

# SCRATCH 2.0 DERS NOTLARI



## SCRATCH NEDİR?

Scratch programı eğlenceli bir ortamda resim, ses, müzik gibi çeşitli medya araçlarını bir araya getirebileceğimiz, kendi animasyonlarımızı, bilgisayar oyunlarımızı tasarlayabileceğimiz ya da interaktif hikayeler anlatabileceğimiz ve paylaşabileceğimiz bir grafik programlama dilidir.

Scratch web sayfası, Scratch' i öğrenebilmemiz için birçok kaynak içermektedir. Web sayfasını ziyaret etmek isterseniz 'http://scratch.mit.edu' adresinden Scratch ana sayfasına ulaşabilirsiniz.

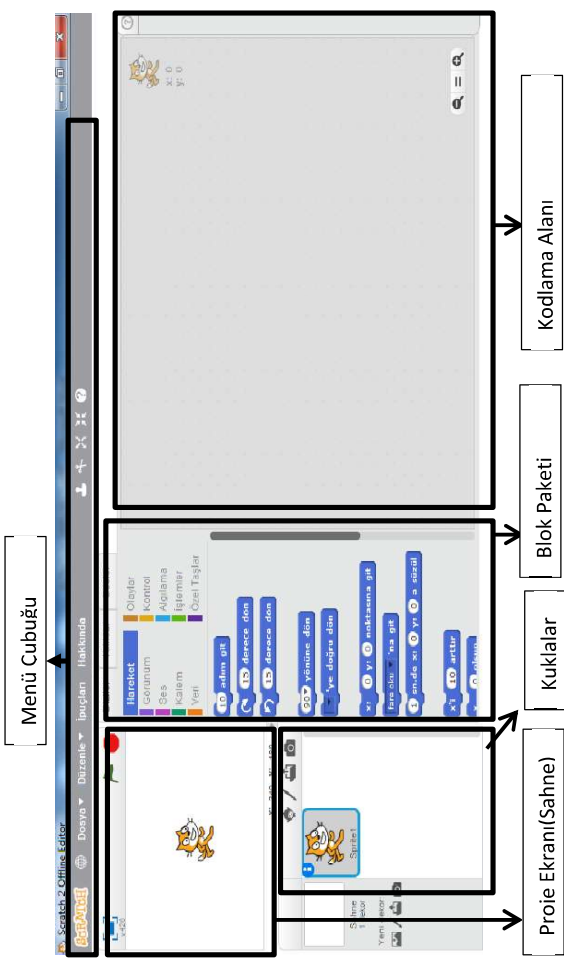
## ÜYE OLMA VE SİTEDE ÇALIŞMA



Scratch.mit.edu adresi açıldığında sağ üstteki **Scratch'a Katıl** bağlantısından siteye üye olabilirsiniz. Daha sonraki kullanimlarımızda aynı sayfadaki **Giriş** bağlantısıyla da sisteme kullanıcı adı ve şifrenizle girebilirsiniz. Bu sayfa üzerindeki **Oluştur** bağlantısıyla doğrudan site üzerinde çalışarak programlar geliştirebilirsiniz.

**Keşfet** bağlantısını kullanarak diğer üyeler tarafından hazırlanan programları hazırlama aşamalarını ve kullanılan komutları görebilirsiniz.

## SCRATCH'IN EKCRANINI TANİYALIM



1. Menü Çubuğu: Scratch'ın yönetim, ayar ve bazı kukla(karakter) komutlarını içerir.
2. Proje Ekranı (Sahne): Bizim sahnemizdir. Projemizi çalıştırdığımızda tasarladığımız her şey burada hayat bulur.
3. Kuklalar(Karakterler): Kukla ekleme silme ve düzenleme işlemlerini yapabildiğimiz kuklaların listelendiği bölümdür.
4. Blok Paketi: Karakterimizi programlamak için kullanabileceğimiz blokların bulunduğu kısımdır.
5. Kodlama Alanı: Blokları sürükleyerek komut dizileri oluşturacağımız alandır.

