

Program nasıl karar verir?

Bugün sketch'lerimiz duruma göre farklı davranmayı öğrenecek.

if / else

Girinti

else if zinciri

İç içe koşullar

SORU

**Skor 10 olunca “Kazandın!” yazsın.
Olmadıysa daire çizsin.**

`draw` her frame aynı satırları çalıştırıyor. Bazen yazı, bazen daire çizmesini nasıl sağlarız?



Günlük hayatta hep yapıyoruz

**Yağmur yağıyorsa
şemsiye al.
Yağmıyorsa alma.**

Koşullu ifade (conditional statement)

Belirli bir koşul doğruysa (`true`) bir kod bloğunu, değilse başka bir bloğu çalıştıran yapıdır.

Bir **soru** var, iki **yol** var. Program cevaba göre yollardan birini seçiyor.

Yazımı: `if`, `else if`, `else`

sketch.js

Kopyala

Büyüt

```
1 if (koşul) {  
2   // koşul doğruysa çalışır  
3 } else if (baskaKoşul) {  
4   // ilk koşul yanlış, bu doğruysa çalışır  
5 } else {  
6   // hiçbirisi doğru değilse çalışır  
7 }
```

- Koşul parantez içinde yazılır ve bir boolean üretir.
- Süslü parantezler `{ }` bloğun sınırını belirler.
- Yukarıdan aşağı bakılır; **ilk doğru** dal çalışır, diğerleri atlanır.
- `else if` ve `else` zorunlu değildir.

Sıcak mı, serin mi?

sketch.js

Kopyala

Düzenle

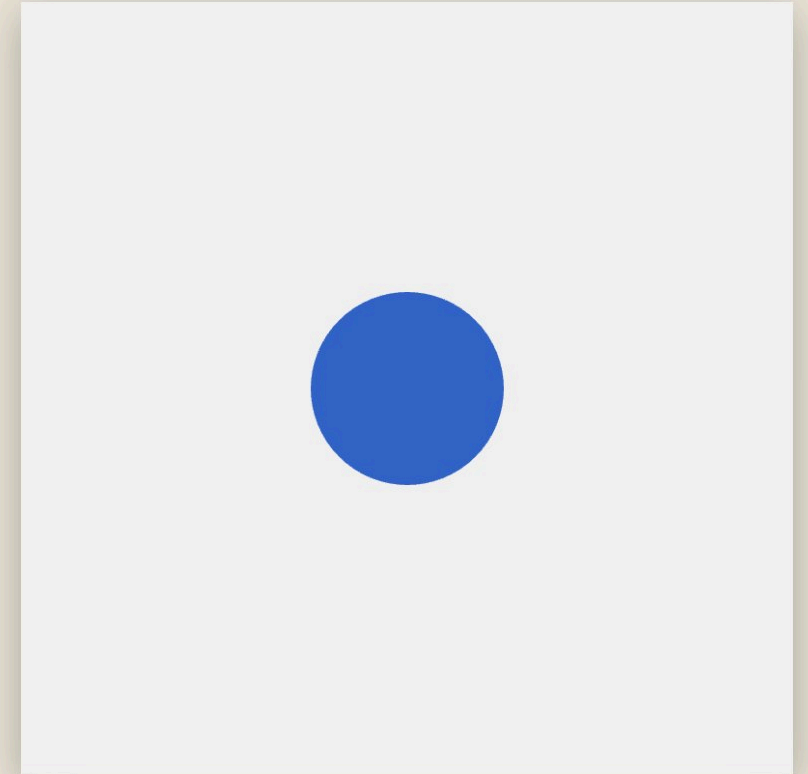
Büyüt

```
1 let sıcaklik; // Değişken – draw dışında tanımlandı (setup'ta değer verilecek)
2
3 function setup() {
4   createCanvas(400, 400);
5   sıcaklik = 18; // Başlangıç değeri
6 }
7
8 function draw() {
9   background(240);
10
11   if (sıcaklik > 25) {
12     fill(200, 80, 50); // Sıcak – turuncu
13   } else {
14     fill(50, 100, 200); // Serin – mavi
15   }
16
17   noStroke();
18   ellipse(200, 200, 100, 100);
19 }
```

