

YAZILIM TASARIMI VE PROGRAMLAMA

# YAZILIMDA KULLANILAN BİLEŞENLER - 2

 DÖNGÜ YAPILARI

 KARAR YAPILARI

# Hatırlayalım & Tartışalım

## ? SORU

Bir problemi çözerken işlemleri gelişigüzel mi, yoksa belirli bir sıraya göre mi yapıyorsunuz?

## 🗨️ TARTIŞMA

Bir işi düzenli ve hatasız yapabilmek için adımların açık ve net olması neden gereklidir?

## KARŞILAŞTIRMA

### ✗ Gelişigüzel

1. Kontrol et
2. Sonucu yaz
3. Planla
4. Malzeme al
5. Başla

### ✓ Düzenli Adımlar

- 1 Problemi anla ▶
- 2 Planla ▶
- 3 Malzeme hazırla ▶
- 4 Uygula ▶
- 5 Kontrol et ▶



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir

# Hayatın İçinden Döngüler

## ? SORU

Her sabah yüzünü yıkaman,  
diş fırçalaman bir **döngü** müdür?

## ÖRNEKLER

- Mevsimlerin sırasıyla gelmesi
- Tohumdan meyveye yaşam serüveni



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir

# Düşünelim

## SORU

Bir spor karşılaşmasında skor arttıkça sonucun ekrana yazılması **tekrarlayan bir işlem** midir?



Her gol / sayı → ekran güncellenir → tekrar bekle → yeni gol... Bu bir **döngüdür!**

MAÇ SKORU

● CANLI



A Takımı

2



B Takımı

2

67:00

DÖNGÜ KAYDI — Her Gol = Ekran Güncellenir

- 1 A Takımı gol attı → 1 - 0
- 2 B Takımı gol attı → 1 - 1
- 3 A Takımı gol attı → 2 - 1
- 4 B Takımı gol attı → 2 - 2



**Evet! Her gol tekrar eden aynı işlemi çalıştırır → Bu bir DÖNGÜDÜR**



@ssinanakdemir



www.sinanakdemir.com.tr

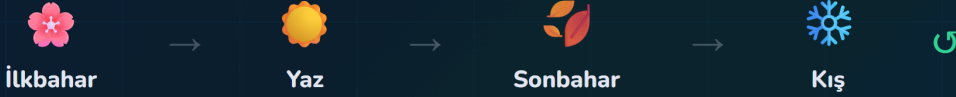


@ssinanakdemir

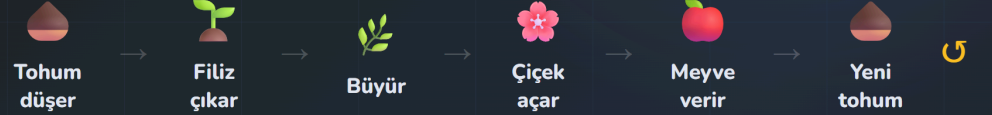
# Döngü Nedir?

 Döngü, bir işin belli bir düzen içerisinde tekrar tekrar yapılmasıdır.

## MEVSİM DÖNGÜSÜ



## BİTKİ YAŞAM DÖNGÜSÜ



Tohum → filiz → büyüme → çiçek → meyve → yeni tohum...  
ve **döngü yeniden başlar!**



Çevrenizi gözlemleyerek bu gibi pek çok döngüyü fark edebilirsiniz. Sizin de çevrenizde gözlemlediğiniz döngüler var mı?



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir

# Programlama ve Döngü



Döngü; **başlangıç ve bitiş arasında** aynı işlemlerin tekrarlanmasıdır. Bilgisayar programlarında da bazı işlemlerin tekrarlanması gerekebilir.



## Tramvay Döngüsü

Belli saat aralığında her dakikada bir aynı duraktan geçer.

1 Tramvay durağa gelir →

2 Yolcular iner / biner →

3 1 dakika bekler →

4 Hareket eder →

Her dakika tekrar eder



## Termostat Döngüsü

Sıcaklık belirli dereceye ulaşana kadar kombiyi çalıştırır.

1 Sıcaklığı ölç →

2 Hedef dereceye ulaştı mı? →

3 Hayır → Kombiyi çalıştır →

4 Evet → Kombiyi durdur →

Hedef dereceye ulaşana kadar tekrar



Bu tür işlemlerde bilgisayar programı içerisinde **döngü komutları** kullanılır.



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir

# Döngü Kullanımının Amacı



Aynı komutları **tekrar tekrar yazmanıza gerek kalmaz.**



Programınızın kodları **daha kısa ve anlaşılır** hale gelir.



