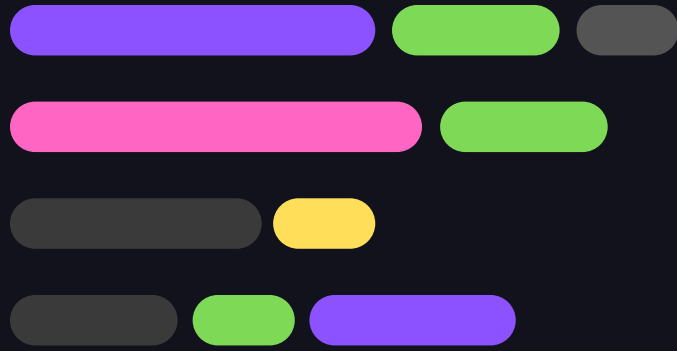




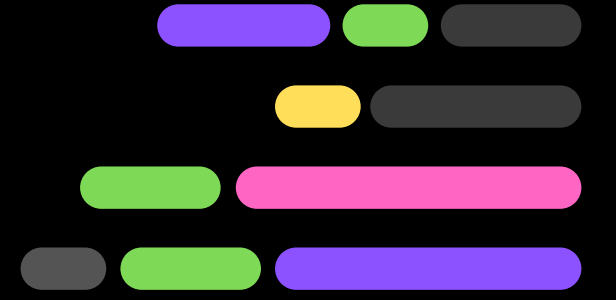
{ TEMEL PROGRAMLAMA II }

Samsun Üniversitesi
Teknik Bilimler Meslek Yüksekokulu
Arka-Yüz Yazılım Geliştirme Pr.





Bu Haftanın Ders Kazanımları



01

Derse Giriş

04

Temel Tekrar : Döngüler

02

Temel Tekrar : Değişkenler

05

Örnek Kodlar

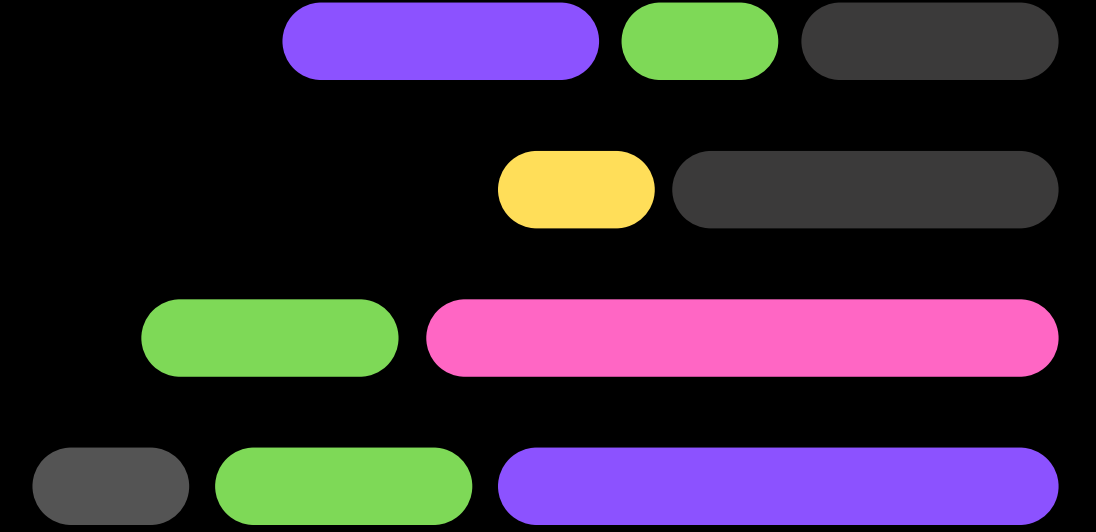
03

Temel Tekrar : Karar yapıları



01

Derse Giriş



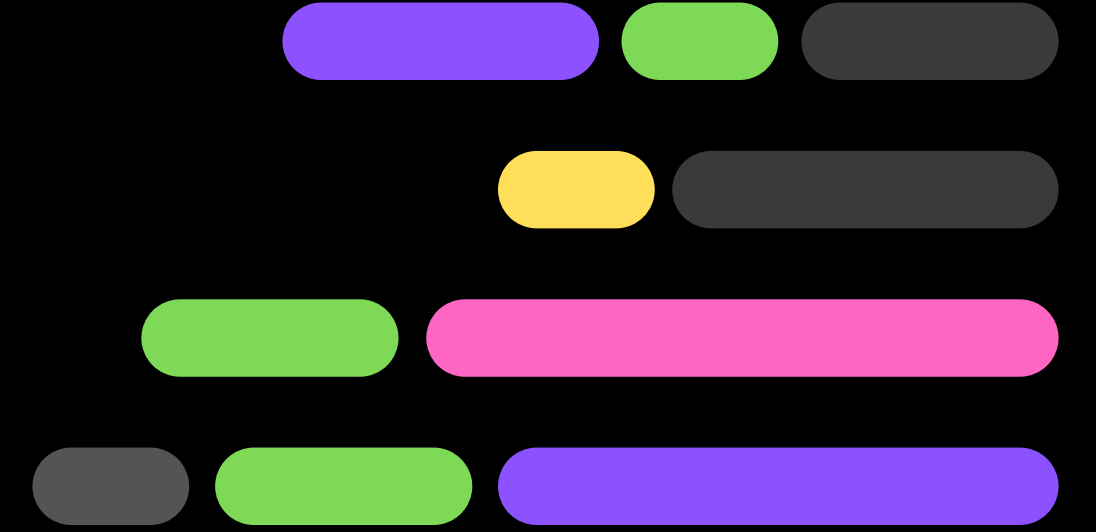
Dersin Amacı :

- Python dilinde temel programlama becerilerini geliştirmek
- Fonksiyonları etkin ve doğru şekilde kullanabilmek
- Metin (string) işlemlerini detaylı şekilde öğrenmek
- Hata yapmanın doğal olduğunu, hatalarla başa çıkmayı öğrenmek
- Kod okunabilirliği ve düzenine dikkat etmeyi alışkanlık haline getirmek



01

Derse Giriş

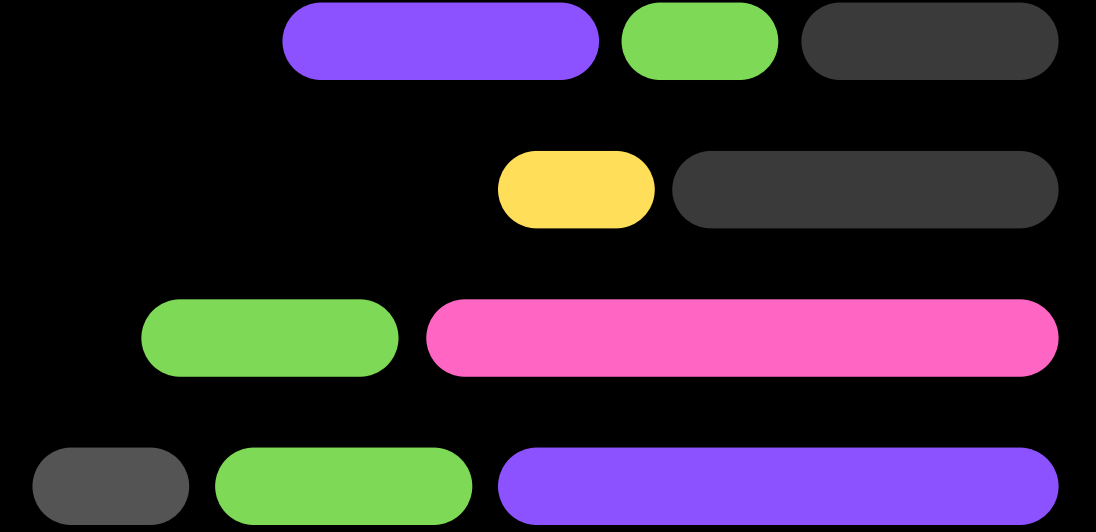


*Kod Yazmayı Biliyorsunuz, Bu Derste 'Düzenli' Kod Yazacağız.
Düzenli kodun temel avantajları nelerdir ?*



01

Derse Giriş



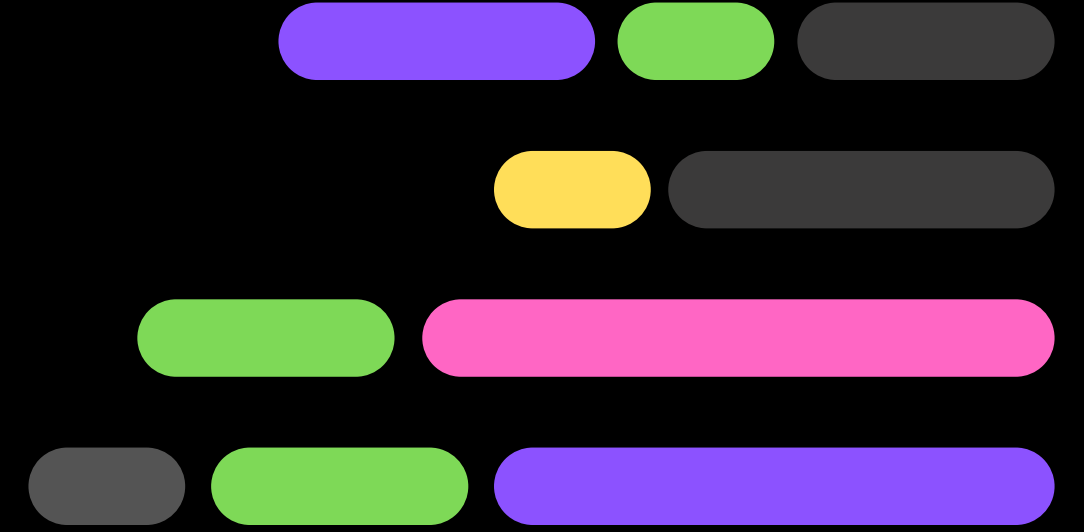
*Kod Yazmayı Biliyorsunuz, Bu Derste 'Düzenli' Kod Yazacağız.
Düzenli kodun temel avantajları nelerdir ?*

