

Bilişim Teknolojileri 6. Sınıf Sınava Hazırlık

Önemli Konu Özetleri ve Hap Bilgiler

1. Algoritma Sıralaması Nasıl Olmalı?

Bir program (örneğin öğrencinin geçip kaldığını bulan program) her zaman belirli bir sırayı izler:

1. **Başla** (Tüm algoritmaların ilk adımı)
2. **Veriyi Al** (Örn: Ortalamayı kullanıcıdan al)
3. **Şartı Kontrol Et** (Eğer ortalama ≥ 50 ise)
4. **Sonuç Yazdır** (Evet ise "Geçti", değilse "Kaldı")
5. **Bitir** (İşlem sonlanır)

2. Sahnenin Tam Merkezi

Scratch sahnesi bir koordinat düzlemidir. Tam orta (başlangıç) noktası her zaman **X: 0** ve **Y: 0** olarak kabul edilir. X yatay eksen, Y dikey eksen temsil eder.

3. En Sağ Nokta (X Sınırı)

Yatay eksen (X eksen) kuklanın gidebileceği mesafeler sınırlıdır. Ekranın en solu **X = -240**, ekranın en sağ **X = 240** değerini alır.

4. En Alt Nokta (Y Sınırı)

Dikey eksen (Y eksen) kuklanın yukarı ve aşağı sınırları şöyledir: Ekranın en üstü **Y = 180**, ekranın en altı **Y = -180** değerini alır.

5. Koordinat Hesaplama Mantığı

Koordinat sorularında en çok kafa karıştıran kısım budur:

1. **değiştir** Ne Demek?

Elinizdeki mevcut sayının üzerine eklemek veya ondan çıkarmak demektir.

x konumunu 30 değiştir (Mevcut X: 50 ise, X = 80 olur)

2. **yap** Ne Demek?

Elinizdeki sayıyı tamamen unutup konumu direkt o yeni sayıya eşitlemektir (sabitlemektir).

x konumunu 100 yap (Mevcut X: 80 ise, X direkt 100 olur)

6. Görünüm Menüsü (Mor Bloklar)

Kuklanın sahnede nasıl görüldüğüyle ilgili tüm işlemler bu gruptan yapılır. Boyut, renk efektleri, konuşma balonları ve kostüm değişimlerinin hepsi mor renkli **Görünüm** sekmesindedir.

Örnekler: **Merhaba!** de renk etkisini **25** değiştir sonraki kostüm

7. Kod Blokları Neden Renkli?

Scratch arayüzünde kodların farklı renklerle kategorize edilmesinin en büyük sebebi **kullanıcıya kolaylık sağlamaktır**. Bu sayede blokların hangi kategoriye ait olduğunu hemen ayırt ederiz ve aradığımız komutu saniyeler içinde bulabiliriz. (Mavi=Hareket, Mor=Görünüm vb.)

8. Projeyi Nasıl Başlatırız?

Scratch projeleri genellikle sahnenin üstündeki **Yeşil Bayrak** simgesine tıklandığında başlar. Bu kod bloğu programın başlamasını tetikler ve sarı renkli **Olaylar** kategorisi altında bulunur.

tıklandığında

9. Yeşil Bayrak ve Kırmızı Buton

Yeşil Bayrak: Yazılan kodları baştan çalıştırır (Projeyi Başlatır).

Kırmızı Buton (Sekizgen): Çalışan tüm kodları, döngüleri ve animasyonları anında durdurur (Projeyi Durdurur).

10. Şartlı İşlemler (Eğer... İse)

Kodların belirli bir şarta göre çalışmasını sağlayan (Örn: "Fareye değiyorsa saklan") karar blokları turuncu renkli **Kontrol** menüsündedir.

eğer **fare imleci** ne değiyor mu? ise

gizlen

11. Operatörler Menüsü (Yeşil)

Matematiksel işlemler ve sayı karşılaştırmaları her zaman yeşil renkli **Operatörler** sekmesinden alınır.

+ **>** **50** **1** ile **10** arasında rastgele sayı seç

12. Çift İhtimalli Şartlar

"Ehliyet Alma" gibi programlarda yaş 18'den büyükse "Alabilirsin", küçükse "Alamazsın" demek istiyoruz. Ortada **iki farklı ihtimal** varsa mutlaka **Eğer ise, Değilse** bloğu kullanılır.

eğer yaş > 17 ise

Ehliyet Alabilirsin de

değilse

Ehliyet Alamazsın de

13. Algılama Menüsü (Değiyor mu?)

Bir kuklanın fare imlecine, ekran kenarına veya başka bir kuklaya dokunup dokunmadığını kontrol eden hissetme blokları açık mavi renkli Algılama menüsündedir.

fare imleci ne değiyor mu?

X rengine değiyor mu?

14. Rastgele Konuma Git

Bu blok çalıştırıldığında kukla yürüyerek gitmez. Sahne içindeki rastgele (bilgisayarın o an uydurduğu) bir X,Y koordinatına **anında ışınlanır**.

rastgele konum 'a git

15. Puanı Artırmak (Değiştir Bloğu)

Oyunda elma yedikçe puanın artması için **Değiştir** bloğu kullanılır.

Puan 'ı 1 kadar değiştir

Eğer Puan 'ı 1 yap dersiniz, ne kadar elma yerseniz yiyin puan hep sabit 1 olarak kalır.

