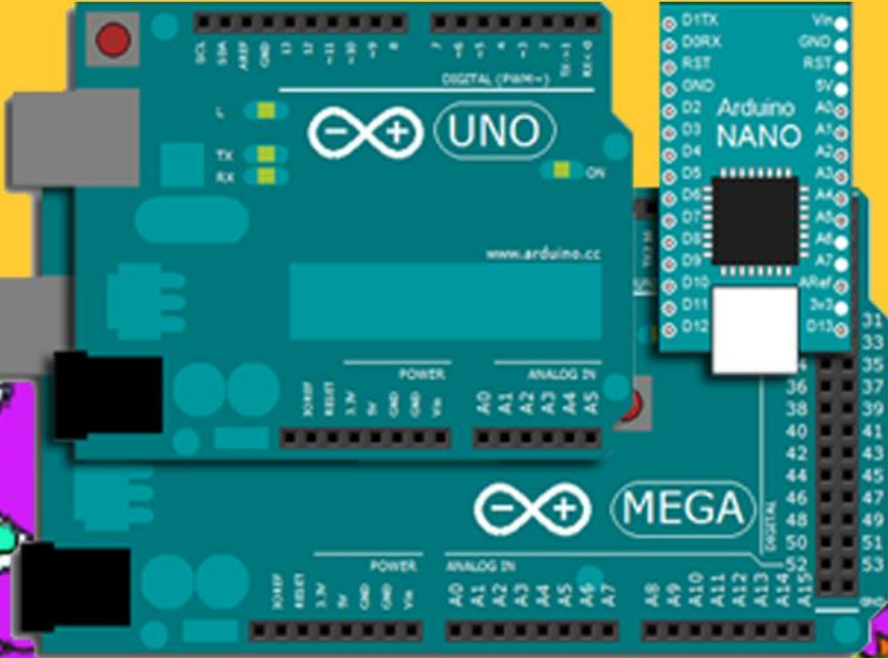




Scratch ile Programlamaya Giriş

ve

Scratch ile Arduino Programlama



Mahmut BORA KARAKUŞ



Bütün ümidim gençliktedir.

Mustafa Kemal ATATÜRK



Hayallerinle Dünyanı Programla

Bu kitap Ali Keşfet projesi kapsamında yazılmıştır.

<http://www.alikesfet.org/>



Programlama Nedir?

Programlama: Bilgisayarın ne yapması gerektiğinin adım adım söylenmesidir. Bilgisayarlar bizim isteklerimiz doğrultusunda çalışan elektronik aletlerdir. İstedığımız iş ve işlemleri yapabilmeleri için programlanmaları gerekir. Programlama ile biz bilgisayara işleri nasıl yapması gerektiğini anlatırız.

Bilgisayarda çalışan bir çok program vardır. Bu programlar programlama dilleri kullanılarak yazılır. İnsanların günlük hayatta iletişim kurmak için nasıl bir dil kullanmaları gerekiyorsa bizim de bilgisayarlar ile haberleşebilmemiz için bir dil kullanmamız gerekir. Bu diller yazılı dillerdir . Üst seviye(kolay) ve alt seviye(zor) diller vardır. Okullarımızda okutulan Bilişim Teknolojileri ve Yazılım derslerinde öğrendiğimiz Scratch programlama dili üst seviye(Kolay) bir programlama dilidir.



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.

Yazar Hakkında



- 1985 Mersin doğumluyum.
- Çanakkale 18 Mart Üniversitesi Bilgisayar ve Öğretim Teknolojileri Öğretmenliği mezunuyum.
- Ahmet Yesevi Üniversitesi Bilgisayar Mühendisliği 2. Sınıf öğrencisiyim.
- Bir devlet okulunda Bilişim Teknolojileri ve Yazılım Dersi öğretmenliği yapmaktayım.



Scratch ile Programlamaya Giriş ve Scratch ile Robot Programlama E-Kitabı Creative Commons Alıntı-Gayriticari-Türetilemez 4.0 Uluslararası Lisansı ile lisanslanmıştır.

www.bilbilimbilisim.com

(Kaynak gösterilmeden paylaşamaz, Ticari olarak kullanılamaz ve Üzerinde değişiklik yapılamaz)

Scratch ile Programlamaya Giriş

Algoritma Nedir?	Sayfa 7
Code.org	Sayfa 8
Scratch Programına Giriş	Sayfa 13
Scratch Editörünü Tanıyalım	Sayfa 15
Scratch Bloklarını Tanıyalım	Sayfa 17
Görünüm Bloklarını Tanıyalım	Sayfa 18
Çalışmamıza Ses Ekleyelim	Sayfa 20
Kalem Menüsü	Sayfa 20
Olaylar Menüsü	Sayfa 22
Kontrol Menüsü	Sayfa 23
İşlemler Menüsü	Sayfa 26
Veri Menüsü	Sayfa 28
İkiz Yaratma	Sayfa 31
Hücre Modeli	Sayfa 33
Balık Yeme	Sayfa 34
Sayı Tahmin Oyunu	Sayfa 36
Basit Pinpon Oyunu	Sayfa 37
Renklerin Dansı	Sayfa 38
Basit Labirent Oyunu	Sayfa 40
Liste Elemanı ile Müzik Besteleme	Sayfa 44
Basit Flappy Bird	Sayfa 46
Scratcher Olmak	Sayfa 49
Bulut Değişken	Sayfa 51
Özel Taşlar	Sayfa 52

Scratch ile Arduino Programlama

Arduino Nedir?	Sayfa 53
Arduino'yu Nasıl Programlarız?	Sayfa 54
Arduino'muzu Tanıyalım	Sayfa 55
Arduino'yu Bilgisayara Bağlama	Sayfa 56
Mblock Programını Tanıma	Sayfa 57
Temel Elektroink	Sayfa 59
Elektronik Devre Elemanları	Sayfa 60
Breadboard Üzerindeki Led'i Yakma	Sayfa 62
Kara Şimşek Devresi	Sayfa 63
Dijital Pinlerden Değer Okuma	Sayfa 64
RGB Led	Sayfa 65
Potansiyometre	Sayfa 66
Sensörler	Sayfa 67
Hava Kararınca Yanan Led	Sayfa 68
Mesafe Sensörü	Sayfa 70
Scratch ile Yazdığımız Programı Arduino'ya Yükleme	Sayfa 71

Algoritma Nedir?

Algoritma: Problemin adım adım çözülmesidir. Bilgisayarda programlar yazılırken belirli bir sıra izlenir ve adım adım yazılır. Bu işlemler sırasında algoritmalar kullanılır. Örnek olarak dişlerimizi fırçalamamızın algoritmasını çıkaralım.



Kirli	Fırçaya	Dişlerini	Temiz
Dişler	Diş Macunu Sür	Fırçala	Dişler

Algoritmamızın her zaman bir başlangıcı ve sonu olmalı. Burada diş fırçalama algoritmamızı başlatan olay kirli dişler. Adım adım düşünecek olursak.

1. Başla (Kirli Dişler)
2. Fırçaya diş macunu sür
3. Dişlerini fırçala
4. Son (Temiz Dişler)

Algoritmalar ile olayları basamaklara böleriz ve sırasıyla bu basamakları izleriz. Böylece yapacağımız işleri daha detaylı düşünür ve daha kolay yaparız.

Kitabımızın bu bölümünde www.code.org internet sitesini ziyaret edeceğiz ve algoritmaları kullanarak Angry Birds oyununu oynayacağız.

Oyunun linki:

<https://studio.code.org/hoc/1>

Oyunun amacı kötü domuzcuğu uzaklaştırmak.

Çalıştığı zaman

ilerle

Angry Birds kötü domuzcuğu uzaklaştırmak istiyor ancak nasıl yapacağını bilmiyor. Biz ise komutları kullanarak Angry Birds'e yardımcı olacağız. Çalıştığı zaman kısmı algoritmamızın başlangıcı.

ilerle

sola dön

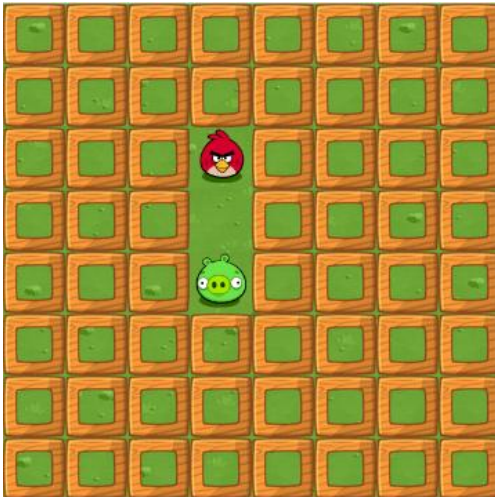
sağa dön


İlerle komutu Angry Birds'i bir adım ileriye götürür. Sola dön ile Angry Birds'in sola, Sağa dön ile de Angry Birds'in sağa dönmesini sağlarız. Burada dikkat etmemiz gereken nokta adım adım düşündürmektir.

Çalıştığı zaman

ilerle

ilerle



Bu bölümde Angry Birds'in domuzcuğa ulaşabilmesi için ilerlemesi gerekmektedir. İki defa "ilerle" komutunu çalıştığı zaman bloğunun altına sürükleyip bırakıyoruz ve  Çalıştır Butonuna basıyoruz.

İlk algoritmamızı geliştirdik. Gördüğünüz gibi çok kolay.

Saat Kodu Bölüm 3

Bölüm linki: <https://studio.code.org/hoc/3>



Angry Birds'e yardım etmeye devam edelim. Şimdi düşünelim Angry Birds domuzcuğa ulaşmak için hangi adımları takip etmeli?

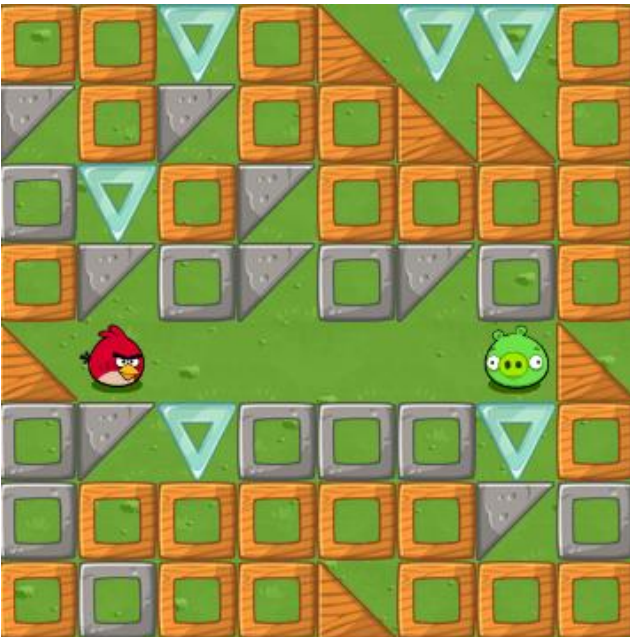
Hep birlikte önce algoritmasını oluşturalım.

1. Başla (Çalıştığı zaman)
2. İlerle
3. İlerle
4. Sağa dön
5. İlerle



Saat Kodu Bölüm 6

Bölüm linki: <https://studio.code.org/hoc/6>



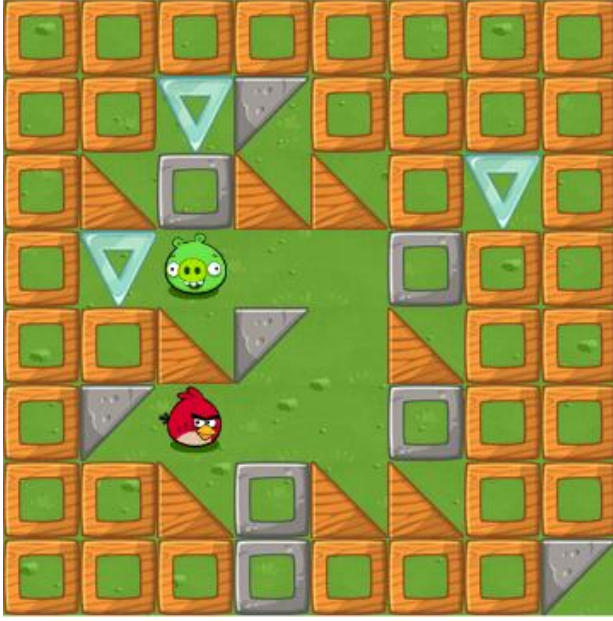
Bilgisayarda program yazarken bazı işleri tekrar tekrar yapmamız gerekebilir. Bu durumda alt alta aynı komutları kullanmak yerine tekrarlanacak komutları bir döngü(tekerrar) komutu kullanarak tekrarlatırız.

Burada angry Birds'in 5 kez ilerlemesi gerekmektedir. 5 kez alt alta "ilerle" komutu kullanmak yerine. Bir döngü komutu kullanarak "ilerle" komutunu 5 kez tekrarlatacağız



Saat Kodu Bölüm 11

Bölüm linki: <https://studio.code.org/hoc/11>



Bazı durumlarda programımızı yazarken tekrar sayısı belirtmek yerine bir durum gerçekleşinceye kadar tekrarları diyebiliriz. Burada da bazı komutları domuzcuğa varana kadar tekrarları dememiz gerekiyor.

Burada bazı komutlar 3 kez tekrarlanıyor hep birlikte tekrarlanacak komutları düşünelim.

Angry Birds “ilerle” , “ilerle” , “sola dön” komutlarını takip etmeli.

Bir önceki örneğimizden farklı olarak burada tekrar sayısı belirtmeyeceğiz. Domuzcuğa varana kadar tekrarları diyeceğiz. Komutlarımız domuzcuğa varana kadar tekrarlanacak.



Saat Kodu Bölüm 14

Bölü linki: <https://studio.code.org/hoc/14>

Kontrol Yapıları (Eğer ise)



Bizim günlük hayatta bazı kararlar almamız gerektiği gibi bilgisayarların da bazı kararlar almaları gerekir. Örneğin; biz havaya bakarız ve eğer hava sıcaksa tişört giyemiz. Burada bir koşulumuz var. Havanın sıcak olması. Koşulumuz gerçekleştiğinde ise bir eylem gerçekleştiriyoruz yani tişört giyiyoruz. Bu bölümde de karakterimiz buna benzer bir şekilde, bir koşula bağlı olarak karar alacak ve uygulayacak.

Karakterimiz çiçeğe varana kadar ilerleyecek ve sola doğru yol olup olmadığını kontrol edecek ve eğer sola doğru yol varsa sola doğru dönecek. Eğer sola doğru yol yoksa sadece ilerleyecek.



Saat Kodu Bölüm 18

Bölüm linki: <https://studio.code.org/hoc/18>

Kontrol Yapıları (Eğer ise / Değilse)



Eğer ise kontrol yapısında koşulumuz gerçekleştiğinde yapılacak işler vardı. Eğer ise/ değilse kontrol yapısında ise koşulumuz gerçekleştiğinde yapılacak işler ve koşulumuz gerçekleşmediğinde yapılacak işler vardır.

Bu bölümde karakterimizin ilerideki yolu kontrol etmesi ve ileride yol varsa ilerlemesi gerekir. Ama ileride yol yoksa bu sefer başka bir karar alarak sola dönmesi gerekir.



Scratch Programına Giriş

Scratch programlama dilini internete bağılı olmadan (offline) ya da internete bağılı olarak (online) kullanabilirsiniz. Biz kitabımızda projelerimizi kolaylıkla kaydetmek ve paylaşmak için online olarak kullanacağız.

İnternete bağılanılmayan (offline) versiyonu kurmak isteyenler aşığıdaki linkten programı bilgisayarlarına kurabilirler.

Scratch programını kurmadan önce Adobe Air programını kurmalısınız.

Adobe Air indirme linki:

<http://get.adobe.com/air/>

Scratch indirme linki:

<https://scratch.mit.edu/scratchr2/static/sa/Scratch-440.exe>

Not: Linkler Windows içindir. Dięer işletim sistemleri için:

<https://scratch.mit.edu/scratch2download/>

Scratch İnternet Sitesine Kayıt Olmak

Scratch ile online olarak projelerimizi geliştirdikten sonra kaydetmek ve paylaşmak için kayıt olamamız gerekmektedir.

Scratch'a kayıt olmak için: <https://scratch.mit.edu/>

Adresine girdikten sonra  butonuna tıklıyoruz.

Gelen menüden kullanıcı adı ve şifre belirliyoruz. Kullanıcı adınızda Türkçeye has karakterler(ş,ç,ğ,ö,ü,ı) kullanmayınız.

Scratch'a Katıl

X

Scratch'da bir hesap açmak hem çok kolay hem de bedava.

Bir Scratch Kullanıcı Adı Belirle

Bir Şifre seç

Şifre Sağlaması



1 2 3 4

Sonraki

Daha sonra 2. Kısımdan doğum yılı, cinsiyet ve ülke bilgilerini giriyoruz.

Scratch'a Katıl

X

Bu sorulara verdiğin cevaplar gizli tutulacak.

Neden bu bilgileri istiyoruz ?

Doğum Ayı ve Yılı

Cinsiyet Erkek Kız

Ülke



1 2 3 4

Sonraki

E-posta kısmı önemli, projelerinizi paylaşabilmeniz için geçerli bir e-posta adresi girmeli ve adresinizi onaylamalısınız.

