

Koordinat sistemi, şekiller ve renkler

Bir şekli tuvalde tam istediğimiz yere, istediğimiz renkte çizmeyi öğreneceğiz.

(0, 0) nerede?

rect, ellipse, line

fill ve stroke

RGB ve alfa

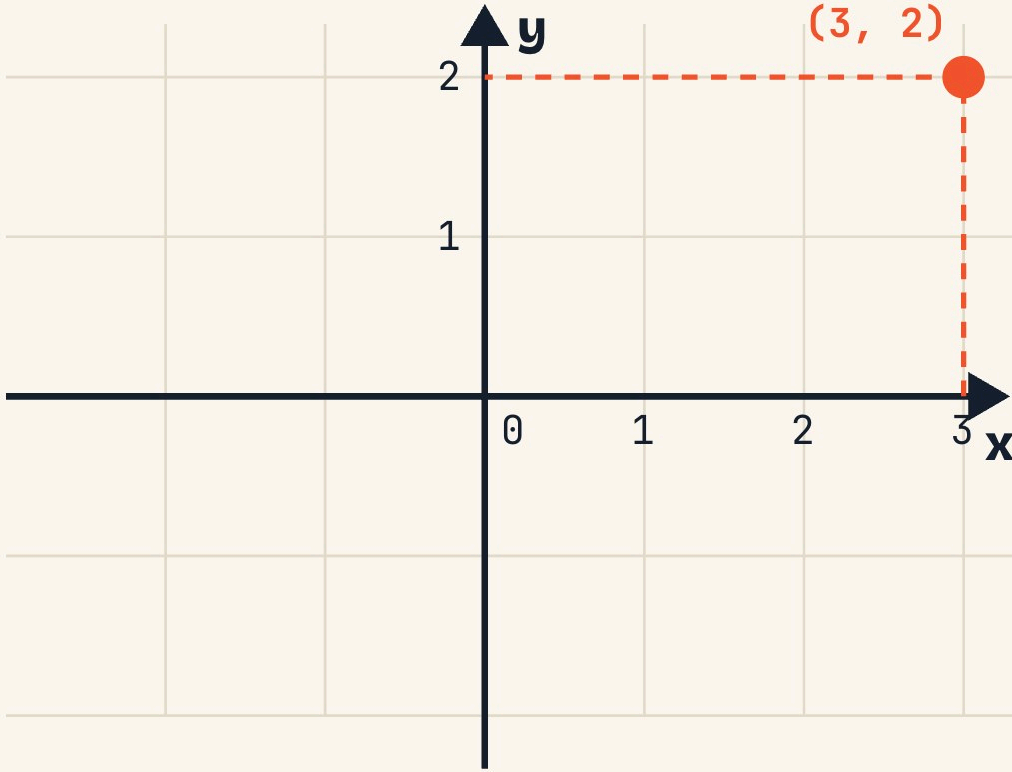
SORU

Matematik dersinde $(3, 2)$ noktasını nasıl buluyordunuz?

Önce hangi yöne, sonra hangi yöne gidiyordunuz? $(0, 0)$ neredeydi?



Hatırlayalım: matematikteki koordinat düzlemi



- $(0, 0)$ noktası, yani **orijin**, ortada.
- **x** sağa doğru artar.
- **y yukarı** doğru artar.
- $(3, 2)$: üç birim sağa, iki birim yukarı.

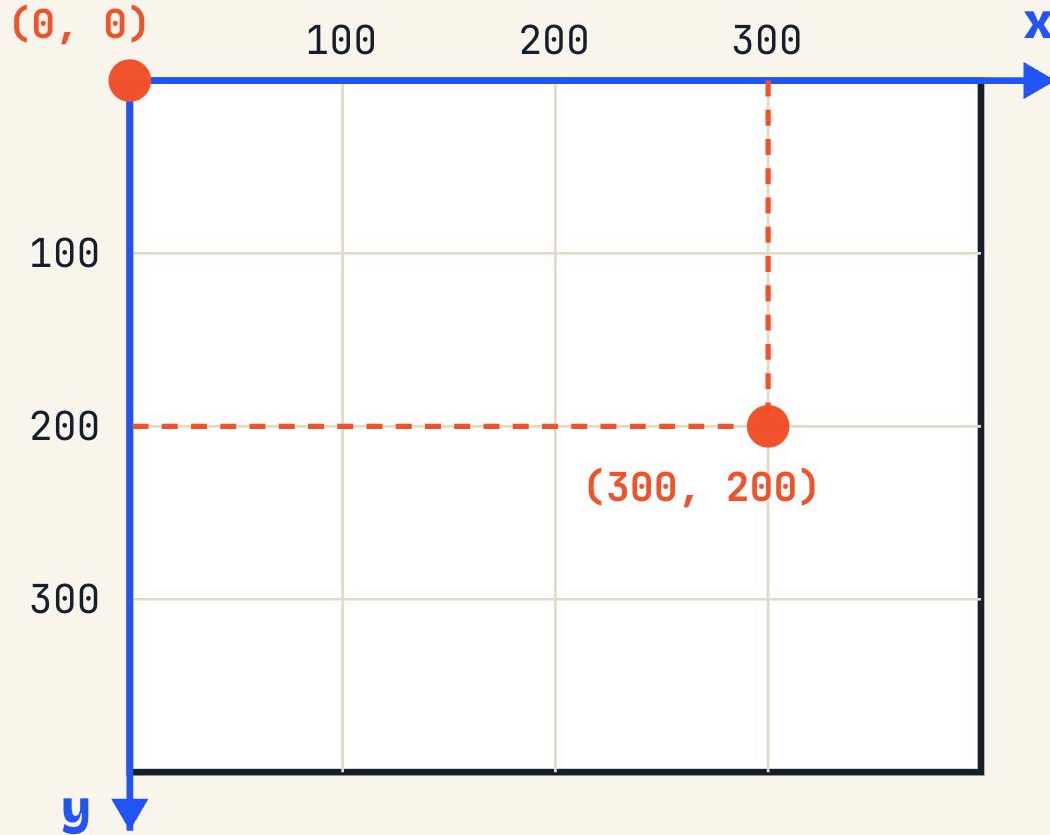
SORU

Peki ekranda (0, 0) nerede?

Bir tahmin yürütün: ortada mı, sol altta mı, başka bir yerde mi? y
hangi yöne artıyor olabilir?



p5.js tuvalinde y aşıağı doęru artar



- Orijin (0, 0) **sol üst köşede**.
- **x** saęa doęru artar. Burası aynı.
- **y aşıağı** doęru artar. Fark burada.

Neden böyle?

Ekrandaki piksel satırları yukarıdan aşıağı numaralanır. Ekranı çizen kütüphanelerin çoęunda düzen budur.

iki dünya, iki y eksenini

y büyüdükçe yukarı çıkarız.



matematik dersi

y büyüdükçe aşağı ineriz.



ekran

Topu yukarı zıplatmak için y'yi...
azaltacağım?!



her yeni oyun programcısı

Bu şaşkınlık bir iki haftada geçer. O zamana kadar şekil ters yöne giderse ilk bakılacak yer y'nin işaretidir.

Neden sol üst? Ekran resmi böyle boyar



Eski tüplü ekranlar görüntüyü soldan sağa, yukarıdan aşağıya satır satır tarardı; tıpkı yazı yazar gibi. İlk boyanan nokta sol üst köşe olduğu için sayma da oradan başlar.