Bilgisayara bir işlemi **kaç kez veya ne zamana kadar** tekrar edeceğini belirtebilirsiniz.



Komutlarındaki **hatayı bulmak ve düzeltmek** kolaylaşır.



Döngüler; tekrarlayan işlemlerin **verimli çalışmasını**, kodun **anlaşılabilir ve yönetilebilir** hale gelmesini sağlar.



@ssinanakdemir



[www.sinanakdemir.com.tr](http://www.sinanakdemir.com.tr)



@ssinanakdemir

# Döngü Yapıları

Döngüler, kod bloklarının **belirli bir sayıya ulaşana** veya **bir koşul sağlanana kadar** tekrar tekrar çalışması temeline dayanır.

## 1 Sayıya Bağlı Döngü

- Tekrar sayısı **bellidir**
- Koşul **sayıya** bakar



### ÖRNEK

Bir turnuvada puan sıralamasına göre **ilk 10'a giren öğrencilerin** belirlenmesi sayıya bağlı döngüdür.

## 2 Koşula Bağlı Döngü

- Tekrar sayısı **belli değildir**
- Bir **duruma bağlıdır** , koşul duruma bakar



### ÖRNEK

Oda sıcaklığı **21°C olana kadar** kombinin çalışması ve hedef sıcaklığa ulaşınca durması koşula bağlı döngüdür.



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir

# Koşula Bağlı Döngüler

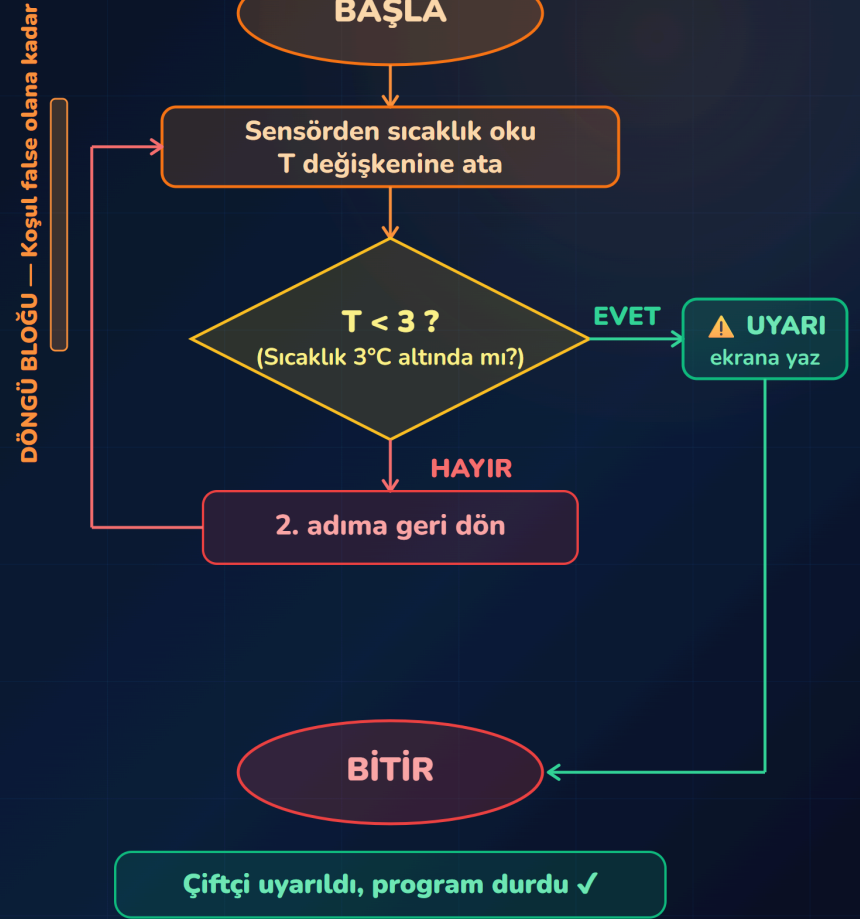
Koşul **TRUE (Doğru)** olduğu sürece döngü çalışır.  
Koşul **FALSE (Yanlış)** olduğunda döngü **sonlanır**.  
Tekrarın ne zaman biteceği **önceden bilinmez**.