● Çıktı

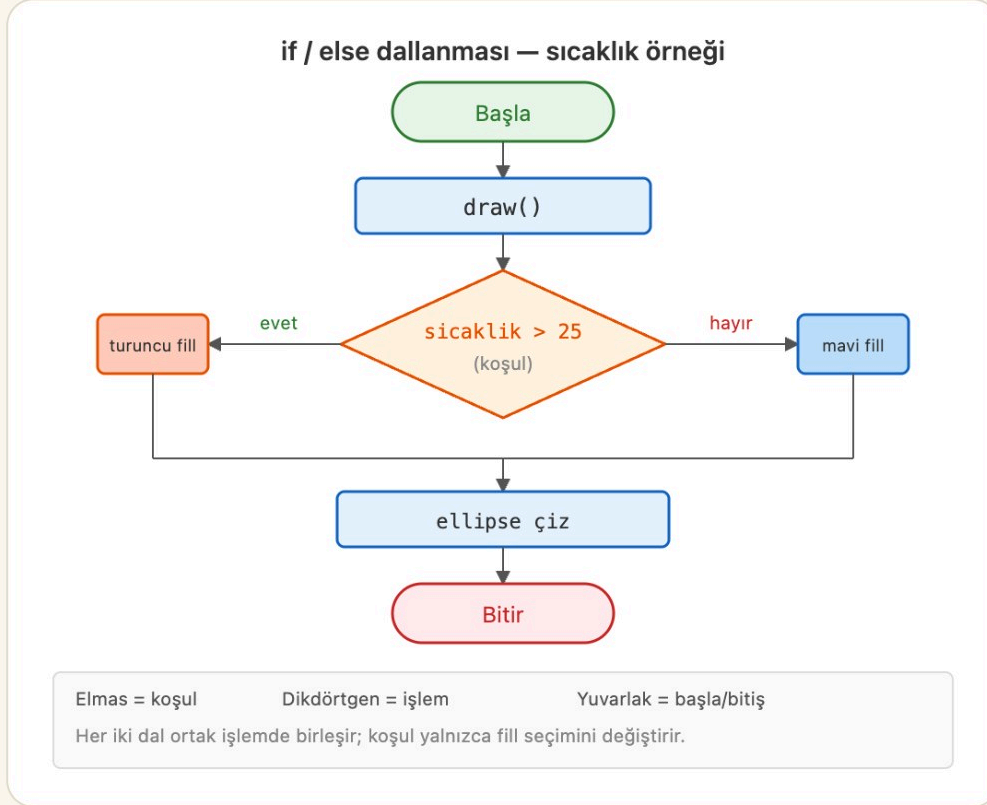
Çalıştır

Büyüt



“Düzenle” ile `sıcaklik = 18` değerini 30 yapın. Sonra tam 25 yapın: hangi renk çıkar, neden?

Kararı çizerek göstermek: akış şeması



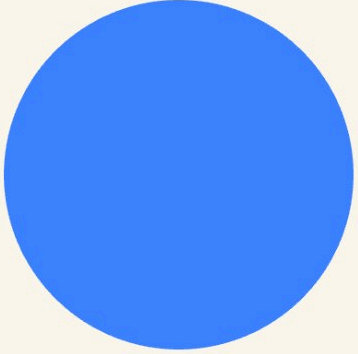
Simge	Anlamı	Bu örnekte
Yuvarlak	Başla / bitiş	Şemanın girişi ve çıkışı
Elmas	Koşul sorusu	sıcaklık > 25
Dikdörtgen	Yapılacak işlem	fill , ellipse