Fareyi tuvalde gezdirin, koordinatı okuyun

sketch.js

Kopyala

Düzenle

Büyüt

```
1 function setup() {  
2   createCanvas(400, 300);  
3 }  
4  
5 function draw() {  
6   background(250);  
7   stroke(200);  
8   line(mouseX, 0, mouseX, height);  
9   line(0, mouseY, width, mouseY);  
10  
11  noStroke();  
12  fill(230, 70, 40);  
13  ellipse(mouseX, mouseY, 14, 14);  
14  
15  fill(30);  
16  textSize(20);  
17  text("x = " + mouseX, 12, 26);  
18  text("y = " + mouseY, 12, 52);  
19 }
```

Çıktı

Çalıştır

Büyüt

```
x = 0  
y = 0
```

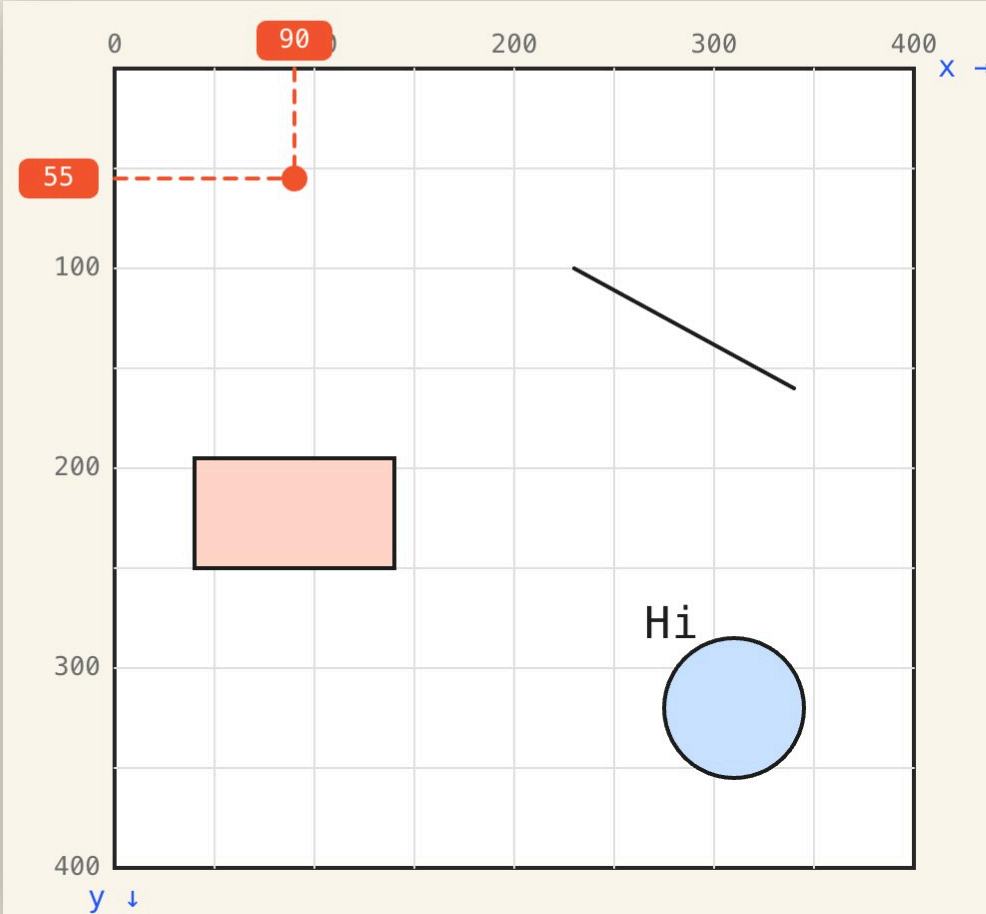
Kodun kendisi şimdilik önemli değil. Sol üst köşeye gidin: x ve y kaç? Sağ alt köşede?

Tuvalde koordinat okumak

Çıktı

Çalıştır

Büyüt



```
point(90, 55)
```

(x, y): noktanın kendisi

```
rect(40, 195, 100, 55)
```

(x, y): sol üst köşe

```
ellipse(310, 320, 70, 70)
```

(x, y): merkez

```
line(230, 100, 340, 160)
```

(x, y): başlangıç ve bitiş

```
text("Hi", 265, 285)
```

(x, y): yazının sol-alt ucu

Bir komutun üstüne gelin.

Turuncu nokta, komuttaki `(x, y)` çiftinin tuvaldeki yeridir; kesikli çizgiler aynı sayıları eksenlerde gösterir.

Görselden beş komut

Komut	(x, y) neresi?	Görseldeki örnek
<code>point(x, y)</code>	Noktanın kendisi	<code>point(90, 55)</code>
<code>rect(x, y, w, h)</code>	Sol üst köşe ; w genişlik, h yükseklik	<code>rect(40, 195, 100, 55)</code>
<code>ellipse(x, y, w, h)</code>	Merkez ; w ve h çap	<code>ellipse(310, 320, 70, 70)</code>
<code>line(x1, y1, x2, y2)</code>	Başlangıç ve bitiş noktası	<code>line(230, 100, 340, 160)</code>
<code>text(metin, x, y)</code>	Yazının sol-alt ucu	<code>text("Hi", 265, 285)</code>

Genişlik ve yükseklik nasıl bulunur?

Eksenlerdeki farktan: $w = 140 - 40 = 100$, $h = 250 - 195 = 55$.

Şekil çizen komutlar

Komut	Ne çizer?	Örnek	Örnekte ne olur?
<code>rect(x, y, w, h)</code>	Dikdörtgen	<code>rect(50, 50, 100, 80)</code>	Sol üst köşesi (50, 50); 100 piksel geniş, 80 piksel yüksek.
<code>ellipse(x, y, w, h)</code>	Elips ya da daire	<code>ellipse(200, 200, 60, 60)</code>	Merkezi (200, 200); çapı 60 olan daire.
<code>line(x1, y1, x2, y2)</code>	Doğru parçası	<code>line(0, 0, 400, 400)</code>	Sol üstten sağ alta bir köşegen.
<code>point(x, y)</code>	Tek nokta	<code>point(100, 150)</code>	(100, 150) konumuna bir nokta.
<code>text(metin, x, y)</code>	Yazı	<code>text("Skor: 7", 20, 40)</code>	Yazı (20, 40) noktasından başlar; hiza sol-alt.

Oyun karakterleri de şekillerden yapılır

● Çıktı

Çalıştır Büyüt



```
✓ ellipse(190, 150, 170, 170)
✓ rect(105, 150, 170, 110)

ellipse(133, 260, 57, 40) × 3
ellipse(155, 140, 46, 56) × 2
ellipse(163, 146, 22, 22) × 2
line(150, 205, 230, 205)
```

Altı komut, bir hayalet. İlk oyunların karakterleri birkaç dikdörtgen ve daireden ibaretti; bugün de prototipler böyle başlar.