✓ **TRUE** → Döngü devam eder

✗ **FALSE** → Döngü biter

## ÖRNEK: SERA SICAKLIK KONTROLÜ

- 1 Program **başlar**
- 2 Sensörden sıcaklık okunur → **T** değişkenine atanır
- 3 **T < 3** mü? — **EVET** → Ekran "UYARI" yazılır → 5. adıma git
- 4 **HAYIR** → Koşul sağlanmadı, **2. adıma geri dön**
- 5 Program **sona erer**



# Sonsuz Döngü

Durdurma koşulu **olmaz** veya **yanlış yazılırsa** döngü **hiç bitmez.**

Bu tür hatalı yapılara **sonsuz döngü** denir ve bir **programlama hatasıdır.**



**Koşul hiçbir zaman FALSE olmaz → Program sonsuza kadar çalışır → Bilgisayar takılır!**

## ÖRNEKLER



**Çocuk & Anne Yaşı**

"Çocuğun yaşı annesinin yaşından büyük olana kadar tekrarlar" → Bu koşul **asla gerçekleşmez!**



**Trafik Işığı**

Işığın **yeşile dönmesi için komut yazılmazsa** araçlar asla hareket edemez → Sistem **döngüde kalır!**

TEKRAR TEKRAR TEKRAR...



**Programlama Hatası!**



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir

• ETKİNLİK


# Düşün & Tartış

 Görev: **1'den başlayarak** rakamları **birer artırarak** 50'ye kadar sıralayan programın akış şemasını hazırlayın. Döngü kullanan ve kullanmayan iki farklı çözüm yolu deneyin.

**1** **Döngüsüz** çözdüğünüzde kaç adım gerekti? Bu yolun **zorluğu** ne oldu?

**2** **Döngü kullanınca** kaç adıma indi? Bu yolun **avantajı** neydi?

**3** 50 yerine **1.000 veya 1.000.000** olsaydı hangi yolu tercih ederdiniz? **Neden?**

 Fikirlerinizi **paylaşın** — hangisi daha verimli, neden?



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir

# POS & PIN

## Akış Şeması

Bir POS cihazında **3 kez** PIN kodu girilmesine izin veren, **3. hatalı girişte** ekrana "**PIN Bloke**" yazan programın komutlarını akış şemasındaki **boş kutulara** yerleştiriniz.

### KOMUTLAR — Doğru Yere Yerleştir

🔒 PIN Bloke

Sayaç = Sayaç + 1

✅ İşlem Onaylandı

Sayaç = 3 mi?

● BAŞLA

PIN Doğru mu?

Sayaç = 0

PIN Gir

● BİTİR

● BİTİR

Tekrar Dene



İpucu: **Şekillere** dikkat et — her komut farklı bir şekle gider!

### AKIŞ ŞEMASI — BOŞLUKLARI DOLDURUN



• ETKİNLİK

# Su Deposu Akış Şeması

Bir su deposunun seviyesi **dolana kadar** her **10 saniyede bir** ölçüm yapan sistemde,  
su seviyesi **100 cm'ye ulaştığında** vanalar kapanmalıdır.  
Bu sisteme ait akış şemasını çiziniz.

1 Bu sistem **hangi tür döngüye** örnektir?  
Neden?

2 Döngünün **başlangıç ve bitiş** koşulu ne olmalıdır?



@ssinanakdemir



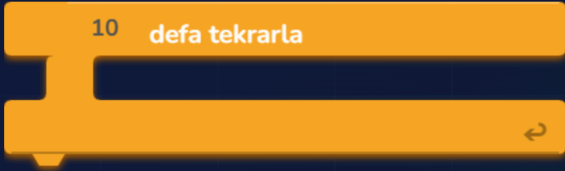
www.sinanakdemir.com.tr



@ssinanakdemir

# mBlock — Döngü Blokları

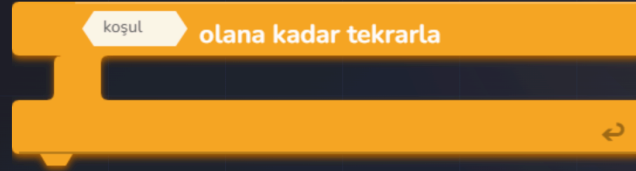
## 12 34 SAYIYA BAĞLI DÖNGÜ



İçindeki komutları **belirtilen sayı kadar** tekrar çalıştırır.  
Kutucuktaki sayıyı değiştirerek tekrar sayısını ayarlayabilirsiniz.

🏆 Örnek: 10 öğrenciyi listele → **10 defa tekrarla**

## 24 KOŞULA BAĞLI DÖNGÜ



Koşul **gerçekleşene kadar** içindeki komutları tekrar çalıştırır.  
Koşul sağlandığı anda döngü durur.