Şemada yolu izleyin

● Çıktı

Çalıştır

Büyüt



sicaklik = 18



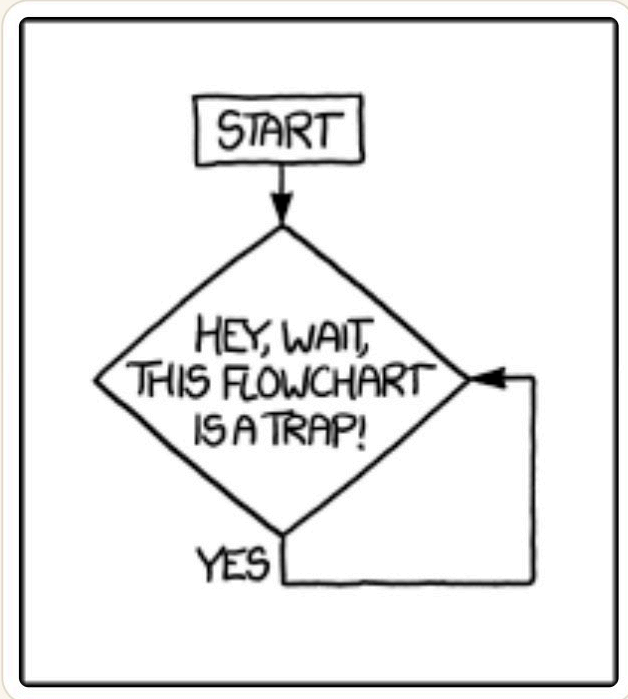
0 25 40

```
graph TD; A[Başla] --> B{sicaklik > 25}; B -- evet --> C(fill(turuncu)); B -- hayır --> D(fill(mavi)); C --> E(ellipse çiz); D --> E; E --> F[ ];
```

The flowchart starts with a red rounded rectangle labeled 'Başla'. It leads to a yellow diamond decision box labeled 'sicaklik > 25'. From the diamond, a grey arrow labeled 'evet' points to a light blue rounded rectangle labeled 'fill(turuncu)'. A red arrow labeled 'hayır' points to a red rounded rectangle labeled 'fill(mavi)'. Both 'fill(turuncu)' and 'fill(mavi)' lead to a red rounded rectangle labeled 'ellipse çiz'. The flowchart ends with an arrow pointing to the right.

Fareyi alttaki çubuğun üstünde gezdirin. Program her seferinde iki yoldan yalnızca birini izler; 25'in tam üstünde ne oluyor?

Her akış şeması bir çıkışa varmalı



Kaynak: xkcd.com/1195, Randall Munroe (CC BY-NC 2.5)

Elmasın içinde şu yazıyor: “Bir dakika, bu akış şeması bir tuzak!”

- Tek cevap “evet” ve ok yine aynı elmasa dönüyor.
- “Hayır” dalı yok, bitiş yok.

Kendi şemanızı çizerken

Her elmastan iki ok çıkmalı ve her yol sonunda bitişe ulaşmalı.

SORU

İki kod da aynı çalışıyor. Hangisini okumak daha kolay?

A

Kopyala

Büyüt

```
if (!oyunBitti) {  
    if (skor >= HEDEF_SKOR) {  
        oyunBitti = true;  
    }  
}
```

B

Kopyala

Büyüt

```
if (!oyunBitti) {  
    if (skor >= HEDEF_SKOR) {  
        oyunBitti = true;  
    }  
}
```

Girinti (indentation)

- Blok içindeki satırları içeri almaktır.
- Hangi satırın hangi bloğa ait olduğu gözüyle görülür.
- İç içe koşullarda girinti yoksa iz çabuk kaybolur.
- Örneklerimizin hepsinde blok gövdesi girintilidir.

JavaScript'te zorunlu değil

Girinti bozuk olsa da kod çoğu zaman çalışır. Hata sessizce okunabilirlikte birikir. "Çalışıyor" demek yetmez.

Python'da zorunlu

Orada girinti sözdiziminin parçasıdır. Yanlışsa program hiç çalışmaz: `IndentationError`.

Girintiyi kim için yapıyoruz?

Çalışıyor işte. Girinti kimin umurunda?



bugün

Bu `}` hangi `if` 'i kapatıyor? Bu kodu kim yazmış böyle?



Üç ay sonra

...ben yazmışım.



dosyanın tarihine bakınca

Kodu en çok okuyacak kişi gelecekteki sizsiniz.

Boşluk mu, Tab mı?

Boşluk (space)

Klavyedeki uzun boşluk çubuğu. Çoğu proje ve p5.js Web Editor 2 ya da 4 boşluk kullanır.

Sekme (Tab)

`Tab` tuşu; genellikle Q'nun solunda, Caps Lock'un üstünde. Editörler bu tuşa basınca boşluk üretecek şekilde ayarlanabilir.

İkisini karıştırmayın

Aynı dosyada hem boşluk hem Tab kullanmak, başka editörlerde satırların kaymış görünmesine yol açar. Birini seçin ve hep onu kullanın.