- Okunabilirlik
- Tekrar kullanılabilirlik



01

Derse Giriş

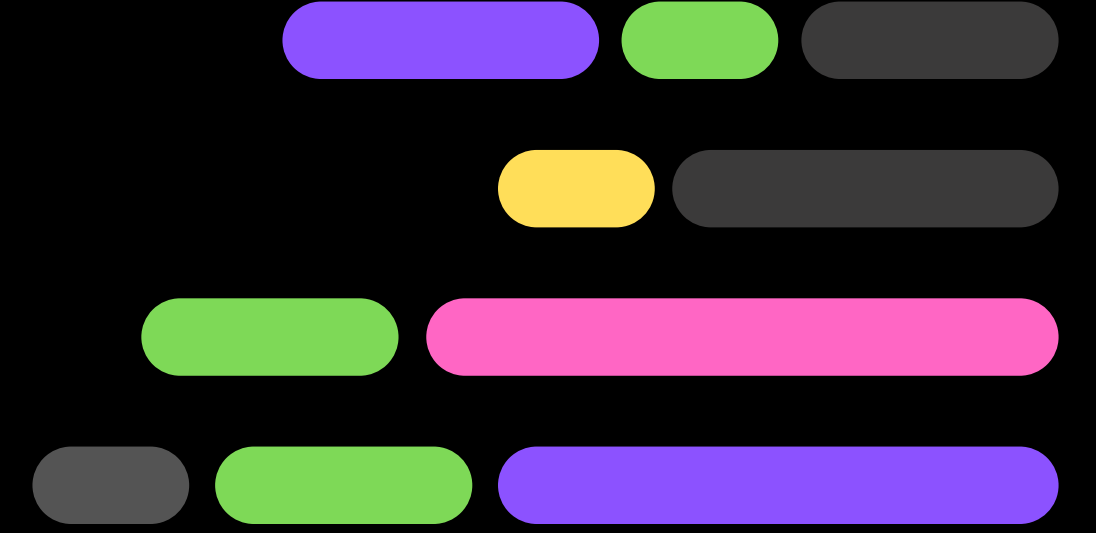


Fonksiyon Merkezli Düşünmeyi Öğreneceğiz



01

Derse Giriş



Fonksiyon Merkezli Düşünmeyi Öğreneceğiz

- Aynı kodu tekrar tekrar yazmak yerine:

Fonksiyon yazacağız

Parametre göndereceğiz

Değer döndüreceğiz (return)

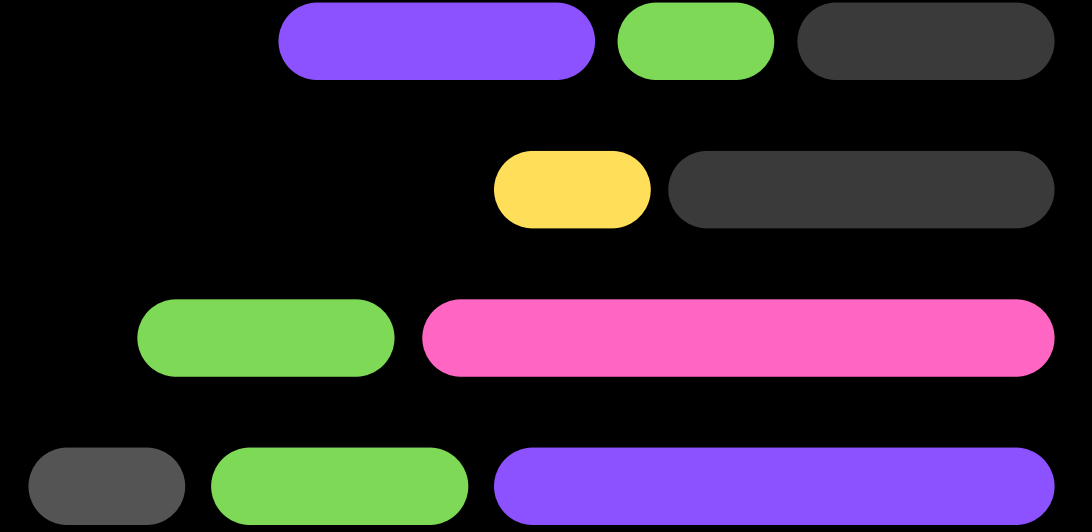
- Küçük problemleri küçük parçalara ayıracağız

“Bir problemi çözmek = fonksiyonlara bölmek”



01

Derse Giriş

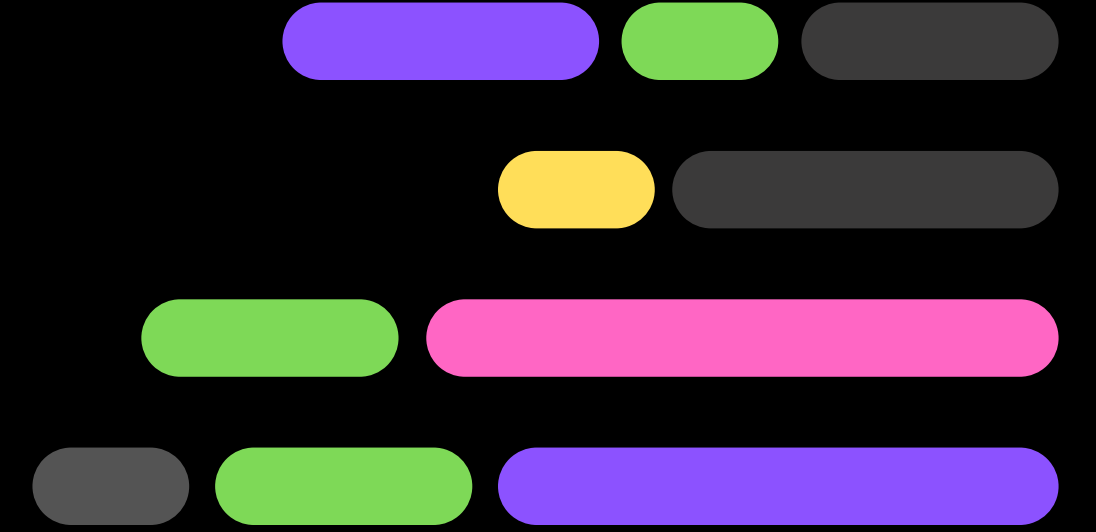


Gerçek Hayat Verileriyle Çalışacağız



01

Derse Giriş



Gerçek Hayat Verileriyle Çalışacağız

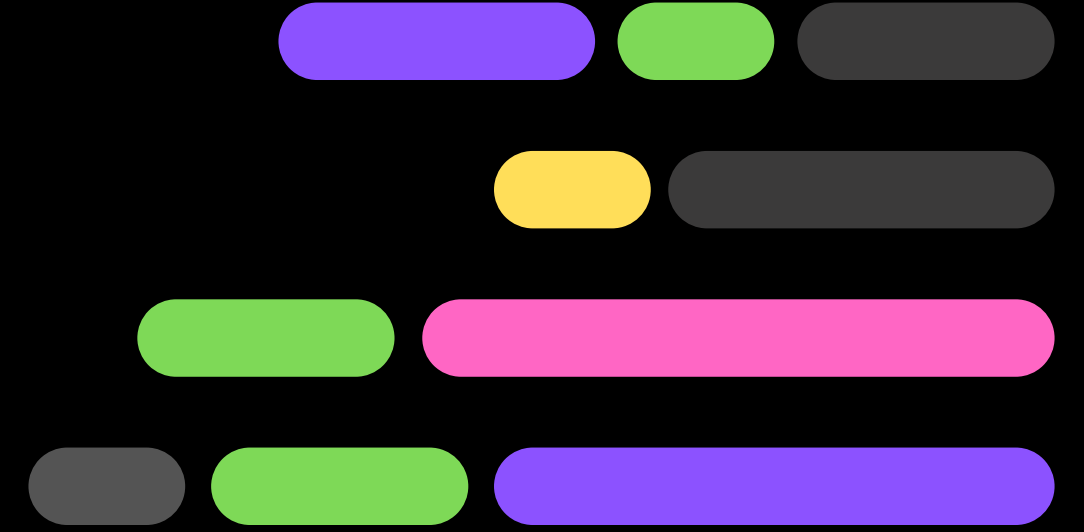
- Metinler (String)
- Tarihler (Datetime)
- Kullanıcıdan gelen hatalı veriler
- Dosyalar (okuma-yazma)

“Artık sadece sayı değil, gerçek hayattaki verilerle çalışacağız.”



01

Derse Giriş

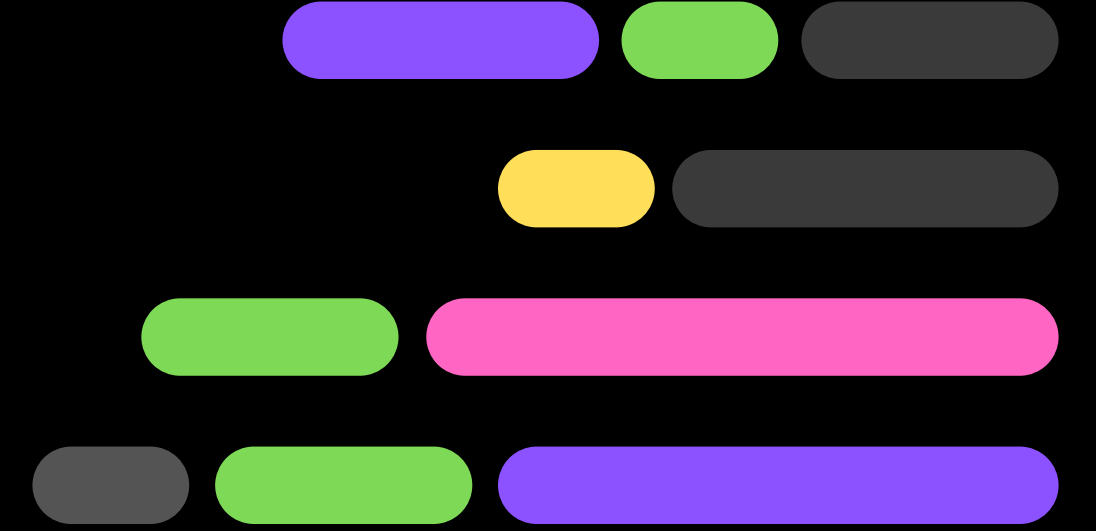


Hatalarla Kavga Etmeyi Öğreneceğiz



01

Derse Giriş



Hatalarla Kavga Etmeyi Öğreneceğiz

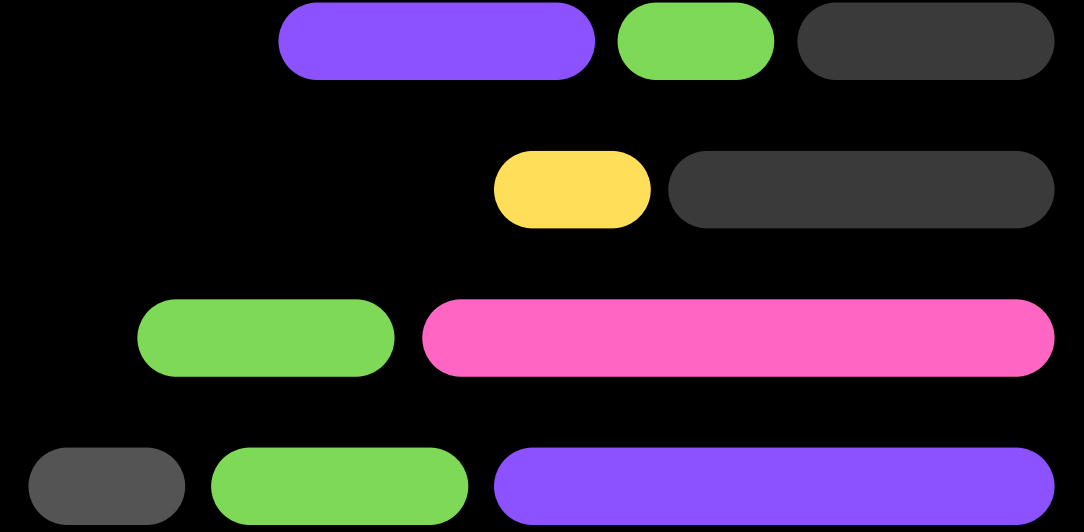
- Program neden çöker?
- Hata türleri nelerdir?
- Hata olunca programı çökertmeden devam ettirebilir miyiz?
- try – except nasıl kullanılır sorularına cevap bulacağız.