SORU

İkisi de (0, 0). Aynı yerde mi görünürler?

sketch.js

Kopyala

Büyüt

```
rect(0, 0, 80, 80);  
ellipse(0, 0, 80, 80);
```

Karenin ne kadarı tuvalin içinde? Dairenin ne kadarı?

rect köşeden, ellipse merkezden çizer

sketch.js

Kopyala

Düzenle

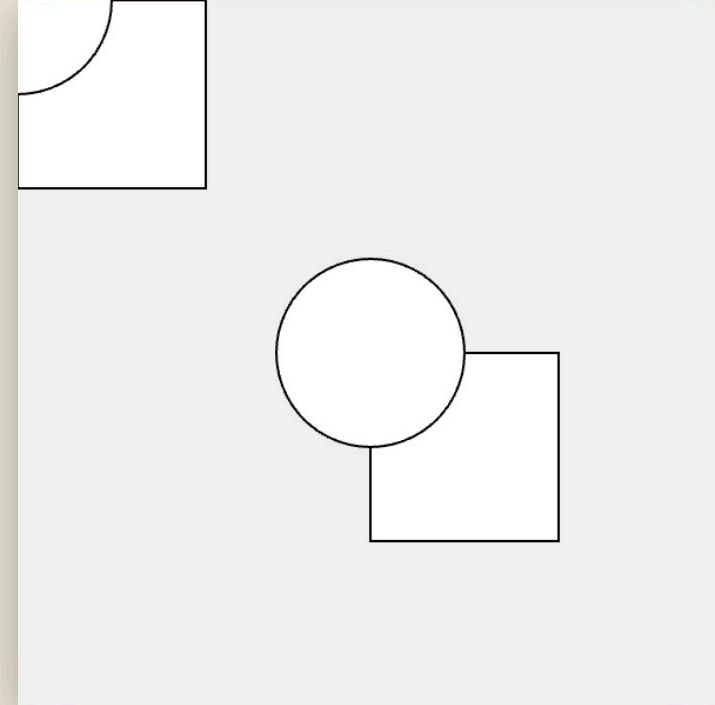
Büyüt

```
1 function setup() {  
2   createCanvas(300, 300);  
3 }  
4  
5 function draw() {  
6   background(240);  
7  
8   rect(0, 0, 80, 80);  
9   ellipse(0, 0, 80, 80);  
10  
11   // Aynı sayılar, tuvalin ortasında  
12   rect(150, 150, 80, 80);  
13   ellipse(150, 150, 80, 80);  
14 }
```

Çıktı

Çalıştır

Büyüt



En sık konum hatası

Karenin tamamı içeride. Dairenin ise yalnızca çeyreği görünüyor; çünkü (0, 0) onun merkezi.

Görünümü ayarlayan komutlar

Komut	Ne yapar?	Örnek
<code>fill(r, g, b)</code>	Sonraki şekillerin iç dolgu rengini ayarlar.	<code>fill(255, 0, 0)</code> → içi kırmızı
<code>stroke(r, g, b)</code>	Sonraki şekillerin kenar çizgisi rengini ayarlar.	<code>stroke(0)</code> → kenar siyah
<code>strokeWeight(n)</code>	Kenar kalınlığını piksel olarak ayarlar.	<code>strokeWeight(2)</code>
<code>noFill()</code>	Dolguyu kapatır; yalnızca kenar çizilir.	İçi boş çerçeve
<code>noStroke()</code>	Kenarı kapatır; yalnızca dolgu görünür.	Kenarsız şekil

Bu komutlar kendileri bir şey çizmez. Kendilerinden **sonra** çizilen şekillerin görünümünü belirlerler.

SORU

İkisi de 100 × 100. Hangisi daha büyük görünür?

sketch.js

Kopyala

Büyüt

```
strokeWeight(1);  
rect(100, 100, 100, 100);  
  
strokeWeight(50);  
rect(100, 300, 100, 100);
```

strokeWeight : kenar, sınırın ortasına oturur

sketch.js

Kopyala

Düzenle

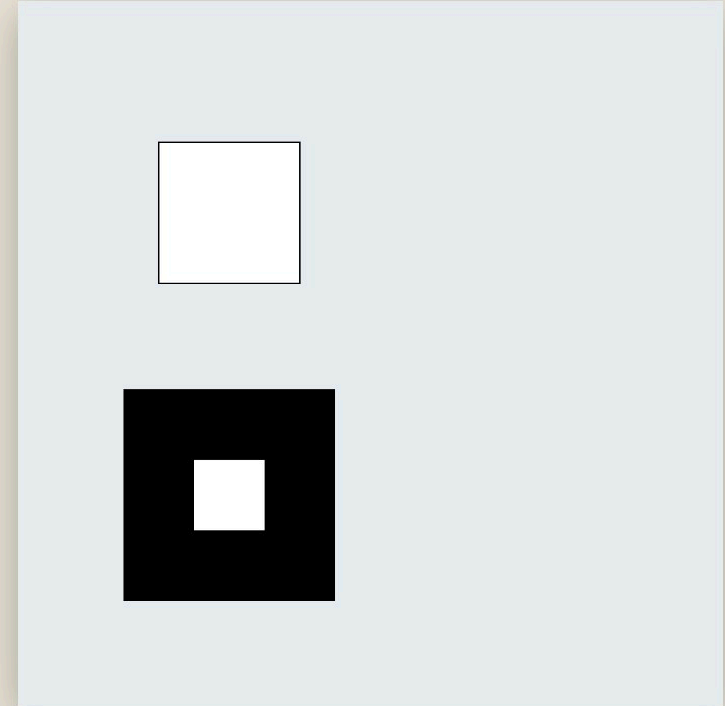
Büyüt

```
1 function setup() {  
2   createCanvas(500, 500);  
3 }  
4  
5 function draw() {  
6   background(230, 235, 240);  
7  
8   strokeWeight(1);  
9   rect(100, 100, 100, 100);  
10  
11  strokeWeight(50);  
12  rect(100, 300, 100, 100);  
13 }
```

Çıktı

Çalıştır

Büyüt



Neden alttaki şişti?

Kenarın yarısı sınırın içinde, yarısı dışında kalır. Kalınlık 50 olunca her yöne yaklaşık 25 piksel taşar.

Aynı etki daire ve dikdörtgende de var

sketch.js · satır 1-17

Kopyala

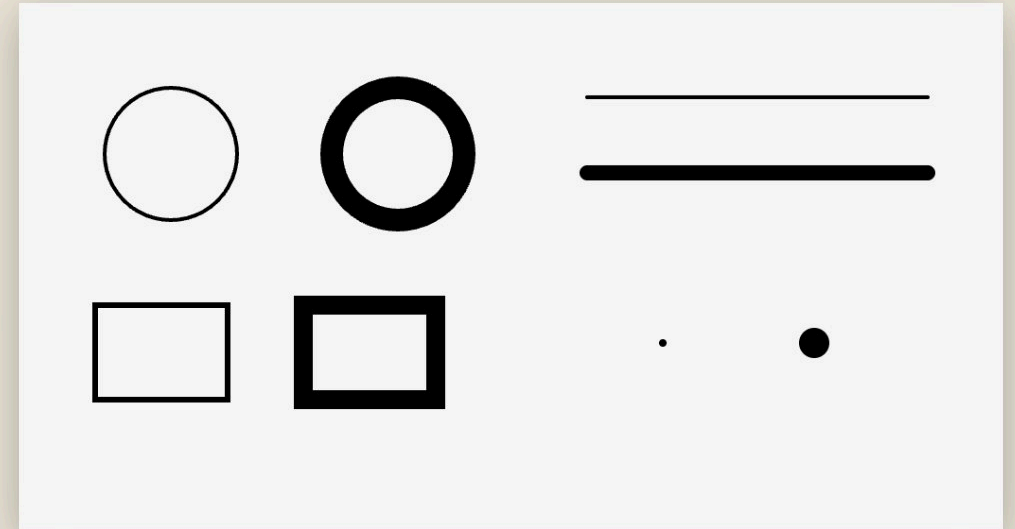
Büyüt

```
1 function setup() {  
2   createCanvas(520, 280);  
3   noLoop();  
4 }  
5  
6 function draw() {  
7   background(245);  
8   noFill();  
9   stroke(0);  
10  
11   strokeWeight(2);  
12   ellipse(80, 80, 70, 70);  
13   strokeWeight(12);  
14   ellipse(200, 80, 70, 70);  
15  
16   strokeWeight(3);  
17   rect(40, 160, 70, 50);
```