Scratch'a Katıl

E-posta adresini gir. Hesabını onaylaman için sana bir e-posta göndereceğiz.

E-posta adresi

E-posta adresini doğrula



1 2 3 4

Sonraki

Scratch'a Katıl

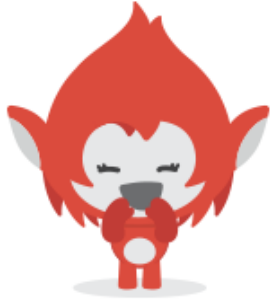
Scratch'e Hoşgeldin! **Kullanıcı adın!**

Kayıt oldun! Artık keşfetmeye ve projeler oluşturmaya başlayabilirsin.

Paylaşmak ve yorum yapmak istiyorsan,
e-posta adresin 'de sana gönderdiğimiz linke tıkla..

Yanlış e-posta mı? E-posta adresini [Hesap Ayarları](#)ndan değiştirebilirsin.

Having problems? [Please give us feedback](#)



1 2 3 4

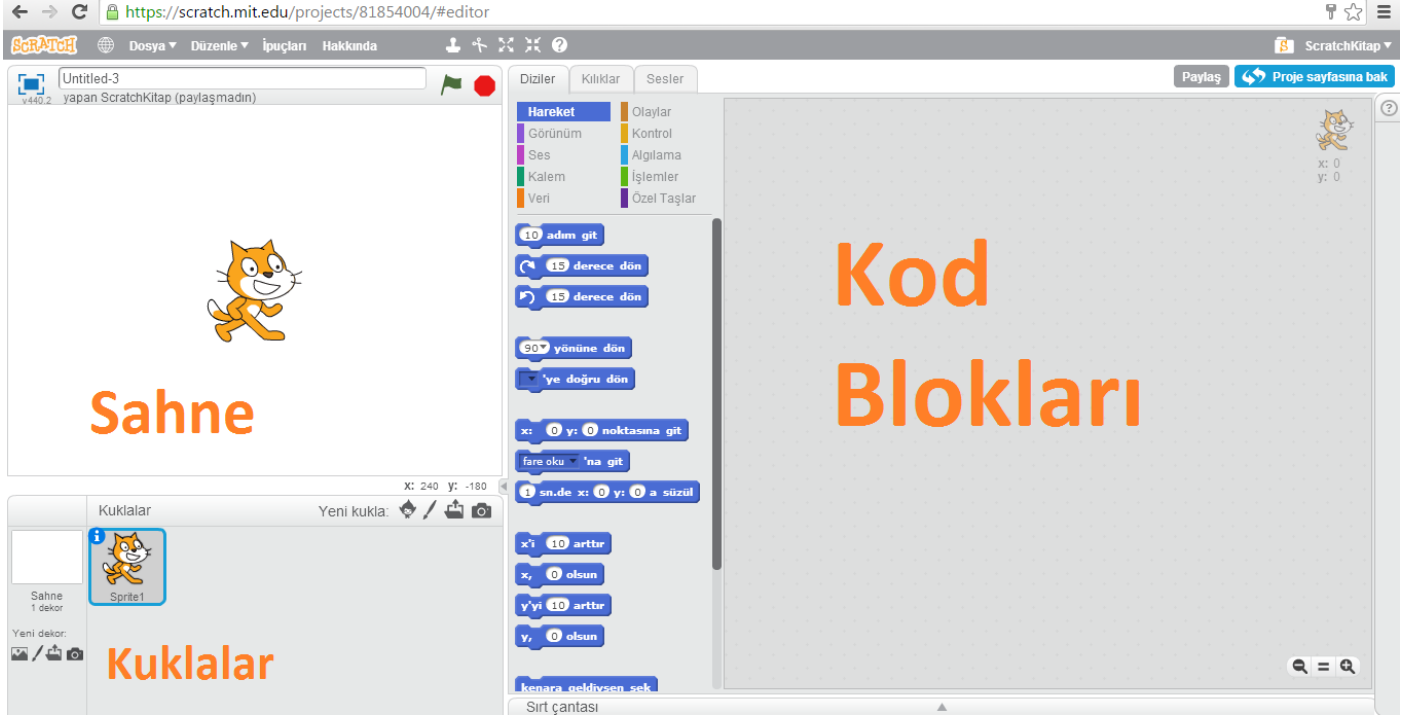
Tamam Başlayalım!

Tebrikler Scratch ailesine katıldın. Scratch ile çalışmak için “Tamam Başlayalım!” butonuna tıklayalım.



Üst menüden “Oluştur” kısmını seçelim.

Scratch Editörünü Tanıyalım



Scratch'da programlar yazılırken kod blokları kullanılır. Her blok bir komutu temsil eder ve lego yapar gibi blokları birleştirirsiniz. Eğer uygun blokları bir araya getirirseniz ortaya çok güzel bir program, animasyon veya oyun çıkar.

Scratch'da kuklalar ve sahneler vardır. Bu kukla ve sahneleri programlayarak projelerimizi gerçekleştireceğiz.

[Youtube Videosu İçin Tıklayınız!](#)



Scratch Bloklarını Tanıyalım

10 adım git

Karakteri 10 adım ileriye götürür.

15 derece dön

Karakteri 15 derece sağa döndürür.

15 derece dön

Karakteri 15 derece sola döndürür.

90 yönüne dön

(90) sağ

(-90) sol

(0) yukarı

(180) aşağı

Karakterin yönünü değiştirir. Örneğin Karaktere 0 yönüne dön 10 adım git komutlarını uygularsak, karakter yukarıya doğru dönüp 10 adım ilerler.



'ye doğru dön

fare oku

Karakteri fare okuna doğru döndürür

x: 0 y: 10 noktasına git

Karakteri $x=0$ $y=0$ noktasına götürür

fare oku 'na git

Karakteri fare okunun konumuna götürür

1 sn.de x: 0 y: 10 a süzül

Belirtilen saniyede karakteri x, y konumlarına götürür

x'i 10 arttır

X'i arttır ile x konumunu arttırırız. X ... olsun ile x konumunu belirleriz. Aynı uygulamaları y konumları için de yaparız.

x, 0 olsun

y'yi 10 arttır

y, 0 olsun

kenara geldiysen sek

Karakterin sahnenin sonuna geldiğinde geri dönmesini sağlar.

[Youtube Videosu İçin Tıklayınız!](#)



Görünüm Bloklarını Tanıyalım

Konuşma blokları ile karakterimize söyletmek istediklerimizi baloncuklar içerisinde söyleriz.

```
Hello! de 2 saniye
Hello! de
Hmm... diye düşün 2 saniye
Hmm... diye düşün
```

Scratch ile
Programlama
Öğreniyorum



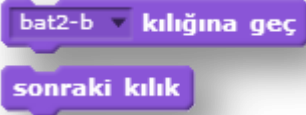
Görün ve gizlen blokları ile karakterimizin sahnede görünmesini veya görünmemesini sağlarız.



Bat2-a kılığı



Bat2-b kılığı



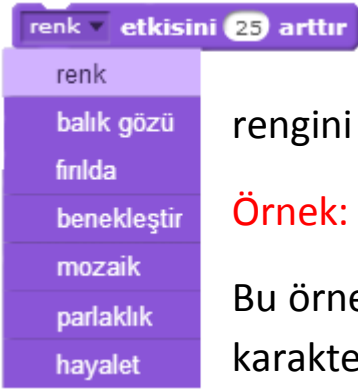
Kılık blokları ile istediğimiz kılıkları seçebilir ya da bir sonraki kılığa geçmesini sağlayabiliriz.

Örnek: Uçan Yarasa

Bu örneğimizde bir uçan yarasa yapacağız.

[Örnek İçin Tıklayınız!](#)

```
tıklanınca
sürekli tekrarla
10 adım git
sonraki kılık
kenara geldiysen sek
kuklanın şekli sağa-sola dönebilsin
```



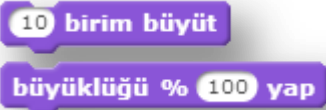
Görsel efekt bloklarımız ile karakterimizin rengini değiştirebiliyoruz veya değişik efektler ekleyebiliyoruz.

Örnek: Görsel Efektler

Bu örneğimizde yeşil bayrağa tıklandığında, 1 tuşuna basıldığında karakterin rengini, 2 tuşuna basıldığında balık gözü etkisini, 3 tuşuna basıldığında ise hayalet etkisini arttıracacağız.



[Örnek için tıklayınız!](#)



Bu bloklar ile karakterimizin büyüklüğünü ayarlayabiliriz.



Bu bloklar ile karakterimizin en alta ya da en üste görünmesini sağlayabiliriz. Detaylı bilgi için videoyu izleyebilirsiniz.

[Youtube Videosu İçin Tıklayınız!](#)



Çalışmamıza Ses Ekleyelim

meow sesini çal

Meow sesini çalar

meow sesini bitene kadar çal

Meow sesini bitene kadar çalar

tüm sesleri durdur

Tüm sesleri çalmayı durdurur



Çalışmamıza Scratch kütüphanesinden ses eklemek için karakterimiz seçiliyken sesler kısmından sesi kütüphaneden seç ile ekleriz.

Daha sonra sesini çal bloğunda eklediğimiz sesi seçeriz



Kütüphaneden ses eklemek için



Ses kaydetmek ve eklemek için



Bilgisayarda kayıtlı sesi eklemek için

Kalem menüsü

temizle

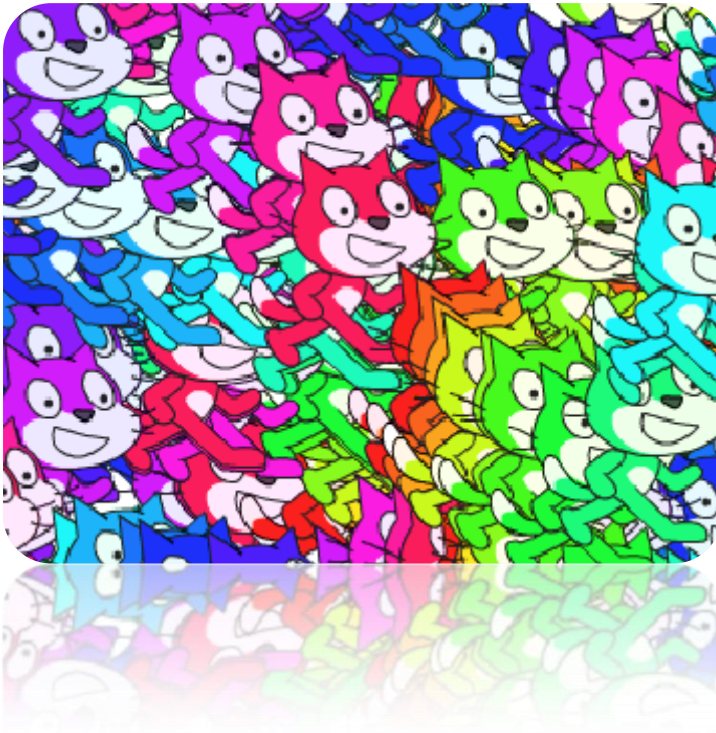
“Temizle” ile kalem efektlerini sileriz. “İz bırak” ile karakterin kopyasını ekrana bırakırız. “Kalemi bastır” ile kalem ile çizmeye başlar , “kalemi kaldır” ile çizmeyi bırakırız.

iz bırak

kalemi bastır

kalemi kaldır

Örnek: Karakterimizle değişik renklerde sahneye iz bırakalım. Bunu yaparken de karakter sürekli fareyi takip etsin ve karakterin rengi değişsin.



```
tıklanınca
temizle
sürekli tekrarla
  fare oku 'na git
  renk etkisini 10 arttır
iz bırak
```

Programımız yeşil bayrak ile başlıyor. Temizle bloğu ile önce kalem ile çizilenleri temizliyor ve sürekli olarak fare okunu takip ederken karakterin rengini değiştirip karakterin kopyasını çıkartıyor.

[Örnek İçin Tıklayınız!](#)

Örnek: Kare çizmece

Bu örneğimizde hareket blokları ve kalem blokları ile karakterimize kare çizdireceğiz.

```
tıklanınca
temizle
kalemi bastır
4 defa tekrarla
  100 adım git
  90 derece dön
```

Programımız yeşil bayrak ile başlar. Önce “temizle” bloğu ile ekranı temizler. “Kalemi bastır” bloğu ile şekil çizmeye hazır hale gelir. Karenin 4 kenarı olduğu için 4 defa 100 adım ilerler ve kare çizmek için 90 derece döner.

[Örnek İçin Tıklayınız!](#)

[Şekil Çizmece Örneği İçin Tıklayınız!](#)

[Youtube Videosu İçin Tıklayınız!](#)



Olaylar menüsü



Scratch'da yazdığımız programlar yeşil bayrak ile başlar. Bazen programımızın belli kısımlarının başlama koşulunun farklı olması gerekebilir. Bu durumda olaylar menüsü bloklarını kullanırız.

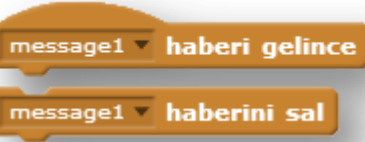
Örnek: Yön tuşları ile karakter kontrolü

Bu örneğimizde yön tuşlarını kullanarak karakterimizin X ve Y konumlarını arttırıp azaltarak hareket ettireceğiz.

[Örnek için Tıklayınız!](#)



Bu bloklar ile karakterlerimiz arasında mesajlaşma yapabiliriz.



[Mesaj Blokları Kullanımı Örneği için Tıklayınız!](#)



Algılama Menüsü

Bilgisayarda program yazarken, bazı olayların bazı durumlar gerçekleştiğinde yapılması gerekebilir. Bu durumda algılama bloklarını kullanacağız. Algılama bloklarını **kontrol** blokları ile birlikte kullanacağız.

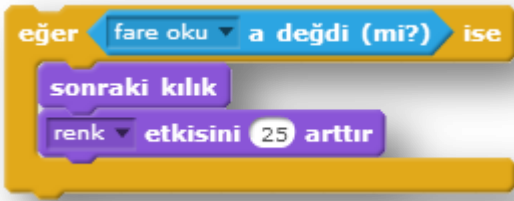
Kontrol menüsü

Eğer ise kontrol yapısı:



Bilgisayarda çalışan programların bazı kararlar almaları gerekebilir. Bu durumda eğer ise kontrol yapısını kullanırız.

Örnek: Eğer ise kontrol yapısı



Bu örneğimizde bir eğer ise yapısı ile karakterimize fare okunun değip değmediğini kontrol edeceğiz ve eğer fare oku değmişse sonraki kılığa geçmesini ve renk efektinin artmasını sağlayacağız.

Not: Bilgisayarlar çok hızlı çalışan elektronik aletlerdir. Biz bu örnekte eğer ise kontrol yapısını tek blok olarak kullanırsak çok hızlı bir şekilde sadece 1 kez kontrol edecek ve program duracaktır. Bunun için kontrol yapısını sürekli bloğunun içine alacağız ve tekrar tekrar kontrol edilmesini sağlayacağız.



[Eğer ise Yapısı Örneği İçin Tıklayınız!](#)

Örnek: Bu örneğimizde eklediğimiz başka bir karakterin kedi karakterine değip değmemesini koşul olarak belirleyeceğiz. Eğer Bat2 karakteri kedi karakterine değerse koşulumuz sağlanmış olacak ve eğer ise yapısı içerisindeki kod bloklarını yapacak.



[Eğer ise kontrol yapısı örneği 2 için tıklayınız!](#)

Eğer ise / değilse kontrol yapısı:



Eğer ise kontrol yapısında sadece koşulumuz gerçekleştiğinde yapılacaklar vardı. Eğer ise / değilse yapısında ise hem koşulumuz gerçekleştiğinde yapılacak olaylar hem de gerçekleşmediğinde yapılacak olaylar vardır.

Örnek: Fare tıkladığında gizlenen, tıklanmadığında ise gözüken kedi



Bu örneğimizde koşulumuz farenin basılı olması. Eğer fare basılı ise koşulumuz gerçekleşmiş olacak ve kedi gizlenecek. Fare basılı değilse koşulumuz sağlanmamız olacak ve değilse kısmında olan görün bloğu çalışacak ve kedimiz görünecek.

[Örnek için Tıklayınız!](#)

olana kadar bekle

Olana kadar bekle bloğu: Bir koşul gerçekleşene kadar programı o adımda tutar. Programın bir sonraki adıma geçmesi için belirlenen koşulun gerçekleşmesi gerekir.

Örnek: Bu örnekte karakterimiz yeşil bayrak tıkladığında X=0 ve Y=0 konumuna gider. Boşluk tuşu basılana kadar bekler. Boşluk tuşuna basıldığında karakter 5 saniyede X=200 ve Y=0 konumuna gider.

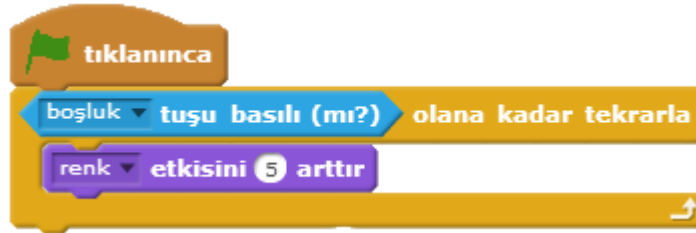


[Örnek için Tıklayınız!](#)

olana kadar tekrarla

Olana kadar tekrarla bloğu: Belli bir koşul gerçekleşene kadar içerisindeki blokları tekrarlar.