SORU

Trafik ışığında üç durum var.
if / else yalnızca iki yol veriyor.

Kırmızı, sarı, yeşil. Üçüncü yolu nereye koyacağız?



else if : araya yeni bir soru

sketch.js · satır 13-34

Kopyala

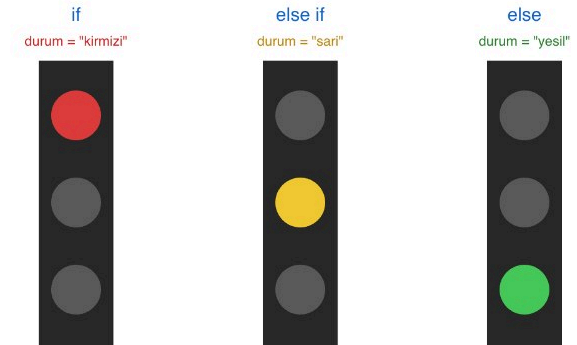
Büyüt

```
13 let durum = "kirmizi";
14 fill(21, 101, 192);
15 text("if", 130, 78);
16 textSize(11);
17 fill(198, 40, 40);
18 text('durum = "kirmizi"', 130, 98);
19 fill(40);
20 rect(100, 110, 60, 230);
21 fill(90);
22 ellipse(130, 154, 40, 40);
23 ellipse(130, 224, 40, 40);
24 ellipse(130, 294, 40, 40);
25 if (durum === "kirmizi") {
26   fill(220, 60, 60);
27   ellipse(130, 154, 40, 40);
28 } else if (durum === "sari") {
29   fill(240, 200, 50);
30   ellipse(130, 224, 40, 40);
31 } else {
32   fill(70, 200, 90);
33   ellipse(130, 294, 40, 40);
34 }
```

Çıktı

Çalıştır

Büyüt



Zincir nasıl çalışır?

- 1 `durum === "kirmizi"` mı? Evetse kırmızıyı yak, zincirden çık.
- 2 Değilse: `durum === "sari"` mı? Evetse sarıyı yak, zincirden çık.
- 3 İkisi de değilse: `else` çalışır, yeşili yak.

Her seferinde yalnızca bir dal

Zincir, uyan ilk dalı çalıştırır. Sonrakilere bakılmaz.

Örnekteki numara

Önce üç lamba da gri çizilir. Zincir yalnızca uygun lambayı yeniden boyar.

Oyunlar sürekli karar verir

Oyun	Koşul	Sonuç
Super Mario	1-Up mantarı alındı	Can bir artar
Pac-Man	Hayaletle çarpışma var ve güç hapi aktif değil	Bir can gider
Tetris	Bir satır tamamen doldu	Satır silinir, skor artar
Futbol oyunu	Top kale çizgisini geçti	Gol; skor güncellenir
Candy Crush	Üç ya da daha fazla aynı renk yan yana	Taşlar patlar, puan gelir

Aynı iskelet birazdan yazacağımız Daire Tıklama oyununda da var: isabet varsa skor artar, hedef skora ulaşılmca oyun biter.

Can barı: `else if` zinciri iş başında

● Çıktı

Çalıştır

Büyüt

`can = 80``if (can > 50)` → yeşil bar`else if (can > 20)` → sarı bar`else if (can > 0)` → kırmızı bar, yanıp söner`else` → bar boş: bayıldı

Fareyi sağa sola götürün. Pokémon'dan dövüş oyunlarına kadar can barlarının renk değiştirmesi böyle bir zincirdir; her an yalnızca bir dal çalışır.

Sevdiğiniz oyundan üç karar

 10 dk

1. Bildiğiniz bir oyundan üç durum seçin.
2. Her birini kısa bir koşul cümlesi olarak yazın: "... ise ... olur."
3. Birini seçip küçük bir `if / else` iskeletine ya da akış şemasına dönüştürün.

SORU

Skor yalnızca can varken ve oyun bitmemişken artsın.

İki ayrı koşul var. Bunu kaç farklı biçimde yazabilirsiniz?



