seçilen Çözüm Yolu ile Problemi Çözmek İçin Gerekli Yönergeleri Oluşturma:

- * Bu adımda numaralandırılmış ve adım adım yönergeler oluşturmanız gerekir.

6. Çözümü Değerlendirme:

*Çözümü test etmek ya da değerlendirmek, sonucun doğruluğunu kontrol etmek anlamına gelir.

ÖZETLE 😊



Bilgisayarlar ile Problem Çözme



Bilgisayarlar ile Problem Çözme

Günlük hayatta karşılaştığımız problemler çok çeşitli olmasına rağmen bilgisayar ile çözebildiğimiz

yalnızca **3 tür** vardır:

Bilgisayarlar ile Problem Çözme

1. Hesaplamalı-matematiksel problemler,
2. Mantıksal problemler,
3. Tekrarlayan-yinelenme sürecini içeren problemler.

Veri Türleri

Çözümler üretebilmek için bilgisayarlar “veri”ye gereksinim duyar.

Bu verilere de girdi ismi verilir

Veri Türleri

Bilgisayarlara ham veri olarak giren verilen çeşitli işlemlerden geçerek çıktıya da bilgiyi oluşturur.

Örneğin

```
*a=int(input("bir sayi "))
```

```
*print (a)
```

Yukarıda yer alan python kodlarında 1.satır girdi
2.satır iste çıktı satırlarıdır.

1) Sayısal Veri

* Sayısal veriler tüm sayı tiplerini içerir. Sayısal veri, hesaplama işlemlerinde kullanılabilen tek veri türüdür.

1) Sayısal Veri

Veri Türü	Veri Seti	Örnek
Sayısal: Tam sayı	Tüm sayılar	66578 -2356
Sayısal: Reel sayı	Tüm reel sayılar ve ondalık sayılar	-56.23 8695.235 0.005

2) Karakter Veri

* Karakter veri seti; tüm tek haneli sayılar (“0”.. “9”), harfler (“a”..“z”, “A”..“Z”) ve özel karakterleri (“#”, “&”, “*”, ..) kapsar.

2) Karakter Veri

Veri Türü	Veri Seti	Örnek
Karakter	Tüm rakamlar, harfler ve özel semboller	"A", "Y", "k", "i", "6", "0", "+", "%"
Dizi	Birden fazla karakterden oluşan kombinasyon	"Bilgisayar", "532-5556633"

3) Mantıksal Veri

*Mantıksal veri, veri setinde yalnızca iki kelime barındırır: doğru ve yanlış. Bu veri evet ya da hayır şeklindeki karar verme süreçlerinde kullanılır.

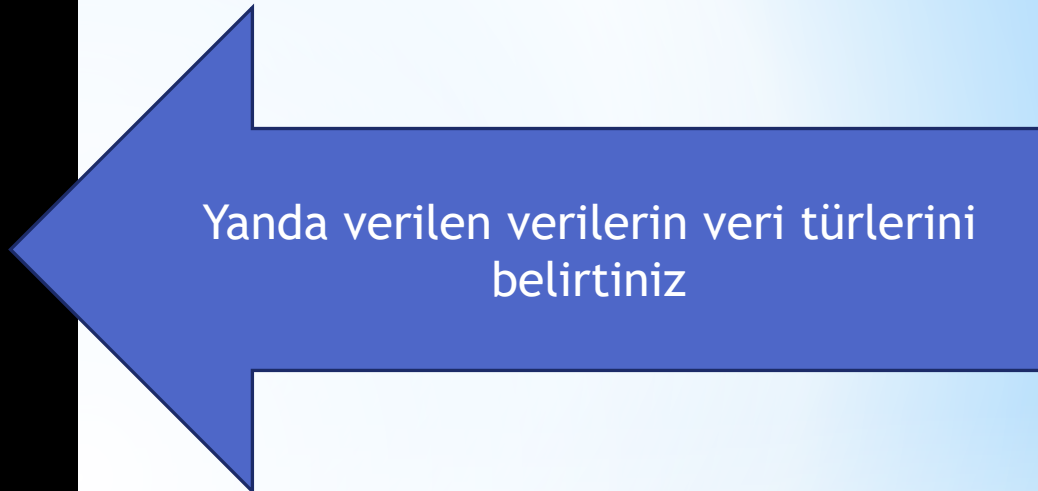
3) Mantıksal Veri

Veri Türü	Veri Seti	Örnek
Mantıksal	Doğru Yanlış True False	Doğru Yanlış True False

Veri Türleri İçin Kurallar

- 1) Tanımladığınız veri genellikle sayısal, karakter, dizi ya da mantıksal olmalıdır.
- 2) Veri türleri karışık kullanılamaz. Örneğin sayısal olarak tanımlanmış bir veri, dizi olarak algılanamaz
- 3) Matematiksel işlemlerde kullanılacak tüm veriler sayısal olarak, diğerleri karakter ya da dizi olarak tanımlanmalıdır.