Örnek: Bu örnekte boşluk tuşuna basılana kadar karakterimizin rengi değişecektir.

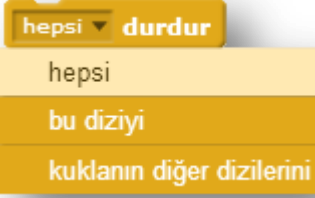


[Örnek için Tıklayınız!](#)

Örnek: Bu örnekte Bat2 karakterimiz Sprite1(kedi) karakterine değene kadar 5 adım ilerleyecek ve sonraki kılığa geçecektir.

[Örnek için Tıklayınız!](#)





Durdur bloğu: Durdur bloğu çalıştığı zaman verilen komut dahilinde çalışan kodları durdurur.

Hepsi: Bütün kod bloklarını durdurur.

Bu diziyi: Sadece bulunduğu kod bloklarını durdurur.

Kuklanın diğer dizilerini: Yazıldığı kod blokları hariç bütün kod bloklarını durdurur.

Örnek: Bu örneğimizde yeşil bayrak ile programımız çalışmaya başlar ve ne zaman boşluk tuşuna basılırsa "hepsi durdur" bloğu çalışır ve program durur.



[Örnek İçin Tıklayınız!](#)



Saniye bekle bloğu: Belirlenen saniye kadar programı duraklatır.

Not: İkiz yaratma blokları ilerleyen sayfalarda işlenecektir.

İşlemler menüsü

Ortaokulda en sevdiğim ders matematikti. Matematği hala çok severim.



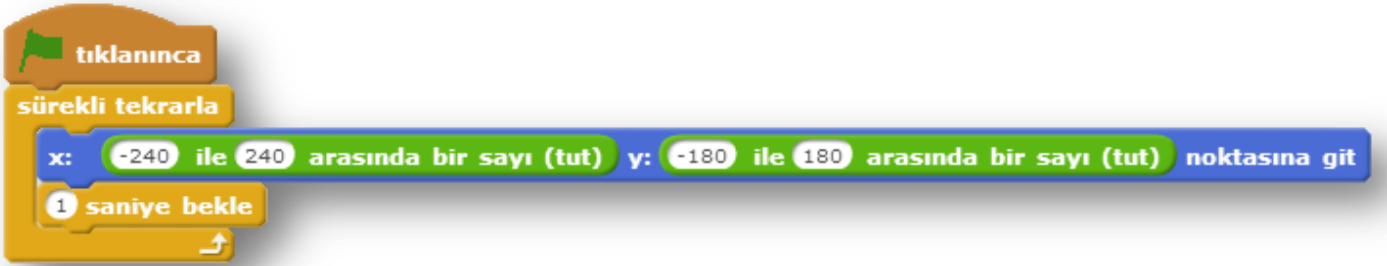
Matematiksel işlemler bazen çok karmaşık gibi görünse de aslında o kadar da karmaşık değildir.

Matematiksel işlemler programlamada önemli bir yer tutar ve Scratch'da bu matematiksel işlemler için bloklarımız mevcut.

1 ile 10 arasında bir sayı (tut)

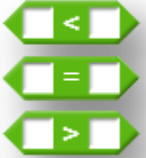
Hepimiz, aklından bir sayı tut oyununu oynamışızdır. Bu oyunda kafamızdan rastgele bir sayı tutuyorduk. Bilgisayarlar bu işi beceremezler ancak onlarda rastgele bir sayı üretebilir. Bu bloğu biz rastgele sayılar üretmek için kullanacağız. Belirlediğimiz sayılar arasında rastgele sayılar üretilecek.

Örnek: Bu örneğimizde X ve Y konumuna git bloğunu kullanacağız. Tek fark kendimiz sayı girmek yerine “arasında bir sayı (tut)” bloğunu kullanacağız. Karakterimiz rastgele bir X ve Y konumuna gidecek.



[Örnek için Tıklayınız!](#)

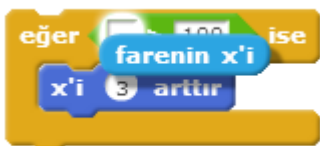
Mantıksal Karşılaştırma Blokları:



Mantıksal karşılaştırma bloklarını kontrol yapısı blokları ile birlikte kullanacağız.

Örnek: Bu örneğimizde farenin konumuna göre karakterimizi hareket ettireceğiz. Eğer farenin konumu $X=100$ den büyükse karakterimizin X konumu artacak. Eğer farenin konumu $X=-100$ den küçükse bu sefer X konumu azalacak. Aynı uygulamaları Y konumu için de yapacağız.

Dört tane eğer ise kontrol bloğu kullanacağız ve farenin konumunu kontrol edeceğiz.



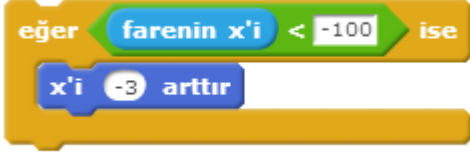
“Farenin x'i” bloğunu [algılama](#) menüsü altında bulabilirsiniz ve sürükleyip bırak yöntemi ile işlemler bloğunda istediğiniz yere yerleştirebilirsiniz. Bu blok

farenin o an bulunduğu X konumunu belirtir.

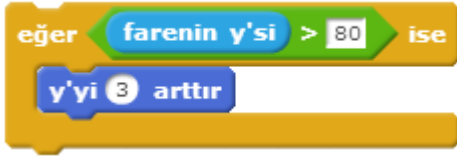


ilerlemesini sağlayacağız.

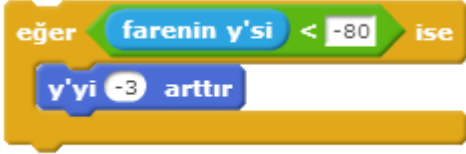
Bu kontrol yapısı bloğuyla farenin X konumu kontrol edeceğiz ve eğer 100'den büyükse karakterimizin X konumunun artmasını ve



Bu kontrol yapısı bloğu ise farenin X konumu eğer -100'den küçükse karakterin X konumu azaltır.



Bu kontrol yapısında da Y farenin konumunu kontrol ediyoruz ve eğer Y konumu 80'den büyükse karakterin Y konumunu arttırıyoruz.



Son kontrol bloğumuzda ise farenin Y konumu kontrol edilir ve eğer -80'den küçükse karakterin Y konumu azalır.

Not: Kontrol bloklarının sürekli çalışması için kontrol bloklarımızı sürekli bloğunun içerisine alacağız.

[Örnek için Tıklayınız!](#)

Veri menüsü

Değişkenler: Bir oyun oynadığınızı düşünün. Oyunda puan aldıkça skorunuz artar. Skorunuzun artmasını sağlayan yapılar değişkenlerdir. Değişkenleri matematikte de sıkça kullanırız. Örneğin bir problem çözerken x değişkenini sıkça kullanırız.

Örnek: $x = 3 \rightarrow x+3x = ?$ Burada x bir değişkendir. X'e 3 değeri atanmış. Bu yüzden x gördüğümüz yere 3 değerini yazarız. Sonuç $= 3 + 3*3 = 12$ olur.

Şimdi x değerini değiştirip aynı problemi tekrar çözelim.

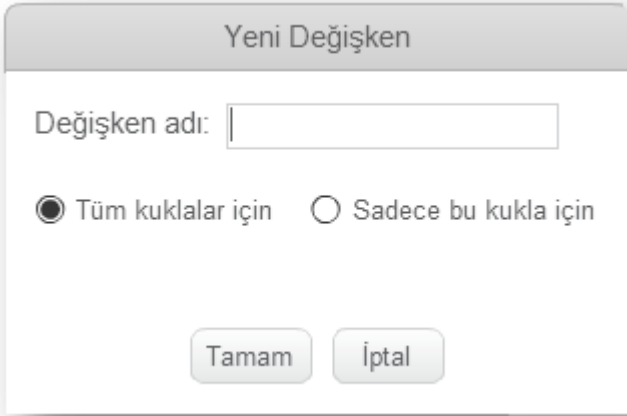
$X = 4 \rightarrow x+3x = ?$ X'e bu sefer 4 değeri atanmış. X bir değişken olduğu için değeri sürekli değişebiliyor. Sonuç = $4 + 3*4 = 20$ olur.

Matematikte sıkça kullanılan değişkenler programlamada da sıkça kullanılır.

Değişken oluşturma:

Bir Değişken Oluştur

Veri menüsü altındaki “Bir Değişken Oluştur” tıklanır.



Gelen pencereden değişkenin adını gireriz.



Tamam dediğimizde değişkenimiz oluşur ve değişkenimiz ile ilgili bloklar veri menüsü altına eklenir.

Örnek: 1'den 100'e kadar sayan program.

Örneğimizde sayaç adında bir değişkin oluşturuyoruz. Kontrol menüsü altındaki “10 kez tekrarla” bloğunu ile bir döngü oluşturacağız ve 100 kez döndürüp sayaç değişkenini her basamakta 1 arttırıp her adımda değişkeni ekrana yazacağız.



Programımız yeşil bayrak ile başlar. Daha sonra 100 kez tekrarla bloğu ile her basamakta değişkenin değerini 1 arttırırız ve görümün menüsündeki “Merhaba de” bloğu ile ekrana yazdırırız.

[Örnek için Tıklayınız!](#)

Örnek: Şimdi basit bir oyun tasarlayalım. Karakterimiz rastgele X ve Y konumuna gitsin. Yeni bir konuma gitmeden önce fare ile karaktere tıklayabilirsek oluşturduğumuz skor değişkeni 1 artsın ve puan kazanalım.



Karakterimizin rastgele bir X ve Y konumuna gitmesini sağlayan bloklar.



Bu bloklar ile de farenin karaktere değmesi ve farenin tıklanmasını kontrol ediyoruz. Eğer fare karakterin üzerinde tıklanırsa oluşturduğumuz skor değışkeni 1 artırıyor ve pop sesi çalınıyor. 1 saniye bekle bloęu ile de skor değışkeninin sadece 1 artmasını sağlıyoruz.



Bazen yazdığımız programda birden fazla koşulun aynı anda gerçekleşmesi gerekebilir. Bu durumda işlemler menüsü altındaki ve bloęunu kullanacağız.



Bu programda skorun artması için farenin hem karaktere değmesi hem de fareye basılı olması gerekmektedir.

[Örnek için Tıklayınız!](#)

Kontrol menüsü- İkiz Yaratma

Scratch programlama dilinde karakterimizin bir kopyasını yaratıp o kopyayı kullanabiliyoruz.

ikiz olarak başladığımda

kendim in ikizini yarat

bu ikizi sil

İkiz yarat bloğu çalıştığında karakterimizin bir ikizi oluşur.

İkiz olarak başladığımda kontrol yapısı ile de bu ikizin hangi eylemlerde bulunacağını belirtiriz.

Bu ikizi sil bloğu çalıştığında ise yaratılmış olan ikizi siler.

Örnek: Bu örneğimizde karakterimizin bir ikizini yaratacağız ve bu ikizlerin sağa sola hareket etmesini sağlayacağız.



Program yeşil bayrak ile başlar. İlk adımda karakterin büyüklüğünü %40 olarak ayarlar ve sürekli bloğu içerisinde kodları döndürmeye başlar. Sürekli bir ikiz yaratır renk etkisini değiştirir ve 1 saniye bekler.

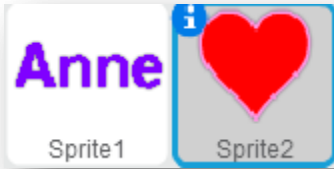
İkiz olarak başladığında ise -90 ile 90 arasında bir yöne döner ve sürekli olarak 10 adım gider.

[Örnek İçin Tıklayınız!](#)

Örnek 2: Hepimiz annelerimizi çok severiz. İsterseniz gelin hep beraber annemiz için bir program yazalım.

Bu örneğimizde bir karakter çizeceğiz ve anne yazacağız ve bir kalp karakteri ekleyeceğiz. Kalp karakterimizin ikizini yaratarak onun ekranda süzülmesini sağlayacağız.

Adım -1



Adım -2

Kalp karakterine istediğimiz animasyonu kazandırmak için gerekli kodları yazacağız.



Program yeşil bayrak ile başlar. İlk adımda X: -200 Y: -150 noktasına gider, büyüklüğü %40 yapar, 1 katman alta iner ve gizlenir. Daha sonra sürekli bloğu içerisindeki kodları tekrarlamaya başlar. Bir ikizini oluşturur, 0.5 saniye bekler ve tekrar bir ikiz oluşturur.

İkiz olarak başladığında ise oluşan ikiz -240 ile 240 arasında rastgele bir X konumuna gider. Görünmeye başlar. Sürekli olarak Y konumunu 5 arttırır hayalet etkisini 1 arttırır.

Buradaki eğer ise bloğu ile de ikizimizin konumu $Y > 140$ olduğunda o ikizi sileriz.

[Örnek İçin Tıklayınız!](#)

[Yılbaşı Animasyon Örneği İçin Tıklayınız!](#)

Hücre Modeli

Bu örneğimizde Scratch'da hücre modeli oluşturacağız. Aşağıdaki linkte hücre modeli oluşturmak için gerekli karakterlerin olduğu Scratch çalışmasının linki mevcut. Siz çekirdek için yazılmış olan kodu sırt çantasına ekleyip diğer karakterlere de ekleyin.

[Hücre Modeli Çalışması](#)



Not: Bu kodu stoplazma için yazmayın çünkü tüm organeller stoplazma üzerinde olduğundan fare oku çekirdeğe değdiğinde aslında stoplazmaya da değmiş olmaktadır.

Bu çalışmayı kendiniz veya arkadaşlarınızla geliştirebilirsiniz.

[Gelişmiş hücre modeli örneği için tıklayınız!](#)

Balık Yeme



Aç balığımızın karnını doyuralım. Büyük balığın küçük balıkları yediği bir oyun tasarlayalım. Bunun için bir tane büyük balık ve üç tane küçük balık oluşturduk. Yazmamız gereken kodlar aşağıdaki gibidir.



Balık Karakterleri İçin Yazılacak Kodlar:

```
when green flag clicked
  show
  turn 90 degrees
  loop
    move 2 steps
    turn random degrees between -20 and 20
    if at edge
      if color is red
        say "Yemek" for 2 secs
        hide
        wait 3 secs
        go to random point between x: -200 and x: 200, y: -200 and y: 200
        show
```

Program yeşil bayrağa basılınca çalışır ve balık karakteri görünür ve 90 yönüne döner.

Sürekli 2 adım giden balık karakteri rastgele bir şekilde -20 ile 20 arasında bir derece döner ve kenara geldiye seker.

Balığın rengi aç balığın ağzının rengine değiştiğinde yemek adında bir haber salarız. Bu haberi aç balık karakterinde kullanacağız. Yemek haberi saldıktan sonra karakterimiz yenmiş efekti vermek için gizlenir, üç saniye bekler ve rastgele bir konuma gittikten sonra yeniden görünür.



Aç balık Karakterleri İçin Yazılacak Kodlar:

Aç balık karakteri ağzının açık olduğu kılığa geçer ve sürekli olarak fare okuna mesafe 10'dan büyükse fare okuna döner ve 3 adım ilerler.

```
when clicked
  open mouth
  loop
    if mouse distance to a > 10
      turn to mouse
      move 3 steps
```

```
when eaten
  play sound
  loop 2
    close mouth
    wait 0.3 seconds
    open mouth
```

Aç balık karakterinin ağzının rengi balık karakterinin ağzının rengine değdiğinde bir haber salmıştık; yemek haberi. Yemek haberi geldiğinde balık yeme animasyonu yapmak için balığın ağzının kapalı olduğu kılığa geçmesini ve 0.3 saniye sonra ağzının açık olduğu kılığa geçmesini ve bu işlemi 2 kez tekrarlamasını söylüyoruz.

[Örnek İçin Tıklayınız!](#)

Sayı Tahmin Oyunu

Şimdi bir basit bir tahmin oyunu tasarlayacağız. Bilgisayara rastgele bir sayı tutturacağız ve kullanıcının klavyeden sayı girerek bu sayıyı tahmin etmesini isteyeceğiz.



Bilgisayara 1 ile 100 arasında rastgele bir sayı tutturduk ve bu değeri "tutulan sayı" değişkenine aktardık.

Olana kadar tekrarla kod bloğunu kullandık. Koşul olarak klavyeden girilen değerin tutulan değere eşit olması dedik.

Bu durumda klavyeden girilen değer tutulan değere eşit olana kadar içerisindeki kod bloklarını tekrarlar.

"..... Diye sor ve bekle" kod bloğunu kullandık. Bu bloğu klavyeden değer girmek için kullanıyoruz. Bu bloğu kullandığımızda ekrana bir giriş kısmı geliyor ve sayımızı giriyoruz. Bu girilen sayı **yanıt** değişkenine kaydediliyor.