İç içe if ya da tek if + &&

İç içe

Kopyala

Büyüt

```
1 if (can > 0) {  
2     if (!oyunBitti) {  
3         skor = skor + 1;  
4     }  
5 }
```

Birleşik koşul

Kopyala

Büyüt

```
1 if (can > 0 && !oyunBitti) {  
2     skor = skor + 1;  
3 }
```

İç içe koşul (nested conditional)

Bir `if` bloğunun içine başka bir `if` yazmaktır. Önce dış koşul, doğruysa iç koşul sorulur.

İki yazım da aynı kararı verir. Arada başka iş yoksa birleşik koşul daha sadedir.

İki yazım, aynı akış

Çıktı

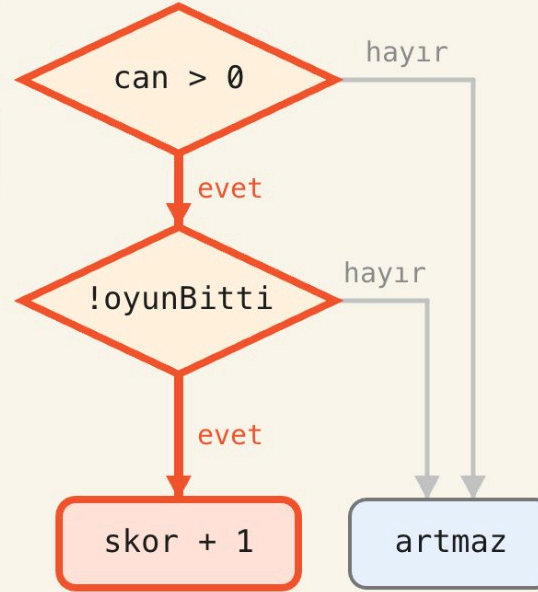
Çalıştır

Büyüt

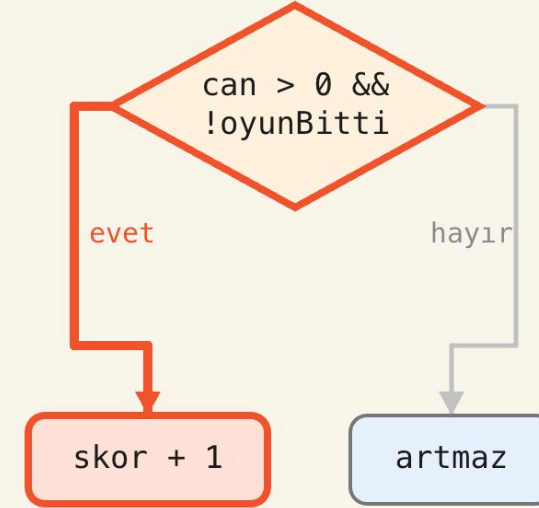
Tıklayın:

`can > 0`
`true``oyunBitti`
`false`

İç içe if



Tek if + &&



İki şemada da sonuç aynı: skor artar

Dört kombinasyonu da deneyin. Turuncu yol, programın o değerlerle izlediği yoldur.

Her dalda farklı mesaj: üç çıkış

sketch.js

Kopyala

Büyüt

```
1 if (can > 0) {  
2   if (!oyunBitti) {  
3     skor = skor + 1;  
4     mesaj = "Devam";  
5   } else {  
6     mesaj = "Oyun bitti";  
7   }  
8 } else {  
9   mesaj = "Can yok";  
10 }
```

Üç yol var: devam, oyun bitti, can yok.

Neden tek `&&` yetmiyor?

Burada `else` 'ler de iş yapıyor. Tek bir birleşik koşul yalnızca iki yol verir.