● Çıktı

Çalıştır

Büyüt



...çizgide ve noktada da

sketch.js · satır 18-30

Kopyala

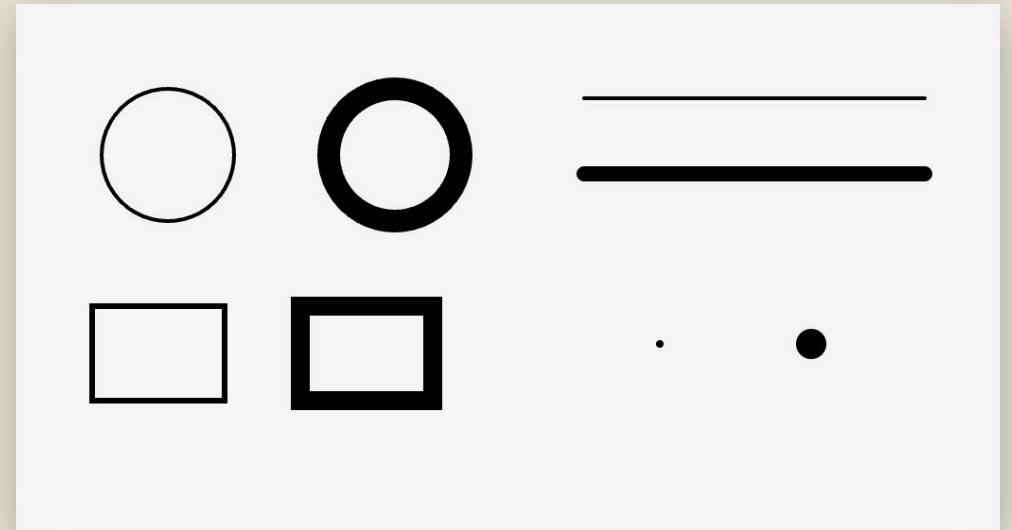
Büyüt

```
18  strokeWeight(10);
19  rect(150, 160, 70, 50);
20
21  strokeWeight(2);
22  line(300, 50, 480, 50);
23  strokeWeight(8);
24  line(300, 90, 480, 90);
25
26  strokeWeight(4);
27  point(340, 180);
28  strokeWeight(16);
29  point(420, 180);
30 }
```

Çıktı

Çalıştır

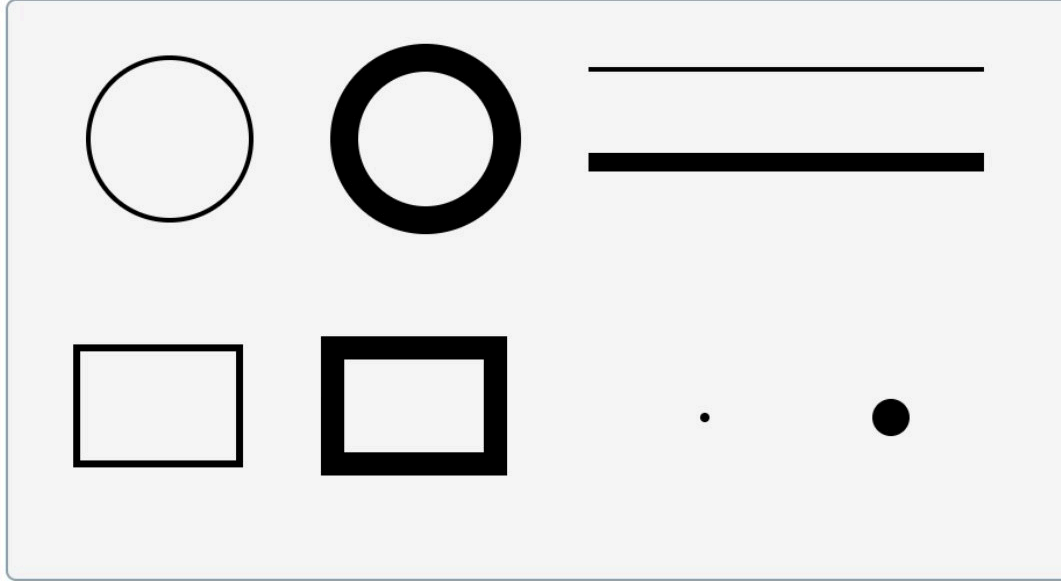
Büyüt



Pikselleri büyütünce

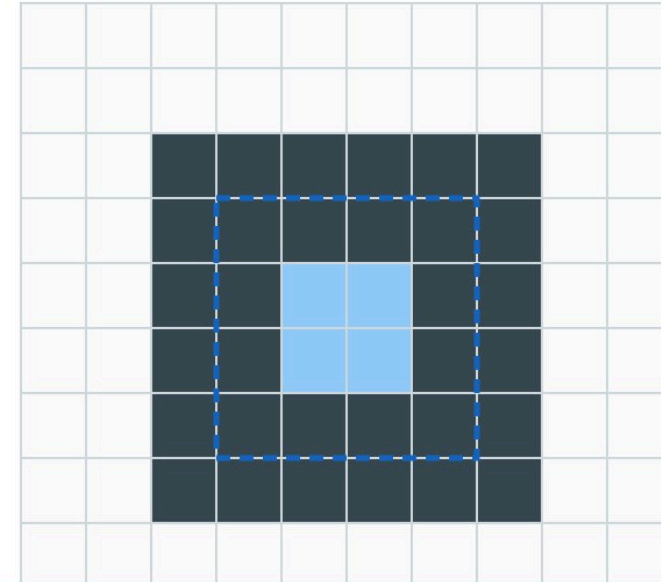
strokeWeight — diğer şekiller ve büyütülmüş piksel

Tuval (kod çıktısı)



Büyütülmüş piksel ızgarası

rect + strokeWeight(2)



dolgu



kenar (stroke)



şeklin boyutu (rect genişlik x yükseklik)

Kenar sınırın ortasına oturur: bir sıra piksel içerde, bir sıra dışarıda. ellipse, line ve point de strokeWeight'ten etkilenir.

Kesik çizgi, verdiğiniz boyut. Koyu kenarın bir sırası içeride, bir sırası dışarıda.

SORU

noFill() ve noStroke() birlikte yazılırsa ne çizilir?

Dolgu yok, kenar da yok. Geriye ne kalır?