Veri	Veri Türü
Ürün satış bedeli: 49.99, 101.50	bilgisayarbilimleri.com
T.C. Kimlik No.: 10654876542	
Ağırlık: 67	
Şirket İsmi: ABC Firması	
Kredi Onayı: Var, Yok	
Posta Kodu: 06110, 34217	
Tarih: 21042017	
IBAN: TR0600006543000012	



Yanda verilen verilerin veri türlerini belirtiniz

ONLINE YAPMAK İÇİN TIKLA 😊

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.

VERİ	VERİ TİPİ
AYŞE	
A	
Okul Numarası	
En İyi Arkadaşın Kim	
Elma Sever Misin	
Cinsiyet	
Yaş	

VERİ	VERİ TİPİ
AYŞE	Karakter Dizisi Veri Tipi
A	Karakter Veri Tipi
Okul Numarası	Sayısal Veri Tipi
En İyi Arkadaşın Kim	Karakter Dizisi Veri Tipi
Elma Sever Misin	Mantıksal Veri Tipi
Cinsiyet	Karakter Veri Tipi
Yaş	Sayısal Veri Tipi

Bilgisayar Veriyi Nasıl Saklar?

Bilgisayar veriyi hafızada saklar.

Her bir deęişken için hafızada belirli bir alan ayrılır ve bu alan her seferinde tek bir deęer saklayabilir.

Kullanıcı, var olan deęer yerine yeni bir deęer atadığında eski deęer silinir.

Sabit ve Değişkenler

Bilgisayarlar problemleri çözmek için süreç boyunca sabit ve değişken olarak adlandırılan veriler kullanır.

“Sabit” olarak tanımlanan veriler problemin çözüm süreci boyunca asla değişmeyen değerlerdir.

Sabit ve Değişkenler

Su 100 °C'de kaynar.

Bugün hava sıcaklığı kaç derece?

>>Yukarıda yer alan cümlelerde
sabitler ve değişkenler nelerdir? <<

Sabit ve Değişkenler

100 °C gibi sayısal değerler sabittir.

Sıcaklık gibi farklı değerler alabilen ifadeler ise değişken olarak adlandırılır.

KISACA SICAKLIK DEĞİŞEBİLİYORKEN
SUYUN 100 DERECEDE KAYNAMASI SABİT
OLARAK HEP AYNIDIR

Sabit ve Değişkenler

* `a=int(input("bir sayı gir be kanka"))`

* `print(a)`

* Yukarıda yer alan `a` değişkendir ve sürekli olarak değişir örnek için <https://repl.it/languages/python3> adresine gidelim ve kodları çalıştıralım her seferinde `a` nın değerinin değiştiğini göreceksiniz.

Sabit ve Değişkenler

* `a=5+3`

* `print(a)`

* Yukarıda yer alan `a` sabittir 1000 keredede çalışsa yine değeri aynı değerdir.

<https://repl.it/languages/python3> adresine gidelim ve kodları çalıştıralım her seferinde `a` nın değerinin aynı kaldığını göreceksiniz.

Değişkenlere isim verirken dikkat edilmesi gereken kurallar

- * 1. Değişkene içerdiği değer ile tutarlı isimler veriniz.
- * 2. Değişkenlere isim verirken boşluk kullanmayınız.
- * 3. Değişkenlere isim verirken bir karakter ile başlayınız.
- * 4. Matematiksel semboller kullanmamaya dikkat ediniz.

Değişkenlere isim verirken dikkat edilmesi gereken kurallar

5) Bazı platformlar desteklemediği için Türkçe karakter kullanımı tavsiye edilmez.

6) Programlama dillerinde kullanılan komut isimleri değişken olarak kullanılamaz. Çok bilinenleri;

*if, for, while, else, do, int, vb.

Değişkenlere isim verirken dikkat edilmesi gereken kurallar

7) Değişken isimlendirmelerinde boşluk karakteri yerine alt çizgi (_) karakteri kullanılabilir ancak değişken isimlendirmede genellikle küçük harfle başlanır ve ikinci bir kelime yazılacaksa ilk kelimenin hemen ardından büyük harfle devam edilir. Buna “Camel Karakter” kullanımı denir.

Örnek: tcKimlikNo

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

Değişkenlere isim verirken dikkat edilmesi gereken kurallar

1. Aşağıdaki problemlerin çözümü için hangi değişkenlere ihtiyaç duyulur?

a) Bir işçinin maaşını hesaplayan programı yazmak,

b) Bir sinema salonu için bilet satış programı yazmak,

c) Bir yolculuk için rezervasyon programı yazmak.