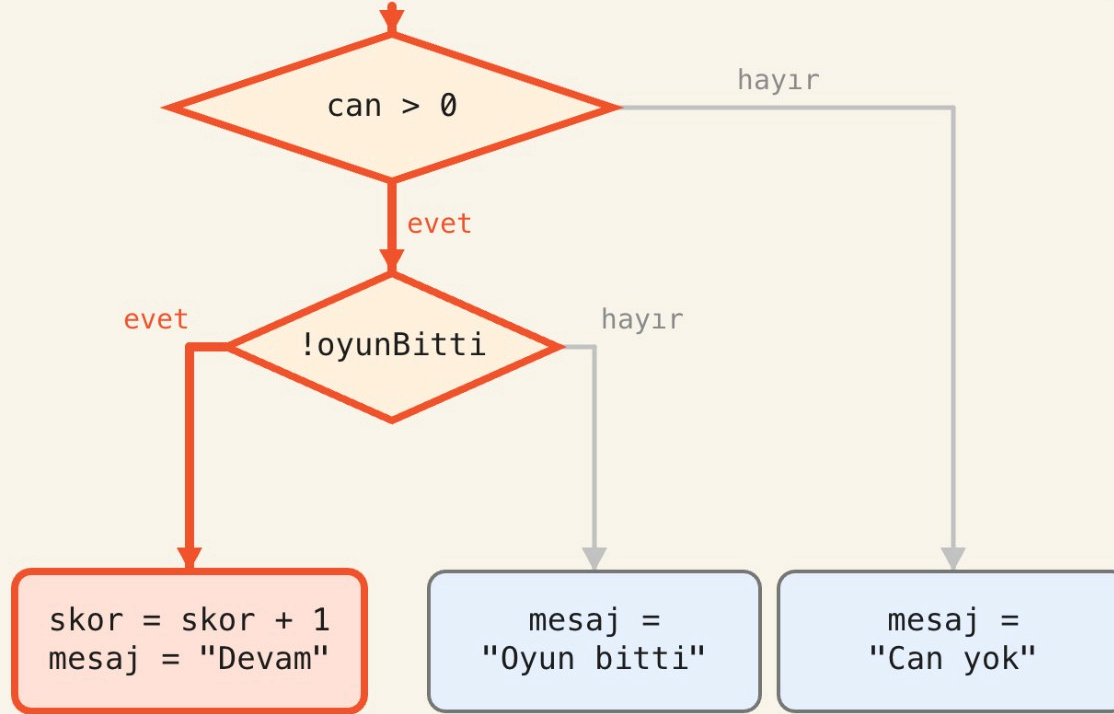
Üç çıkışlı akış: yolu izleyin

• Çıktı

Çalıştır

Büyüt

Tıklayın:

`can > 0
true``oyunBitti
false`

mesaj: Devam

Önce tahmin edin: `can > 0` false iken `oyunBitti` 'yi değiştirmek sonucu etkiler mi?

Dışarıda birleşik koşul, içeride skor eşiği

sketch.js

Kopyala

Büyüt

```
1 if (can > 0 && !oyunBitti) {  
2   if (skor >= 10) {  
3     mesaj = "Seviye atlandı";  
4     oyunBitti = true;  
5   } else {  
6     mesaj = "Skor yetersiz";  
7   }  
8 } else {  
9   mesaj = "Oyuna devam edilemez";  
10 }
```

Her düzeydeki koşul tek bir karşılaştırma olmak zorunda değil.

Birleştirilebilir

Dış ya da iç koşul `&&`, `||`, `!` ile kurulmuş bir ifade olabilir.