Dört kombinasyon (1/2)

sketch.js · satır 1-19

Kopyala

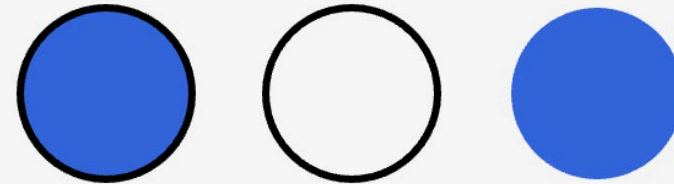
Büyüt

```
1 function setup() {  
2   createCanvas(400, 200);  
3   noLoop();  
4 }  
5  
6 function draw() {  
7   background(245);  
8  
9   // 1) fill() + stroke()  
10  fill(50, 100, 220);  
11  stroke(0);  
12  strokeWeight(3);  
13  ellipse(50, 100, 70, 70);  
14  
15  // 2) noFill() + stroke()  
16  noFill();  
17  stroke(0);  
18  ellipse(150, 100, 70, 70);  
19
```

Çıktı

Çalıştır

Büyüt



`fill` ve `stroke` her zaman şekilden **önce** yazılır.

Dört kombinasyon (2/2)

sketch.js · satır 20-29

Kopyala

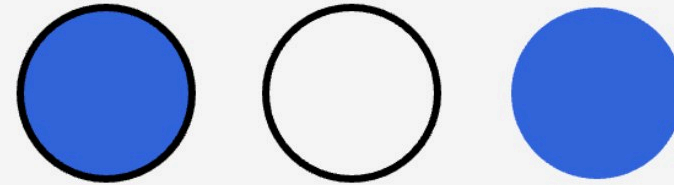
Büyüt

```
20 // 3) fill() + noStroke()
21 fill(50, 100, 220);
22 noStroke();
23 ellipse(250, 100, 70, 70);
24
25 // 4) noFill() + noStroke() → daire görünmez
26 noFill();
27 noStroke();
28 ellipse(350, 100, 70, 70);
29 }
```

● Çıktı

Çalıştır

Büyüt



Dördüncü daire çiziliyor ama görünmüyor: dolgusu da kenarı da kapalı.

Ayar, çizim sırasında değişebilir (1/2)

sketch.js · satır 1-25

Kopyala

Büyüt

```
1 function setup() {  
2   createCanvas(620, 200);  
3   noLoop();  
4 }  
5  
6 function draw() {  
7   background(245);  
8  
9   // ① kırmızı dolgu + kenar  
10  fill(220, 50, 50);  
11  stroke(0);  
12  strokeWeight(2);  
13  rect(20, 70, 70, 50);  
14  
15  // ② yalnız kenar  
16  noFill();  
17  stroke(0);  
18  ellipse(140, 95, 64, 64);  
19  
20  // ③ mavi dolgu, kenarsız  
21  fill(50, 100, 220);  
22  noStroke();  
23  ellipse(230, 95, 64, 64);  
24  
25  // ④ yeşil dolgu + kenar
```

Çıktı

Çalıştır

Büyüt



Ayar, çizim sırasında değişebilir (2/2)

sketch.js · satır 26-46

Kopyala

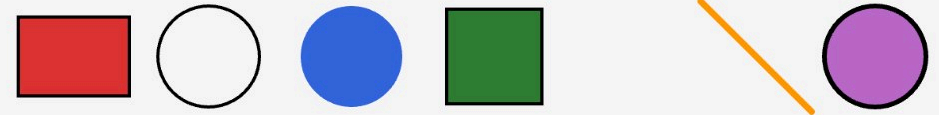
Büyüt

```
26 fill(46, 125, 50);
27 stroke(0);
28 strokeWidth(2);
29 rect(290, 65, 60, 60);
30
31 // ⑤ görünmez
32 noFill();
33 noStroke();
34 ellipse(400, 95, 64, 64);
35
36 // ⑥ turuncu çizgi (line'da fill kullanılmaz)
37 stroke(255, 152, 0);
38 strokeWidth(4);
39 line(450, 60, 520, 130);
40
41 // ⑦ mor dolgu + kenar
42 fill(186, 104, 200);
43 stroke(0);
44 strokeWidth(3);
45 ellipse(560, 95, 64, 64);
46 }
```

Çıktı

Çalıştır

Büyüt



Kural: her şekil, çizildiği andaki ayarı kullanır

Ayar kalıcıdır

`fill` ve `stroke` bir kez çağrılınca sonraki bütün şekiller aynı rengi kullanır. Yeni renk istiyorsanız komutu yeniden yazın.

Geriye işlemez

Yeni ayar, daha önce çizilmiş şekilleri değiştirmez.

Sıra

Önce `fill` / `stroke`, sonra `rect` / `ellipse` / `line`.

`line` ve dolgu

Çizginin içi yoktur; `fill` onu etkilemez. Rengini `stroke` belirler.

RGB renk modeli

RGB (Red–Green–Blue)

Ekran rengini kırmızı (R), yeşil (G) ve mavi (B) ışığın karışımıyla ifade eden modeldir. p5.js'te her bileşen 0 ile 255 arasında bir tam sayıdır: 0 kapalı, 255 tam güç.



```
fill(255, 0, 0)
```

Yalnızca kırmızı açık.



```
fill(0, 255, 0)
```

Yalnızca yeşil açık.



```
fill(0, 0, 255)
```

Yalnızca mavi açık.

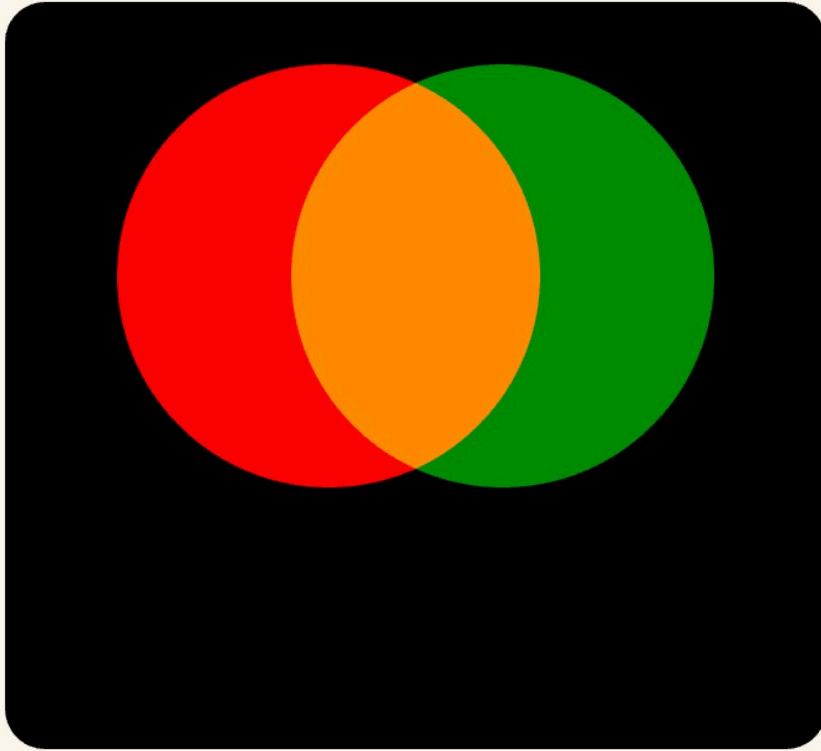
Ekran ışıkla çalışır: bileşenler birleştikçe renk **aydınlanır**. Üçü de 255 ise beyaz, üçü de 0 ise siyah.