## Scratch 2.0 Ders Notları

### MENÜ ÇUBUĞU

	<b>Dil ayarla:</b> Bu ikonu kullanarak kullandığımız blok dilini değiştirebilir farklı dillerde kullanabiliriz.
	<b>Kopyasını Çıkar:</b> Var olan karakterimizi çoğaltabiliriz.
	<b>Sil:</b> İstedğimiz bir karakteri silebiliriz.
	<b>Büyüt:</b> Bu simgeyi seçerek karakterimizin üzerine tıkladığımızda karakterimizi istediğimiz kadar büyütülebiliriz.
	<b>Ufult:</b> Bu simgeyi seçerek karakterimizin üzerine tıkladığımızda karakterimizi istediğimiz kadar küçülebiliriz
	<b>Yardım:</b> Scratch ve komut blokları hakkında yardım sağlar.

### DOSYA MENÜSÜ

Dosya ▾
Yeni
Open
Kaydet
Save as
Share to website
Check for updates
Quit

det'e tıklarız.

**Share to website(Web'de paylaş):** Hazırlana projenin scratch.mit.edu adresindeki sayfamıza aktararak internette paylaşmamız için kullanılır. Bunu kullanabilmek için önce üye olmamız gerekir. Komutu verdiğimizde açılan pencerede kullanıcı adı şifre ve proje adı bilgilerinin girerek işlemi tamamlarız.

**Check for Updates:** Scratch'ın yeni versiyonunu indirmek için kullanılır.

**Quit(Çıkış):** Scratch programından çıkış yapmamızı sağlar.

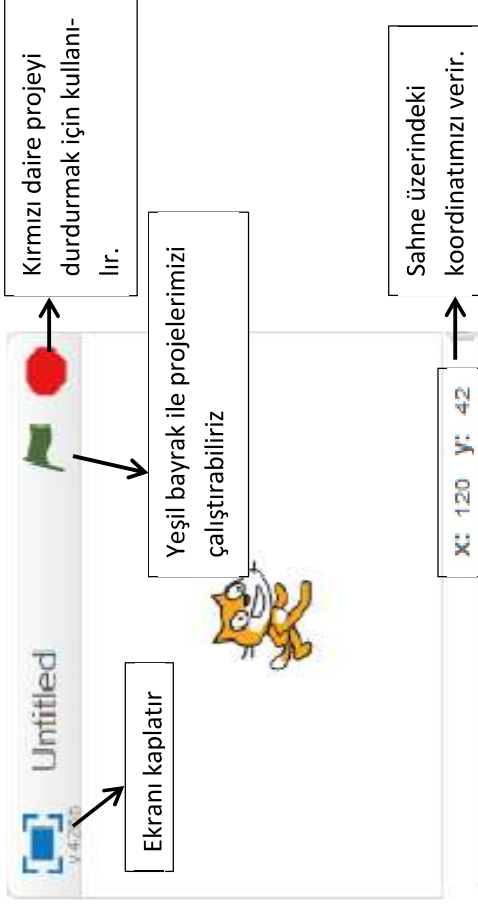
### DÜZENLE MENÜSÜ

**Silmeseydim:** Sildiğimiz komut bloklarını geri alır.

**Küçük sahne düzeni:** kodlama esnasında sahnenin küçük ya da büyük görünmesini sağlar. Tik işareti olursa sahne küçük olur.

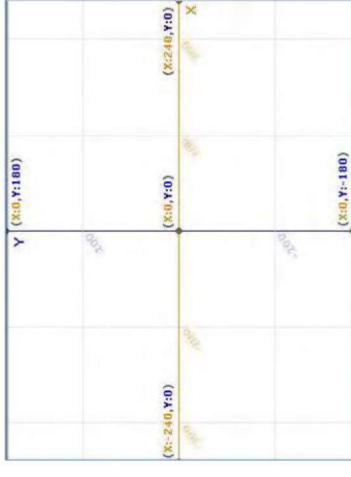
**Jet hızında:** Projeyi hızlı çalıştırmak için kullanılır. Örneğin bu modda hareketler hızlanır.

### SAHNE



Hazırladığımız projemizin canlandığı yerdir. Oyunlarımız, animasyonlarımız, hikayelerimiz bu ekranda hayat bulur.

Ekran 480 birim genişlikte ve 360 birim uzunluktadır. Scratch ekranı aslında bir koordinat düzlemidir. Scratch programı açıldığında karşımıza çıkan kedi karakteri başlangıçta (0,0) noktasındadır.