"Eğer ise" kod bloğunu kullandık. Burada ise girilen sayının tutulan sayıdan büyük mü yoksa küçük mü olduğu kontrol ediliyor. Küçükse daha büyük sayı girilmesini, büyükse daha küçük sayı girilmesini istiyor.

"Olana kadar tekrarla" bloğu girilen sayı ile tutulan sayı eşit olana kadar tekrarlar. Ne zaman girilen sayı tutulan sayıya eşit olur o zaman döngüden çıkılır. Döngüden çıkılınca bu kod bloğu çalışır. Ekrana "Tebrikler Bildiniz!" yazar.

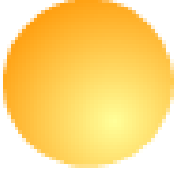
[Örnek İçin Tıklayınız!](#)

Basit Pinpon Oyunu

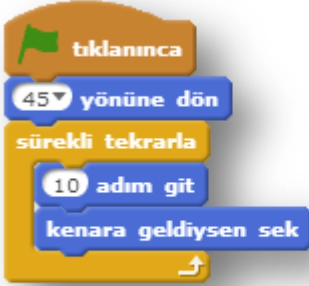
Bu oyunumuzda pinpon topunu sektireceğiz.



Kuklayı kütüphaneden seç kısmından bir top karakteri seçiyoruz.



Pinpon topu için yazılacak kodlar



Yeşil bayrak tıklanıldığında top 45 yönüne döner ve sürekli olarak 10 adım gider eğer kenara gelmişse seker.



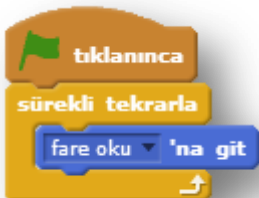
Bu blokta da topun diğer karakterimize değip değmemesini kontrol ediyoruz. Eğer değmişse 180 derece dön komutu ile geldiği yönün tersi bir yöne çeviriyoruz.

Sonraki kılığa geçmesini sağlıyoruz.

1 saniye bekle bloğunu bu işlemi hızlı bir şekilde üst üste yapmaması için koyuyoruz. Eğer 1 saniye bekle bloğunu koymazsak üst üste birden fazla sayıda 180 derece dönebilir.



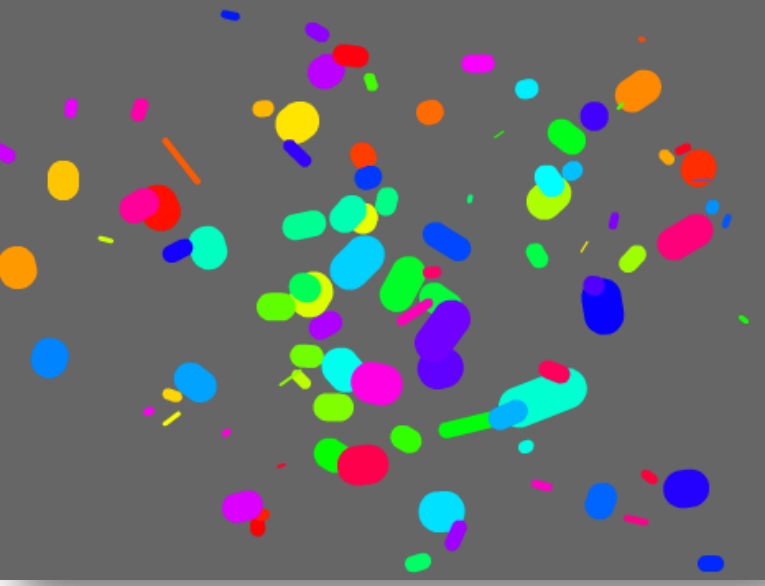
Paddle için yazılacak kodlar



Bu kod ile Paddle'nin sürekli olarak fare okuna gitmesini sağlıyoruz.

[Örnek İçin Tıklayınız!](#)

Renklerin Dansı



Kalem menüsü altındaki kod bloklarını kullanarak bir birinden güzel çalışmalar yapabiliriz. Yapacağınız çalışmalar tamamen sizin hayal gücünüzle orantılı.

```
tıklandınca
kalem tonunu 50 yap
x , 0.05 olsun
y , 5 olsun
kalem rengini 0 yap
büyüklüğü % 15 yap
100 defa tekrarla
  kendim in ikizini yarat
  gizlen
  kalem rengini 2 arttır
```

```
tıklandınca
sürekli tekrarla
  temizle
```

Program yeşil bayrakla başlıyor. Kalem tonu menüsü ile kalemin tonunu ayarlıyoruz. İki adet değişkenimiz var bu değişkenlere 0.05 ve 5 değerlerini atıyoruz. Bu değişkenleri şekillerimizi oluşturmada kullanacağız. Kalemin rengini başlangıçta 0 yapıyoruz.

100 defa tekrarla bloğu ile belirli bir sayıda ikiz yaratıyoruz. Her ikiz yarattığımızda da kalem rengini 2 arttırıyoruz.

Scratch ekranı çok hızlı çalışmaz. Biz temizle bloğunu kullandığımızda ekran bir saniye aralıklarla temizlenecek. Bu özelliği kullanarak biz renkli şekillerimizin uzunluğunu arttırıp azaltabileceğiz.

```
ikiz olarak başladığımızda
x: 0 y: 0 noktasına git
kalem kalınlığını 1 ile 25 arasında bir sayı (tut) yap
kalemi bastır
sürekli tekrarla
x ile y arasında bir sayı (tut) sn.de x: -240 ile 240 arasında bir sayı (tut) y: -180 ile 180 arasında bir sayı (tut) a süzül
```

Oluşturduğumuz ikizler oluşturduğumuz andan itibaren çalışmaya başlar. İkiz olarak başladığımızda bloğuyla ikizlerin ne yapması gerektiğini programlıyoruz.

İlk olarak ikizlerin X:0 , Y:0 noktasına gitmesini sağlıyoruz.

Kalem kalınlığını 1 ile 25 arasında rastgele bir değer yapıyoruz. Kalem ile oluşan şeklin büyüklüğünü belirliyoruz.

Kalemi bastırıyoruz. Burada şekilleri çizecek olan karakter gizli olduğu için sadece kalem ile çizilen şekiller gözükecek.

Karakterimizin sürekli olarak X ve Y değişkenimizin değerlerinin arasında bir saniyede rastgele X ve Y noktalarına gitmesini sağlıyoruz. Kalem basılı olacağına geçtiği yerleri kalem ile boyayacaktır. Ancak ekranı temizle komutu sürekli çalıştığı için ekranın her yere boyanmayacak!



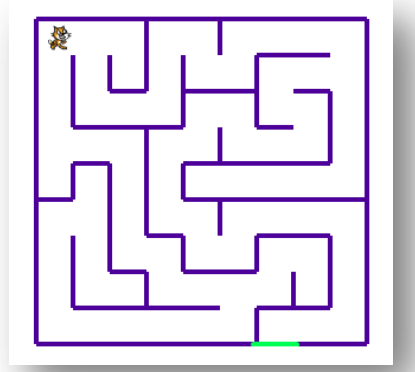
Algılama menüsü altında bir süre ölçer bulunmakta. Bu blok program çalışmaya başladığından itibaren geçen süreyi saniye olarak saklamaktadır. Biz bu özelliği kullanarak 15 saniye geçtikten sonra Y değişkeninin değerini düşürüyoruz. 30 saniye geçtikten sonra biraz daha düşürüyoruz ve Y değişkeninin değeri 0 oluyor. Bu sayede saniyede X ... Y... konumuna git bloğunda bu değişkeni kullandığımız için karakterimiz bir noktadan bir noktaya daha çabuk süzülüyor ve kalem ile çizdiği çizgi daha uzun oluyor.

Daha sonra süre ölçer > 80 olana kadar bekle bloğu ile 80. saniyeye kadar bekliyoruz ve Y değişkeninin değerini eski haline getiriyoruz ve süre ölçeri sıfırlıyoruz.

[Örnek için Tıklayınız!](#)

[Yağmur Efektini Örneği için Tıklayınız!](#)

Basit Labirent Oyunu



Labirent Oyunu Yazısı İçin Yazılacak Kodlar:

```
tıklanınca  
görün  
dance snare beat sesini bitene kadar çal
```

Yeşil bayrak tıklandığında görün diyoruz ki daha önce gizlenmişse görünsün.

Daha sonra dance snare beat sesini çalar.

```
başla haberi gelince  
gizlen
```

Oyna butonuna basıldığında başla haberi salınıyor. Oyna haberi salındığında gizlenmesi için başla haberi geldiğinde gizlen diyoruz.



Oyna Butonu İçin Yazılacak Kodlar:

```
tıklanınca  
görün  
x: 0 y: -50 noktasına git  
100 defa tekrarla  
3 sn.de x: x konumu + -30 ile 30 arasında bir sayı (tut) y: y konumu + 30 a süzül  
3 sn.de x: x konumu + -30 ile 30 arasında bir sayı (tut) y: y konumu + -30 a süzül  
bu kukla tıklanınca  
gizlen  
başla haberi sal
```

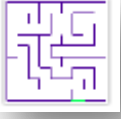
Kukla tıklandığında gizlenir ve başla haberini salar. Oyun başlar.

Yeşil bayrak tıklanınca görünür ve X:0 , Y:-50 noktasına gider.

100 defa

3 saniyede X konumuna -30 ile 30 arasında bir sayı ekler ve o X konumuna gider, Y konumuna 30 ekler ve o Y konumuna gider.

3 saniyede X konumuna -30 ile 30 arasında bir sayı ekler ve o X konumuna gider, Y konumuna -30 ekler ve o Y konumuna gider.



Labirent Karakteri İçin Yazılacak Kodlar:

```
when green flag clicked  
hide  
when green flag clicked  
show  
go to x:0 y:0 spot
```

Yeşil bayrak tıklandığında gizlenir.

Başla haberi salındığında görünür ve X:0 ,Y:0 noktasına gider.



Kedi Karakteri İçin Yazılacak Kodlar:

Başla haberi geldiğinde görünür. Katman olarak en üste çıkar ve büyüklüğü %20 yapar.

```
when green flag clicked  
show  
go to top layer  
set size to 20%  
repeat (forever)  
if (up arrow pressed) then  
y = y + 5  
if (color changed) then  
y = y + 5  
if (down arrow pressed) then  
y = y - 5  
if (color changed) then  
y = y - 5  
if (left arrow pressed) then  
x = x - 5  
if (color changed) then  
x = x - 5  
if (right arrow pressed) then  
x = x + 5  
if (color changed) then  
x = x + 5  
if (color changed) then  
y = y - 20  
play birthday sound  
say "kazandın" for 4 seconds
```

Burada karakterin yön kontrolünü yapıyoruz. Eğer ise yapıları ile yukarı,aşağı,sol,sağ ok tuşlarının basılmasını kontrol ediyoruz. Eğer yukarı ok tuşu basılmışsa Y konumunu 5 arttırıyoruz ve karakter yukarıya hareket ediyor. İçindeki eğer ise yapısı ile de labirente değmesini kontrol ediyoruz. Eğer labirente değişorsa gittiği yönün tam ters yönde gitmesini sağlıyoruz ve karakterimiz labirent duvarlarının içinden geçemiyor.

Burada labirentin tamamlanmasını kontrol ediyoruz. Karakterimiz yeşil renge değerse Y konu -20 arttırıyoruz birthday sesini çalıyoruz ve kazandın diyoruz.

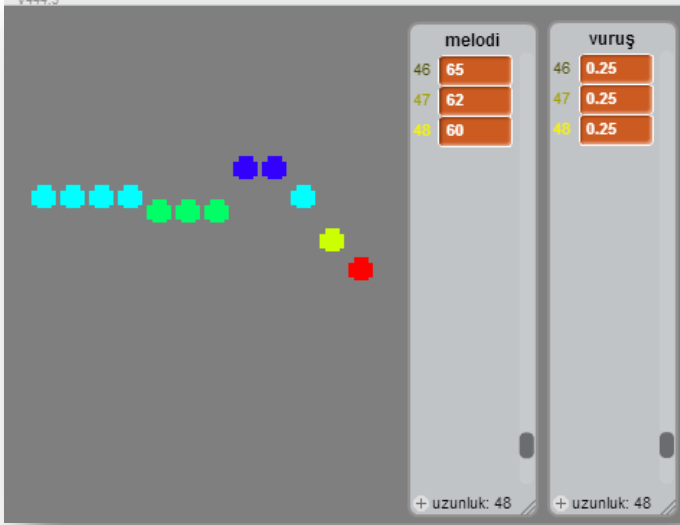


Burada yeşil bayrak tıklandığında gizlen diyoruz.

X: -120, Y:124 konumuna git diyoruz ve kedi karakterinin başlangıç konumunu ayarlıyoruz.

[Örnek için Tıklayınız!](#)

Liste elemanı ile Müzik Besteleme

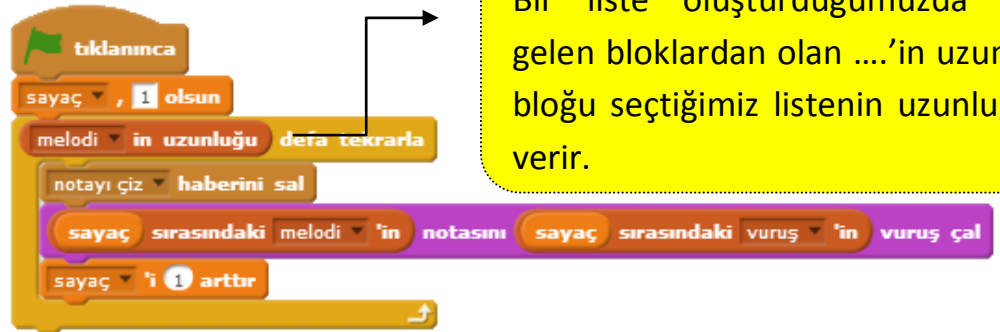


Veri menüsü içinde değişken oluşturma kısmının altında “Bir Liste Oluştur” kısmı vardır. Biz listeleri verileri bilgisayar hafızasında düzenli ve sıralı bir şekilde saklamak için kullanırız.

Şimdi bu listeleri kullanarak, listemizde sakladığımız nota ve vuruşlar ile bir melodi oluşturacağız.



Nota Çalıcı İçin Yazılacak Kodlar:



Bir liste oluşturduğumuzda hazır gelen bloklardan olan ...'in uzunluğu bloğu seçtiğimiz listenin uzunluğunu verir.

Program yeşil bayrak ile başlar. Oluşturduğumuz sayaç değişkenini 1 yapıyoruz. Melodi'nin uzunluğu kadar tekrarlar bloğuyla melodi listesinin uzunluğu kadar tekrarlatıyoruz. Notayı çiz haberini çalıyoruz.

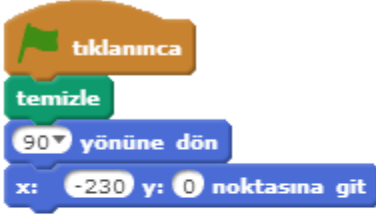


Bu blok ile listenin herhangi bir sırasındaki kayıtlı veriyi çağırabiliyoruz. Burada sayaç ismindeki değişken ile önce 1. Sıradaki veriyi alıyoruz, sayaç bir artıyor ve 2. Sıradaki veriyi okuyoruz



Burada nota kısmına melodi listesinden okuduğumuz sıradaki veriyi vuruş kısmına da vuruş listesinde okuduğumuz sıradaki veriyi ekliyoruz.

Nota Çizici İçin Yazılacak Kodlar:



Program yeşil bayrak ile başlar. Önce ekranı temizler.
90 yönüne döner ve X:-230 , Y:0 noktasına gider.



Notayı çiz haberi gelince. X konumunu kontrol eder, eğer X konumu sıfırdan büyükse X konumunu eski konumuna getirir yani -230 yapar ve ekranı temizler.

Bir önceki notanın üstüne gelmemek için 20 adım gider.

Y konumunu, renk etkisini, benekleştir etkisini ayarlar.

iz bırak ile o noktaya damgasını bırakır.

Not: Notaları değiştirmek için editör açıkken sahneden listenin ilgili satırlarını değiştirebilirsiniz.

[Örnek için Tıklayınız!](#)

Basit Flappy Bird Oyunu

Şimdi hep birlikte basit bir Flappy Bird oyunu tasarlayalım.

Öncelikli olarak Flappy Bird oyununu yapabilmek için gerekli olan karakterlerin olduğu Scratch çalışmamızı bilgisayarımıza indirip kendi hesabımıza yükleyelim.

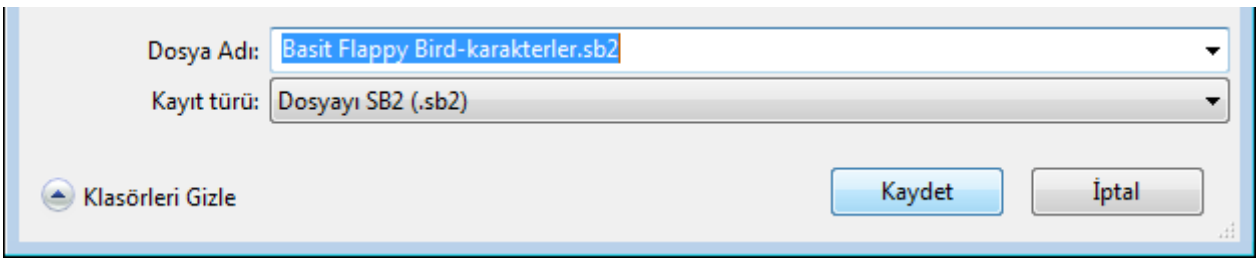
Aşağıdaki linke tıklayalım.