“Hata almak kötü değil, yakalayamamak kötü.”



01

Derse Giriş

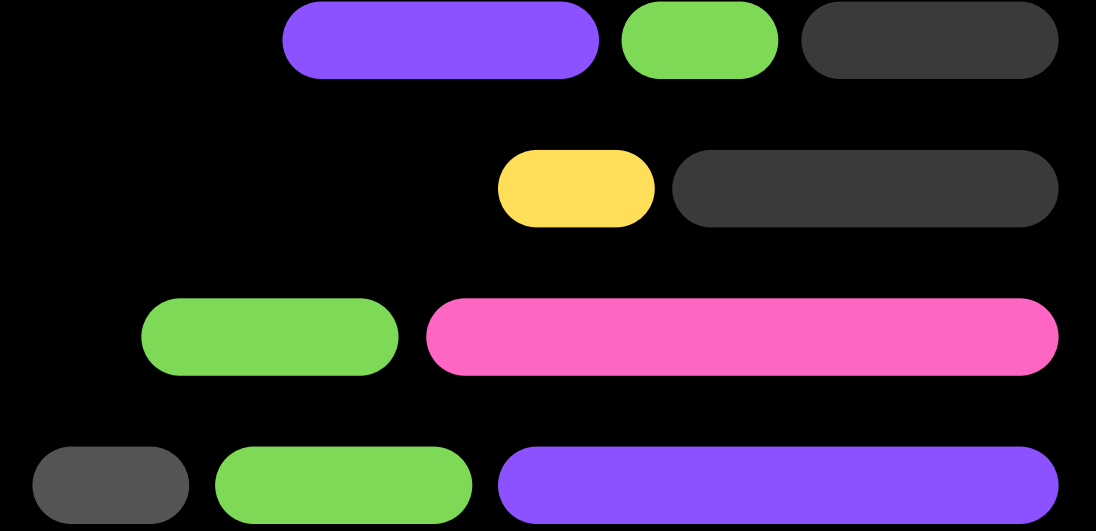


Küçük Ama Gerçek Programlar Yazacağız



01

Derse Giriş



Küçük Ama Gerçek Programlar Yazacağız

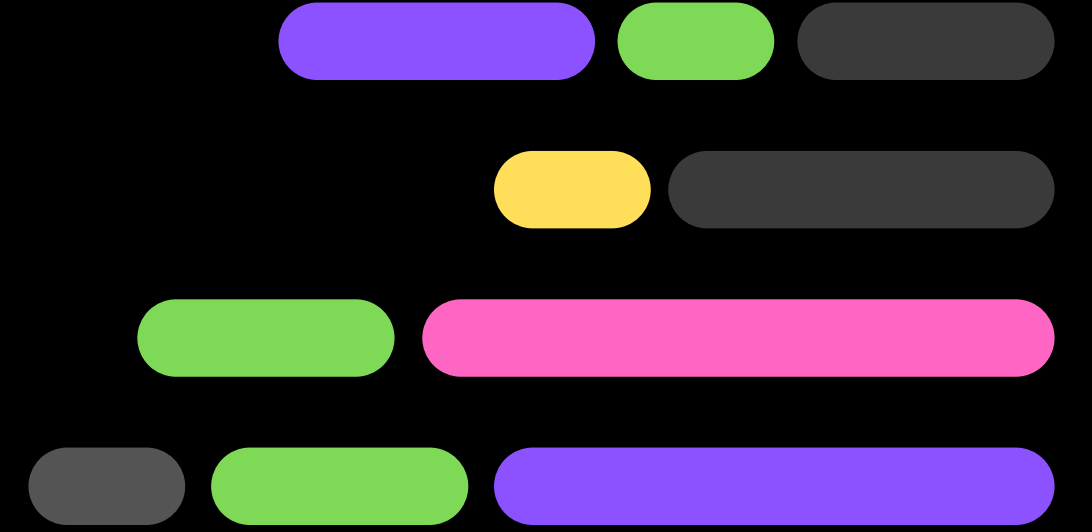
- Metin işleyen programlar
- Dosyaya veri kaydeden uygulamalar
- Hataları yöneten mini sistemler

“Bu dersin sonunda yazdığınız kod küçük ama gerçek olacak.”



01

Derse Giriş

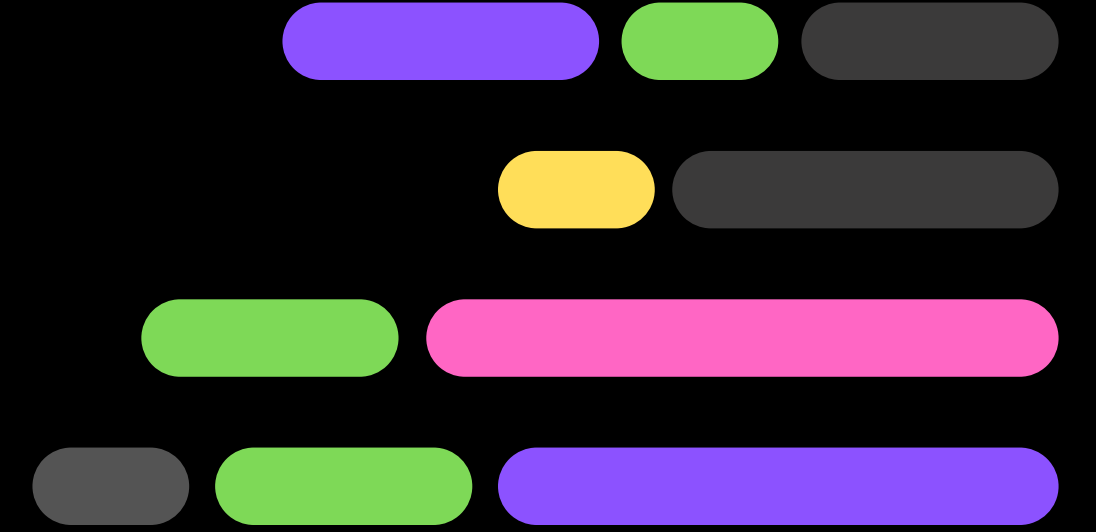


Büyük Konulara Hazırlık Yapacağız



01

Derse Giriş



Büyük Konulara Hazırlık Yapacağız

- Nesneye Yönelik Programlama'ya altyapı
- Kod standartları ve düzen
- Mini SOLID (S ve O) farkındalığı

“Bu ders, yazılım hayatınızın temelini atıyor.”



01

Derse Giriş

Ders değerlendirme sistemi :

%20 - Proje

%30 - Vize

%50 - Final

YOKLAMALARA DİKKAT EDİLECEK !

Temel Programlama 2 - Değerlendirme Sistemi



Proje

Final

20%
Proje

50%
Final

30%
Vize

Vize

Proje
(20%)

Vize
(30%)

Final
(50%)



01

Derse Giriş



Proje Hakkında

- 3'er kişilik gruplar
- 10 -15 dk sunum
- Kod ve çıktısı detaylı anlatılacak
- AI'dan faydalanılabilir fakat AI'a YAPTIRILMAYACAK.



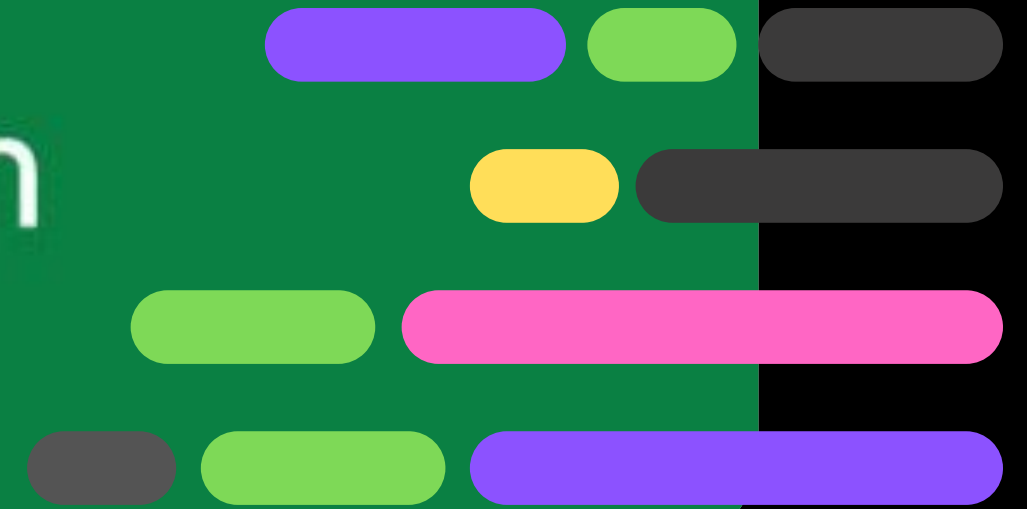
01

Derse Giriş



Google Classroom

313ihcog



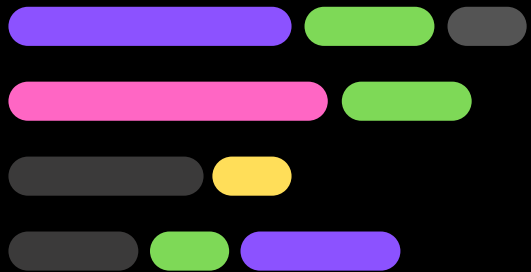
Temel Tekrar : Değişkenler



02

Temel Tekrar : Değişkenler

Değişken Nedir, neden kullanırız ?



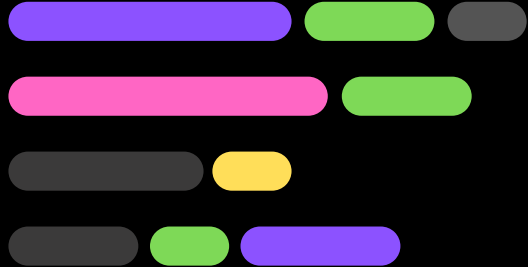


02

Temel Tekrar : Değişkenler

Değişken Nedir, neden kullanırız ?