## Scratch 2.0 Ders Notları

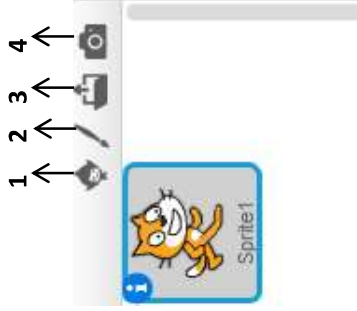
### SAHNE VE DEKOR AYARLARI


- 1. Dekor Kütüphaneden Seç:** Kütüphaneden dekor eklemek için kullanılır.
- 2. Yeni Dekor Çiz:** Bir çizim arayüzü sayesinde dekoru siz çizersiniz.
- 3. Dekor Bilgisayarından Seç:** Bilgisayarınızda kayıtlı bir resmi dekor olarak eklemek için kullanılır.
- 4. Kameradan Yeni Dekor Oluştur:** Bilgisayara bağlı kamera yardımıyla çekilen fotoğrafı dekor olarak ekler.



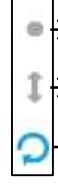
### KUKLALAR(KARAKTERLER)

- 1. Kuklayı Kütüphaneden Seç:** Kütüphaneden yeni kukla eklemek için kullanılır.
- 2. Yeni Kukla Çiz:** Bir çizim arayüzü sayesinde yeni kuklayı siz çizersiniz.
- 3. Kuklayı Bilgisayarından Seç:** Bilgisayarınızda kayıtlı bir resmi yeni kukla olarak eklemek için kullanılır.
- 4. Kameradan Yeni Kukla Oluştur:** Bilgisayara bağlı kamera yardımıyla çekilen fotoğrafı yeni kukla olarak ekler.



Kuklamanın sol üst köşesindeki  tuşuna tıklanarak açılan bölümde kuklayla ilgili ayarlar yapmak için kullanılır. Bu ayarlar;

- 1. Kukla adı:** Kuklamanın adını değiştirmek için kullanılır.
- 2. Konum ve Yön bilgisi:** Kuklamanın koordinatlarını(konum) ve yönünü belirtir bu bölümden kukla yönü değiştirilebilir.



**3. Dönme Özellikleri:** Karakterin nasıl döneceğini belirlediğimiz bölümdür. Bunlar;

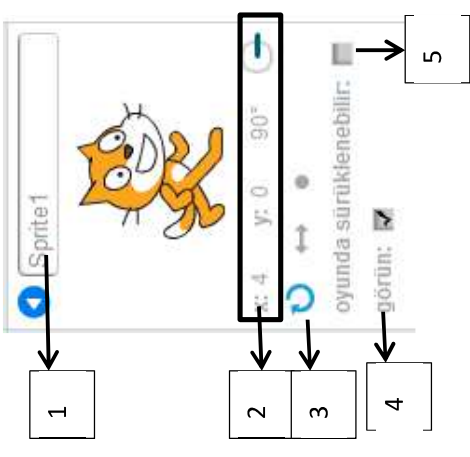
- 1. Tıklanırsa karakter kostümünü 360° döndürme özelliği kazanır.**

2. Tıklanırsa karakter kostümünü sağa-sola dönebilir

3. Tıklanırsa karakter kostümünü hiç bir şekilde dönmez.

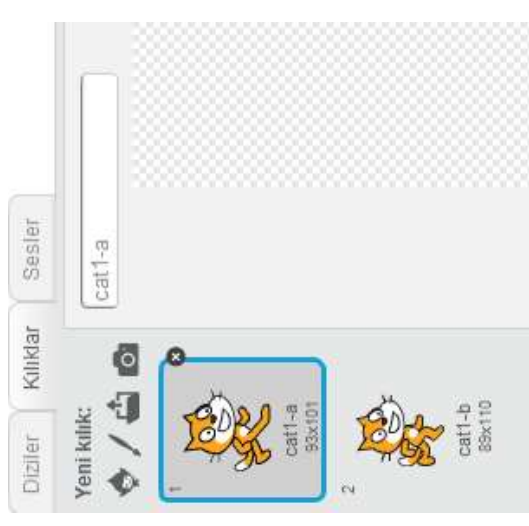
**4. Görün:** Proje çalıştırıldığında kuklanın ekranda görünüp görünmeyeceğini ayarlar.

**5. Oyunda Sürüklenebilir:** Proje çalıştırıldıktan sonra kuklanın fare ile tutularak sürüklenip sürüklenemeyeceğini ayarlar.