[Flappy Bird karakterler için tıklayınız!](#)

Daha sonra dosya menüsünden bilgisayarımıza indiri seçin.



Gelen menüden kayıt yerini seçip bilgisayarımıza kaydediyoruz.



Daha sonra kendi hesabımız açıkken dosya menüsü bilgisayarımızdan yükle kısmından kendi hesabımıza indirdiğimiz dosyayı yüklüyoruz. Artık karakterlerin olduğu çalışma sizin hesabınızda.

Bird karakteri için yazılacak kodlar:



```
when green flag clicked
  triumph sound effect
  score set to 0
  x: -100 y: 0 to here
  loop
    flap sound effect
```

Yeşil bayrak basıldığında oluşturduğumuz skor değişkenini sıfırlıyoruz. Karakterimizin X: -100 Y:0 konumuna gitmesini sağlıyoruz ve sürekli olarak kanat çırpma efekti için sonraki kılığa geçmesini sağlıyoruz.

```
if mouse clicked
  y: +8
  flap sound effect
else
  y: -3
```

Bu kısımda bir Eğer ise/Değilse yapısı kullandık. Fareye basılıysa Bird karakterinin Y konumu 8 artar ve flap sesini çalar. Değilse yani fareye basılı değilse bu sefer Y konumu -3 azalır. Yani Fareye basılınca Bird karakteri uçar basılı olmadığında ise aşağıya doğru düşer.

```
if hit wall
  dead sound effect
  say Game Over! for 2 seconds
  stop all
```

Burada Bird karakterinin engel karakterine değdiğini kontrol ediyoruz. Eğer değerse dead sesi çalıyor, Oyun Bitti diyor ve bütün blokları durduruyoruz.

```
when green flag clicked
  wait 3 seconds
  loop
    if mouse clicked
      y: +8
      flap sound effect
    else
      y: -3
    if hit wall
      dead sound effect
      say Game Over! for 2 seconds
      stop all
```

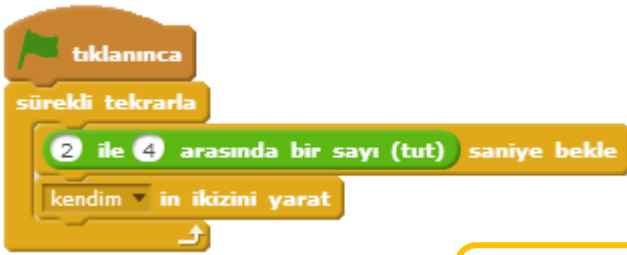
Başlangıçta 3 saniye bekleme koyuyoruz.

Eğer ise bloklarımızı sürekli bloğunun içine yazmamız gerekiyor. Aksi takdirde sadece bir kez çalışırlar.

Engel karakteri için yazılacak kodlar:



Engel karakterimizin X konumunu ayarlıyoruz ve gizlen diyoruz.



2 ile 4 saniye arasında değişen bir değerde karakterimizin ikizini yaratıyoruz.



Daha sonra ikiz olarak başladığında engelin yukarıda veya aşağıda olmasını ayarlamak için Y konumu 80 ile -80 arasında bir sayı olsun diyoruz. Görün komutu ile ikizin görünmesini sağlıyoruz.

Sürekli olarak -5 adım gitmesini yani geriye doğru gitmesini söylüyoruz. Bu oyunda kuş karakteri yerinde kalıyor engel karakteri hareket ediyor.

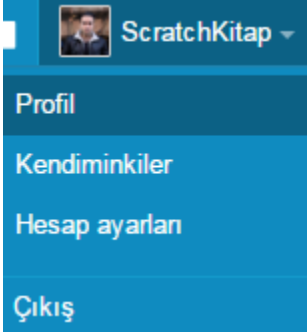
Bir eğer ise bloğu ile konumunun -230 daha küçük olup olmadığını kontrol ediyoruz. Eğer -230'dan daha küçükse yani sahnenin sonuna doğru gelmişse ikizi siliyoruz.

Bir başka eğer ise bloğu ile de engel karakterinin X konumunun -100 olup olmadığını kontrol ediyoruz. Eğer -100 olmuşsa yani Bird karakteri ile aynı X konumuna gelmişse skor değişkenini 1 arttırıyoruz ve pop sesini çalıyoruz.

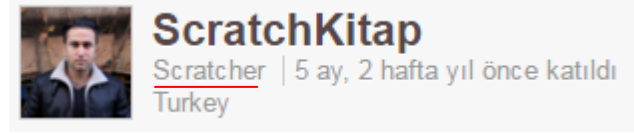
[Örnek için Tıklayınız!](#)

Scratcher(Scratchçı) Olmak

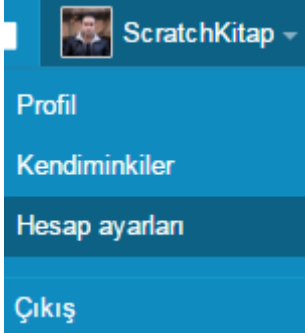
Scratch internet sitesine üye olduğunuzda site size bir rütbe verir. Yeni üyeler new Scratcher (yeni Scratchçı) olur. Bazı koşulları yerine getirirseniz Scratcher olursunuz. Bazı özellikleri kullanmak için Scratcher olmak gerekir.



İsminize tıkladığınızda profil kısmına tıklarsanız rütbenizi görebilirsiniz.



Scratcher olmak için öncelikli olarak e-posta hesabınız doğrulamalısınız. Scratch internet sitesine kayıt olurken verdiğiniz e-posta hesabına bir mail gider. Hesabınıza girerek o maili onaylamalısınız. Eğer mail gelmemişse ya da problem olmuşsa tekrar gönder diyebilirsiniz.



Hesap ayarları menüsüne girin. E-posta kısmına tıklayın ve tekrar gönder butonuna tıklayın e-mailinize yeni bir onay maili gelecektir.

Eğer mailinizi yanlış girdiyse ya da unuttuysanız.

Type new Email:

E-postayı Değiştir

Type new Email kısmından yeni bir mail adresi girin. Eğer mail adresi ile bir sıkıntı yaşıyorsanız babanızın mail adresini yazabilirsiniz.

Not: Eğer E-postanızı onaylamazsanız paylaşım yapamaz ve yorum yazamazsınız!

E-postanızı onayladıktan sonra bazı şartları yerine getirmelisiniz.

1. En az 2 proje paylaşmalısınız.
2. Birkaç yorum yazmalısınız.
3. Bazı projeleri beğenmeli ve favorilere eklemelisiniz.
4. Scratch'a kayıt olalı en az 2 hafta olmalı.

Koşulları yerine getirdikten sonra profilinize girdiğinizde [become a scratcher](#) yazısı çıkacaktır. Yazıya tıklayın ve yazılanları onaylayın.

Bulut Değişken

Normal değişken kullandığınızda değişen değişken değerleri program sonlandığında sıfırlanır. Diğer insanların verilerini değişkenlerde saklamak isterseniz bulut değişken kullanmalısınız.

Not: Bulut değişkenleri kullanmak için Scratcher olmanız gerekir.

Bulut Değişken Örneği:

Yeni Değişken

Değişken adı:

Tüm kuklalar için Sadece bu kukla için

Cloud variable (stored on server)

Tamam İptal

Bir değişken oluşturuyoruz. Ancak bu sefer Cloud variable kısmını seçiyoruz.



Bulut değişkenin yan tarafında bulut işareti vardır.



Fare basılı ise ve fare oku karaktere değişirse kostüm2'ye geçer, bulut değişkenin değerini 1 arttırır, 0.5 saniye bekler ve tekrar eski kostüme yani kostüm1'e geçer.

[Örnek İçin Tıklayınız!](#)

Bulut değişken ile diğer kullanıcıların yaptıkları işlemlerde değişkenlerde kaydedilebilir.

[Hangi Takımı Tutuyorsunuz Örneği İçin Tıklayınız!](#)

Özel Taşlar

Özel taşlar menüsü ile kendi kod bloklarımızı oluşturabiliyoruz.



Özel taşlar menüsüne geldikten sonra “Bir Taş Oluştur” seçeneğine tıklıyoruz.



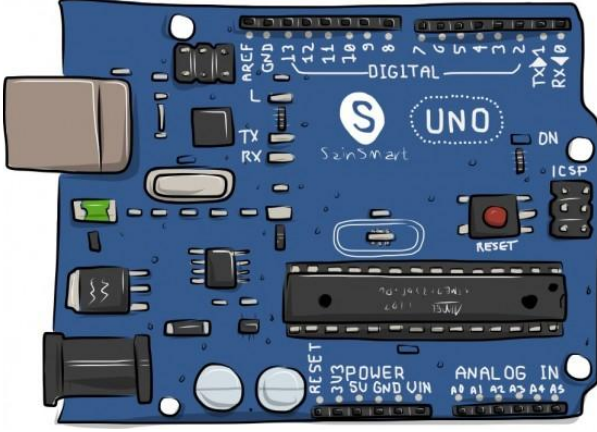
Taşımızın ismini belirliyoruz ve Tamama tıklıyoruz.



Tanımla kısmının altına kodumuz çalıştığında yapmasını istediğimiz kodları yazıyoruz.

[Özel taşlar ile zıpla bloğu yapım örneği için tıklayınız!](#)

Arduino Nedir?



Arduino, mikroişlemci temelli programlanabilir bir geliştirme kartıdır. Arduino ile kendi robotlarınızı yapabileceğiniz gibi kendi elektronik devrelerinizi de oluşturabilirsiniz.

Not: Arduino'ya elektronik bir beyin diyebiliriz. Bu beyini projelerimizde kullanabilmek için programlamamız gerekmektedir.



Arduino'yu Nasıl Programlarız?

Arduino'yu programlamak için bir programlama diline ihtiyacımız var. Biz kitabımızda Arduino'yu programlamak için Scratch programlama dilini kullanacağız.



S4A, Arduino'yu programlamak için oluşturulmuş Scratch üzerine inşa edilmiş bir programdır. S4A'yı kullanarak Arduino'yu kolayca programlayabilirsiniz.

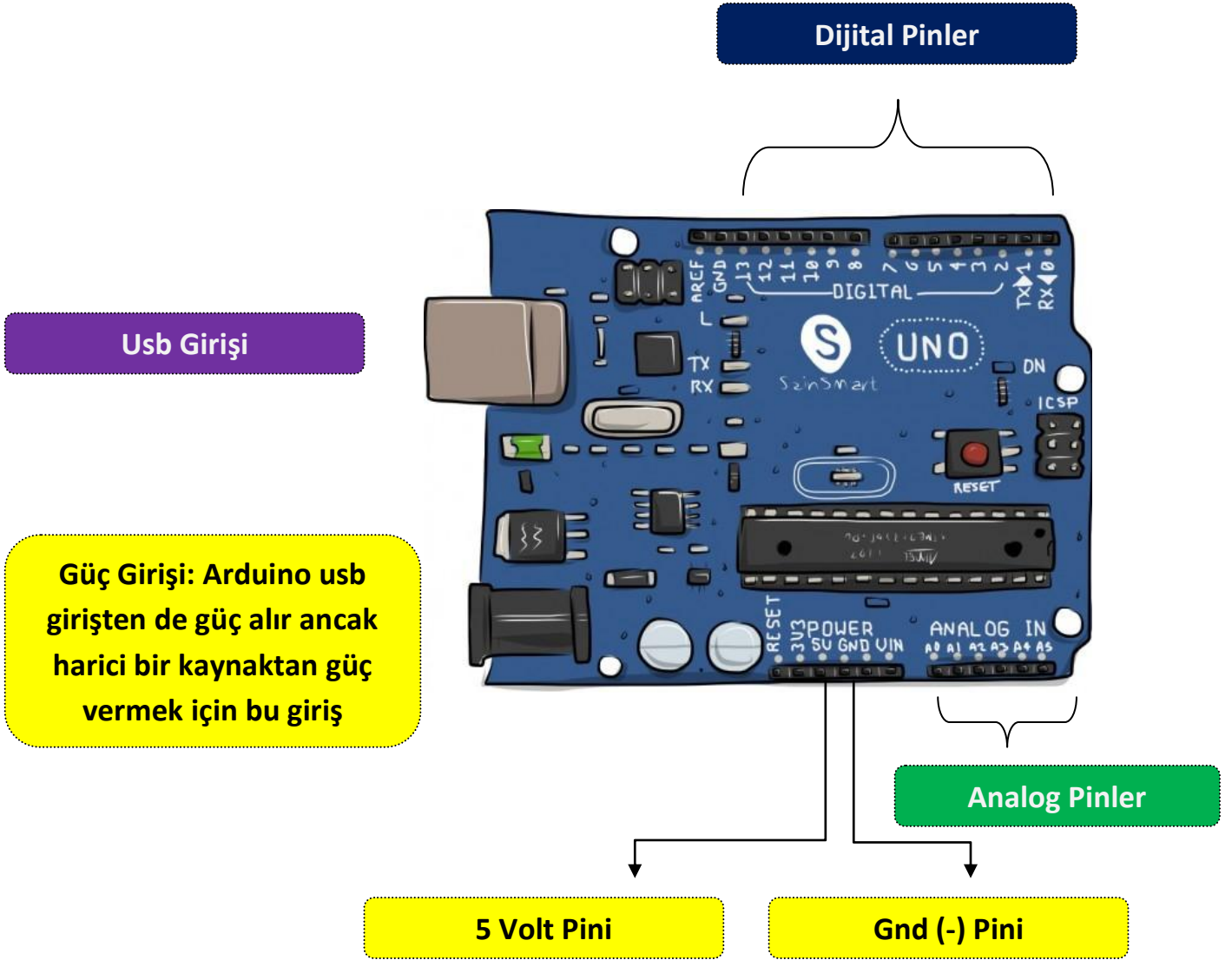
[S4A ile Arduino Programlama E-Kitabı için Tıklayınız!](#)



Mblock programı Makeblock firmasının geliştirmiş olduğu Scratch programlama dili üzerine inşa edilmiş bir programdır. Şu anda Arduino programlamada kullanılan en popüler görsel programlama aracıdır. Biz de kitabımızda Arduino programlarken genellikle bu programı kullanacağız.

[Mblock Programını İndirmek için Tıklayınız!](#)

Arduino'muzu Tanıyalım

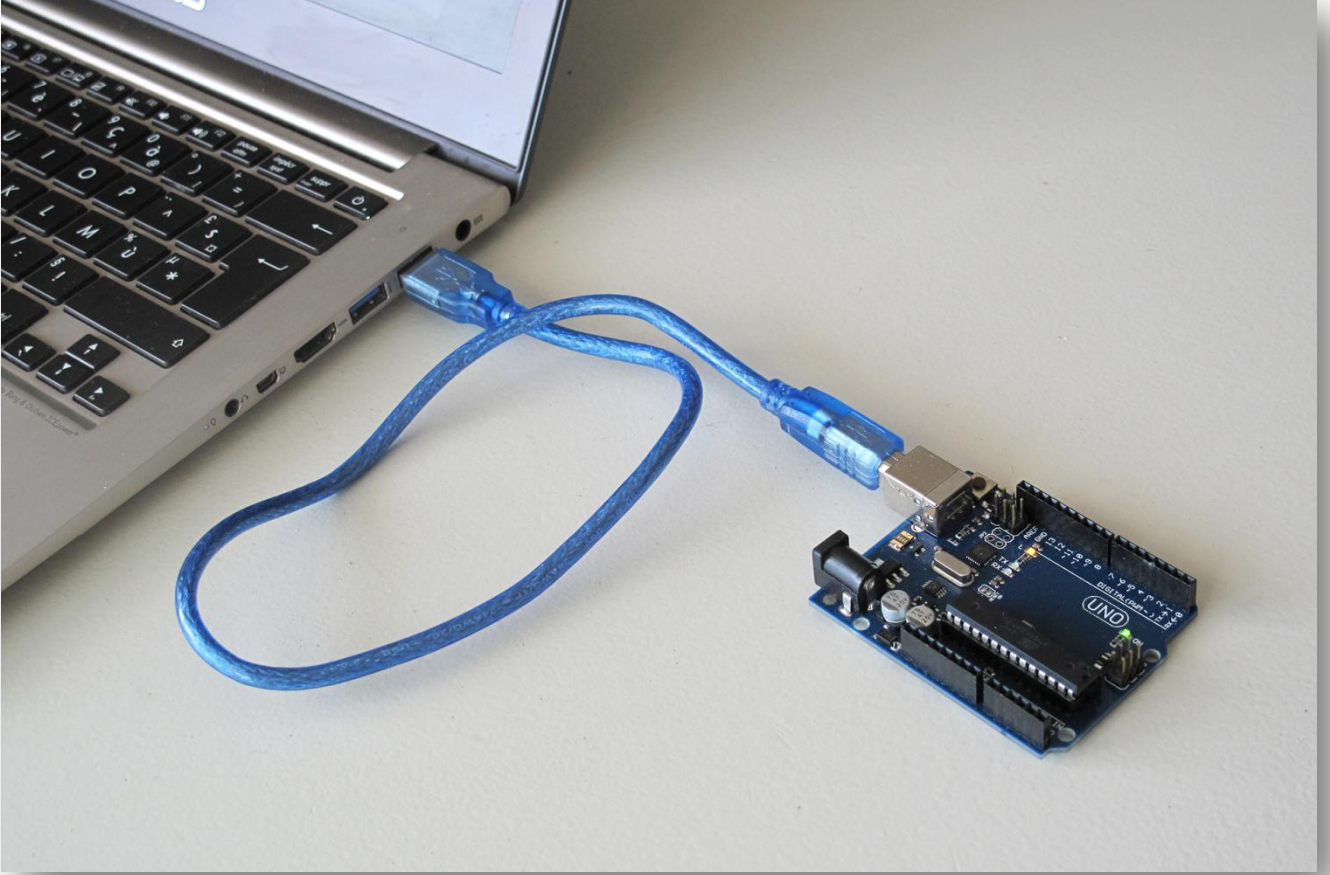


Usb Girişi: Arduino'yu bilgisayara bağlamak için kullandığımız giriş.

