

# PROGRAMLAMA

## Programlama Nedir?

Programlama, bilgisayarın ne yapması gerektiğinin adım adım söylenmesidir. Örneğin; fareye tıklandığında sayfayı kapat, boşluk tuşuna basıldığında oyundaki karakteri zıplat. Bir bilgisayar, bu işleri nasıl yapması gerektiğini bilmez. Bu yüzden bilgisayarın bir işi yapabilmesi için programlanması gerekmektedir.

Bilgisayarlarda çalışan birçok program vardır. Resimleri açmak veya internete bağlanmak için bilgisayardaki bu programları kullanırız. Bu programlar bilgisayarın çalışması için gereken kılavuzlardır. Bilgisayardaki bu programları oluşturmak için programlama dillerini kullanırız. İnsanların günlük hayatta iletişim kurmak için nasıl bir dil kullanmaları gerekiyorsa bizim de bilgisayarlar ile haberleşebilmemiz için bir dil kullanmamız gerekir. Bu dillerden bir tanesi de Scratch programlama dilidir.

Programlama dillerinin, bizim kullandığımız dillerden farkı, yazılı olmalarıdır. Programlama dillerini kullanarak adım adım bilgisayarın yapması gereken işleri yazarız. Bilgisayarlar da bu işleri sırasıyla bizim belirttiğimiz şekilde yaparlar.



Scratch MIT (Massachusetts Institute of Technology) üniversitesi tarafından geliştirilen bir görsel programlama dilidir. Bu dil özellikle 9-16 yaş grubu öğrenciler için oluşturulmuştur. Bu dili kullanarak kendi oyununuzu programlayabilir, kendi animasyonlarınızı oluşturabilir ve yaratıcılığınız dahilinde birbirinden güzel çalışmalar üretebilirsiniz.

Scratch'da program yazmak çok kolaydır. Scratch'da bilgisayara verilecek komutlar, bloklar şeklindedir. Size düşen ise lego yapar gibi bu blokları birleştirmektir.

## Niçin Programlama Öğrenmeliyiz?

Dünya çok hızlı bir şekilde gelişmekte ve bu gelişme her alanda kendini göstermektedir. Bugünün öğrencileri, yarın meslek hayatına atıldığında karşılaşacakları dünya çok farklı olacaktır. Belki de sizin uğraş edindiğiniz meslek bugün olmayan bir meslek olabilir. İngiliz Hükümetince yapılan bir araştırmaya göre, 2030 yılının en popüler meslekleri içerisinde "**vücut parçası imalatçılığı**" da bulunuyor. Bu gelişim sonucunda yeni meslekler ortaya çıkacağı gibi mevcut olan mesleklerde bu değişimden etkileneceklerdir. Örneğin; cerrahi operasyonlar robotlarla yapılacak ve robotik cerrahi alanında uzman olmayan doktorlar cerrahi operasyonlar yapamayacaklar.



Dünyadaki gelişmelerin zorunlu kılması üzerine bugün gelişmiş ülkelerin birçoğunda programlama eğitimi ilkokuldan itibaren verilmeye başlanmıştır. Ülkemizde de kodlama dersinin ilkokullardan itibaren zorunlu olması planlanmaktadır. Bu derslerin ilkokuldan itibaren verilmesinin amacı, öğrencilerin hepsinin birer bilgisayar mühendisi olması değildir. Bu derslerin öğrencilere verilmesinin nedeni 21. Yüzyılın zorunlu kıldığı yeteneklerin öğrencilere kazandırılmasının sağlanmasıdır.

Bu yeteneklerden bazıları şunlardır:

- Bilgisayarca düşünme

Bilgisayarları üretim amaçlı olarak hayat problemlerinin çözümünde kullanabilmek için gerekli olan bilgi, beceri ve tutumlara sahip olmak.

- Analitik düşünme

Bir problemi çözmek için, bilgileri küçük parçalara ayırmak ve küçük parçalar üzerinden problemi çözmek.

- Yaratıcı düşünme

Elde mevcut bulunan bilgi ve imkanlarla yeni ve işe yarar bir şeyi meydana getirmek.

- Problem çözme

Hayatta karşılaşılan güçlük, zorluk ve problemlere çözüm getirebilme becerisi.

Facebook, Twitter, Apple gibi büyük şirketlerin kurucularının başarılı olmalarının altında yatan ortak etkenlerden biri de küçük yaşlarda programlama öğrenmeleridir. Bu programlama yetenekleri onları diğer insanlardan daha avantajlı kılmıştır. Bu nedenden ötürü bugün öğrencilere küçük yaşlardan itibaren programlama öğretilmektedir. Programlama öğrenmek, olaylar arasındaki ilişkiyi görme becerisini, sistematik düşünme becerisini ve problem çözme becerisini artırır. Bilgisayarda program yazarken algoritmalar geliştirilir. Algoritma problemin adım adım çözülme sürecidir. Bu da bize olayları detaylı düşünebilme ve çözebilme yeteneği kazandırır.

“Bu ülkedeki herkes bilgisayar programlamayı öğrenmeli çünkü insana düşünmeyi öğretir.”

Steve JOBS

## Algoritma Nedir?

Algoritma, bir işin yapılması için izlenecek adımlardır. Bilgisayarda programlar yazılırken algoritmalar kullanılmaktadır. Algoritmalarla biz bir işi parçalara böleriz ve sırası ile o parçaları çözeriz. Bu durum bizim olayları detaylı bir şekilde düşünmemizi ve kolayca çözmemizi sağlar.

Hepimiz her gün sabah kahvaltı yaparken, dişimizi fırçalarken ya da uyumaya giderken algoritmaları kullanırız. Algoritma bu yaptığımız işlerin yapılış sıralamasıdır.

## Hayatımızdaki Algoritmalar

### Ayakkabı Giyme Algoritması:



1. Ayakkabıyı Raftan Al

2. Ayakkabıyı Giy

3. Bağcıklarını Bağla

### Diş Fırçalama Algoritması:



1. Kirli Dişler

2. Fırçaya Diş Macunu Sür

3. Dişlerini Fırçala

4. Temiz Dişler

### Çiçek Dikme Algoritması:



1.Saksıyı

2. Toprağı

3.Tohumu

4. Tohumu

5. Çiçeği

6. Güneşli

Toprakla

Karıştır

Dik

Toprakla

Sula

Bir

Doldur

Karıştır

Yere

Koy

# 2

## Scratch ile Programlama Öğreniyorum

Scratch, kendi oyunlarınızı, çizgi filmlerinizi ve programlarınızı yapabileceğiniz görsel bir programlama dilidir. Kolay olduğu kadar eğlencelidir.

Bu bölümde Scratch programını ve Scratch bloklarını tanıyacağız ve örnek uygulamalar geliştireceğiz.

# SCRATCH ile PROGRAMLAMA ÖĞRENİYORUM

## Scratch Nedir?

Scratch, programlarınızı görsel olarak yazabileceğiniz bir programlama dilidir. Scratch ile programlama öğrenmek hem basit hem de zevklidir. Scratch'da kod yazmak için blokları kullanırız. Blok sistemi sürükle bırak yöntemine dayandığı için programlarınızı hızlı ve kolay bir şekilde oluşturabilirsiniz. Scratch'da oyun programlamak lego oynamaya benzer. Lego oyunundaki gibi blokları birleştirirsiniz. Eğer doğru blokları birleştirirseniz ortaya çok güzel bir program, animasyon veya oyun çıkar.