Değişken, bir veriyi bilgisayarın hafızasında bir isimle saklamaktır.





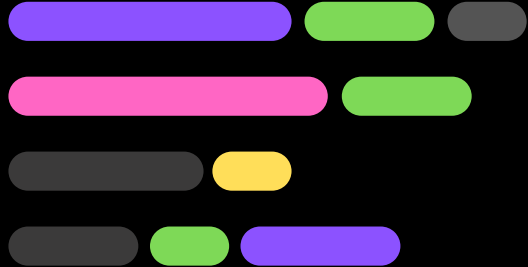
02

Temel Tekrar : Değişkenler

Değişken Nedir, neden kullanırız ?

Değişken, bir veriyi bilgisayarın hafızasında bir isimle saklamaktır.

- Tekrar Kullanım
- Okunabilirlik
- Veri Yönetimi

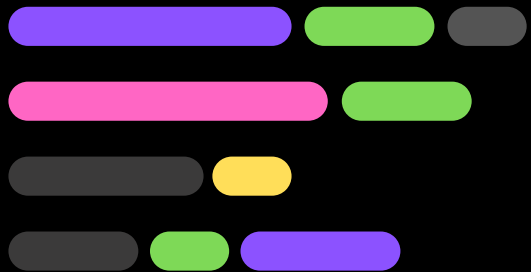




02

Temel Tekrar : Değişkenler

Bir programda hiç değişken kullanmazsak ne olur?



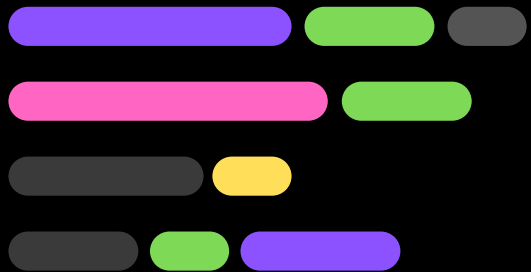


02

Temel Tekrar : Değişkenler

Bir programda hiç değişken kullanmazsak ne olur?

- Kod karmaşık olur
- Değişiklik zor olur
- Okunmaz





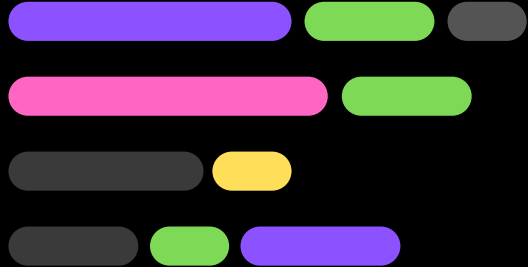
02

Temel Tekrar : Değişkenler

Bir programda hiç değişken kullanmazsak ne olur?

- Kod karmaşık olur
- Değişiklik zor olur
- Okunmaz

"Değişkenler, programın hafızasıdır."

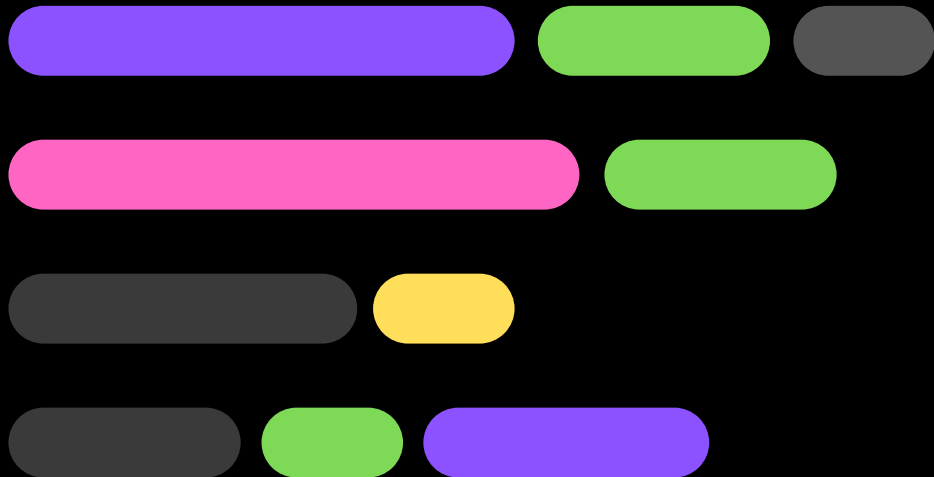
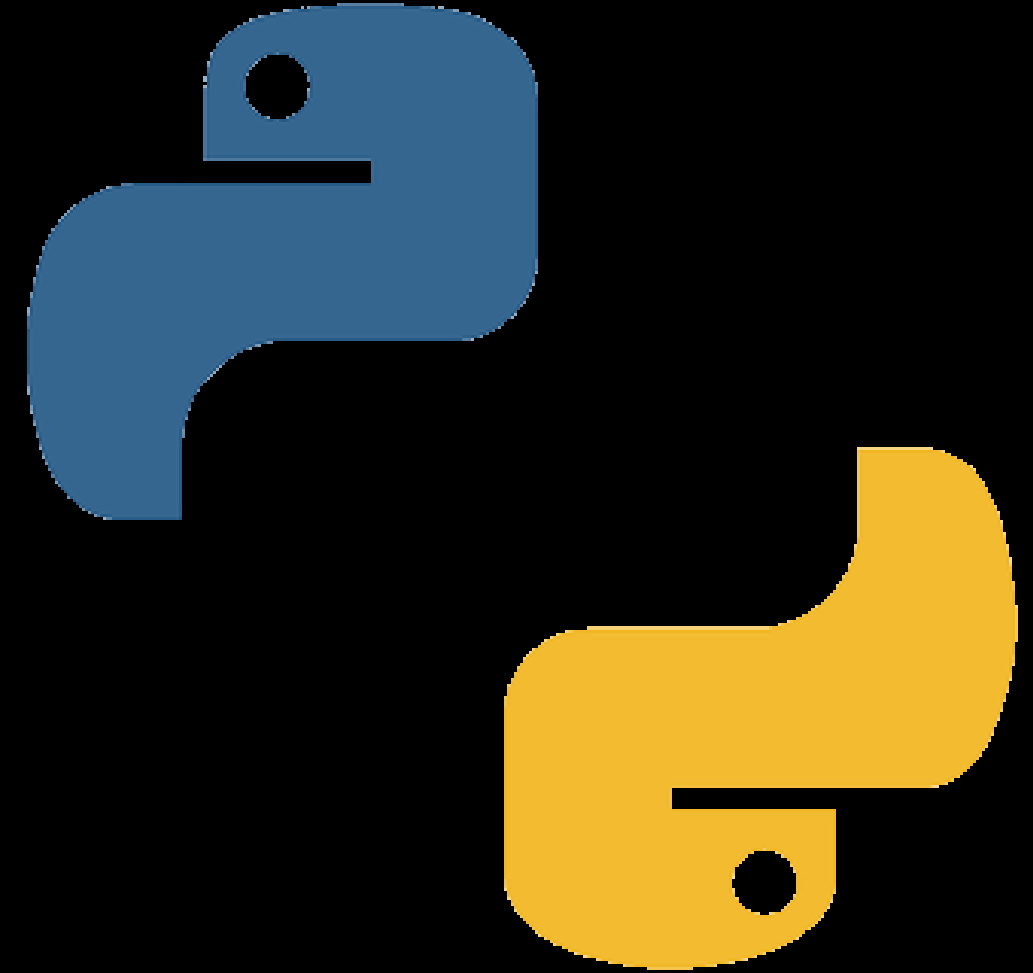




02

Temel Tekrar : Değişkenler

İsmlendirme Kuralları
Nelerdir?



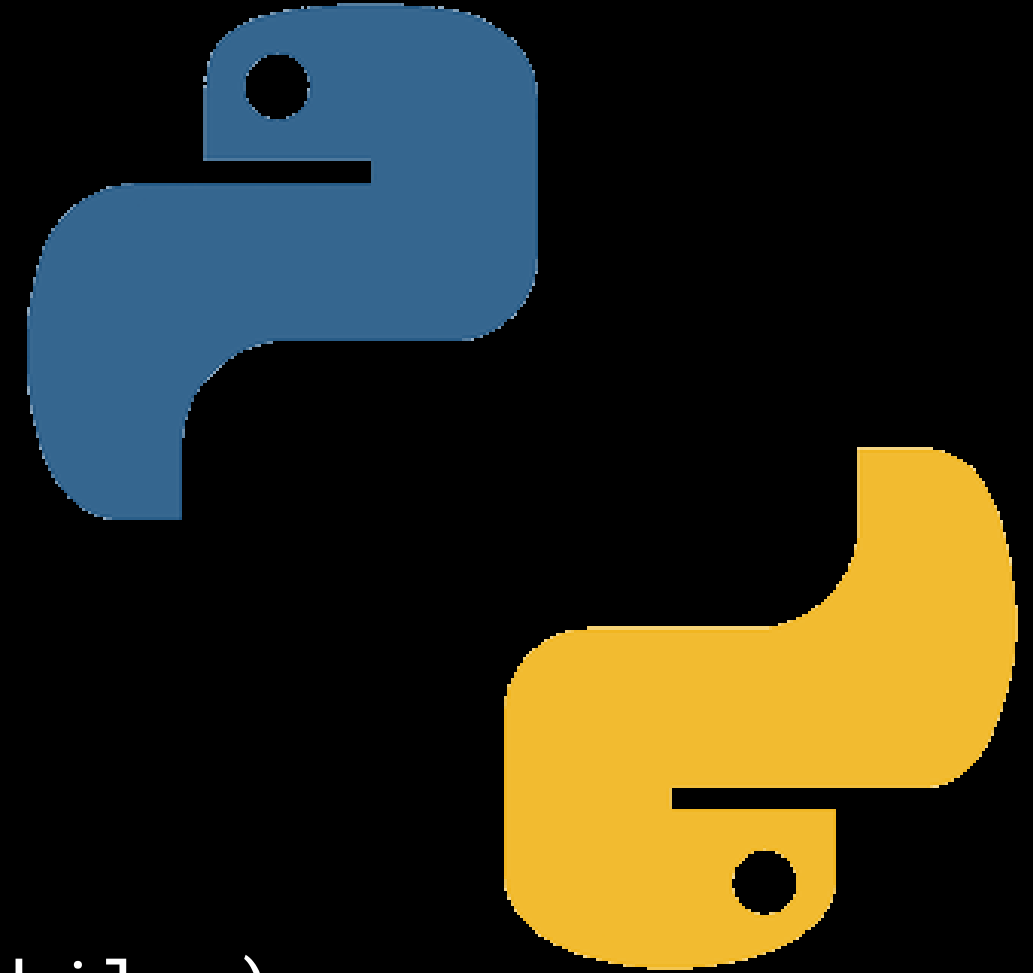


02

Temel Tekrar : Değişkenler

İsmlendirme Kuralları

- snake_case
 - Anlamlı isimlendirmeler
 - Değişken adları harf veya _ ile başlar
 - Boşluk olmaz
 - Türkçe karakter kullanma (ş, ğ, ü, ç, ö, ı, İ)
 - Sayı ile başlayamaz
- Python anahtar kelimeleri kullanılmaz (if, for, while...)