### KILIKLAR(KOSTÜMLER)

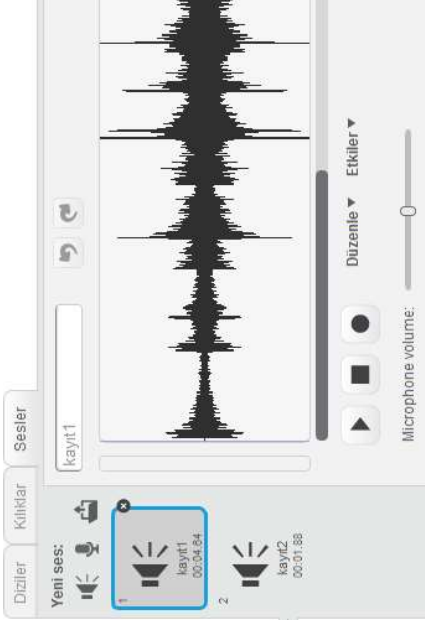
Karakterimizin başka görünümünün olmamasını istiyorsak kütüphaneden ekleyebilir, kendimiz çizebilir, bilgisayarımızda kayıtlı bir resmi içe aktarabiliriz ya da kamera ile çekebiliriz. Var olan kostümleri kostümün sağ üst köşesindeki 'x' butonuna basarak silebiliriz, seçili kostümü sağdaki düzenleme alanında düzenleyebiliriz. Kostüm üzerinde sağ tık yaparak kopyalayabilir ve silebiliriz.



## Scratch 2.0 Ders Notları

### SESLER

Karakter için var olan sesleri görebilmek için bu bölüme tıklayınız. Bu bölümden yeni sesler ekleyebilir (🔊📁) sırasıyla kütüphaneden ses ekler, ses kaydeder, bilgisayarda kayıtlı bir sesi ekler, var olan sesi dinleyebilir, düzenle ve etkiler altındaki komutlarla düzenleyebilir veya silebilirsiniz.



### BLOK PAKETİ (DİZİLER)

Blok Paletinde karakterleri programlayabilmek için kullanılan bloklar vardır. 10 ana kategoride toplanmış blok grubu bulunmaktadır.

### HAREKET

Hareket bloğunda yer alan blokların işlevleri tablodaki gibidir.

	Karakteri öne ya da arkaya doğru hareket ettirir.
	Karakterin istenilen derece kadar saat yönünde dönmelerini sağlar.
	Karakterin istenilen derece kadar saat yönüne tersine dönmelerini sağlar.
	Karakterin hangi yöne döneceğini belirler. (0=yukarı, 90=sag,180=aşagi, -90=sol)

	Karakteri 'mause işaretçisi' veya başka karakterler varsa istenilen 'karakter' e doğru döndürülür.
	Karakteri istenilen x ve y konumuna getirir.
	Karakter 'mause işaretçisi' veya başka karakterler varsa istenilen 'karakter' ile aynı konuma gönderilir.
	Karakter belli bir süre içerisinde belirlenen x ve y konumuna hareket eder.
	Karakterin bulunduğu konumun x değeri istenilen değer kadar artırılır veya azaltılır.
	Karakterin x konumunu istenilen değere çeker.
	Karakterin bulunduğu konumun y değeri istenilen değer kadar artırılır veya azaltılır.
	Karakterin y konumunu istenilen değere çeker.
	Karakter ekranın kenarına değdiği an karakteri ters yöne döndürür.
	Karakterin sağa-sola dönme, etrafında dönme ve hiç dönmemeye ayarlarını yapar
	Karakterin x pozisyonu bilgisini verir. Bu seçeneğin işaretlendiğinde x değerini ekranda görebiliriz.
	Karakterin y pozisyonu bilgisini verir. Bu seçeneğin işaretlendiğinde y değerini ekranda görebiliriz.
	Karakterin yön bilgisini verir. Bu seçeneğin işaretlendiğinde yön bilgisini ekranda görebiliriz.

### GÖRÜNÜM

Görünüm bloğunda yer alan blokların işlevleri tablodaki gibidir.

## HAREKET KODLARI

10 adım git

→ Kuklayı yönü doğrultusunda girilen değer kadar (10 adım ileri) ileri götürür.