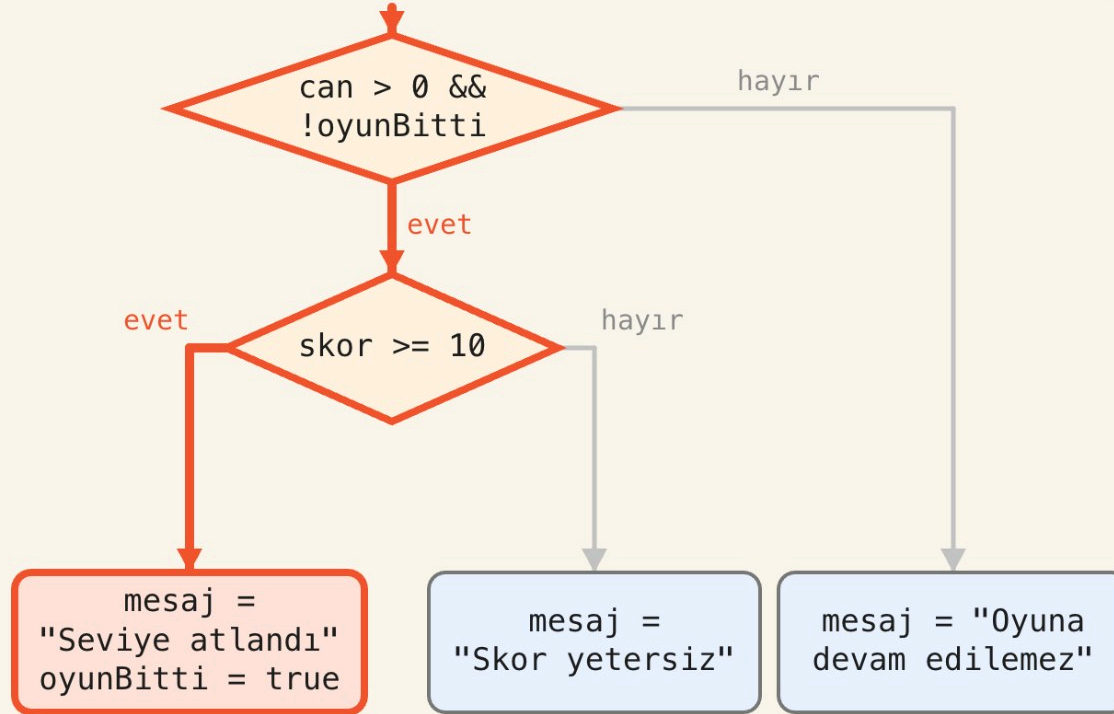
Birleşik koşullu akış: yolu izleyin

• Çıktı

Çalıştır

Büyüt

Tıklayın:

`can > 0
true``oyunBitti
false``skor >= 10
true`

mesaj: Seviye atlandı

Üç düğme, sekiz kombinasyon. Kaç tanesi "Seviye atlandı" mesajına ulaşır?

İki küçük not

Boş `else` yazmayın

`else` yalnızca gerçekten yapılacak bir iş varsa eklenir. `else { }` gereksizdir.

`||` için de benzer eşdeğerlik var

“Can bitti ya da süre bitti → oyunu bitir” kuralı art arda iki `if` ile ya da `if (can === 0 || sure === 0)` ile yazılabilir.

Konsolda hangi satırlar, hangi sırayla görünür?

sketch.js

Kopyala

Büyüt

```
1 let x = 15;
2 let y = 8;
3 let aktif = true;
4 let seviye = 2;
5
6 console.log("basla");
7
8 if (x > 10 && y < 10) {
9   console.log("bolge-icinde");
10 } else {
11   console.log("bolge-disinda");
12 }
13
```

```
14 if (!aktif || seviye === 1) {
15   console.log("beklet");
16 } else if (seviye >= 2 && x === 15) {
17   console.log("ilerle");
18 } else {
19   console.log("bekle");
20 }
21
22 let sonuc = (x >= 10 && y !== 0) || !aktif;
23 console.log(sonuc);
```

Adım adım izleyelim

sketch.js

Kopyala

Düzenle

Büyüt

```
1 let x = 15;
2 let y = 8;
3 let aktif = true;
4 let seviye = 2;
5
6 console.log("başla");
7
8 if (x > 10 && y < 10) {
9   console.log("bolge-icinde");
10 } else {
11   console.log("bolge-disinda");
12 }
13
14 if (!aktif || seviye === 1) {
15   console.log("beklet");
16 } else if (seviye >= 2 && x === 15) {
17   console.log("ilerle");
18 } else {
19   console.log("bekle");
20 }
21
22 let sonuc = (x >= 10 && y !== 0) || !aktif;
23 console.log(sonuc);
```

● Konsol

Çalıştır

Büyüt

```
başla
bolge-icinde
ilerle
true
```

Oyun kuralından koda

 15 dk

1. Üç kısa oyun kuralı yazın. Örnek: “can sıfırsa bitir”, “skor 10 olunca seviye atla”.
2. Bir yapay zekâya bu kuralları `if / else if / else` iskeletine çevirmesini söyleyin.
3. Gelen kodu denetleyin: `=` ile `===` doğru mu? `&&` doğru yerde mi? Hatalı atamaları düzeltin.

Kodunuzu okunabilirlik açısından gözden geçirtin

 15-20 dk

1. Bugün yazdığınız bir sketch'i kopyalayın (örneğin sıcaklık rengi).
2. Bir yapay zekâya şunu sorun: girinti tutarlı mı, değişken adları anlamlı mı (camelCase), sabitler BÜYÜK_HARF mi, gereksiz karmaşık yer var mı?
3. Kısa, madde madde öneri isteyin. "Kodun mantığını değiştirme" deyin.
4. Önerileri uygulayıp yeniden çalıştırın. Davranış aynı kalmalı; yalnızca okunuş iyileşmeli.