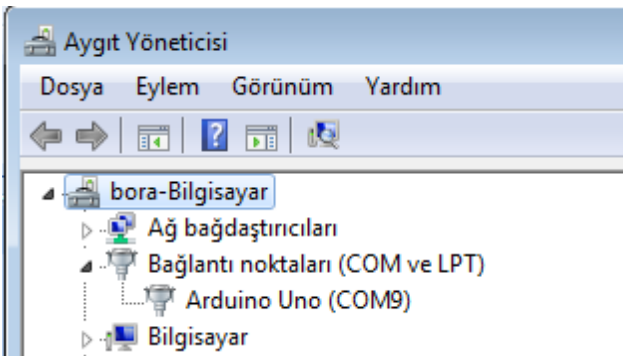
Dijital Pinler: Arduino'muzu çevresiyle etkileşim halinde tutabilmek için pinleri kullanırız. Dijital pinleri Arduino'dan dışarıya akım vermek ya da gelen akımları akım için kullanırız. Örneğin; Arduino'ya bağladığımız bir Led'i yakmak için dijital pinleri kullanırız.

Analog Pinler: Arduino'ya bağladığımız sensörleri okumak için analog pinleri kullanırız.

Arduino'yu Bilgisayara Bağlama

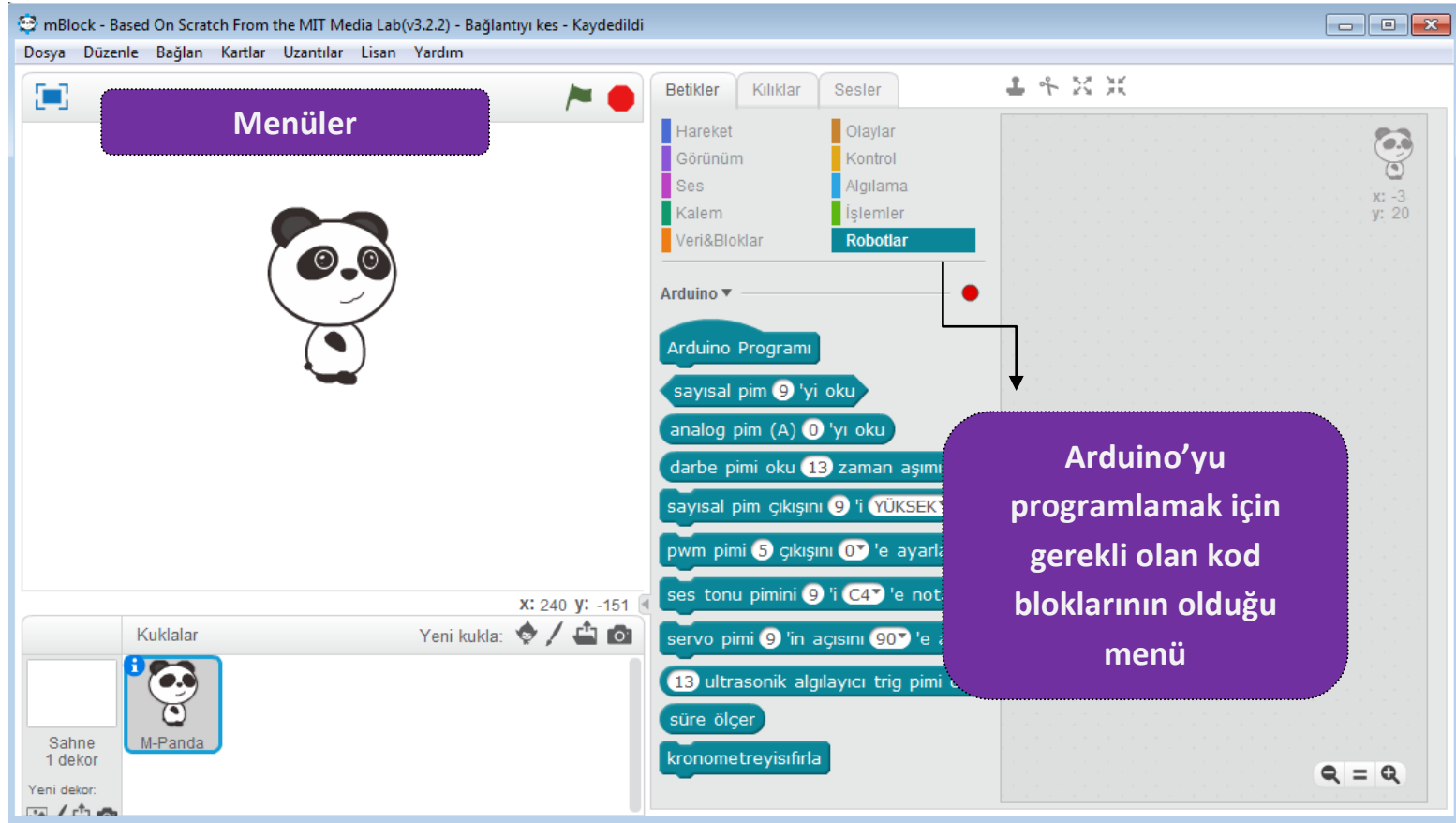


Arduino'yu bilgisayarımıza usb kablosu aracılığıyla bağlarız. Bilgisayarınızın Arduino'yu tanıyıp tanımadığını anlamak ya da nasıl bağlandığını görmek için bilgisayarımıza sağ tıklayın, özellikler menüsünden **aygıt yöneticisini** açın.



Arduino bilgisayara Com bağlantı noktasından bağlanır. Bende bağlantı noktası Com 9 olarak gözükmekte. Sizde farklı bir Com noktası gözükebilir.

Mblock Programını Tanıyalım



Arduino'yu Mblock ile Programlama

Bağlan

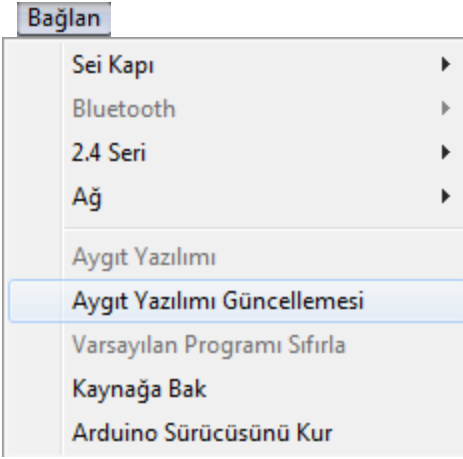
Sei Kapı

COM9

Öncelikli olarak Arduino'yu bilgisayara bağlayın. Mblock programını açın. Bağlan menüsünden seri kapı seçeneği ile Arduino'nun bağlı olduğu Com numarasını seçin.

[Youtube Videosu İçin Tıklayınız!](#)





Daha sonra yine bağlan menüsü altında bulunan “Aygıt Yazılımı Güncellenmesi” kısmından Arduino’yu programlamamız için gerekli programı Arduino’ya yüklüyoruz.

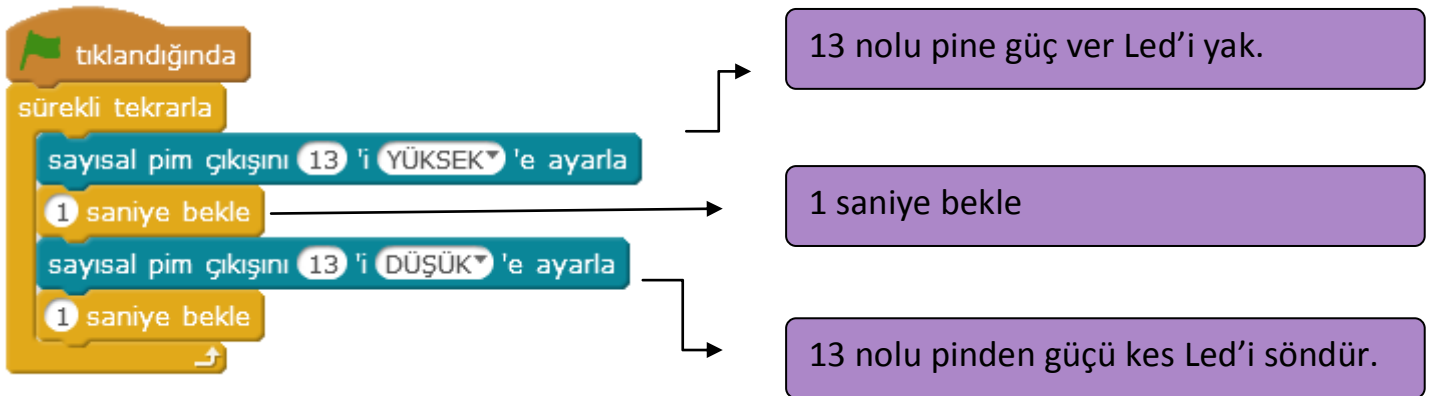


Yükleme bitti yazısı çıktıktan sonra Arduino’muzu programlamaya başlayabiliriz.

İlk Programımızı Yazalım

Arduino üzerinde dijital 13 nolu pine bağlı bir Led mevcut. Biz bu Led’i Scratch ile programlayacağız.

LED’ler elektrik enerjisini ışığa dönüştüren yarı iletken devre elemanlarıdır.



Not: Bağlan menüsünden Arduino’nun bağlı ve aygıt yazılımının yüklü olduğundan emin olun!

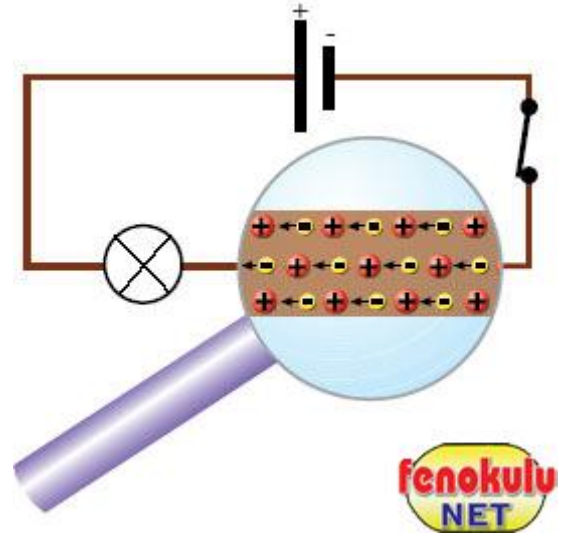
Temel Elektronik

Arduino programlanırken pek fazla elektronik bilgisi gerekmez. Ancak elektronik hakkında temel bilgiler edinmekte fayda var.

Elektronik, elektronik aygıtları çalıştırmak için küçük elektrik akımlarının nasıl kumanda edilmesi gerektiğini inceleyen bilim dalıdır. Kullandığımız bilgisayar,televizyon,akıllı telefonlar hepsi elektronik bilimi sayesinde hayatımıza girdi. Bizde kitabımızda bazı elektronik kavramlarını ve elektronik devre elemanlarını inceleyeceğiz. İnceleyeceğimiz devre elemanları, kullandığımız bütün elektronik aletlerde bulunan parçalardır.

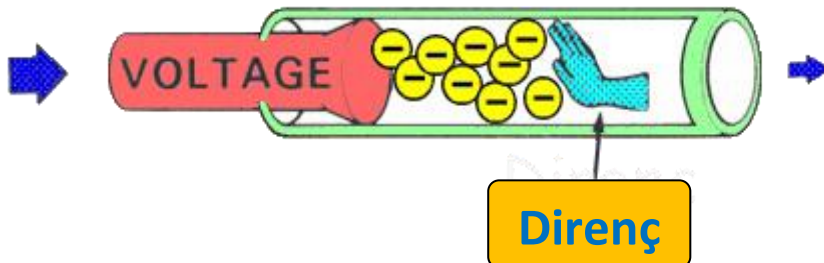
Elektrik Akımı

Elektrik akımı, elektron adı verilen çok küçük parçacıkların bir tel boyunca akışıdır. Bütün maddeler elektron içerir. Elektronlar her şeyi oluşturan atomların bir parçasıdır.

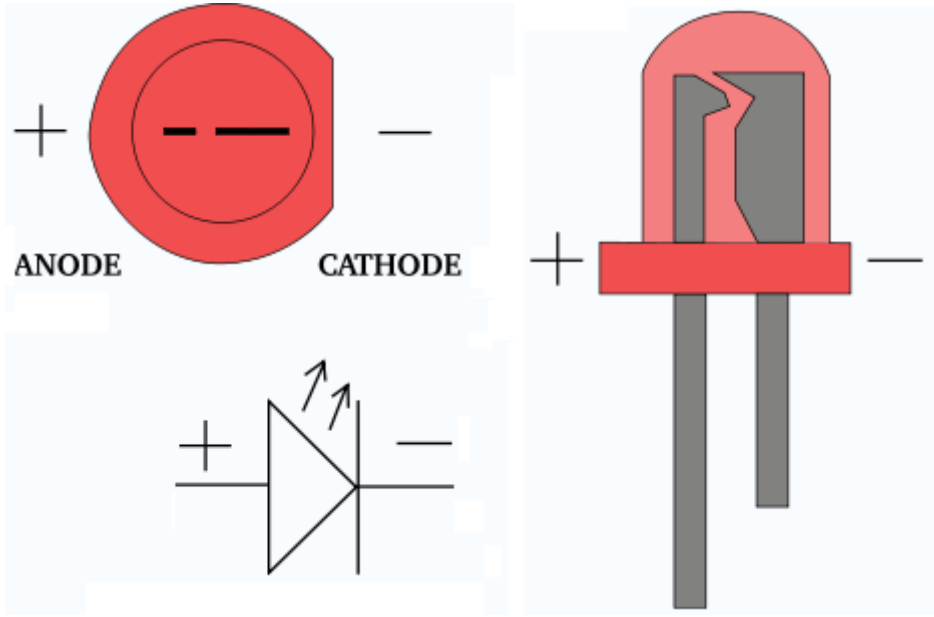


[Detaylı Bilgi İçin Tıklayınız!](#)

Elektronlar pilin (+) ucundan (-) ucuna doğru hareket ederler. Bu hareket sonucu bir gerilim oluşur, biz bu gerilime Voltaj diyeceğiz ve V ile göstereceğiz. Elektronların (+) uctan (-) uca doğru hareketine akım diyeceğiz ve I ile göstereceğiz. Akım (I) ve gerilim (V) bir devrenin temel bileşenlerinden ikisidir. Üçüncü temel bileşen dirençtir. Direnç, kimi maddelerin elektron akışına yani akıma karşı gösterdiği zorluktur.

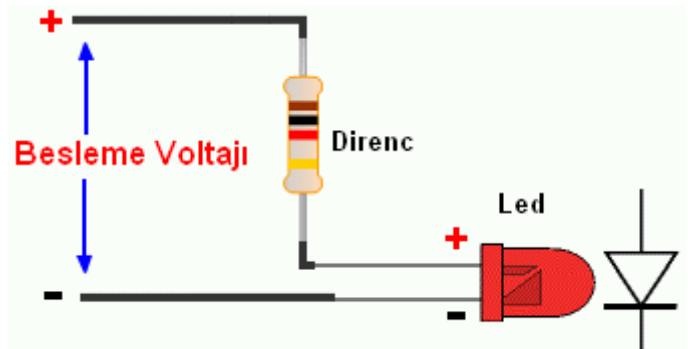


Elektronik Devre Elemanları

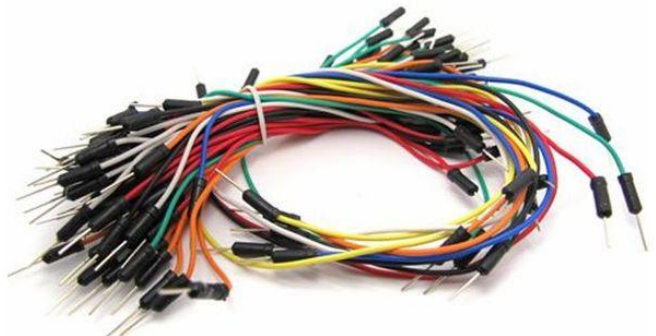


Led'ler, üzerinden akım geçtiğinde ışık yayan devre elemanlarıdır. Uzun ucu (+) kısa ucu (-). Yalnız dikkat etmemiz gereken husus: Asla Led'imizi direnç olmadan devremize bağlamayacağız. Çünkü Led'ler hassas devre elemanlarıdır, uzun süre yüksek akıma maruz kaldığında yanabilir ve bir gaz açığa çıkabilir! Bu gaz tehlikelidir ve asla solumayınız. Direnç akıma zorluk göstererek akımı azaltacak ve Led'imize uygun hale getirecektir.