Eğer kuklayı yönünün tersi doğrultusunda hareket ettirmek istersek değerın başına eksi işareti konur. mesela -25 gibi

Not: burada adım diye bahsedilen ölçü birimi aslında pixeldir.

15 derece dön

→ Kulanın yönünü saat yönünde 15° derece **daha** döndürür. Eğer kuklayı saat yönünün tersinde döndürmek istersek başına – işareti konur. mesela -80 gibi

15 derece dön

→ Kulanın yönünü saat yönünün tersine 15° derece **daha** döndürür.

Eğer kuklayı saat yönünde döndürmek istersek başına – işareti konur. mesela -80 gibi...

90 yönüne dön

(90) sağ  
(-90) sol  
(0) yukarı  
(180) aşağı

→ Kulanın yönünü direk olarak belirtilen yöne döndürür. Kutucuğun içine istenilen yön yazılabilir. mesela 135 veya 210 gibi...

fare oku 'ye doğru dön

fare oku  
elma

→ Kuklanın yönünü direk olarak belirtilen kukla yönüne (elma gibi) veya fare oku (işaretçisi) yönüne doğru dönmesini sağlar

x: 0 y: 0 noktasına git

→ kuklayı sahne üzerinde (koordinat sistemine göre) belirtilen (x,y) konumuna götürmeyi sağlar.

fare oku 'na git

fare oku  
rastgele konum  
elma

→ Kuklayı fare okunun olduğu yere veya sahnede rastgele bir konuma veya herhangi başka bir kuklanın (elma gibi) olduğu yere gitmesini sağlar

1 sn.de x: 0 y: 0 a süzül

→ Kuklayı belirtilen (x,y) konumuna belirtilen saniye süresince yavaş yavaş gitmesini sağlar

x'i 10 arttır

→ Kuklanın sahnedeki konumunun x eksenindeki karşılığını belirtilen değer kadar arttırır. Başka bir ifadeyle sağa hareket ettirir. negatif bir değer girilirse x eksenini karşılığını azaltır yani sola hareket ettirir. Kuklayı yatay hareket ettirmek için kullanılabilir.

y'yi 10 arttır

→ Kuklanın sahnedeki y eksenindeki karşılığını belirtilen değer kadar arttırır. Başka bir ifadeyle yukarıya hareket ettirir. negatif bir değer girilirse y eksenini karşılığını azaltır yani aşağıya hareket ettirir. Kuklayı dikey hareket ettirmek için kullanılabilir.

y, 0 olsun

→ kuklanın sahnedeki y eksenini karşılığını direk olarak belirtilen değer yapar. Değerler değiştirildikçe kuklayı dikey olarak hareket ettirir

kenara geldiysen sek

→ Bu kod çalıştığında kukla, eğer sahne kenarındaysa yönlendirmesi sahne içine doğru değişir veya yansır veya seker

kuklanın şekli sağa-sola dönebilsin

sağa-sola dönebilsin  
hiç dönmesin  
her yöne dönebilsin

→ Kuklanın dönüş sınırları ayarlanır.

x konumu

→ Kuklanın sahnede bulunduğu konumun x eksenini karşılığı değeri saklar.

y konumu

→ Kuklanın sahnede bulunduğu konumun y eksenini karşılığı değeri saklar.

yönü

→ kuklanın hangi yöne dönük olduğunun açısını veri olarak saklar

## İŞLEMLER KODLARI

+

→ İçerisine girilen değerlerin **toplamını** alır. değişken girilirse değişkenleri toplar

-

→ içerisine girilen değerlerle **çıkarma** işlemi yapar. değişken girilirse değişkenleri çıkarr

\*

→ içerisine girilen değerlerle **çarpma** işlemi yapar. değişken girilirse değişkenleri çarpır

/

→ içerisine girilen değerlerle **bölme** işlemi yapar. değişken girilirse değişkenleri böler

1 ile 10 arasında bir sayı (tut)

→ içerisine girilen değerler arasında rastgele bir sayı tutar. eğer 0 ile 0.99 girilirse bu iki sayı arasındaki bir ondalıklı sayıyı rastgele tutar.

<

→ girilen değerler doğruysa doğru(true) değerini, yanlışsa yanlış (false) değerini tutar.

Örnek: 5<8 → true , 60<20 → false

=

→ girilen değerler doğruysa doğru(true) değerini, yanlışsa yanlış (false) değerini tutar.