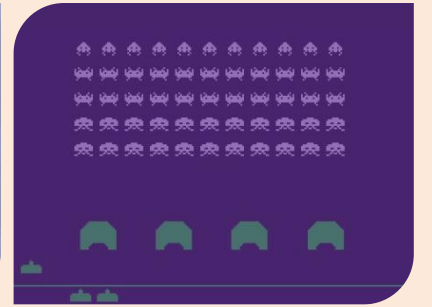
Scratch, günümüzde programlama eğitiminde en çok kullanılan programlama dillerinden biridir. Dünyanın en saygın üniversitelerinden biri olan Harvard'da bilgisayar bilimine giriş dersinde Scratch kullanılmaktadır. Scratch çok kullanılan ve sevilen bir programlama dili olduğu için Scratch kullanırken destek bulmakta zorlanmazsınız. Sürekli güncellenen ve gelişen yapısı ile de çağa ayak uyduran Scratch, programlama öğrenirken sizin en büyük yardımcınız olacaktır.



**Hayal Et**



**Programla**



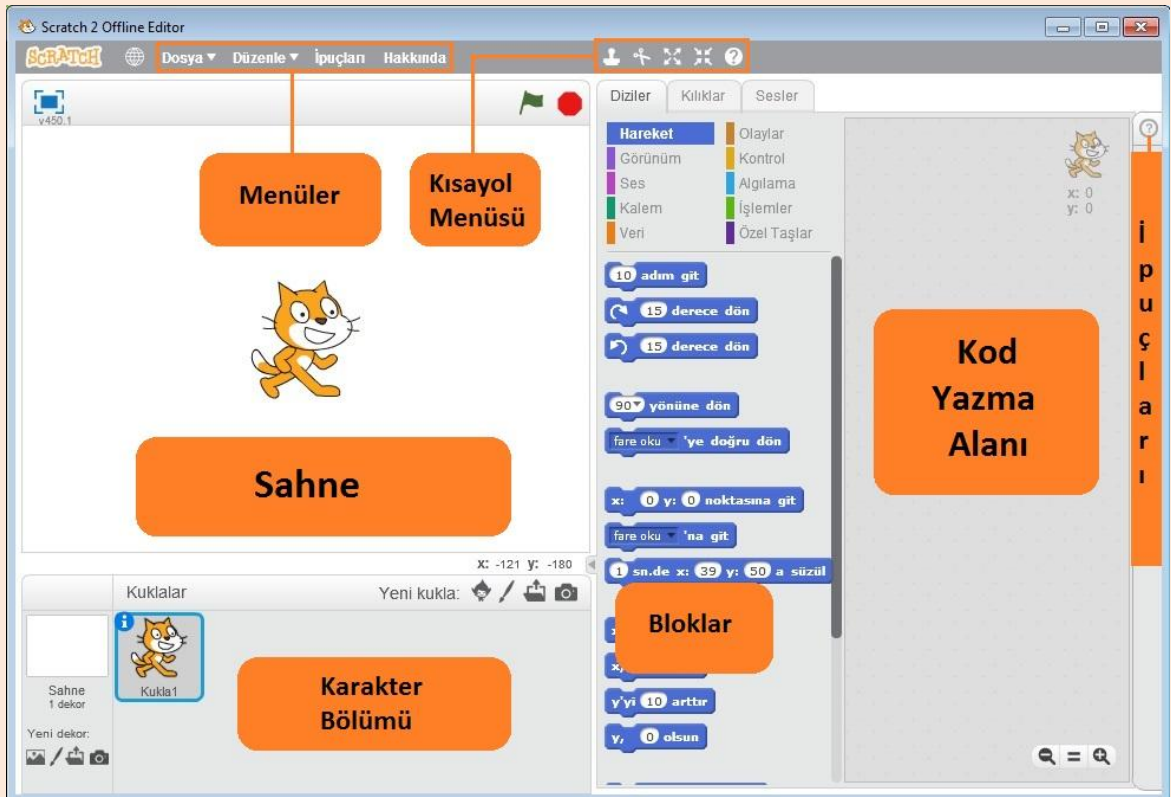
**Paylaş**

## Scratch Programına Giriş

Scratch programını internete bağlı olmadan (offline) ya da internete bağlı olarak (online) kullanabilirsiniz. İnternet üzerinden program yazmak için <https://scratch.mit.edu> adresini ziyaret edebilirsiniz.

Scratch programını indirip bilgisayarınıza kurmak için, <https://scratch.mit.edu/scratch2download> adresini ziyaret ederek programı kurabilirsiniz. Scratch programını kurmadan önce Adobe AIR programını kurunuz.

## Scratch Offline Editör



Scratch'da bloklarımızı sürükleyip bırak yöntemiyle kod yazma alanına taşıyoruz. Yazdığımız kodlar etkisini sahne kısmında gösterir.



## Sahne

Tam Ekran Butonu

Başlat Butonu

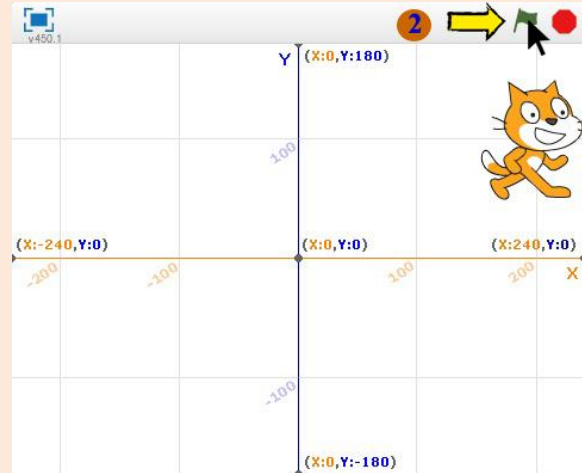
Durdur Butonu



Fare imlecinin o anki koordinatı

Sahne üzerindeki karakterin bulunduğu konumu kontrol etmek için bir koordinat sistemi vardır. Bu koordinat sistemini, yatay ve dikey, iki doğru olarak düşünebiliriz. Yatay doğruya X doğruyu, dikey doğruya ise Y doğruyu diyeceğiz. X doğrusunda 480 nokta, Y doğrusunda ise 360 nokta vardır.