16. Sonsuz Döngü (Sürekli Tekrarla)

İçine yerleştirilen kodların, program durdurulana kadar **baştan sona hiç durmadan** çalışmasını sağlar. Kontrol (Turuncu) grubundadır.

sürekli tekrarla

10 adım git

sonraki kostüm

17. Değişken İsimlendirme Sanatı

Programlamada bir değişken oluştururken isimler rastgele verilmemelidir. **İsim, içindeki veriyi anlatmalıdır.** "İki sayının toplamı" programı için şu bloklar oluşturulmalıdır:

Sayı1 Sayı2 Toplam

18. Hızlı Koordinat Pratiği

Başlangıç **X: 100, Y: 100**

1- **x konumunu 50 değiştir** → 100'e 50 ekle = 150

2- **y konumunu -100 yap** → Mevcut değeri boşver, direkt -100'e sabitle.

Son Konum: X: 150, Y: -100

19. Değişken ve İşlem Pratiği

sayı 1 'i **70** yap

sayı 2 'i **30** yap

ortalama 'yı **sayı 1 + sayı 2 / 2** yap

sonuç 'u **ortalama - sayı 2** yap

Eğer **sonuç > 10** ise **sonuç 'u 50 yap** değilse **sonuç 'u 0 yap**

Sayı1=70, Sayı2=30. Toplamı 100, ortalaması 50 olur. Sonuç değişkeni ilk başta 50-30=20 olur. 20 sayısı 10'dan büyük olduğu için sonuç en son **50** olarak kalır.

20. Yürüme Animasyonu Nasıl Yapılır?

Bir kuklayı yürütmek için sadece **10 adım git** demek yetmez. Ayaklarını oynatıyormuş gibi bir animasyon oluşturmak için hareket komutunun yanına şu blok eklenmelidir:

sonraki kostüm

21. Uçan Kuş Hatası

Bir kuş kuklası sadece ileriye doğru donuk bir şekilde gidiyorsa, kanat çırpma komutu unutulmuş demektir. Çözüm olarak koda Görünüm menüsünden **sonraki kostüm** bloğu eklenmelidir.

22. Sahne Dışına Çıkılmamak

Kuklanın sahne dışına çıkıp kaybolmasını önlemek için **kenara geldiysen sektir** komutu kullanılır. Bu komut kukla kenarlara çarptığı an yönünü tersine çevirir.

23. Kenardan Dönünce Tepetaklak Olmak

Kukla kenara çarpıp döndüğünde baş aşağı (tepetaklak) dönebilir. Bunu engellemek ve dik durmasını sağlamak için **dönüş stilini sol-sağ yap** ayarı kullanılmalıdır.

24. İşlemleri Neden Bekletiriz?

Bilgisayarlar komutları çok hızlı çalıştırır. Eğer art arda çok hızlı kostüm değiştirir veya diyalog kurarsak gözümüz bunu yakalayamaz. Görebilmek için Kontrol menüsünden **1 saniye bekle** eklenir.

25. Kuklalar Arası Konuşma (Haber Sal)

İki kuklayı konuştururken "Saniye bekle" yöntemi kullanmak kaymalara sebep olur. Profesyonel iletişim şudur:

KEDİ (Gönderen Kukla):

tıklandığında

merhaba de

haber1 haberini sal

KÖPEK (Alıcı Kukla):

haber1 haberini aldığımda

Selam, hoş geldin! de

26. Birbirini Sıfırlayan Kodlar

Bir kuklaya peş peşe şu komutlar verilirse:

x konumunu -10 değiştir

x konumunu 10 değiştir

Sonuç: +10 ve -10 birbirini nötrler ve kukla yerinden hiç kıpırdamamış gibi görünür.

27. 'Adım Git' Bloğunun Yönü

10 adım git komutu her zaman sağa gitmez. Kukla o anki **yöneldiği açı (burnu)** nereyi gösteriyorsa o tarafa doğru ilerler. (0 ise yukarı, -90 ise sola gibi).

28. Açı Bulma Uygulaması (Bölüm 1)

Projelerdeki o büyük açı kodunda **İç İçe Eğer İse Değilse** kullanılmıştır. Çalışma mantığı bir şelale gibidir:

eğer **Açı < 90** ise

haber1 haberini sal → *Dar Açı*

değilse

(Açı 90 veya daha büyükse buraya gir - Devamı aşağıda)

Kural: Program her zaman en üstteki soruyu sorar. Cevap EVET ise işlemi yapar ve aşağılara ASLA İNMEZ.

Kullanıcı 50 girerse: 1. Soru: "Açı (50) 90'dan küçük mü?" Cevap: **Evet**. O zaman "Dar Açı" der ve program biter.

29. Açı Bulma Uygulaması (Bölüm 2)

Şelale kuralına ve "Değilse" kısmının içine bakalım:

değilse

eğer **Açı = 90** ise

haber2 haberini sal → *Dik Açı*

değilse

haber3 haberini sal → *Geniş Açı*

Açı **90** girilirse: $90 < 90$ değildir. İçteki eğer'e bakar: $90 = 90$ mı? Evet (**Dik Açı**).

Açı **120** girilirse: $120 < 90$ değildir, $120 = 90$ da değildir. En sondaki Değilse çalışır (**Geniş Açı**).