Örnek: 5=5 → true, 9=1 → false

>

→ girilen değerler doğruysa doğru(true) değerini, yanlışsa yanlış (false) değerini tutar.

Örnek: 23>17 → true, 46>89 → false

ve

true ve true → true  
true ve false → false  
false ve true → false  
false ve false → false

doğru ve doğru → doğru  
doğru ve yanlış → yanlış  
yanlış ve doğru → yanlış  
yanlış ve yanlış → yanlış



veya

true veya true → true  
true veya false → true  
false veya true → true  
false veya false → false

doğru ve doğru → doğru  
doğru ve yanlış → doğru  
yanlış ve doğru → doğru  
yanlış ve yanlış → yanlış

değil

→ içerisindeki ifadenin tersini yapar. Yani true ise false olur. false ise true olur

hello ile world i birleştir

→ içerisindeki iki farklı metni tek bir metin haline getirir. Eğer değişken kullanılmışsa değişkenlerin içerisindeki metinleri birleştirir.

1 . harfi world in

→ Belirtilen metnin (world) sayıyla belirtilen sıradaki harfini (w) değer olarak tutar. Değişken kullanılmışsa değişkenin içerisindeki metnin harfini tutar.

world in uzunluğu

→ Belirtilen metnin kaç karakterden oluştuğu değerini tutar.

mod

→ 1.kutucuktaki sayının 2. kutucuktaki sayıya bölümünden kalanı tutar.

karekök 9

→ belirtilen değerın karekökünü alma işlemini yapar. Menüüne bakıldığında sinüs, kosinüs, mutlak değer alma gibi birçok matematiksel işlemleri yapma özelliği vardır.

i yuvarla

→ Ondalıklı yani virgüllü bir sayının en yakın tam sayı değerini bulur.

## KONTROL KODLARI

1 saniye bekle

→ Kod dizisinin çalışmasını belirtilen saniye kadar bekletir

10 defa tekrarla

→içerisine eklenen kodları belirtilen sayı kadar tekrarlar. Bu örnekte 10 defa tekrarlayacaktır.

sürekli tekrarla

→ içerisinde konulan kodları yazılım çalıştığı sürece sürekli tekrarlar

eğer ise

→koşul doğru (true) olduysa "ise" bölümünde yer alan kodları, yanlış(false) olduysa "değilse" kısmındaki kodları çalıştırır.

değilse

olana kadar bekle

→ Koşul doğru (true) olana kadar yazılımın çalışmasını bekletir.

olana kadar tekrarla

→ Koşul doğru(true) olana kadar içerisinde eklenen kodları sürekli tekrar eder. Doğru(true) olduktan sonra tekrardan(döngüden) çıkar.

hepsi durdur

hepsi  
bu diziyi  
kuklanın diğer dizilerini

→ "Hepsi" seçeneği seçiliyse diğer kuklalarla birlikte tüm yazılımı durdurur.

"bu diziyi" seçeneği seçiliyse sadece bu kodun çalıştığı kod dizisini durdurur. diğer kod dizilerini etkilemez.

"kuklanın diğer dizilerini" seçeneği seçiliyse, bu kodun çalıştığı dizi haricindeki kuklanın tüm dizilerini durdurur. Diğer kuklaların dizilerini hiç etkilemez

kendim in ikizini yarat

→ Kuklanın kendi ikizini (kopyasını) çıkarır. kodun menüsünden diğer kuklalar da seçilebilir ve böylece diğer kuklaların da kopyaları yani ikizleri çıkartılabilir

bu ikizi sil

→ kodun çalıştığı kuklaya ait tüm ikizleri/kopyaları siler

ikiz olarak başladığımda

→ Kodun çalıştığı kuklanın ikizi başlatıldığı taktirde bu kodun altına eklenen kodlar da çalışmaya başlar.