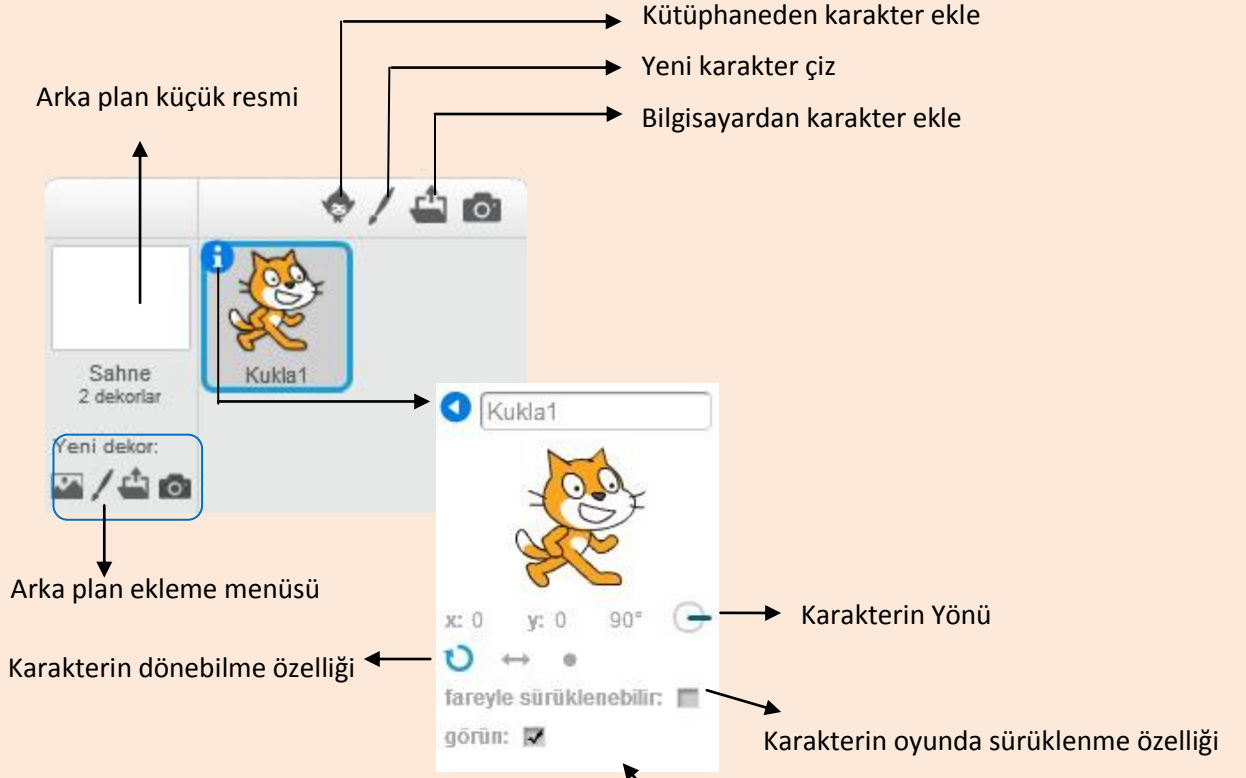
Karakterin bulunduğu konum X ve Y doğrularının kesişim noktasıdır. Burada karakterimiz X=100, Y=100 noktasındadır.



Koordinat sistemini programlarımızda sıkça kullanacağız. Koordinat sistemi ile karakterimizin hareketlerini kontrol ederiz. Karakterin ileriye doğru hareket etmesi aslında X konumunun artması, geriye doğru hareket etmesi ise aslında X konumunun azalmasıdır. Benzer bir durum Y koordinatı içinde geçerlidir. Bir oyunda karakter zıpladığında aslında Y konumu önce artar daha sonra azalarak eski konumuna gelir.



## Karakter Bölümü



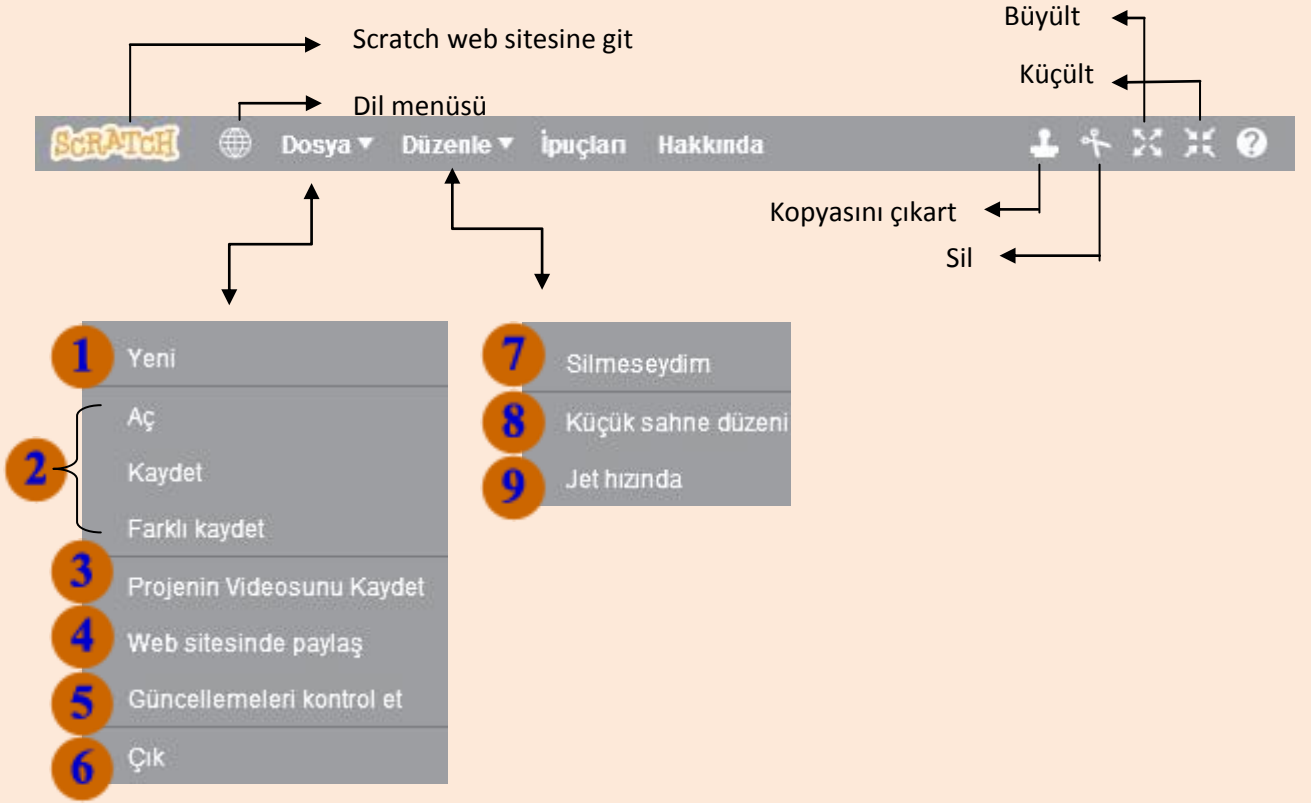
Scratch'da oyunlarımızı yaparken karakterler kullanacağız. Bu karakterleri programlayarak projelerimizi hayata geçireceğiz.

- 1 Karakterimizin içindeki tüm kodlar ve içeriklerle birlikte bir kopyasını çıkartır.
- 2 Karakterimizi siler.
- 3 Karakterimizi bilgisayara kaydeder.
- 4 Karakterimiz sahnede üzerinde görünüp gizlenebilir. Bu menü ile karakterimizi gizleyebiliriz.