Üç ışığı siz karıştırın

● Çıktı

Çalıştır

Büyüt



R = 255

G = 140

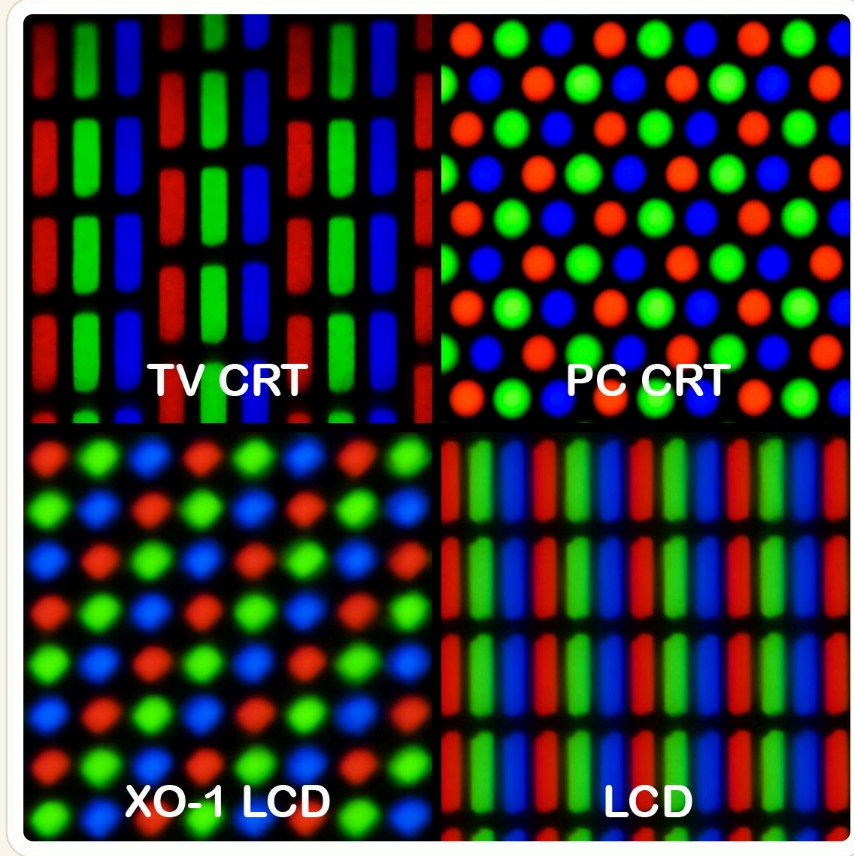
B = 0



```
fill(255, 140, 0);
```

Çubukları sürükleyin. Üçü de 255 olunca beyaz, üçü de 0 olunca siyah. Sarıyı bulmak için hangi ikisini açmak gerekir?

Ekrana büyüteçle bakınca



Farklı ekran türlerinde pikseller. Kaynak: [Wikimedia Commons](#), Pengo
(CC BY-SA 3.0)

Her piksel aslında üç küçük ışıktır: kırmızı, yeşil, mavi.

- `fill(255, 140, 0)` : kırmızı ışık sonuna kadar açık, yeşil yarıda, mavi kapalı.
- Gözümüz bu üçünü ayrı ayrı seçemez; tek bir turuncu görür.

Deneyin

Telefon ekranına bir damla su damlatın: damla mercek gibi davranır, renkli noktalar görünür.

Sayıları okuyalım

	Komut	R / G / B	Sonuç
	<code>fill(255, 255, 255)</code>	Üçü de tam güçte	Beyaz
	<code>fill(0, 0, 0)</code>	Üçü de kapalı	Siyah
	<code>fill(128)</code>	Tek sayı = gri ton (R=G=B=128)	Orta gri
	<code>fill(220, 50, 50)</code>	Kırmızı baskın, diğerleri düşük	Soluk kırmızı
	<code>fill(255, 255, 0)</code>	Kırmızı + yeşil	Sarı
	<code>fill(0, 255, 255)</code>	Yeşil + mavi	Cyan
	<code>fill(255, 0, 255)</code>	Kırmızı + mavi	Magenta

Rengi yazmanın üç biçimi ve alfa

Biçim	Örnek	Anlamı
Gri ton (tek sayı)	<code>background(220)</code>	0 siyah ... 255 beyaz
RGB (üç sayı)	<code>fill(50, 100, 220)</code>	Kırmızı, yeşil, mavi
RGBA (dört sayı)	<code>fill(255, 0, 0, 120)</code>	Dördüncü sayı alfa

Alfa (alpha) kanalı

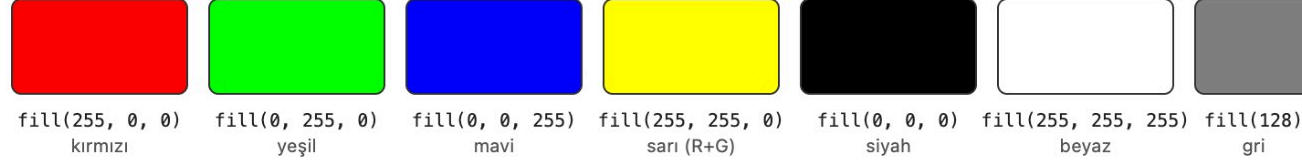
Rengin ne kadar opak ya da saydam olduğunu belirleyen dördüncü bileşendir. 0 tamamen görünmez, 255 tamamen opak, aradaki değerler yarı saydam.

`stroke` ve `background` da aynı sayı biçimlerini kullanır.

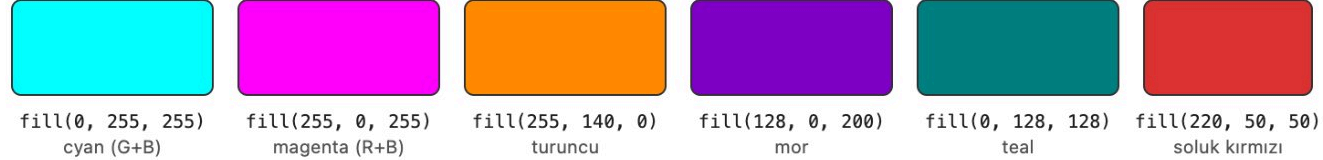
RGB ve alfa, yan yana

RGB örnekleri ve alfa kanalı

Temel renkler



Bazı renk karışımları

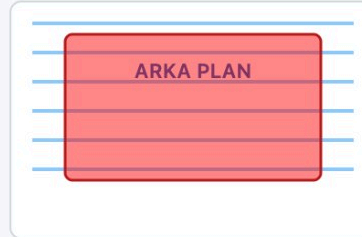


Alfa kanalı: aynı RGB (255, 0, 0), farklı saydamlık

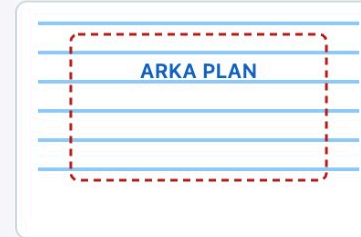
Arka plandaki çizgiler / yazı, alfa düştükçe daha net görünür



fill(255, 0, 0, 255)
alfa = 255 — opak



fill(255, 0, 0, 120)
alfa = 120 — yarı saydam



fill(255, 0, 0, 0)
alfa = 0 — görünmez

En altta aynı kırmızı: alfa 255, 120 ve 0.