## OLAYLAR

tıklanınca

→ Programın başlatılmasından itibaren otomatik olarak çalışmasını istediğimiz kodlar bu olayın altına eklenmesi gerekir.

boşluk tuşu basılınca

→ Klavyeden bir tuşa basılmasından itibaren çalışmasını istediğimiz kodlar bu olayın altına eklenmesi gerekir.

bu kukla tıklanınca

→ Eğer bir kuklaya tıklanmasından itibaren çalışmasını istediğimiz kodlar varsa bu olayın altına eklenmesi gerekir.

dekor dekor1 olunca

→ Sahnenin dekoru seçenekte belirtilen dekor (dekor1) olmasından itibaren çalışmasını istediğimiz kodlar varsa bu olayın altına eklenmesi gerekir

haber1 haberini sal

→ Bu kod çalıştırıldığında yazılımın tüm kuklalarına bir haber salar. Haberin ismi seçenekte ne belirtilmişse odur. (haber1)... Kuklalardaki kodlarda aşağıdaki "haberi gelince" olayı kullanılmışsa bu haberi algıladığından itibaren kendisine eklenmiş olan kodlar çalışmaya başlar.

haber1 haberi gelince

→ bu kodun yaptığı şudur; Yazılım çalıştığı esnada seçenekte belirtilen "haber1" haberi salınırsa bu kod etkilenir ve bu kodun altındaki kodlar çalışmaya başlar.

ses şiddeti > 10 olunca

→ Ses şiddeti belirtilen değerın üzerine çıkarsa bu olay kodu tetiklenir ve olay kodun altındaki kodlar çalışmaya başlar.

## ALGILAMA KODLARI

**fare oku** a değdi (mi?) → Kuklanın fare okuna, kenara ve diğer kuklalara değip değmediğini kontrol eder.

**rengine değdi (mi?)** → Kuklanın seçilen renge değip değmediğini kontrol eder. (Rengi değiştirmek için renk üzerine tıklayıp ekrandan renk seçin.)

**renge** rengine değdi (mi?) → Kuklanın üzerinden seçilen bir rengin seçilen başka bir renge değip değmediğini kontrol eder.

**fare oku** a mesafe → kuklanın fare okuyla veya diğer kuklalarla arasındaki mesafeyi değer olarak tutar

**What's your name? diye sor ve bekle** → Kodun üzerindeki kutucuğa yazılan yazıyı kuklaya soru olarak sahnede yazdırır ve bekletir. kullanıcının verdiği yanıt ise **yanıt** kodunda değer olarak tutulur.

**boşluk** tuşu basılı (mi?) → kodun üzerindeki menüden seçilebilecek herhangi bir klavye tuşuna basılıp basılmadığını kontrol eder.

**fareye basılı (mi?)** → Fareye basılıp basılmadığını kontrol eder.

**farenin x'i** → Farenin x koordinatını değer olarak tutar.

**farenin y'si** → Farenin y koordinatını tutarç

**ses şiddeti** → Ses şiddetinin yüksekliğini değer olarak tutar. ortam sessizken değer sıfırdır. ses yükseldikçe maksimum 100'e kadar değer tutar

**video hareketi**, bu kukla üzerindeki → Bilgisayar kamerasındaki görüntünün kukla veya sahne üzerindeki hareket oranını veya hareket yönünü değer olarak tutar.

**videoyu** aç → bilgisayarın kamera görüntüsünün sahnede açılmasını, kapatılmasını veya simetri görüntüsünün açılmasını sağlar.

**video saydamlığı** % 50 olsun → Bilgisayarın kamera görüntüsünün istenilen oranda saydamlaştırılmasını sağlar.

**süre ölçer** → Yeşil bayrağa basılmasından itibaren kaç saniye geçtiğini tutar. Her bayrağa basıldığında sıfırlanır.

**süre ölçeri sıfırla** → Süre ölçerin değerini sıfırlar.

**x konumu** değeri Kukla2 'in → Seçilen menüye göre belirtilen kuklanın x koordinat noktasını, y koordinat noktasını, gidiş yönünü, sahnedeki kılık numarasını, sahnedeki kılığının adını, yüzde(%) oran cinsinden büyüklüğünü ve kuklada çalan sesin şiddetini değer olarak tutar.

**şimdiki dakika** → Scratch'in çalıştığı bilgisayarın (menü sıralamasına göre) yılını, ayını, gününü (sayı olarak), haftanın gününü (pazartesi, Salı gibi), saatini, dakikasını, saniyesini değer olarak tutar.

**2000'den beri geçen gün** → 01.01.2000 tarih ve 00:00 saatten şu anki tarih ve saate kadar geçen gün sayısını verir.

## SAHNEDEKİ KOORDİNAT SİSTEMİ VE HAREKET YÖNLERİ