🌡️ → Örnek: Sıcaklık 100°C olana kadar ısıt **olana kadar tekrarla**

YAZILIM TASARIMI VE PROGRAMLAMA

# KARAR YAPILARI

 Eğer / Değilse

 Koşul & Sonuç

 @ssinanakdemir

 [www.sinanakdemir.com.tr](http://www.sinanakdemir.com.tr)

 @ssinanakdemir

# Hatırlayalım

Bir yazılım hazırlanırken programın karşılaşılabileceği **her durum önceden düşünülür** ve programa eklenir. Çünkü bilgisayarlar insanlar gibi **düşünemez, sezemez** ya da aklına gelince karar veremez.

Onlar sadece kendilerine **öğretilen kurallara** göre çalışır. Yazdığımız her blok aslında bilgisayara verdiğimiz bir **kuraldır**. Eğer bir durum için kural yazılmamışsa, bilgisayar o durumda **ne yapacağını bilemez**.

## ŞİMDİ DÜŞÜNELİM



Programda **eksik kurallar** olursa neler olur?



Fikirlerinizi arkadaşlarınızla paylaşın ve tartışın.



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir

# Karar Yapıları Nedir?



Karar yapıları, belirli bir **koşul ifadesine göre** bilgisayarın **farklı işlemler yapmasını** sağlar. Bir koşul ifadesi **doğru ya da yanlış** olabilir.



**TRUE — DOĞRU**

## Koşulun Doğru Olması

Şartın sağlandığını gösterir.

Bilgisayar bu duruma ait komutu çalıştırır.



**FALSE — YANLIŞ**

## Koşulun Yanlış Olması

Şartın sağlanmadığını gösterir.

Bilgisayar farklı bir yolu izler ya da durur.



Karar yapıları sayesinde program

**her duruma uygun**

farklı tepkiler verebilir — tıpkı insanların günlük kararları gibi.



@ssinanakdemir



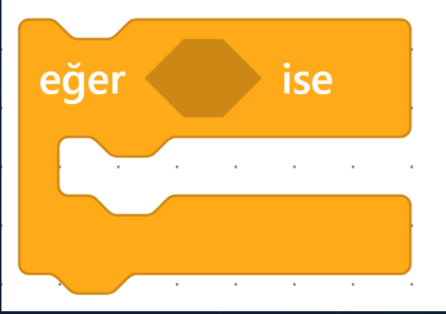
www.sinanakdemir.com.tr



@ssinanakdemir

# Karar Yapısı Kod Blokları

## 1 Eğer İse Bloğu



Verilen **koşul sağlanıyorsa** "eğer" bloğu içindeki kodlar çalışır.

Koşul sağlanmıyorsa bu blok **atlanarak** diğer işlemler devam eder.

## 2 Eğer İse Değilse Bloğu



Koşul **sağlanıyorsa** "eğer" bloğundaki kodlar;  
koşul **sağlanmıyorsa** "değilse" bloğundaki kodlar çalışır.



@ssinanakdemir



[www.sinanakdemir.com.tr](http://www.sinanakdemir.com.tr)



@ssinanakdemir

# Karar Yapısı — Örnek Uygulamalar

## Sınav Notu Kontrolü

Eğer **Not > 50** ise

"Tebrikler, Geçtiniz!" de

Değilse

"Maalesef, Kaldınız." de

Bu yapıda şart sağlanırsa (50'den büyükse) ilk bölüm, sağlanmazsa ikinci bölüm çalışır.

## Engelden Kaçan Robot

Eğer **Mesafe < 15** ise

Geri Git ve Dur

**Sadece Eğer:** Robot sadece önünde bir engel gördüğünde tepki verir. Engel yoksa yoluna devam eder.



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir

# mBlock ile Uygulama Zamanı

1 Eğer klavyeden "Boşluk" tuşuna basılıyorsa panda "Miyav" sesini çıkarsın, basılmıyorsa sessiz kalsın.

2 Eğer panda fare imlecine değiyorsa boyutunu %150 yap, değmiyorsa normal boyutuna (%100) döndür.

3 Defa tekrarla döngüsünü kullanarak pandaya bir Kare çizdirin. (4 defa tekrarla: Git ve Dön).

4 Sürekli tekrarla döngüsü ile pandanın sahnede sürekli sağa-sola gitmesini ve kenara gelince sekmesini sağlayın.

# SON



Haftaya: Blok Tabanlı Ortamda Yazılım Geliştirme ve mBlock programını detaylıca göreceğiz.



@ssinanakdemir



www.sinanakdemir.com.tr



@ssinanakdemir