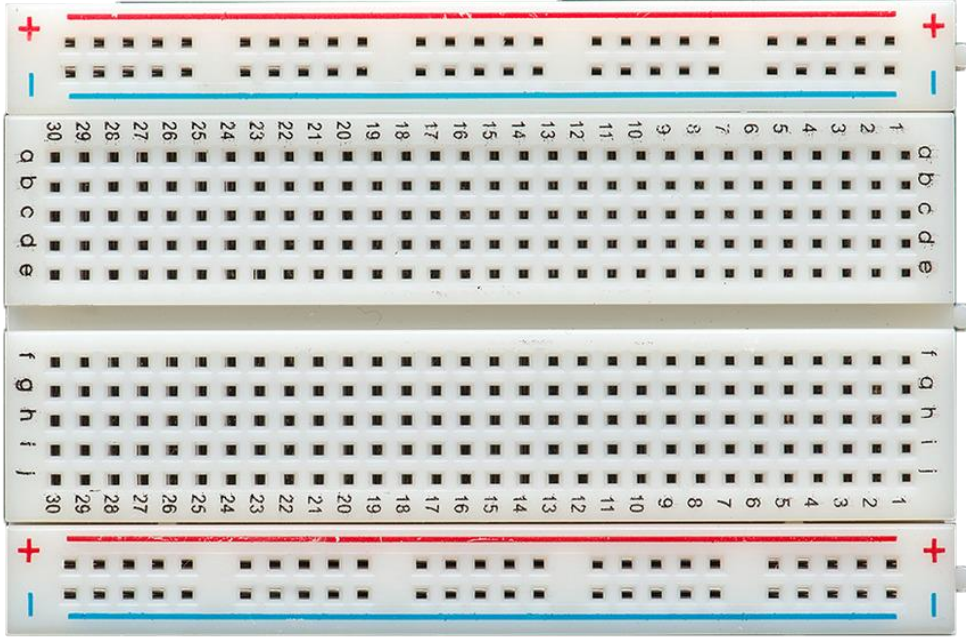
Dirençler akıma zorluk gösteren devre elemanlarıdır. Arduino ile pinlerden vereceğimiz akım Led'ler için yüksek değerdedir. Dirençler bu değeri düşürerek Led'e uygun hale getirirler.



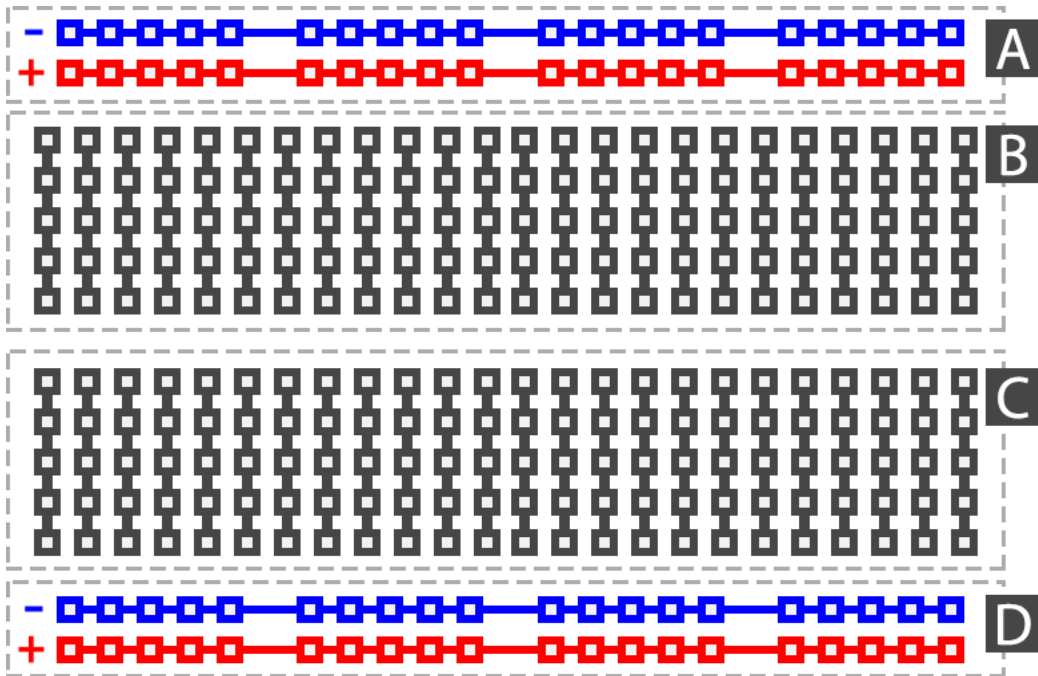
Jumper kablo Devre elemanlarını birbirine bağlamak için kullanılır. Bir ucunu Arduino'muza diğer ucunu da breadboarda bağlayacağız.



Breadboard

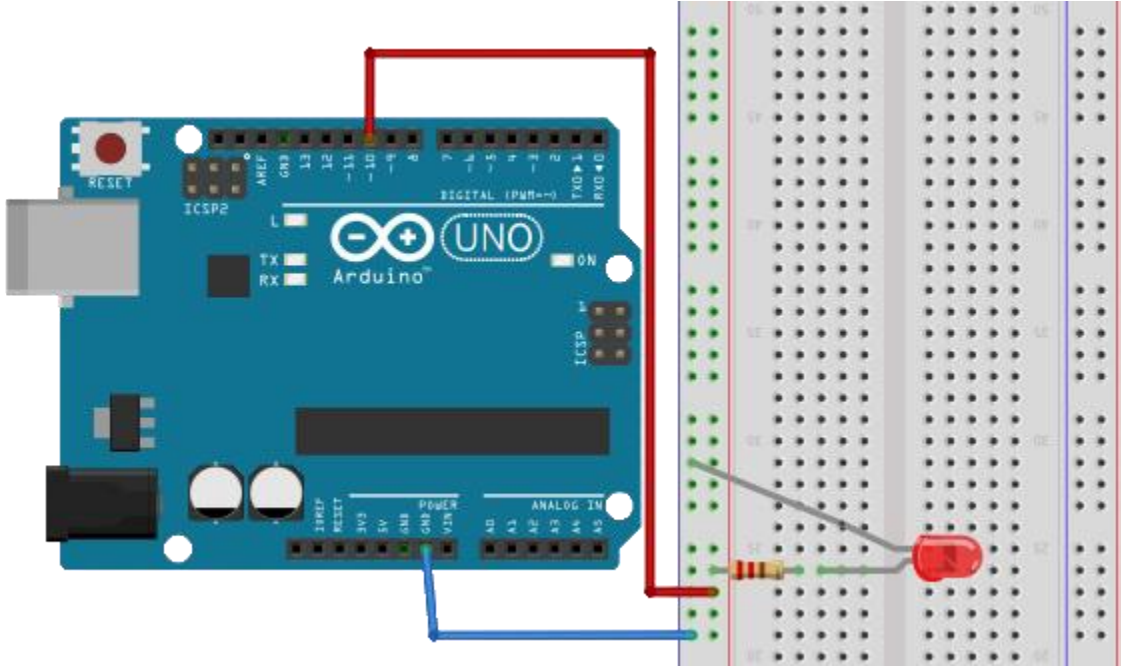


Breadboard devre elemanlarını birbirine bağlamak için kullanılır. Tekrar tekrar kullanabilirsiniz.



A ve D kısımları yatay olarak birbirine bağlıdır. B ve C kısımları ise dikey olarak birbirine bağlıdır.

Breadboard Üzerindeki Led'i Yakıp Söndürme



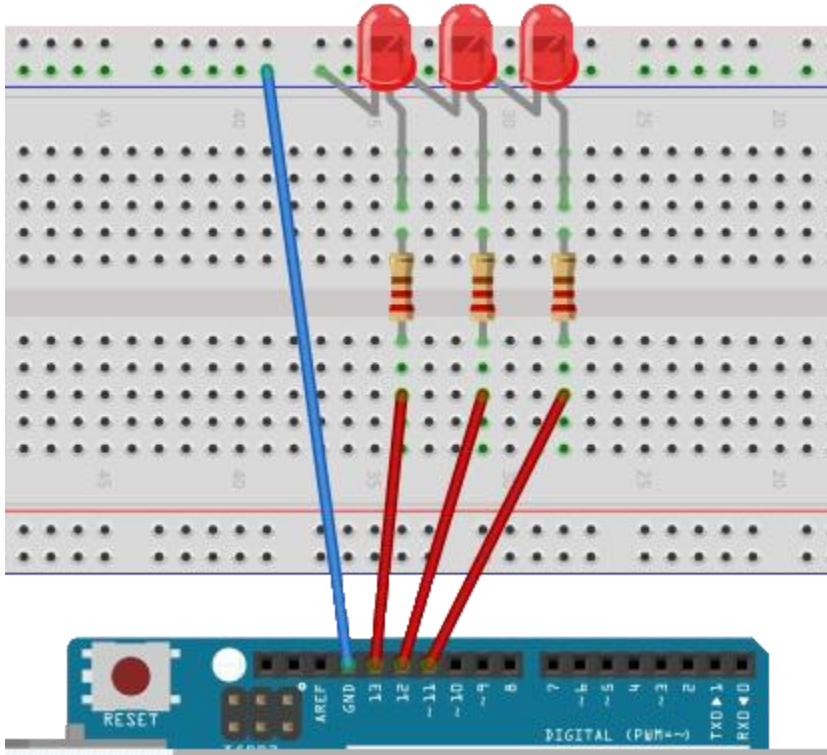
Gerekli Malzemeler

- 1 Adet Arduino Uno
- 1 Adet breadbord
- 1 Adet Led
- 1 Adet 220 Ohm ya da 330 Ohm direnç

220 Ohm direncin rengi kırmızı, kırmızı, kahverengidir. 330 Ohm direncin ise rengi turuncu, turuncu, kahverengidir.

```
when clicked  
  repeat  
    set digital pin output to 10 to HIGH  
    wait 1 seconds  
    set digital pin output to 10 to LOW  
    wait 1 seconds
```


Kara Şimşek Devresi



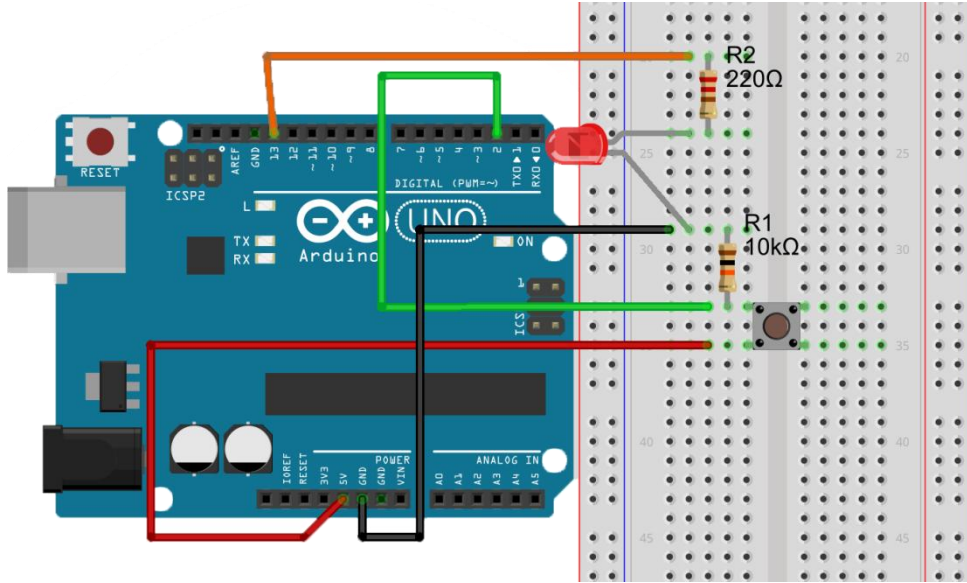
```
when clicked
  repeat (forever)
    set digital pin 13 to HIGH
    wait 0.3 seconds
    set digital pin 13 to LOW
    set digital pin 12 to HIGH
    wait 0.3 seconds
    set digital pin 12 to LOW
    set digital pin 11 to HIGH
    wait 0.3 seconds
    set digital pin 11 to LOW
    set digital pin 12 to HIGH
    wait 0.3 seconds
    set digital pin 12 to LOW
```

Gerekli Malzemeler

- 1 Adet Arduino Uno
- 1 Adet breadbord
- 3 Adet Led
- 3 Adet 220 Ohm ya da 330 Ohm direnç

Dijital Pinlerden Deęer Okuma

Arduino ile dijital pinlerden akım verebildiđimiz gibi gelen akımları da okuyabiliyoruz. Burada bir adet buton kullanarak buton basıldıđında gelen akımı okuyarak butonun basılı olup olmadıđını anlayacađız.



Gerekli Malzemeler

- 1 Adet Arduino Uno
- 1 Adet breadbord
- 1 Adet Led
- 1 Adet buton
- 1 Adet 220 Ohm ya da 330 Ohm direnç
- 1 Adet 10 Kilo Ohm direnç (kahverengi,siyah,turuncu)

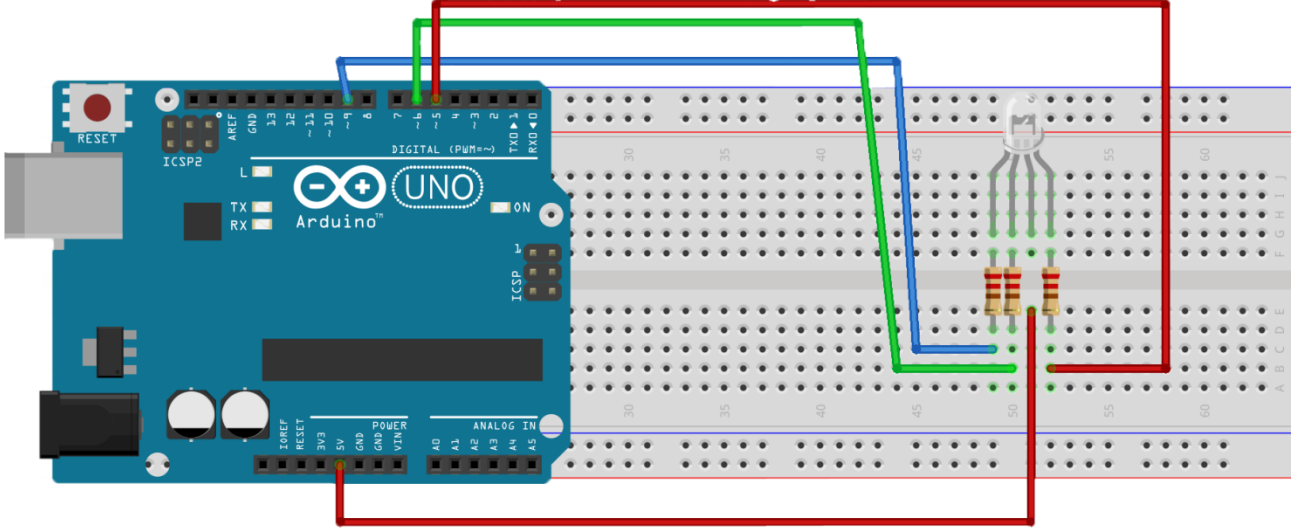


Youtube Videosu İçin Tıklayınız!



RGB Led

Rgb Led'ler üç ana rengi kullanarak(Kırmızı, Yeşil, Mavi) bütün renklerde yanabilen Led'lerdir.



Gerekli Malzemeler

- 1 adet Arduino Uno
- 1 adet Breadboard
- 1 adet RGB Anot(+) Led
- 3 adet 220Ω direnç
- 4 adet Jumper kablo

İki tür RGB Led var. Anot(+) ve Katot(-). Biz devremizde 5v'ye bağlanan anot Led kullandık. Led'in uzun ucu 5v'ye bağlanacak. Türkiye'de genellikle anot Led satılır.

m tuşu basılınca

pwm pimi 9 çıkışını 0'e ayarla

pwm pimi 6 çıkışını 255'e ayarla

pwm pimi 5 çıkışını 255'e ayarla

y tuşu basılınca

pwm pimi 9 çıkışını 255'e ayarla

pwm pimi 6 çıkışını 0'e ayarla

pwm pimi 5 çıkışını 255'e ayarla

k tuşu basılınca

pwm pimi 9 çıkışını 255'e ayarla

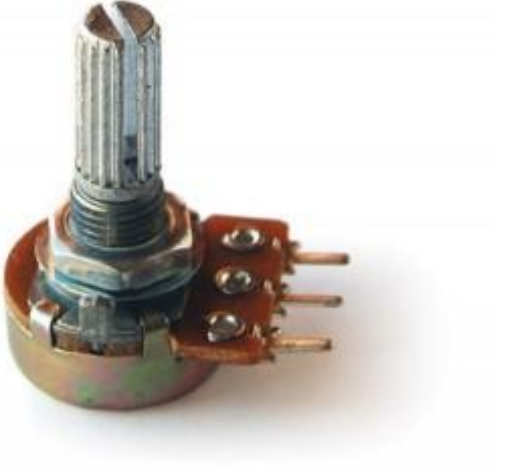
pwm pimi 6 çıkışını 255'e ayarla

pwm pimi 5 çıkışını 0'e ayarla

Youtube Videosu İçin Tıklayınız!