Kod zamanı

 25 dk

1. Sıcaklık örneğine üçüncü bir renk ekleyin: 10'un altı mavi, 10–25 arası yeşil, 25'in üstü turuncu.
2. Kayan daire tuvalin sağ yarısına geçince rengi değişsin.
3. Yanıp sönen ışık: bir daire bir saniye kırmızı, bir saniye yeşil olsun. İpucu: `frameCount % 120`.
4. **Zorlu:** İlk dersteki söz. Top sağa gitsin; kenara değince yönünü tersine çevirsin. İpucu: `x` ve `hiz` diye iki değişken.

skor = 10 iken konsola ne yazılır?

sketch.js

Kopyala

Büyüt

```
1 if (skor > 5) {  
2   console.log("A");  
3 } else if (skor > 8) {  
4   console.log("B");  
5 } else {  
6   console.log("C");  
7 }
```

☒ Yalnızca A☐ B A ve B☐ C Yalnızca B☐ D C

İlk doğru dal çalışır, zincir biter. İkinci koşul da doğru ama ona hiç bakılmaz. Sıralama önemli.

Bu kodda sinsi bir hata var. Ne olur?

sketch.js

Kopyala

Büyüt

```
1 let skor = 3;  
2  
3 if (skor = 10) {  
4   console.log("Kazandın!");  
5 }
```

A Hiçbir şey yazılmaz.

☒ "Kazandın!" yazılır ve skor 10 olur.

C Program hata verip durur.

Tek `=` atamadır: skor kutusuna 10 koyar. Koşul da doğru sayılır. Soru sormak için `===`.

can = 0, oyunBitti = true. mesaj ne olur?

sketch.js

Kopyala

Büyüt

```
1 if (can > 0) {  
2   if (!oyunBitti) {  
3     skor = skor + 1;  
4     mesaj = "Devam";  
5   } else {  
6     mesaj = "Oyun bitti";  
7   }  
8 } else {  
9   mesaj = "Can yok";  
10 }
```

A "Devam"

B "Oyun bitti"

✓ "Can yok"

Dış koşul `can > 0` yanlış. İçerideki `if` 'e hiç girilmez; dıştaki `else` çalışır.

Merak edenler için kaynaklar

PATT VİRA

if, else if, else

Koşullu ifadeleri görsel örneklerle anlatan ders.

[pattvira.com > intro-to-creative-coding > conditionals-if-else-if-else](https://pattvira.com/intro-to-creative-coding/conditionals-if-else-if-else)

THE CODING TRAIN

Conditional Statements

if ile ilk kararlar.

[thecodingtrain.com > 3-conditionals > 1-conditionals](https://thecodingtrain.com/3-conditionals/1-conditionals)

THE CODING TRAIN

else, else if, &&, ||

Zincir ve birleşik koşullar tek videoda.

[thecodingtrain.com > 3-conditionals > 3-else-if-and-or](https://thecodingtrain.com/3-conditionals/3-else-if-and-or)

HAPPY CODING

If Statements

Yazılı anlatım ve denenebilir örnekler (İngilizce).

[happycoding.io > tutorials > p5js > if-statements](https://happycoding.io/tutorials/p5js/if-statements)

Bugün ne öğrendik?

- ✓ `if` bir soru sorar; doğruysa bloğu çalışır, değilse `else`.
- ✓ `else if` ile ikiden fazla yol kurulur; uyan ilk dal çalışır.
- ✓ Girinti zorunlu değil ama okunabilirlik için şart. Boşluk ile Tab karışmaz.
- ✓ Akış şeması: yuvarlak başla/bitiş, elmas koşul, dikdörtgen işlem.
- ✓ İç içe `if` ile `&&` çoğu zaman aynı kararı verir.
- ✓ `if (skor = 10)` bir sorudur sanmayın; atamadır.

SIRADAKİ DERS

İlk oyunumuz

Fare, mesafe ve Daire Tıklama.

Sonraki sunum →

← Önceki