Alfa: arkadaki şekil görünür mü?

sketch.js

Kopyala

Düzenle

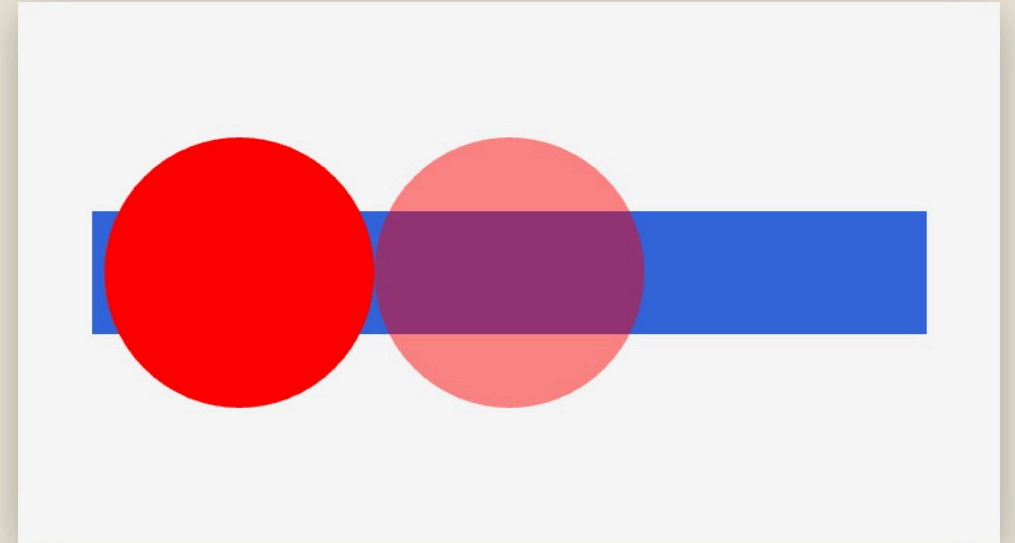
Büyüt

```
1 function setup() {  
2   createCanvas(400, 220);  
3   noLoop();  
4 }  
5  
6 function draw() {  
7   background(245);  
8   noStroke();  
9  
10  fill(50, 100, 220);  
11  rect(30, 85, 340, 50);  
12  
13  fill(255, 0, 0); // opak  
14  ellipse(90, 110, 110, 110);  
15  
16  fill(255, 0, 0, 120); // yarı saydam  
17  ellipse(200, 110, 110, 110);  
18  
19  fill(255, 0, 0, 0); // görünmez  
20  ellipse(310, 110, 110, 110);  
21 }
```

● Çıktı

Çalıştır

Büyüt



Alfa oyunlarda nerede işe yarar?

Üst üste binen şekiller

Arkadaki çizim kısmen görünür.

Vurgu efektleri

Bir nesnenin üstüne soluk bir parlama koymak.

HUD (Heads-Up Display)

Ekranın üstünde ya da köşesinde duran skor, can gibi bilgi paneli.

`fill(255, 0, 0)` opak kırmızı; `fill(255, 0, 0, 80)` arkasındaki çizimin kısmen görüldüğü soluk kırmızıdır.

Alfa iş başında: hayalet, hasar, gece

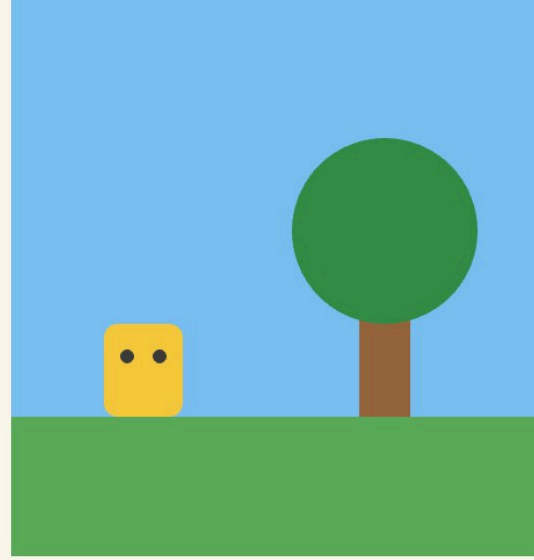
● Çıktı

Çalıştır

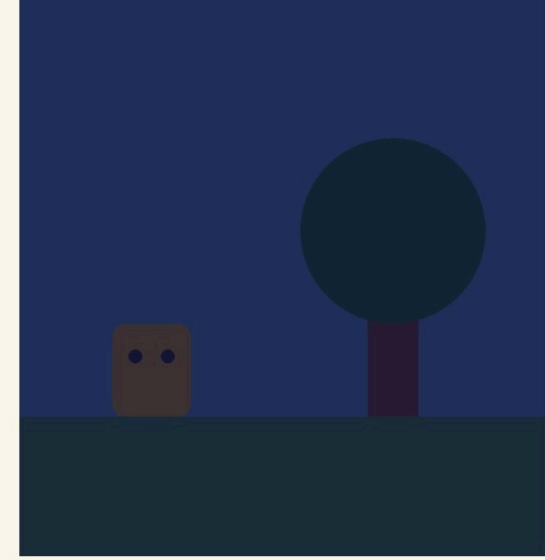
Büyüt



`fill(255, 120)`
yarı saydam hayalet



`fill(255, 0, 0, 000)`
hasar alınca parlama



`fill(10, 10, 50, 201)`
gece ile gündüz

Üçünde de arkadaki sahne aynı; üstüne yarı saydam bir şekil çiziliyor. Alfa sayısı değiştikçe arkası daha çok ya da daha az görünür.

Renk sayılarını ezberlemek gerekmez

Renk seçici (color picker)

Bir rengi tıklayınca R, G, B değerlerini verir. Google'da "color picker" araması, HTML Color Codes, Colors ya da tarayıcının geliştirici araçları iş görür. Üç sayıyı `fill`, `stroke` ya da `background` içine yazmanız yeter.

Başka renk modelleri de var

HSB (ton, doygunluk, parlaklık; p5.js'te `colorMode(HSB)`) ve HSL. Bu derste RGB ve alfa ile yetiniyoruz.