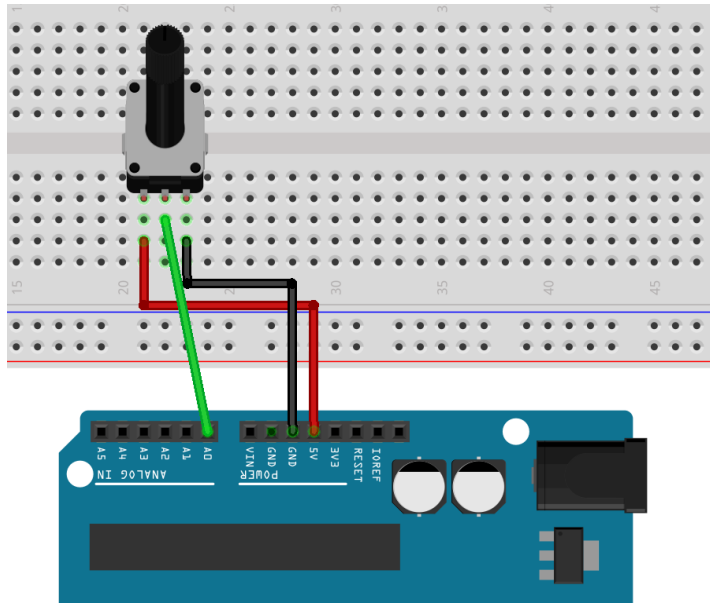
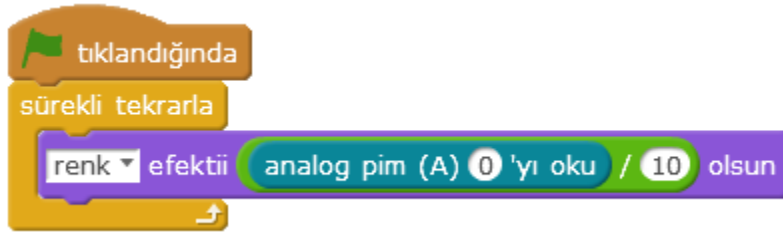


Potansiyometre



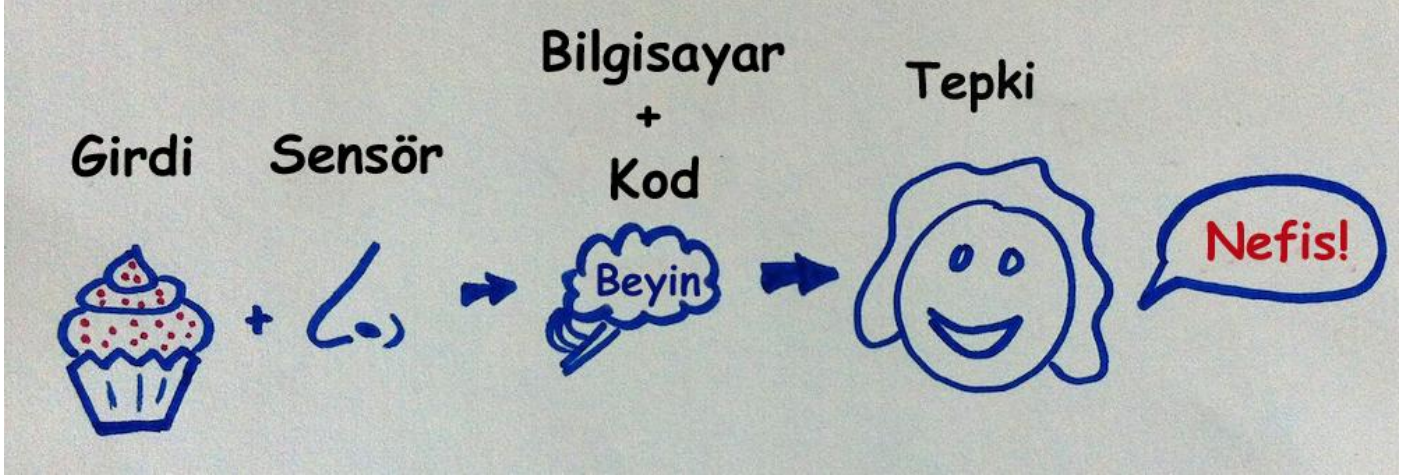
Potansiyometreler, ayarlanabilir dirençlerdir. Kullandığımız cihazlardaki sesi arttırıp azaltmaya yarayan düğmeler potansiyometre ile yapılır. Potansiyometreyi ayarlayarak direnci azaltırız ses artar, direnci arttırırız ses azalır.

Potansiyometrenin üç bacağı vardır. En soldaki bacağı (+) ya , orta bacağı analog 0'a, en sağdaki bacağı ise (-) ye bağlayarak uygulamalarımızı yapacağız.

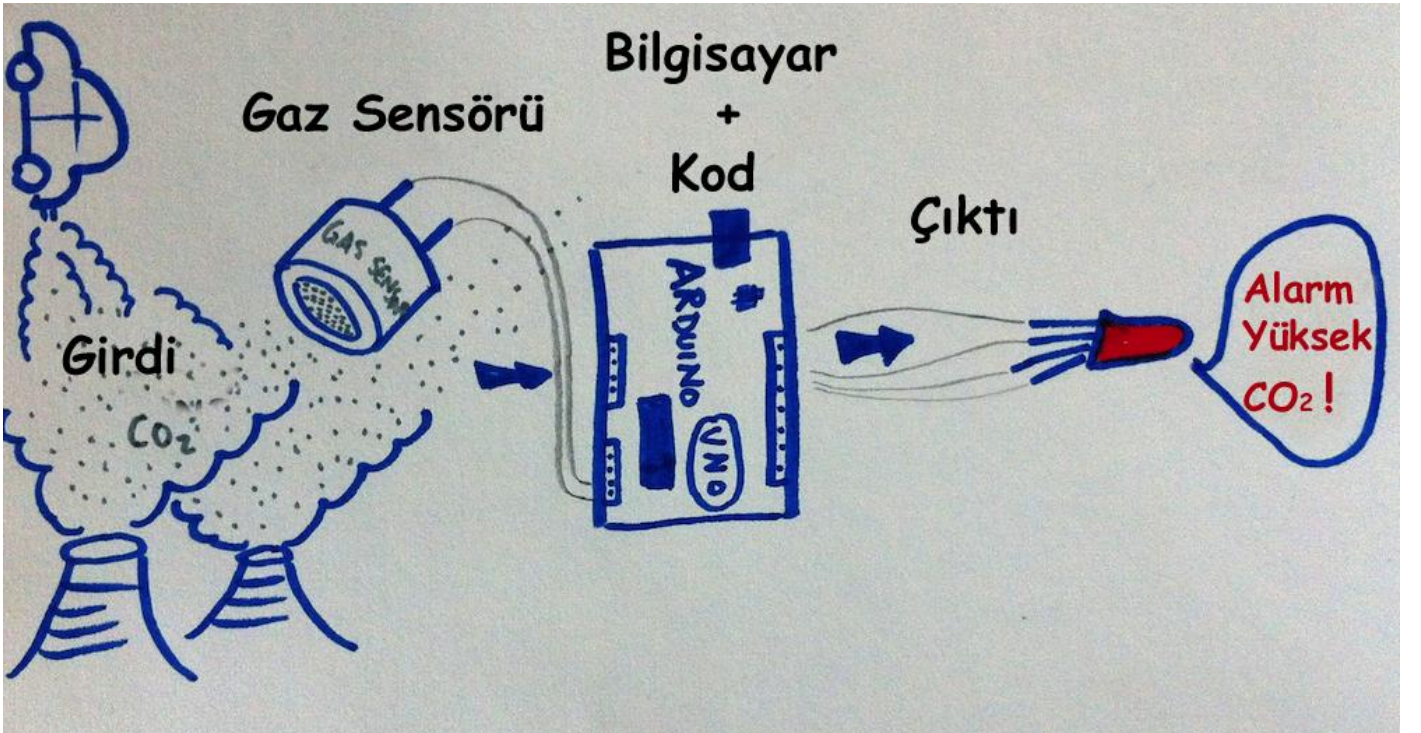


Sensörler

İnsanlar çevrelerindeki dünyayı ve çevrelerinde olup bitenleri algılamak için duyu organlarını kullanır.

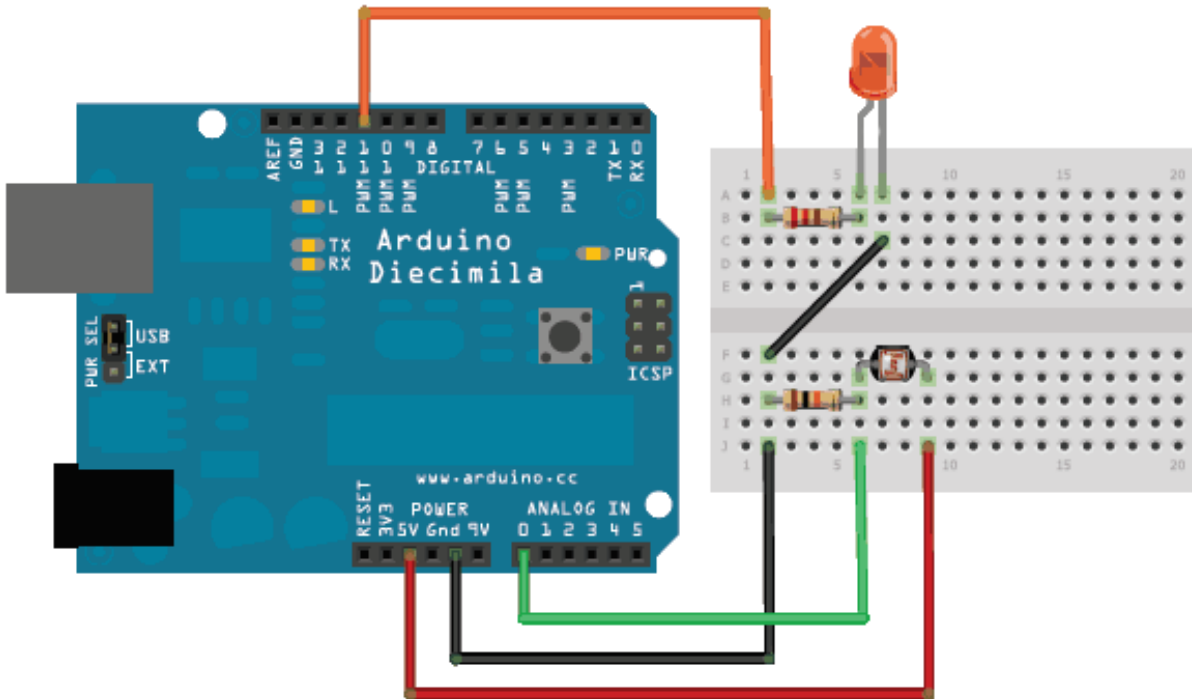


Makineler (Bilgisayarlar, Robotlar) ise çevrelerinde olup bitenleri algılamak için sensörleri kullanır.



Hava Kararınca Yanan Led

LDR'ler ışığa duyarlı dirençlerdir. Aydınlıkta dirençleri minimum, karanlıkta ise maksimumdur. Biz bu özelliğini kullanarak gece olduğunda yani karanlıkta yanan Led yapacağız. Hava kararınca Ldr'nin direnci maksimum olacak ve yazacağımız kod ile bu durum gerçekleştiğinde Led yanacak.



Gerekli Malzemeler

- | | |
|---------------------------------|---|
| 1 adet Arduino uno | 1 adet 220Ω direnç (Led'e bağlanacak) |
| 1 adet breadboard | 1 adet 10000Ω (10KΩ) direnç (Ldr'ye bağlanacak) |
| 1 adet Ldr(foto direnç) | 1 adet Led |
| 5 adet erkek-erkek jumper kablo | |



Eğer sensör değeri 500'den küçük ise, Led'imizin bağlı olduğu digital 11 pini aktif olacak yani Led yanacak. Sensör değeri 500'den küçük değilse, digital 11 pini pasif olacak, giden akım kesilecek, Led sönecek.

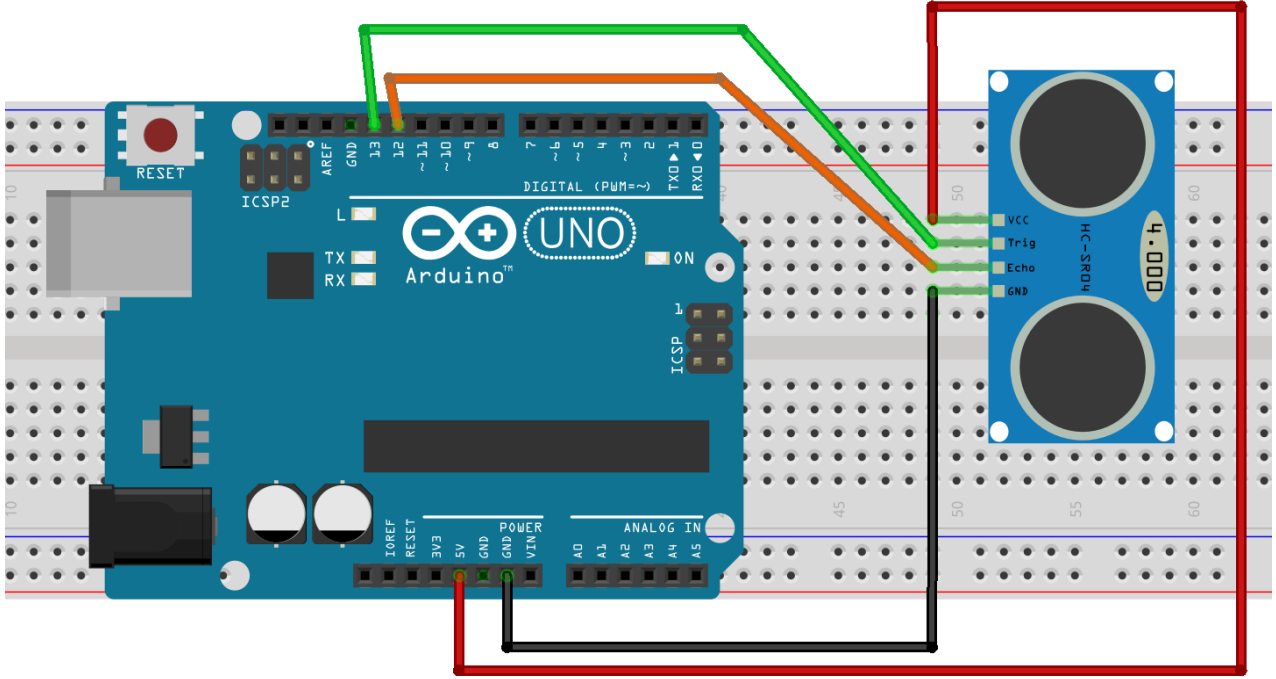
Bu uygulamada LDR'nin bir ucu Arduinonun analog 0 pinine bağlıdır ve buradan değeri okumaktadır. Bu değeri "analog pim A(0) (analog 0 pinine bağlı sensörün değeri)" komutu ile kontrol ederiz.

Bu pinden 0 ile 1023 arası değerler okunur. 1023 aydınlık, 0 karanlık. Işığın değerine göre de 0-1023 arası değer okunur.

Burada bir sürekli bloğunun içerisinde bir eğer ise değilse yapısı kullandık. Bir koşul belirledik. Sensör değeri 500'ün altında mı , değil mi?

Mesafe Sensörü

Ultrasonik mesafe sensörünü projelerimizde cisimlerle olan mesafeleri ölçmek için kullanırız. Engelleri algılayıp aşan robotlarda sıkça kullanılır. Biz arabalardaki sistemlere benzer bir mesafe sensörü devresi yapacağız.



```
🚩 tıkladığında  
sürekli tekrarla  
mesafe , 13 ultrasonik algılayıcı trig pimi oku 12 pimi göster olsun
```

Ultrasonik sensörden okunan değeri tutan kod bloğu

Scratch ile Yazdığımız Programı Arduino'ya Yükleme

Scratch ile yazdığımız programlar normalde bilgisayar üzerinden çalışır. Program çalışmayı durdurduğunda ya da Arduino'nun bilgisayar ile bağlantısını kopardığımızda yazdığımız program çalışmayı durdurur.

Mblock programının Arduino kipi özelliği ile yazdığımız kodları Arduino'ya yükleyebiliriz ve Arduino üzerinden çalışmasını sağlayabiliriz.

[Youtube Videosu İçin Tıklayınız!](#)