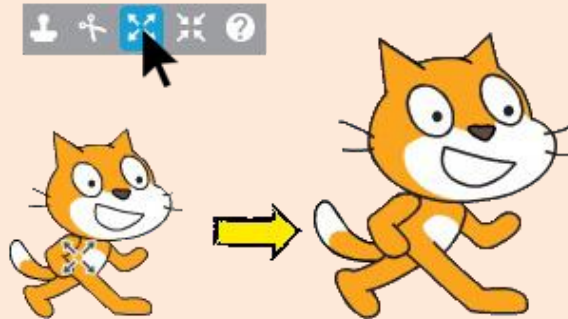


**Karaktere sağ tıklandığında açılan menü**

## Menüler



Menüleri kullanarak 1 yeni bir çalışma sayfası açabiliriz. 2 Çalışmamızı kaydedebilir ya da mevcut olan bir çalışmayı açabiliriz. 3 Yaptığımız projenin videosunu kaydedebiliriz. 4 Web sitesinde herkese açık bir şekilde paylaşabiliriz. Bunun için Scratch web sitesine kayıt olmamız gerekmektedir. 5 Program için güncelleme olup olmadığını kontrol edebiliriz. 6 Programdan çıkış yapabiliriz. 7 Yanlışlıkla sildiğimiz blokları geri getirebiliriz. 8 Sahneyi küçültüp kod yazdığımız alanı büyütebiliriz. 9 Programımızın çalışma hızını arttırabiliriz.



### Büyült butonu kullanımı

## Diziler



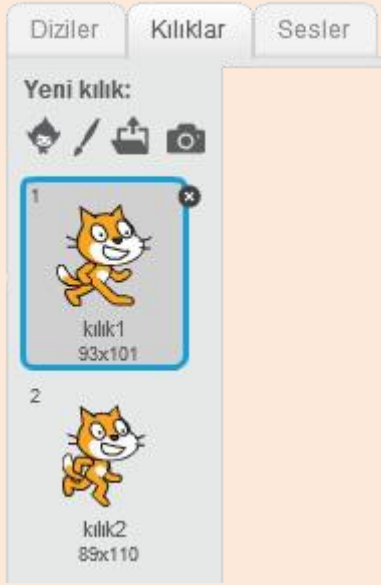
Scratch'da programlarımızı yaparken blokları kullanacağız. Bu bloklar Diziler menüsü altındadır. Bloklar kendi içlerinde gruplara ayrılmıştır. Örneğin; hareket menüsü karakterin sahnede hareketini sağlayan komutları içeren blokları barındırırken, Kontrol menüsü eğer ise gibi kontrol bloklarını bünyesinde barındırır.

Scratch'da kod blokları en üstten çalışmaya başlar. En üstte **Olaylar** menüsü blokları bulunur. Daha sonra sırasıyla alt kısımdaki kodlar çalışmaya başlar. Kod bloğu çalıştırdıktan sonra sıra bir alttakine geçer ve son bloğa kadar bu şekilde çalıştırılır.



Kodlar adım adım çalıştırılır. ❶ Programlar genelde yeşil bayrağa basılınca başlar. Sırasıyla diğer adımlara geçilir. Yeşil bayrağın altındaki ilk kod bloğu **10 adım git** olduğu için karakter ❷ 10 adım ileri gider. Daha sonra bir alttaki kod bloğu çalıştırılır. ❸ Karakter ekrana "Merhaba" yazar.

## Kılıklar



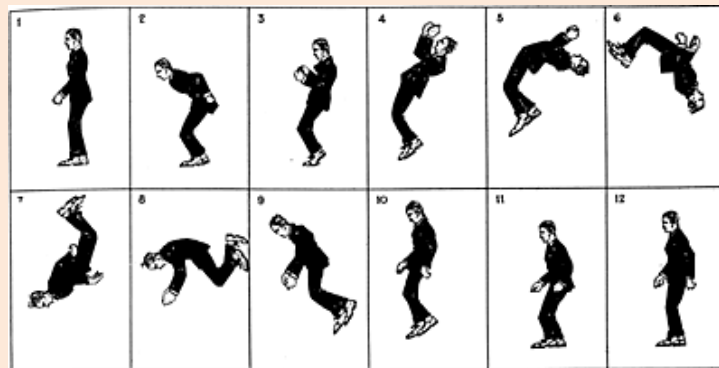
Karakterimize efekt vermek için kılıkları kullanırız. Karakterimize uçma ya da yürüme efekti verebiliriz. Kedi karakterimizin standart olarak iki kılığı vardır. Kılık1 ve kılık2 arasında geçiş yaparak yürüme efekti veririz. Karakterimiz önce kılık1 sonra kılık2'de gözüktüğünde, gözümüz onu yürüyor olarak görür.

## Çizgi Filmler Nasıl Yapılır?

Kağıtlarla çizgi filmler oluşturulurken bir çok resim çizilir. Her resimde karakterimiz bir hareketin devamını yapar. Bu resimler arka arkaya eklenir ve daha sonra hızlı bir şekilde aralarında geçiş yapılır. Gözümüz karakteri gerçekten hareket ediyormuş gibi görür.



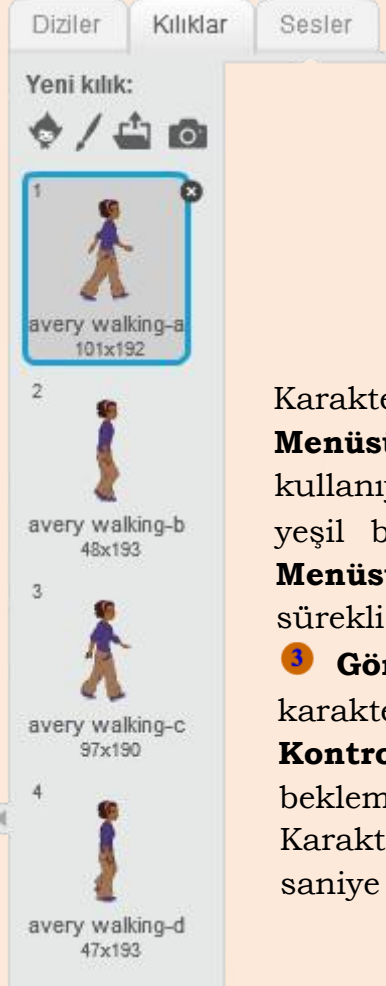
Scratch'da buna benzer bir şekilde kılıklar arasında geçiş yaparak karakterimize yürüme, zıplama ya da uçma efekti verebiliriz.



**Ters takla atan bir karakterin kılıkları**

Kütüphaneden karakter ekle butonu ile **Avery Walking** karakterini ekleyelim.

Karakterimiz, yürüme efekti verebilmek için dört adet kılığa sahip.



Karakterimize yürüme efekti vermek için **1 Olaylar Menüsü** altındaki “Yeşil bayrak tıklanınca” komutunu kullanıyoruz. Bu programımızın başlangıcı. Program biz yeşil bayrağa tıkladığımızda başlayacaktır. **2 Kontrol Menüsü** altındaki **sürekli tekrarla** bloğu ile efektimizin sürekli devam etmesini sağlayan bir döngü oluştururuz. **3 Görünüm Menüsü** altındaki **sonraki kılık** bloğu karakterimizin bir sonraki kılığa geçmesini sağlar. **4 Kontrol Menüsü** altındaki **1 saniye bekle** bloğu ile araya bekleme koyarak efektin çok hızlı gerçekleşmesini önleriz. Karakter bir sonraki kılığa geçerken her seferinde 0.2 saniye bekler.

Bu kısımdaki yazıları ya da rakamları değiştirebilirsiniz!