"İyi isimlendirme, yazılmayan yorum satırıdır."



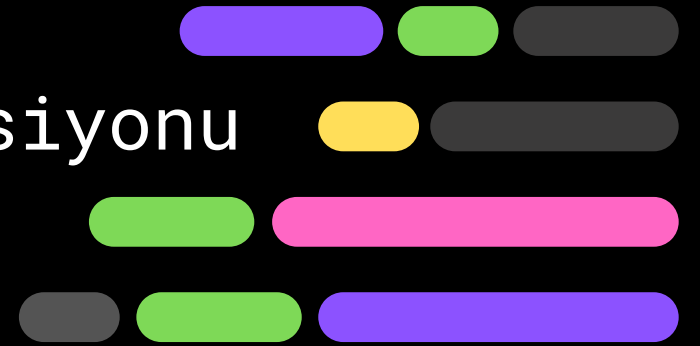
02

Temel Tekrar : Değişkenler

Temel Veri Tipleri

- int
- float
- str
- bool

Verinin tipini öğrenmek için hangi fonksiyonu kullanırız?



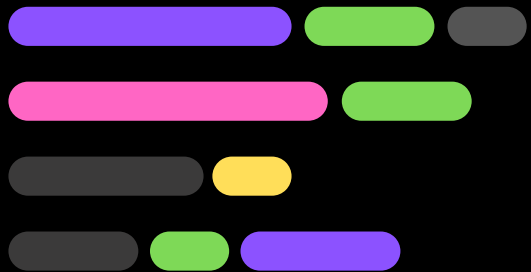


02

Temel Tekrar : Değişkenler

Tip Dönüşümü

```
yas = int("20")  
fiyat = float("19.5")  
yas_str = str(25)
```





02

Temel Tekrar : Değişkenler

Tip Dönüşümü

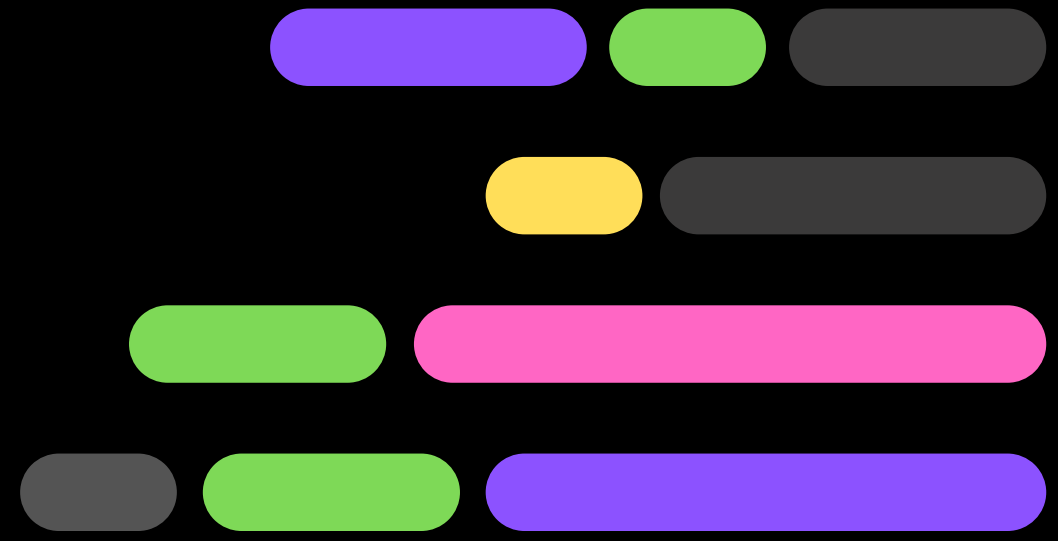
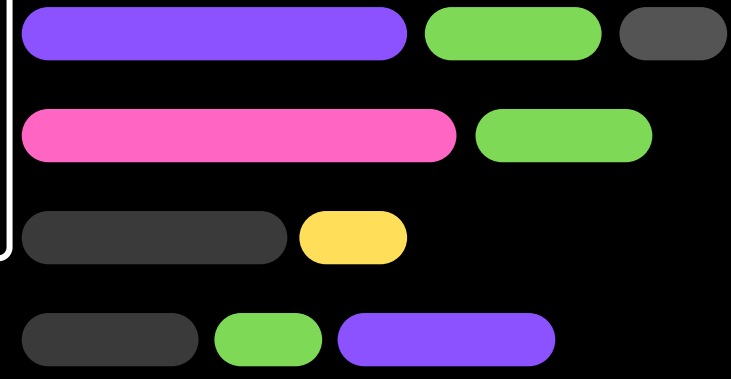
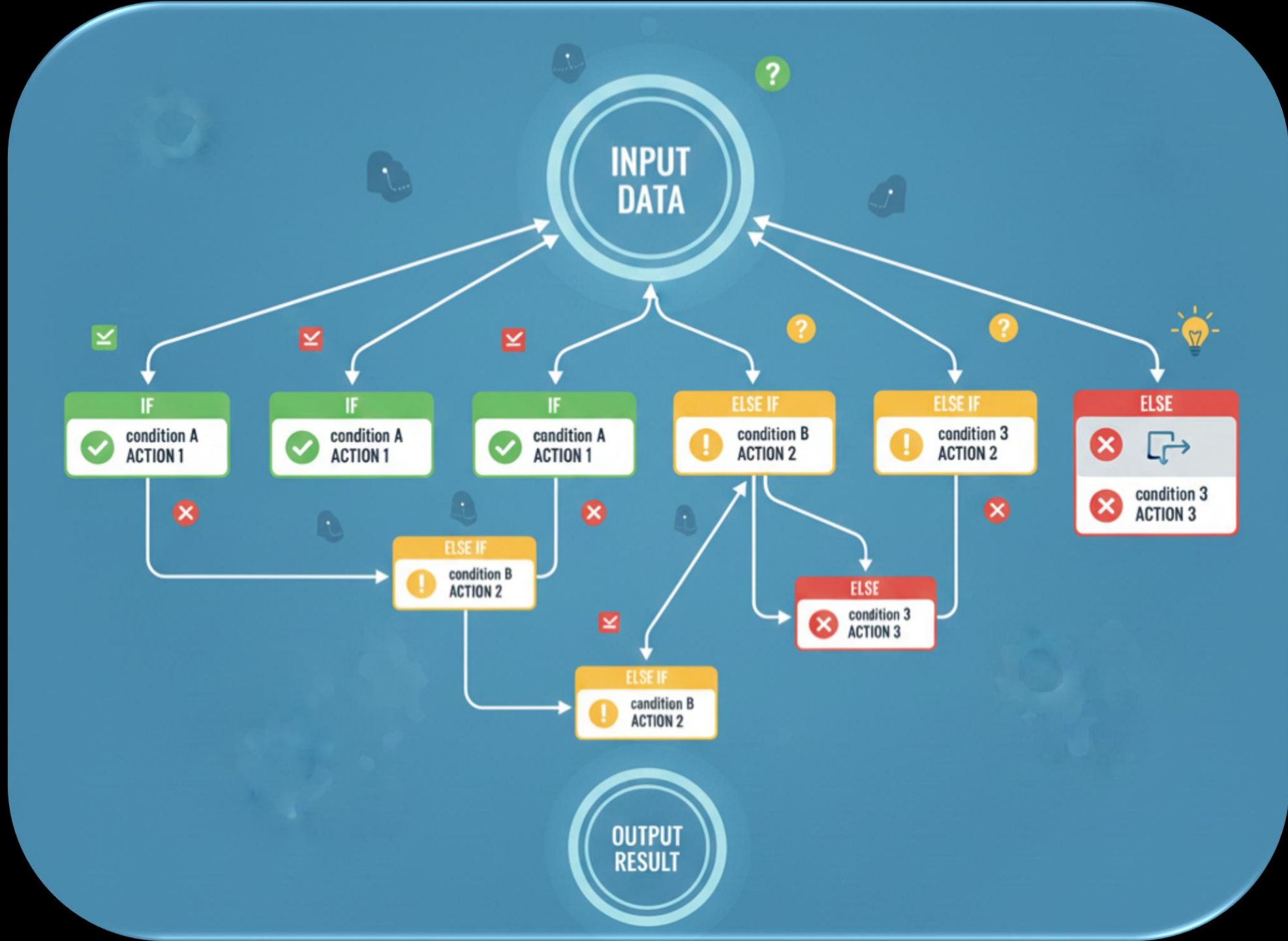
```
yas = int("20")  
fiyat = float("19.5")  
yas_str = str(25)
```

`input()` hangi tip veri tutar ?

Temel Tekrar : Karar Yapıları

03

Temel Tekrar : Karar Yapıları



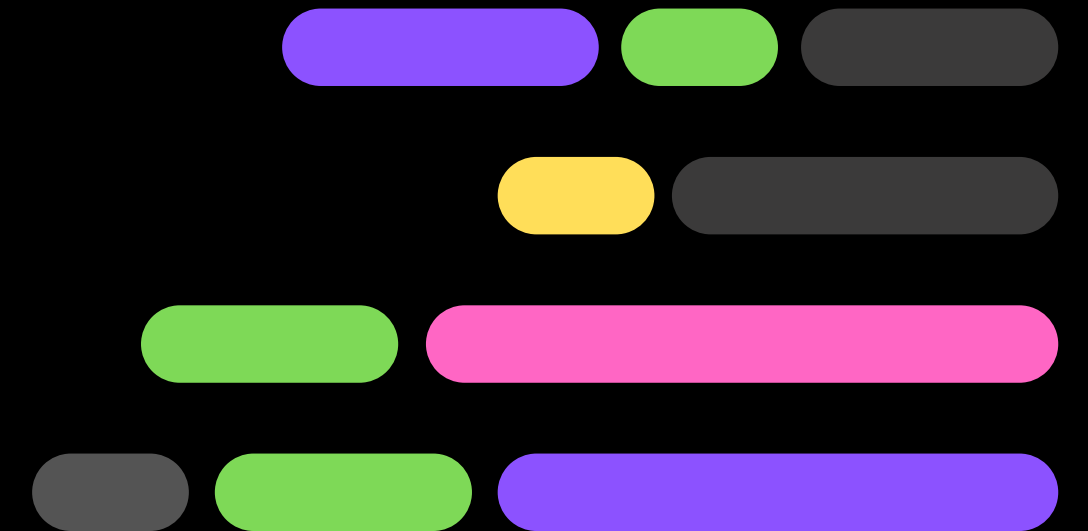
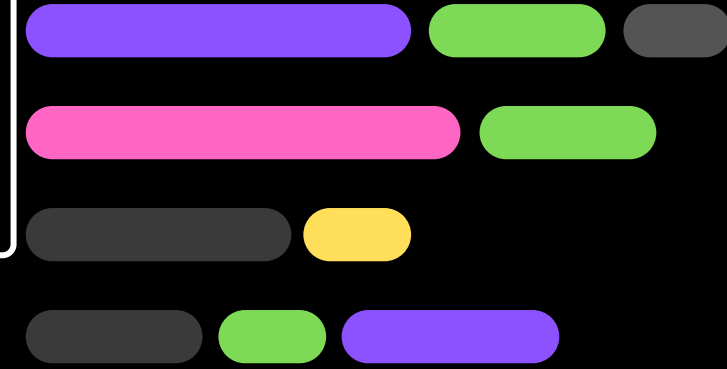