Hepsi bir arada

sketch.js

Kopyala

Düzenle

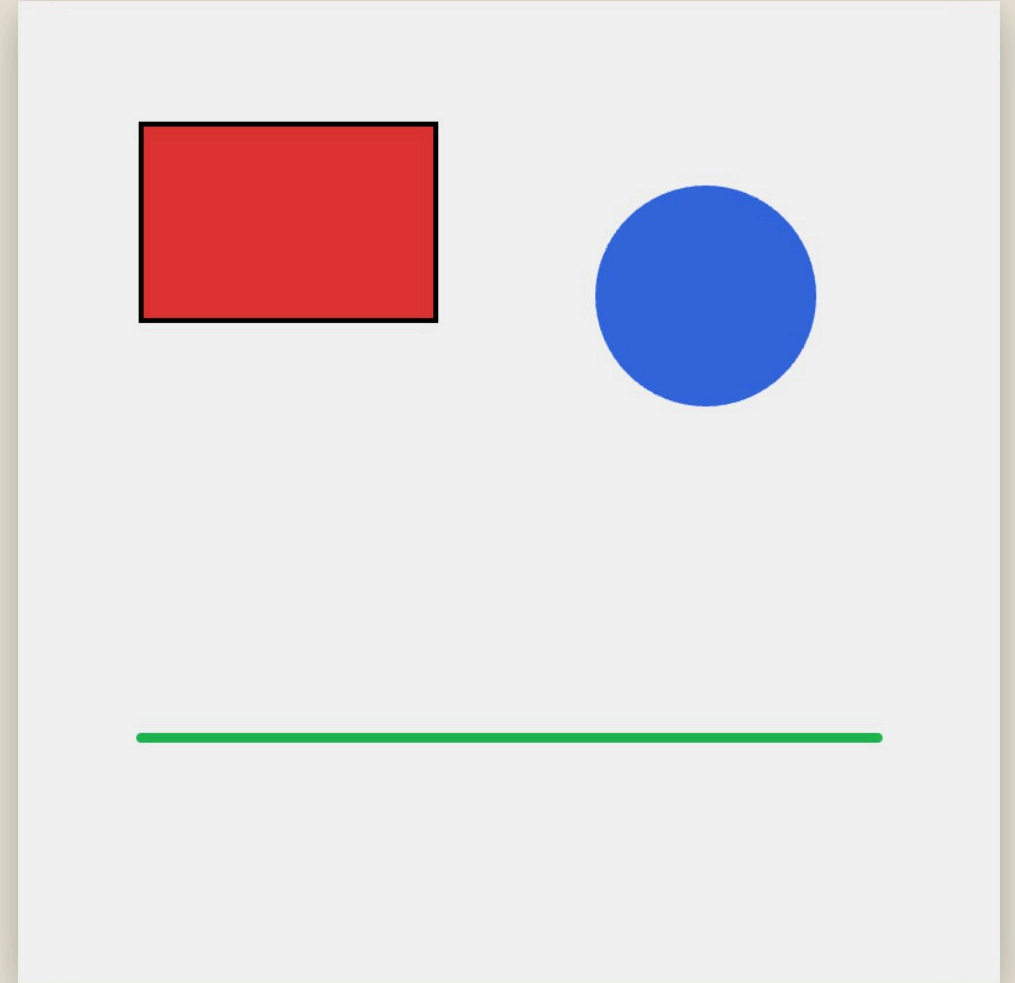
Büyüt

```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(240);  
7  
8   // Kırmızı dolgulu, siyah kenarlı dikdörtgen  
9   fill(220, 50, 50);  
10  stroke(0);  
11  strokeWeight(2);  
12  rect(50, 50, 120, 80);  
13  
14  // Mavi dolgulu daire  
15  fill(50, 100, 220);  
16  noStroke();  
17  ellipse(280, 120, 90, 90);  
18  
19  // Yeşil çizgi  
20  stroke(30, 180, 80);  
21  strokeWeight(4);  
22  line(50, 300, 350, 300);  
23 }
```

Çıktı

Çalıştır

Büyüt



Amiral Battı: koordinat söyleme oyunu



Bir öğrenci “sütun 3, satır 2” desin, bir diğeri tıklasın. Oyunlardaki her nesnenin yeri de böyle bir sayı çiftidir: Minecraft’ta F3’e basınca karakterin koordinatları ekrana gelir.

Koordinat tahmini

 15-20 dk

1. 400×400 tuvalin tam ortasına bir daire çizmek için x ve y ne olmalı? Önce kâğıda yazın.
2. Bir yapay zekâdan aynı tuval için köşe ve merkez örnekleri isteyin. Tahmininizle karşılaştırın.
3. “ y aşağı artar” kuralını unutunca çıkan bir hata senaryosu isteyin. Önerdiği düzeltmeyi kendi sketch’inizde deneyin.

Çizin bakalım

 15 dk

1. 400×400 tuval. Çapı 80 olan bir daire tam ortada dursun.
2. Aynı daire sol alt köşeye içeriden değsin. Merkezin y değeri kaç olmalı?
3. Dört köşeye 50×50 birer kare koyun; her biri farklı renkte olsun.
4. Tuvalin iki köşegenini kalın, turuncu çizgilerle çizin.
5. Ortadaki dairenin üstüne yarı saydam, sarı, daha büyük bir daire ekleyin.

Zorlu görev: dart tahtası ve ev

⌚ 20 dk

Dart tahtası

Aynı merkezli dört daire: en dışta siyah, sonra beyaz, sonra mavi, en içte kırmızı. Hangi sırayla çizmelisiniz? Sırayı ters çevirince ne oluyor?

Ev

Kare bir gövde, iki line ile çatı, bir kapı ve kenarsız, yarı saydam bir pencere. Evin tamamı tuvalin alt kenarına otursun.

İpucu: sonra çizilen şekil, öncekinin üstüne gelir.

400 × 300 tuvalin sağ alt köşesine içeriden oturan 60 × 40 dikdörtgen hangisi?

A `rect(400, 300, 60, 40)`

☒ `rect(340, 260, 60, 40)`

C `rect(370, 280, 60, 40)`

D `rect(340, 300, 60, 40)`

`rect` sol üst köşeden çizer: $x = 400 - 60$, $y = 300 - 40$.

Üçüncü daire nasıl görünür?

sketch.js

Kopyala

Büyüt

```
1 fill(255, 0, 0);  
2 ellipse(100, 100, 80, 80);  
3  
4 stroke(0, 0, 255);  
5 fill(0, 255, 0);  
6 ellipse(200, 100, 80, 80);  
7  
8 noStroke();  
9 ellipse(300, 100, 80, 80);
```

A Kırmızı, kenarsız

B Yeşil, mavi kenarlı

☒ **C** Yeşil, kenarsız

D Görünmez

Son `fill` yeşildi ve hâlâ geçerli. `noStroke` yalnızca kenarı kapattı.

`strokeWeight(20)` ile çizilen `rect(100, 100, 100, 100)` ekranda kaç piksel genişlik kaplar?

A 100

B 110

✓ 120

D 140

Kenar sınırın ortasına oturur: solda 10, sağda 10 piksel dışarı taşar.

Merak edenler için kaynaklar

PATT VİRA

Pixel & Coordinate System

Piksel ve koordinat sistemini sıfırdan anlatan ders.

[pattvira.com > intro-to-creative-coding > pixel-coordinate-system](https://pattvira.com/intro-to-creative-coding/pixel-coordinate-system)

THE CODING TRAIN

Shapes & Drawing

rect, ellipse, line ve koordinatlar.

[thecodingtrain.com > 1-intro > 3-shapes-drawing](https://thecodingtrain.com/1-intro/3-shapes-drawing)

PATT VİRA

Color

RGB, gri tonlar ve saydamlık.

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

THE CODING TRAIN

Color

fill, stroke, background ve alfa.

[thecodingtrain.com > 1-intro > 4-color](https://thecodingtrain.com/1-intro/4-color)

Bugün ne öğrendik?

- ✓ Tuvalde (0, 0) **sol üst köşe**; x sağa, y **aşağı** artar.
- ✓ `rect` sol üst köşeden, `ellipse` merkezden konumlanır.
- ✓ `fill`, `stroke`, `strokeWeight`, `noFill`, `noStroke` kendilerinden sonraki şekilleri etkiler.
- ✓ Kalın kenar, şeklin sınırının iki yanına taşar.
- ✓ Renk: tek sayı gri, üç sayı RGB, dördüncü sayı alfa. Hepsi 0–255.

SIRADAKİ DERS

Şekiller nasıl hareket eder?

setup, draw, frameCount ve ilk animasyon.

Sonraki sunum →

← Önceki