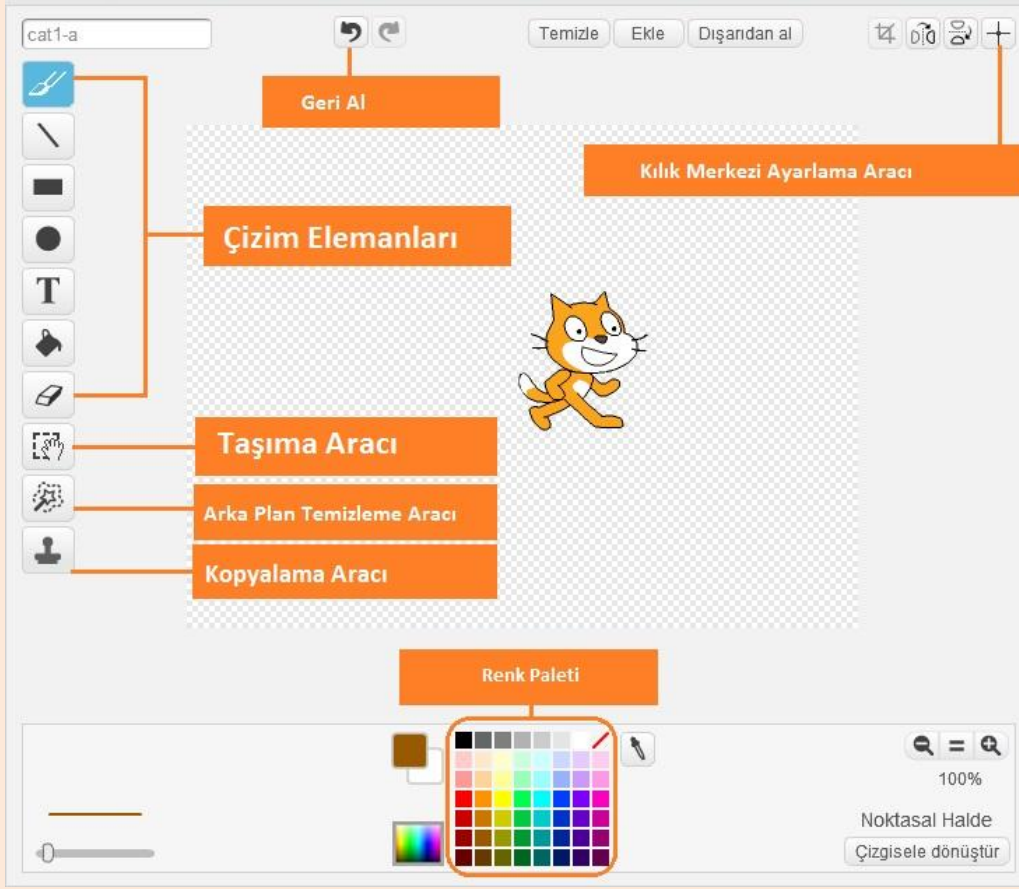


Olaylar

Kontrol

Görünüm

## Resim Düzenleme Aracı



Scratch karakterlerinin kılıklarını düzenlemek için Windows işletim sistemindeki Paint programına benzer bir Resim Düzenleme Aracı mevcuttur. Bu aracı kullanarak kılıklarımız rengini ve şeklini değiştirebilir ya da yazı ekleyebiliriz.

Resim düzenleme aracının noktasal ve çizgisel olarak iki türü mevcut. İsteğimiz dahilinde birini kullanabiliriz.



### Karakterin rengini değiştirme

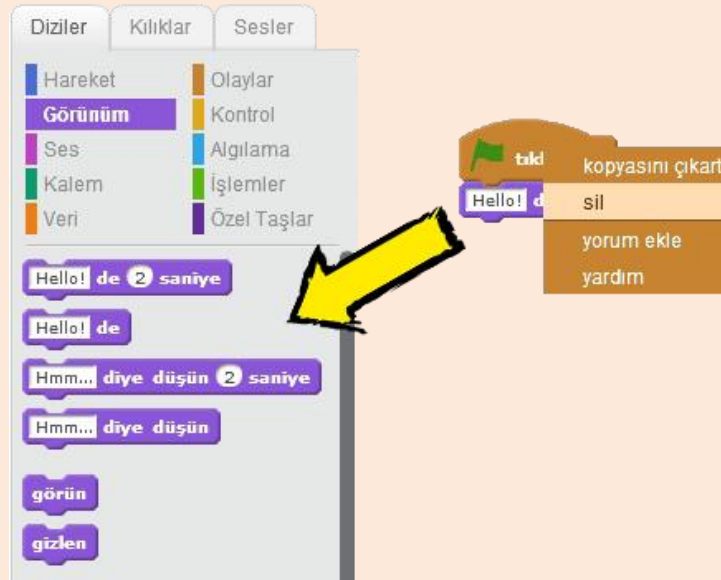
## Sesler



Programlarımıza 1 kütüphanedeki hazır sesleri ekleyebildiğimiz gibi 2 kendi kaydettiğimiz sesleri ya da 3 bilgisayarda kayıtlı bulunan ses dosyalarını ekleyebiliriz.



Programımıza bir sesi eklemek için öncelikle o sesi **Sesler** menüsünden eklemelisiniz. Ekledikten sonra **sesini çal** bloğu ile o sesi kullanabiliriz.



Kullanmadığımız kod bloklarını silmek için iki yöntemimiz mevcut. Birinci yöntem, kod bloğuna sağ tıklayıp silmek. İkincisi ise kod bloğunu sürükleyip diziler menüsündeki kod bloklarının olduğu yere bırakmak.

# HAREKET ve GÖRÜNÜM

## Hareket Blokları

Scratch'da bir sahne düzeni mevcut. Biz yazdığımız programlarla sahne üzerindeki karakteri kod blokları ile kontrol ederiz. Karakteri kontrol etmek için kullanılan kod blokları ise kendi içerisinde gruplara ayrılmıştır. Bu gruplar içerisinde en üstte bulunan ve mavi renge sahip olan bloklar hareket bloklarıdır. Hareket bloklarını, karakteri sahne üzerinde hareket ettirmek için kullanırız.



İlk olarak karakterimizi hareket ettirelim. Hareket bloklarından olan **10 adım git** git bloğu ile karakterimizin ileriye doğru hareket ettiririz. Burada yeşil bayrak tıkladığında karakter 10 adım ilerleyecektir.



Eğer karakterimizin sahne üzerinde sürekli olarak hareket etmesini istiyorsak. Bir tekrarlama bloğu kullanmamız gerekmektedir. Kontrol menüsü altındaki **sürekli tekrarla** bloğu ile karakterimizin sahne üzerinde sürekli hareket etmesini sağlayabiliriz.

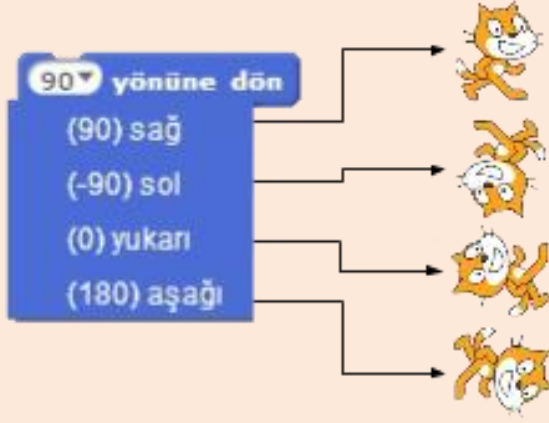


Karakterimiz sahne üzerinde hareket ederken sahne sonuna geldiğinde ilerleyemeyecektir. Bu sorunu çözmek için bir **kenara geldiysen sek** bloğu ile sahne sonuna geldiğinde geri dönmesini sağlarız.