03

Temel Tekrar : Karar Yapıları

```
yas = 20  
  
if yas >= 18:  
    print("Reşit")
```

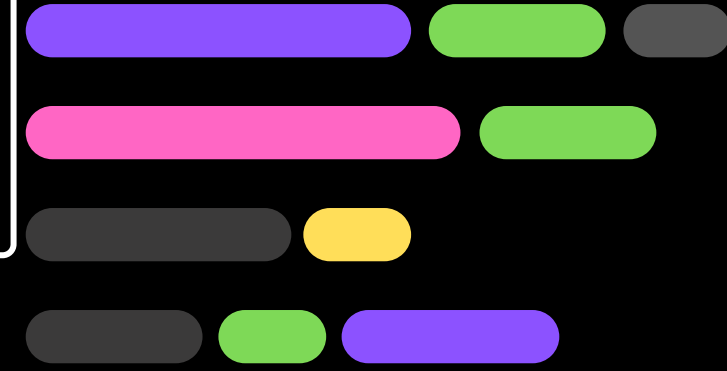
Karar yapılarına neden ihtiyacımız var ?





03

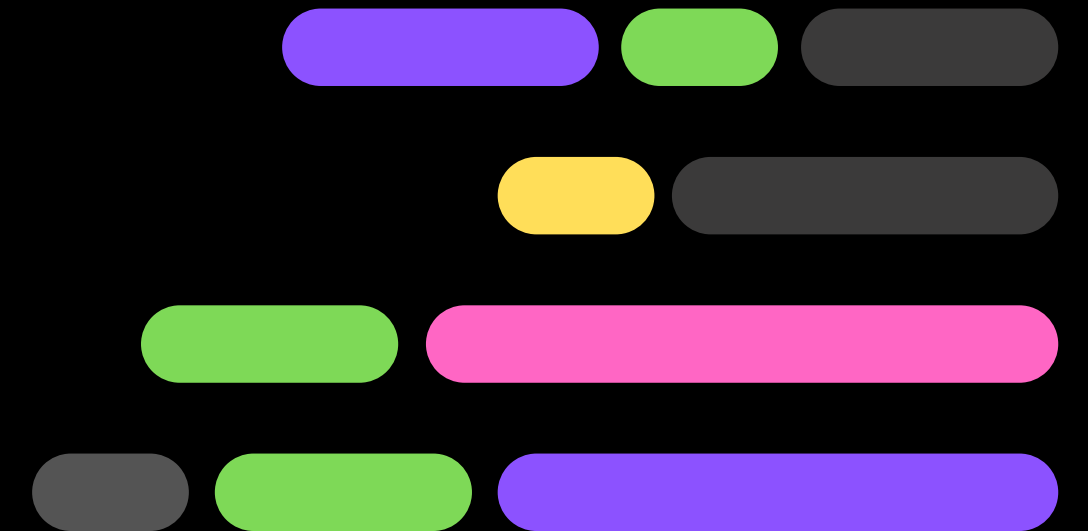
Temel Tekrar :
Karar Yapıları



Karar yapıları bool (True / False) ile çalışır.

```
yas = 20  
print(yas >= 18)
```

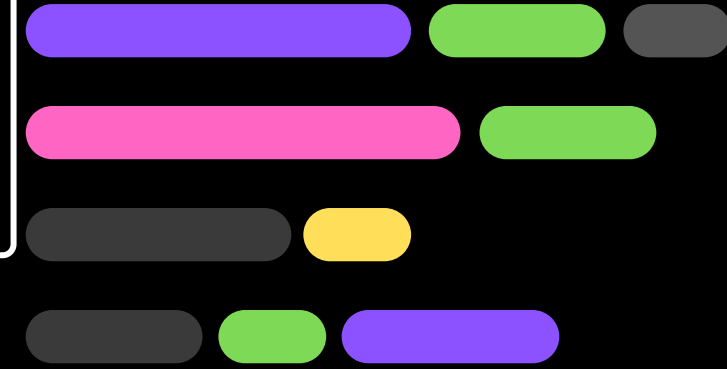
Kodun çıktısı nedir ?





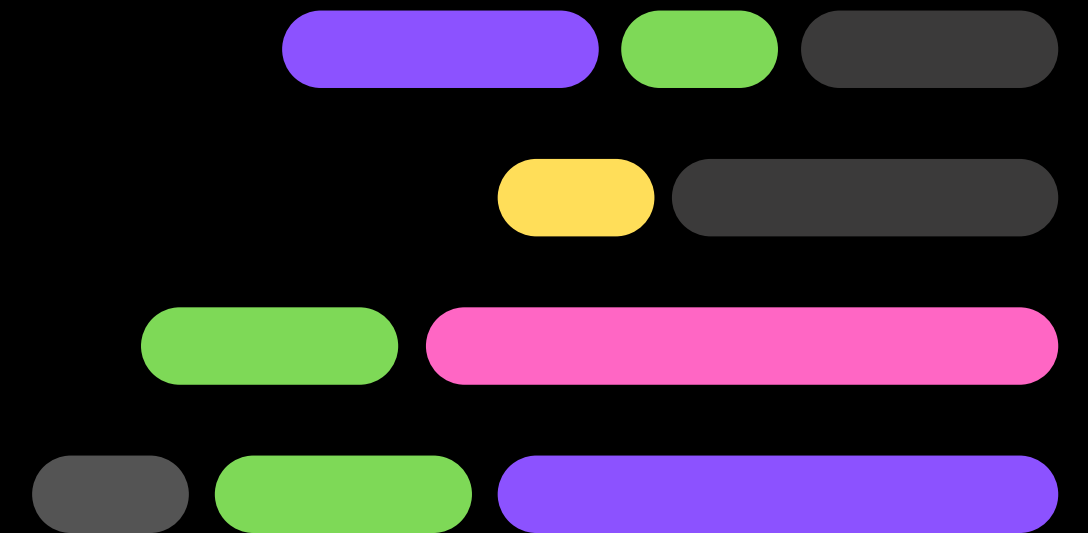
03

Temel Tekrar :
Karar Yapıları



Python if - else Yapısı

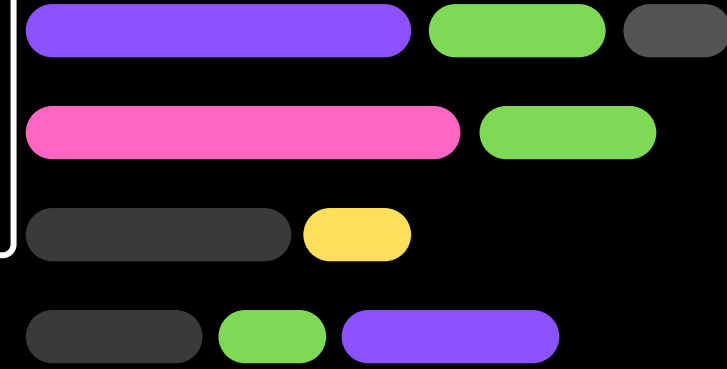
```
if sayi == 0:  
    print("Sıfır")  
elif sayi < 0:  
    print("Negatif")  
else:  
    print("Pozitif")
```



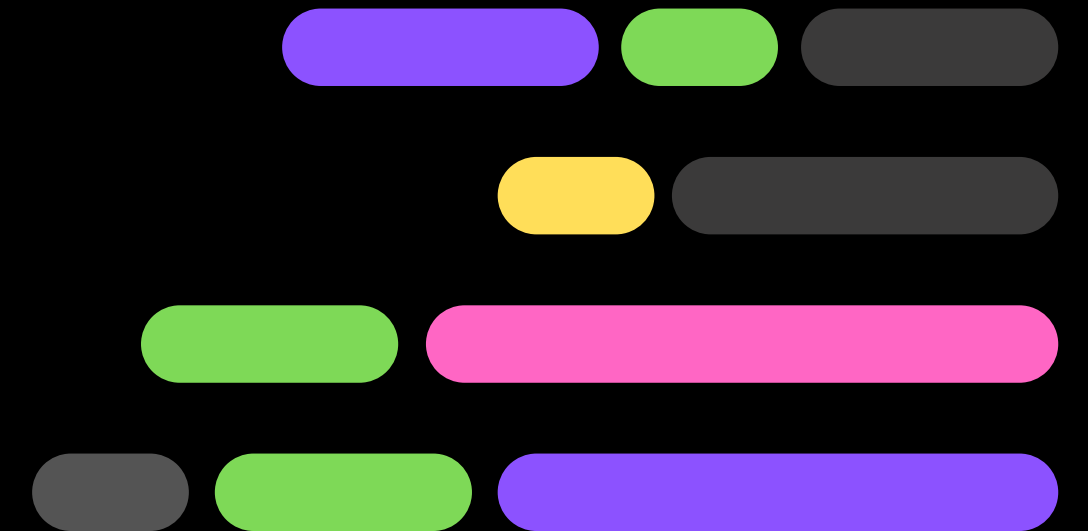


03

Temel Tekrar :
Karar Yapıları



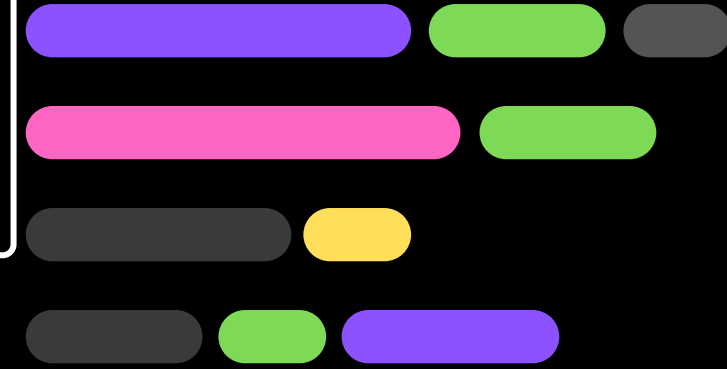
Karşılaştırma operatörleri nelerdir?