2. Siz de doğru ve yanlış değişken isimlerine örnekler veriniz.

Ufak ufak kod yazalım 😊

a) Bir işçinin maaşını hesaplayan program

```
gunlukyevmiye=int(input("Yevmiye gir  
kanki"))
```

```
gun=int(input("kaç gün çalıştın"))
```

```
maas=gunlukyevmiye*gun
```

```
print("maaşınız:",maas)
```

Ufak ufak kod yazalım 😊

b) Bir sinema salonu için bilet satış programı yazalım

```
filmsec=input("gitmek istediğiniz filmi yazınız please :)")
```

```
if filmsec=="recep ivedik":
```

```
print("50 TL ödeyiniz")
```

```
if filmsec=="end game":
```

```
print("20 TL ödeyiniz")
```

```
if filmsec=="bilgisayarbilimleri.com":
```

```
print("30 TL ödeyiniz")
```


FONKSİYONLAR

Fonksiyonlar, belirli işlemleri yürüten ve sonuçları döndüren bir işlem kümesidir. Genellikle bilgisayar dilinde oluşturulur.

Fonksiyonlar, kısaca işlemlerimizi daha hızlı yapmamızı sağlarlar.

FONKSİYONLAR

Fonksiyon Adı	Açıklama	Örnek Kullanım ve Çıktı
<code>sqrt()</code>	İstenilen değerin karekök'ünün bulunmasını sağlar.	<code>math.sqrt(16)</code> → 4
<code>exp()</code>	e (Euler sabiti) sayısının istenilen kuvvetinin alınmasını sağlar.	<code>math.exp(2)</code> → 7.389056
<code>log()</code>	<code>log(x,y)</code> fonksiyonumuz iki parametre alır. İlk parametremiz olan x logaritması, alınacak sayıyı; ikinci parametre olan y taban sayısını temsil etmektedir.	<code>math.log(2,2)</code> → 1.0
<code>log10()</code>	<code>log(x,y)</code> fonksiyonundan tek farkı taban olarak 10 sayısının sabit olmasıdır.	<code>math.log10(10)</code> → 1.0
<code>cos()</code>	<code>cos(x)</code> , x derecesinin kosinüs değerini verir.	<code>math.cos(45)</code> → 0.5253
<code>pow()</code>	<code>pow(x,y)</code> fonksiyonu x sayısının y. kuvvetinin alınmasını sağlar.	<code>math.pow(2,2)</code> → 4
<code>degress()</code>	<code>degress(x)</code> fonksiyonu x açısını radyandan dereceye çevirmeye yarar.	<code>math.degress(45)</code> → 2578.310078088
<code>radians()</code>	<code>radians(x)</code> fonksiyonu x açısını dereceden radyana çevirmeye yarar.	<code>math.radians(45)</code> → 0.7853981633
<code>fabs()</code>	<code>fabs(x)</code> fonksiyonu x değerinin mutlak değerinin alınması işlemini gerçekleştirir.	<code>math.fabs(-5)</code> → 5.0

ÖRNEKLER

```
*print (pow(5,2))
```

```
*print(round(28.51))
```

```
*siniflistesi =  
['ayşe', 'fatma', 'ayşe', 'ayşe', 'burak']
```

```
print(siniflistesi.count('ayşe'))
```

```
Print(len('bilgisayarbilimleri.com'))
```

KENDİMİZ FONKSİYON YAZALIM

```
def fonksiyon(isim,not1,not2):  
    ortalama = (not1+not2)/2  
    print (isim,"adındaki öğrencinin  
not ortalaması:",ortalama)  
fonksiyon(not1=34,isim="celil",not  
2=78)
```

OPERATÖRLER

Bilgisayara, verileri nasıl işleyeceğini belirtmek gerekir.

Bu işlem için operatörler kullanılır. “Operatörler” verileri, ifade ve eşitlikler ile birleştirir.

OPERATÖRLER

Operatör	Bilgisayar Sembolü	Örnek	
Matematiksel		İşlem	Sonuç
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
İlişkisel**		İşlem	Sonuç
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
Mantıksal		İşlem	Sonuç
Değil	NOT	NOT True	False
Ve	AND	True AND True	True
Veya	OR	True OR False	True

OPERATÖRLER

1) $a=5+7$

`print (a)`

2) $a=(3+4+21)/7$

$b=(9*4)/(2+1)-6$

`print(a*b)-(a+b)`

OPERATÖRLER

Değişken	İşlem	Çıktı
x=10 y=15 z=20	$x+y-z$	5
	$x-y*z$	-290
	$z \text{ mod } y$	5
	$(x-y)*5$	-25
	$x < y$	True
	$x < y \text{ AND } x < z$	True
	$\text{NOT } (x > 0)$	False
	$x <> y \text{ OR } z \geq 20$	True

BASİT BİR SORU 😊

Puanları 68, 80, 40
olan öğrencinin puan
ortalamasını
hesaplayınız.