Karakterimizin sahne sonuna gelip geri döndüğünde yönü değişeceği için ters dönecektir. Ters dönmesini engellemek içinse **kuklanın şekli** kod bloğunu kullanacağız.



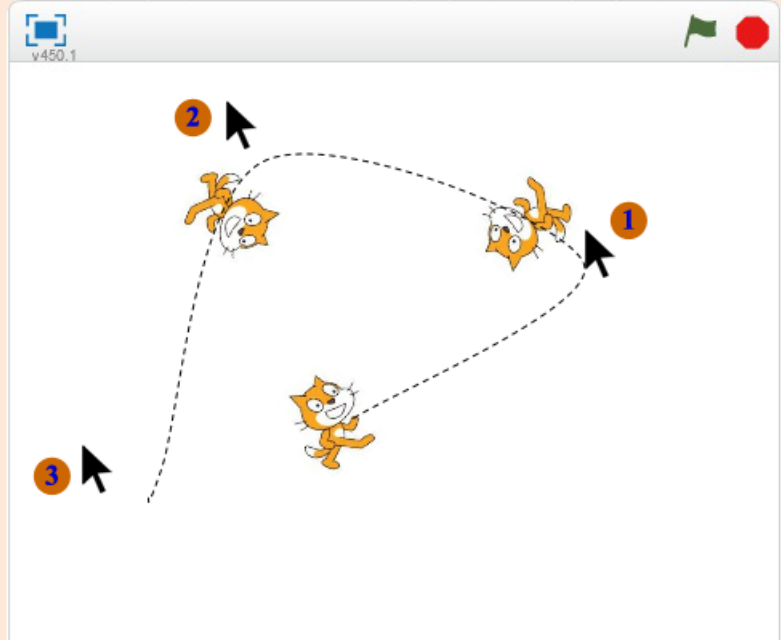


Karakterimizin sahne üzerinde yönleri vardır. Bu yönleri ayarlayarak karakterimizin hareketini kontrol ederiz. Her yönün bir açı karşılığı vardır. Karakterimize **90 yönüne dön** diyebildiğimiz gibi ara değerlerde verebiliriz. Örneğin; **45 yönüne dön** dediğimizde 0 ile 90 arasında yani yukarı ve sağ arasında bir yöne döner.

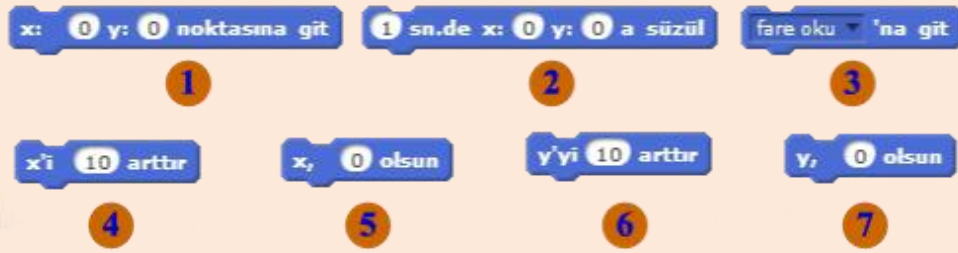


Karakterimizin küçük resmindeki info(bilgi) işaretine tıklarsak, karakter ile ilgili bilgileri görebiliriz.

- 1 Karakterimizin o anki X ve Y koordinatı
- 2 Karakterimizin o anki yönü
- 3 Her yöne dönebilsin
- 4 Sadece sağa sola dönebilsin
- 5 Hiçbir yöne dönmesin



İstedığımız takdirde karakterimizi belli bir yöne değil de fare okuna doğru döndürerek karakterimizi hareket ettirebiliriz.



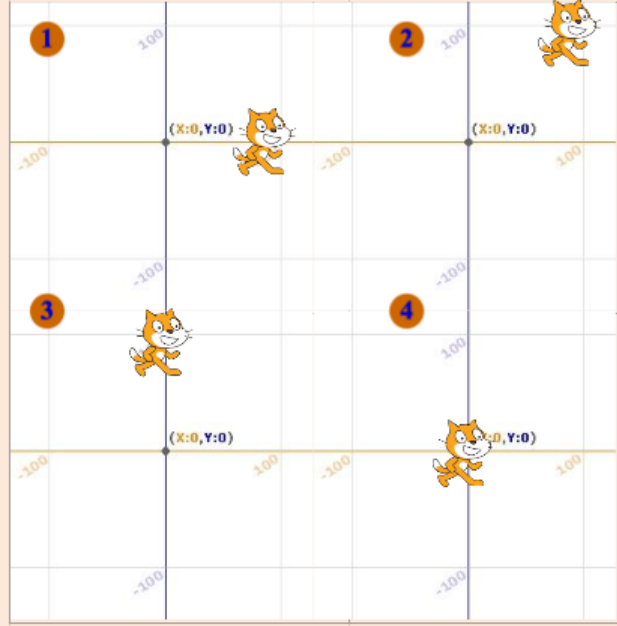
Sahne üzerinde bulunan her noktanın bir koordinatı vardır. Bu koordinat noktalarını kullanarak karakterimizi sahne üzerinde istediğimiz herhangi bir noktaya taşıyabiliriz.

1 Karakterimizi sahne üzerinde istediğimiz X ve Y konumuna taşıyabiliriz. 2 Bir animasyon ya da çizgi film tasarlıyorsak karakterimizin belirttiğimiz konuma yavaşça gitmesi gerekebilir. Bu durumda karakterimizin belirlediğimiz konuma belli bir saniyede gitmesini sağlayabiliriz. 3 Karakterimizin belli bir konuma değil de fare okunun bulunduğu konuma gitmesini sağlayabiliriz. 4 Koordinat düzleminde X konumunun artması karakteri ileriye götürür. Kod bloğundaki “10” rakamını negatif yaparsak “-10” bu sefer X konumu azalacağı için karakter geriye doğru gidecektir. 5 Karakterimizin belirlediğimiz X konumuna gitmesini sağlayabiliriz. 6 X konumu için yaptığımız işlemler Y konumu içinde geçerlidir. Y konumu artan karakter yukarıya doğru hareket edecektir. Kod bloğundaki “10” rakamını negatif yaparsak “-10” bu sefer Y konumu azalacağı için karakter aşağıya doğru hareket eder. 7 Karakterimizin Y konumunu istediğimiz konum olarak ayarlayabiliriz.

```

tıklanınca
x: 100 y: 0 noktasına git
1 saniye bekle
x: 100 y: 100 noktasına git
1 saniye bekle
x: 0 y: 100 noktasına git
1 saniye bekle
x: 0 y: 0 noktasına git

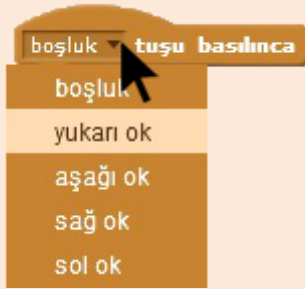
```



## Yön Tuşlarıyla Karakter Kontrolü

Programımız genel olarak yeşil bayrağa tıklayınca başlar; ancak bazı kod bloklarının başlaması için başka olayların gerçekleşmesi gerekebilir. Örneğin; fareye tıklamak, boşluk tuşuna basmak gibi. Bu örneğimizde kod bloklarımız yön tuşlarına basıldığında çalışacak.