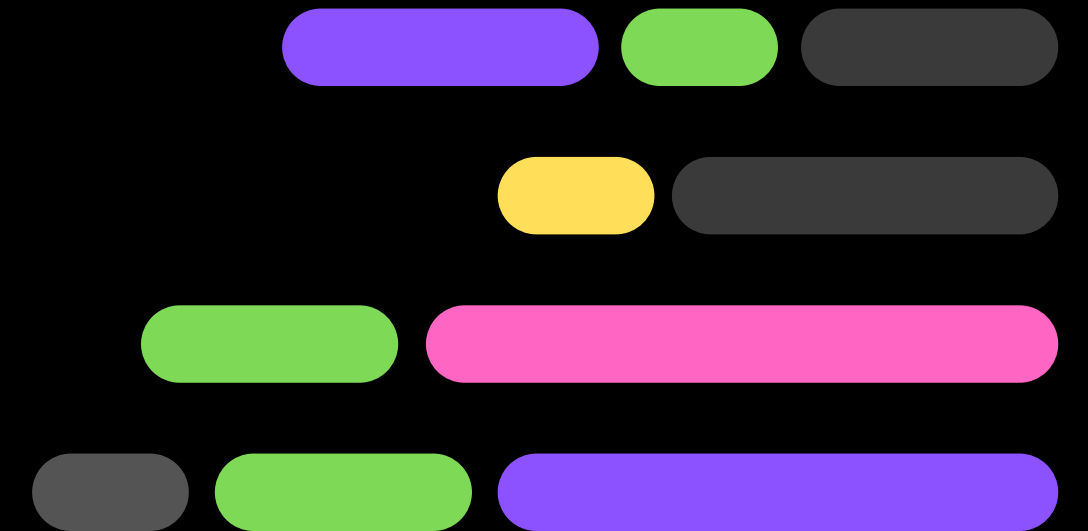
03

Temel Tekrar : Karar Yapıları



Karşılaştırma operatörleri

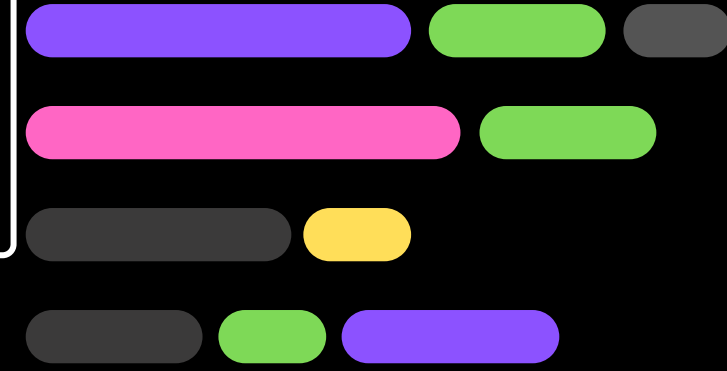
- ==
- !=
- <
- <=
- >
- >=



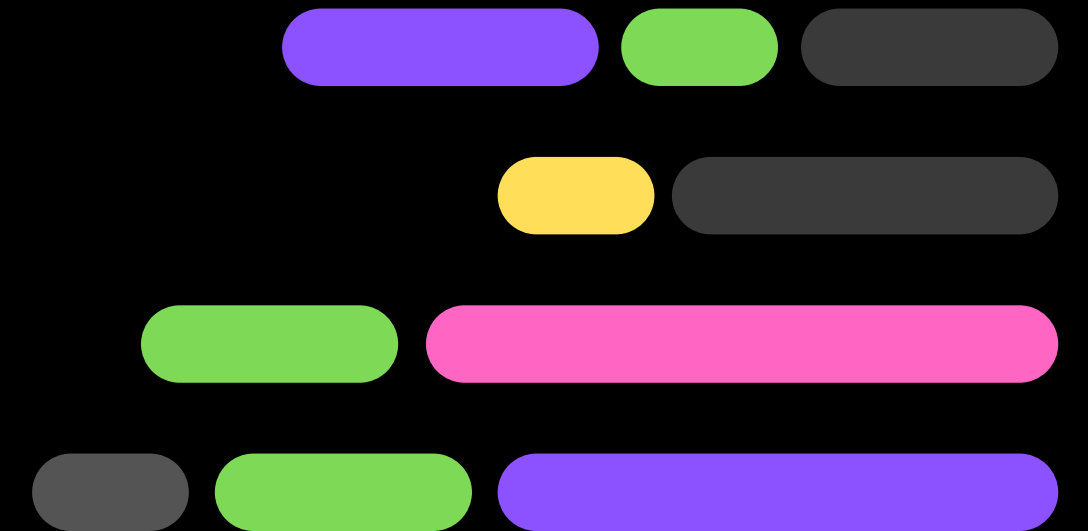


03

Temel Tekrar :
Karar Yapıları



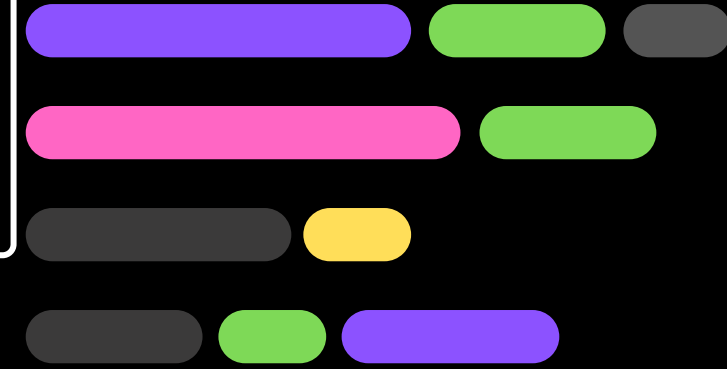
Mantıksal operatörler nelerdir ?





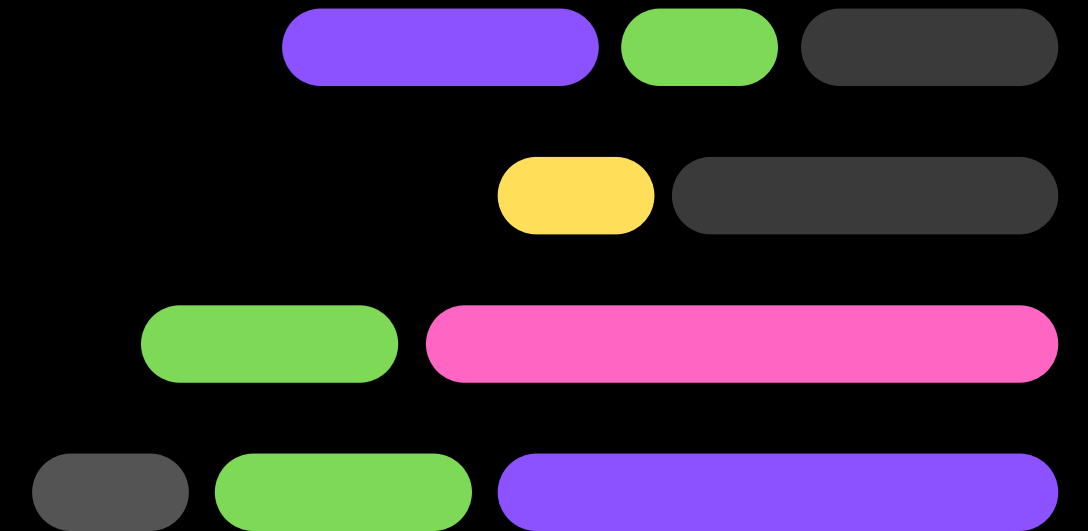
03

Temel Tekrar :
Karar Yapıları



Mantıksal operatörler

- **and**
- **or**
- **not**



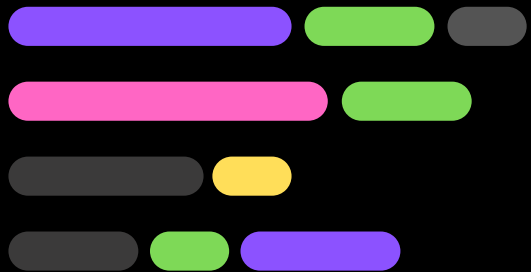
Temel Tekrar : Döngüler



04

Temel Tekrar : Döngüler

Neden Döngüye İhtiyacımız Var?





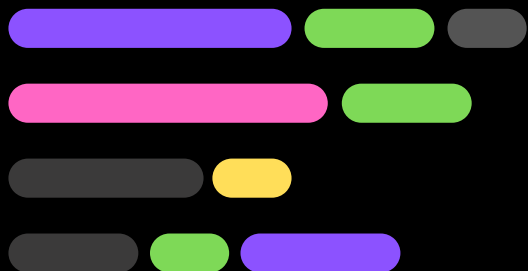
04

Temel Tekrar : Döngüler

```
print("1")  
print("2")  
print("3")  
print("4")  
print("5")
```



```
for i in range(5):  
    print(i+1)
```



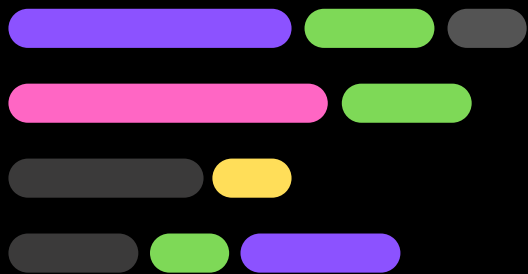


04

Temel Tekrar : Döngüler

range() Kullanımı

```
range(5)  
range(1, 6)  
range(0, 10, 2)
```



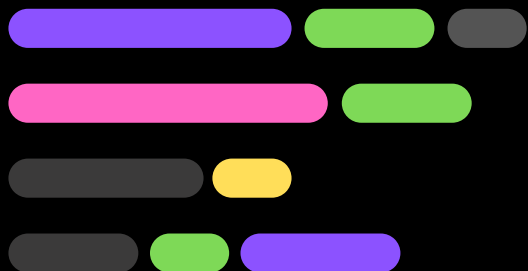


04

Temel Tekrar : Döngüler

for döngüsünde listeler

```
isimler = ["Ali", "Ayşe", "Mehmet"]  
  
for isim in isimler:  
    print(isim)
```





04

Temel Tekrar : Döngüler

for döngüsünde string değerler

```
isimler = "SAMU"  
  
for isim in isimler:  
    print(isim*5)
```