Önce **Olaylar** menüsünden **boşluk tuşuna basılınca** kod bloğunu alalım.



Boşluk tuşunu başka tuşlarla değiştirebiliriz. Seçtiğimiz tuş basılınca kod bloğunun altına yerleştirdiğimiz kod blokları çalışacaktır.

```

yukarı ok tuşu basılınca
y'yi 10 arttır

```

```

aşağı ok tuşu basılınca
y'yi -10 arttır

```

```

sağ ok tuşu basılınca
x'i 10 arttır

```

```

sol ok tuşu basılınca
x'i -10 arttır

```

Yukarıdaki kodları, kod yazma alanına yazalım. Bu kodlar karakterimizin sahne üzerindeki hareketini kontrol edecektir. Yukarı ok tuşu basıldığında karakterin Y konumu 10 artacak, karakter yukarıya doğru hareket edecektir. Aşağı ok tuşu basıldığında Y konumu -10 artacak yani 10 azalacaktır. Y konumuna benzer şekilde sağ ok tuşu karakterin X konumunu arttırırken, sol ok tuşu karakterin X konumunu azaltacaktır. Bu şekilde karakterimiz sahne üzerinde ileri, geri,yukarı,aşağı yönlerinde hareket edecektir.

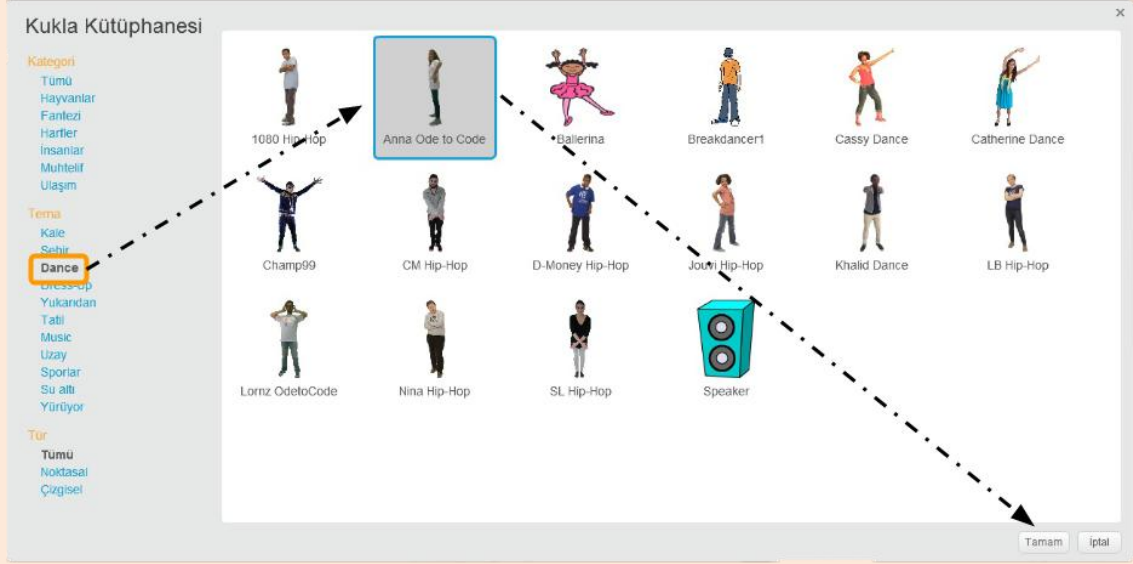
## Görünüm Blokları

Karakterimizin sahne üzerinde nasıl görüneceğini, görünüm blokları ile ayarlarız. Görünüm blokları ile karakterimizi konuşturabilir, rengini ya da kılığını değiştirebilir, ekranda görünmesini veya gizlenmesini sağlayabiliriz.



## Dans Partisi

Programımız için **kuklayı kütüphaneden seç** butonuna tıklayalım. Dance bölümünden bir karakter ekleyelim.

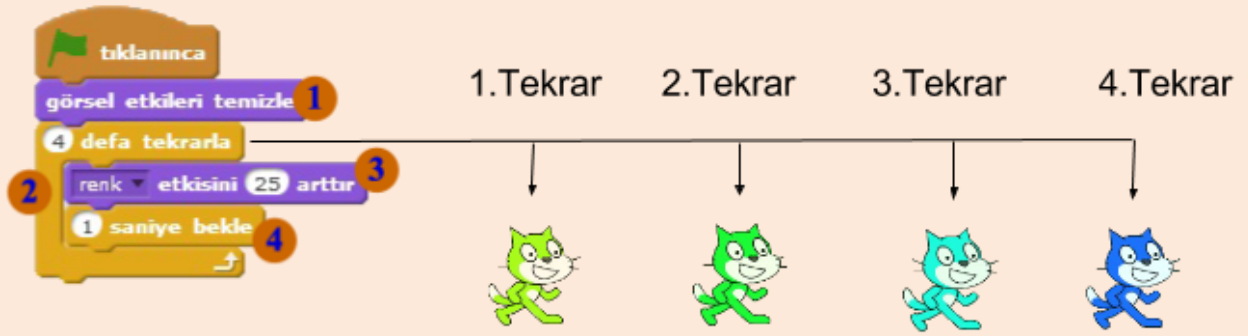


boşluk tuşu basılınca  
sonraki kılık

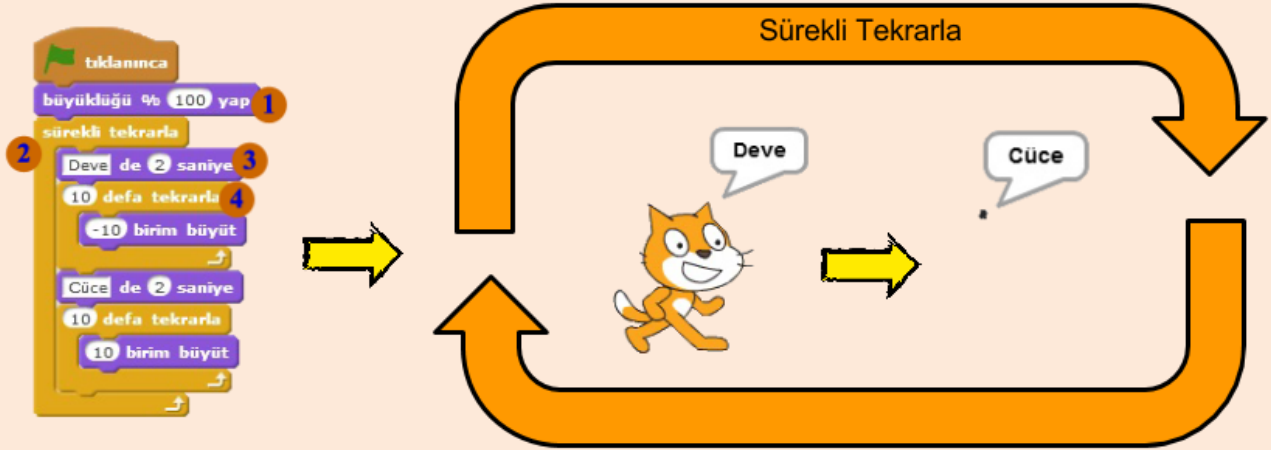
Yan taraftaki kod bloğu, karakteri ekleyince otomatik olarak eklenir. Karakterimizin 12 adet kılığı mevcut. Eklediğimiz **sonraki kılık** kod bloğu ile kılıklar arasında geçiş yaparak dans etme animasyonu yaparız. Boşluk tuşuna her bastığımızda karakterimizin kılığı değişecektir.

renk etkisini 25 artır  
renk  
balık gözü  
fırında  
benekleştire  
mozaik  
parlaklık  
hayalet

Görünüm bloklarını kullanarak karakterimizin rengini, parlaklığını ya da ekranda görünme etkisini(hayalet) ayarlayabiliriz.



Programımızda ① **görsel etkileri temizle** bloğu ile daha önce değiştirilmiş görsel etkileri sıfırlarız. Uyguladığımız bütün görsel etkiler silinir. ② **Kontrol menüsü** altında bulunan **10 defa tekrarla** kod bloğu ile istediğimiz sayıda döngü oluşturabiliriz. Biz programımızda 4 defa tekrarla diyerek dört adet döngü oluşturduk. Döngü içerisinde her adımda ③ **renk etkisini 25 arttır** kod bloğu çalışır. Her döngüde karakterimizin rengi değişir. ④ Kontrol menüsü altında bulunan **1 saniye bekle** kod bloğunu programımızın çok hızlı çalışmaması için kullanırız. Bilgisayarlar çok hızlı çalışan elektronik aletlerdir ve bizim gözümüz çok hızlı olan değişimleri seçemez. Bu yüzden araya bir saniye bekleme koyarak programı yavaşlatırız.

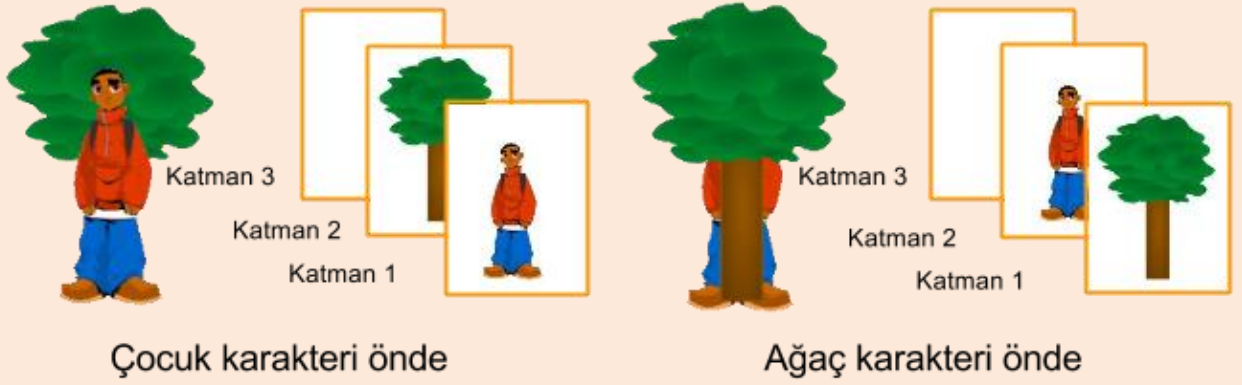


Karakterimizin büyüklüğünü ayarlamak istediğimizde **büüklüğü %... yap** ve **10 birim büyüt** kod bloklarını kullanırız.

Programımız yeşil bayrağa basılınca başlar. ① İlk adımda **büüklüğü %100 yap** kod bloğuyla büyüklüğü orijinal büyüklüğe getiririz. ② İkinci adımda sürekli tekrarla döngüsü çalışır. Sürekli tekrarla döngüsü içerisinde bulunan kod blokları program çalıştığı sürece sürekli tekrarlanır. ③ Sürekli tekrarla bloğu çalışmaya başladıktan sonra ilk olarak en üstte bulunan **Deve de 2 saniye** kod bloğu çalışır ve

karakterimiz ekrana Deve yazısını yazar. 4 Daha sonra bir alttaki kod bloğuna geçer ve **10 defa tekrar** kod bloğu çalışmaya başlar. On defa - **10 birim büyüt** kod bloğu çalışır. Bu kod bloğunu karakterimizin büyüklüğünü arttırmak ya da azaltmak için kullanırız. Burada -10 birim büyüt dediğimiz için her adımda karakterimizin büyüklüğü 10 birim azalacaktır ve karakterimiz küçülecektir. Daha sonra bir alt satıra geçilerek **Cüce de 2 saniye** kod bloğu ile ekrana Cüce yazdırılacaktır. Sürekli tekrarlar bloğumuzun içindeki son kod bloğu ile on defa, on birim büyüt kod bloğu çalışacak ve karakterimiz büyüyerek eski haline dönecektir. Bu durum program çalıştığı sürece sürekli olarak tekrarlanacaktır.

## Katmanlar



Sahne üzerinde bir katman yapısı vardır. En üst katmanda bulunan karakter en önde gözükürken onun altındaki katmanda bulunan karakter ikinci sırada gözükür.



**Üste çık** komutu ile karakteri en üst katmana çıkarırken **1 katman alta in** komutu ile bir alt katmana indiririz.

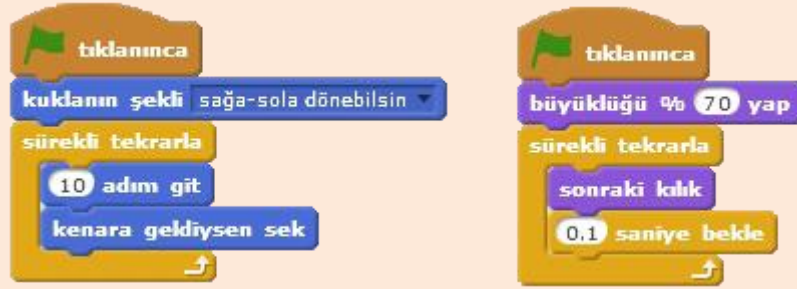
## Uçan Yarasa



Bat2-a Kılığı

Bat2-b Kılığı

Şimdi **kuklayı kütüphaneden seç** butonu ile yeni bir karakter ekleyelim. Hayvanlar menüsü altında bulunan Bat2 karakterini ekleyelim.



Bat2 karakteri için yazılacak kodlar

## Paralel Programlama

Paralel programlama, iki kod bloğunun aynı anda çalıştırılmasıdır. Scratch'da aynı anda çalışmasını istediğimiz kod bloklarını ayrı yeşil bayrak bloklarının altına yerleştiririz. Bu bloklar yeşil bayrağa tıklandığında aynı anda çalışmaya başlar.

Yeşil bayrağa tıklandığında **kuklanın şekli sağa-sola dönebilsin** ve **büyüklüğü %70 yap** kod blokları aynı anda çalışmaya başlar. Daha sonra iki sürekli tekrarla bloğu da birlikte çalışmaya başlar. Karakterimiz aynı anda 10 adım giderken sonraki kılığa geçer ve uçma etkisi verilir. Bu işlemler paralel olarak yürütülür.