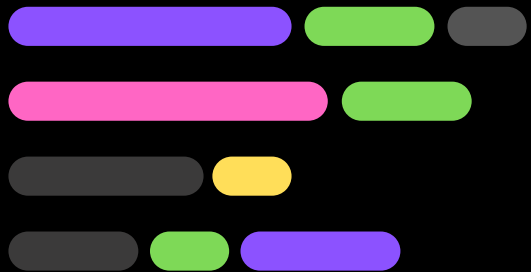
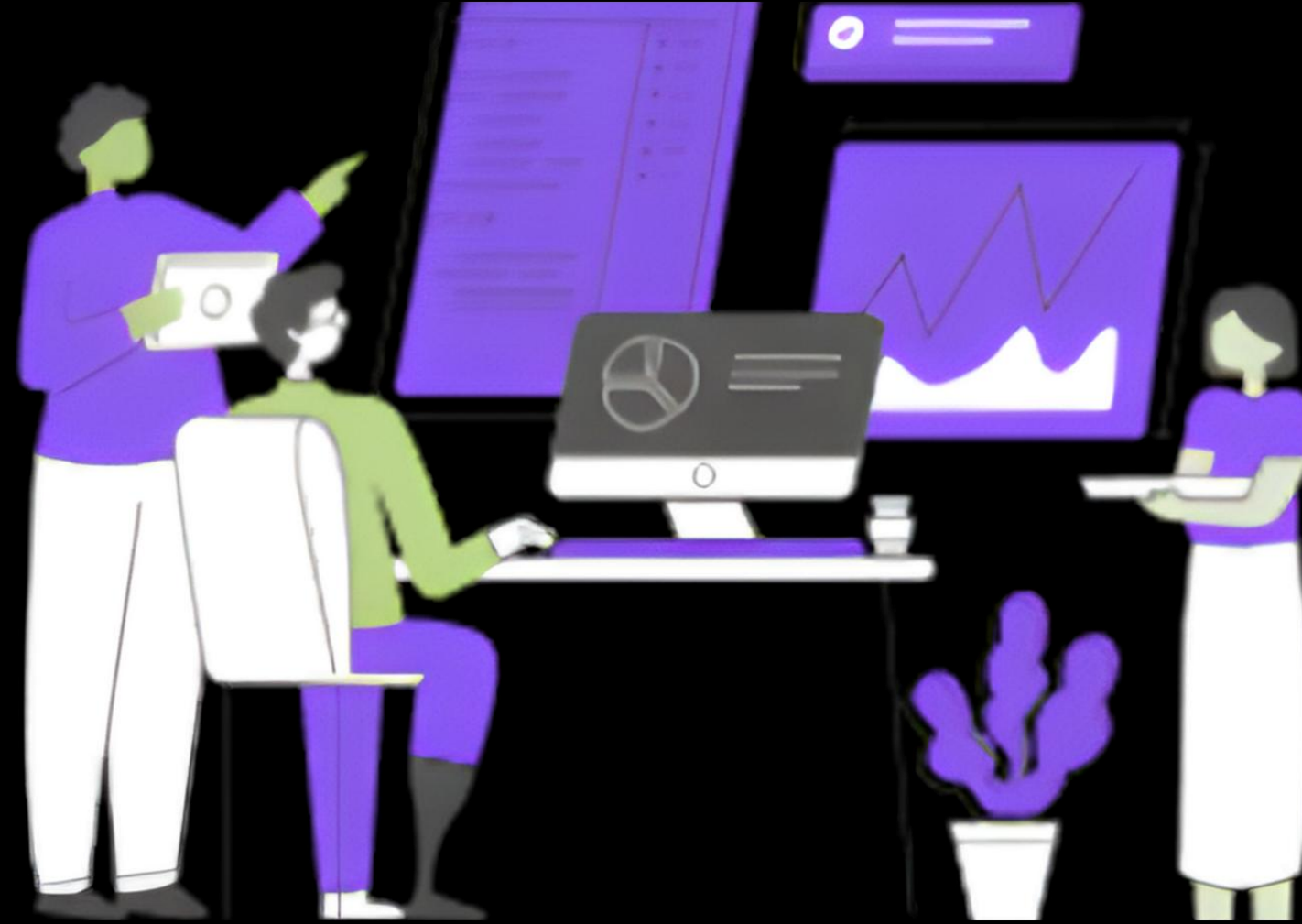




04

Temel Tekrar : Döngüler

Temelde for döngüsü ve while döngüsünün farkı nedir?



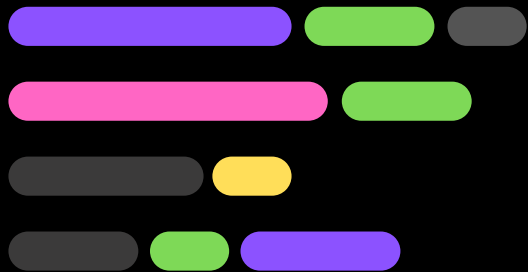
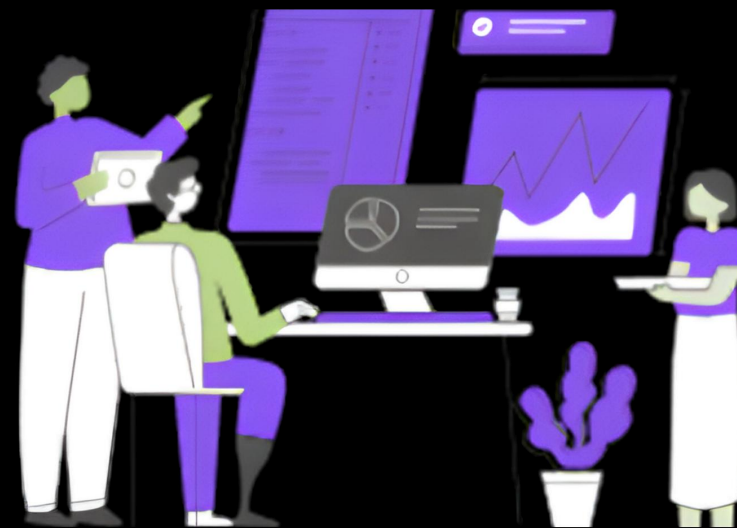


04

Temel Tekrar : Döngüler

Temelde for döngüsü ve while döngüsünün farkı nedir?

Kaç kez döneceği belliyse → for
Koşula bağlıysa → while

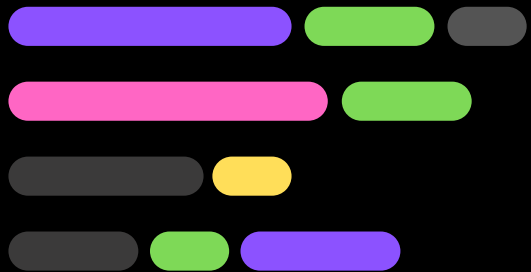




04

Temel Tekrar : Döngüler

Sonsuz While Döngüsü; while True:

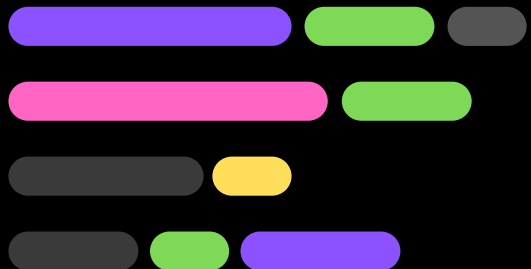




04

Temel Tekrar : Döngüler

break & continue



Örnek Kodlar



05

Örnek Kodlar

```
print("=== SAYI TAHMİN OYUNU ===")
```

```
hedef = 37  
deneme = 0  
maks_deneme = 7
```

```
while deneme < maks_deneme:  
    tahmin = int(input("Tahmininiz (0-100): "))  
    deneme += 1  
  
    if tahmin < 0 or tahmin > 100:  
        print("0-100 arası bir değer giriniz. (Bu deneme sayıldı.)")  
        continue  
  
    if tahmin == hedef:  
        print(f"Tebrikler! {deneme}. denemede bildiniz")  
        break  
    elif tahmin < hedef:  
        print("Daha büyük bir sayı dene.")  
    else:  
        print("Daha küçük bir sayı dene.")  
  
    kalan = maks_deneme - deneme  
    print(f"Kalan deneme: {kalan}")
```

```
if deneme == maks_deneme and tahmin != hedef:  
    print(f"Hak bitti ✘ Doğru cevap: {hedef}")
```



05

Örnek Kodlar

```
toplam = 0
sayac = 0

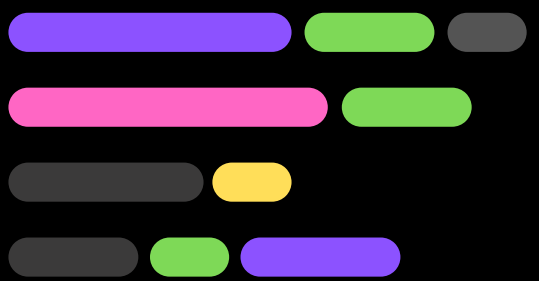
while True:
    sayi = int(input("Sayı gir (bitirmek için 0): "))

    if sayi == 0:
        break

    if sayi < 0:
        print("Negatif sayı atlandı.")
        continue

    toplam += sayi
    sayac += 1

print(f"Girilen pozitif sayı adedi: {sayac}")
print(f"Sayıların toplamı: {toplam}")
```





05

Örnek Kodlar

```
for i in range(1, 11):  
    if i % 2 != 0:  
        continue  
    print(i)
```

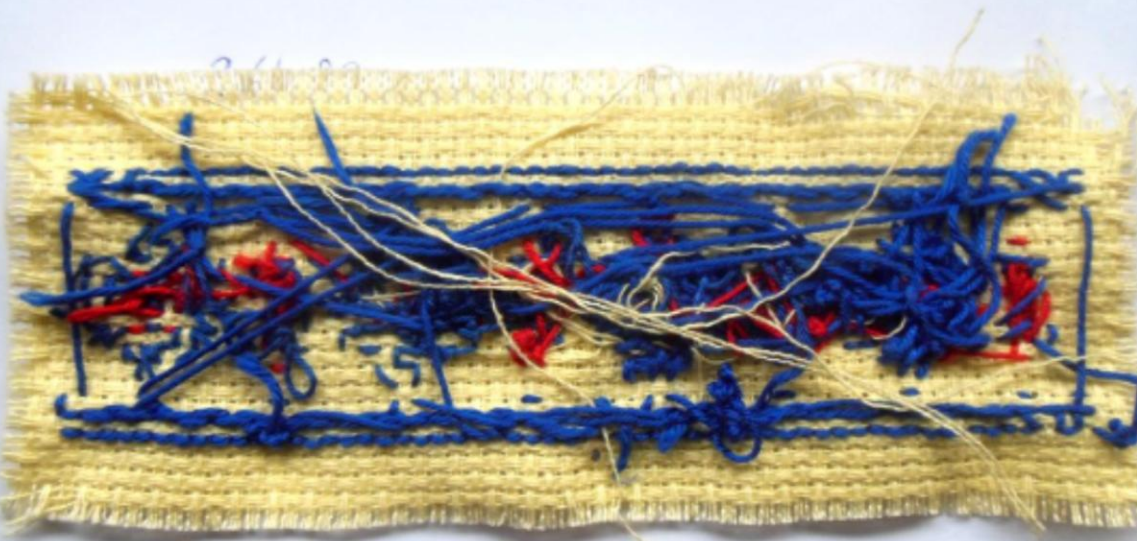




Frontend



Backend



Son...