

2013

SCRATCH EĞİTİMİ DERS NOTLARI



TÜBİTAK

Bilişim ve Bilgi Güvenliği İleri Teknolojiler Araştırma Merkezi
Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü
Matematiksel ve Hesaplamalı Bilimler Birimi |

TÜBİTAK BİLGEM UEKAE
17.06.2013



Bu Eğitim,

Programlamanın Temellerini Sizlere Eğlenceli ve Kolay Bir Programlama Ortamında Öğretmek Amacıyla Hazırlanmıştır.

Hazırlayanlar

Şükran Külekci

Fatma Çelik

Esna Koçyiğit

H. Gökçe Macit



PROGRAMLAMA NEDİR?

Genel olarak programlama;

Kullanıcıların isteklerini karşılayacak Muhasebe, Stok, Oyun, İşletim Sistemi gibi programları oluşturacak alt yapıyı (Kodları) hazırlamaktır.

Yani programlama, bilgisayarımıza istediklerimizi yaptırabilmemiz ve bilgisayarımız ile aynı dili konuşabilmemiz için gereklidir.

ALGORİTMA NEDİR?

Algoritma bir uygulamanın, işin, oluşun vb. işlem basamaklarını teker teker yazarak çözümlene ve sonuca varmada izlenecek yoldur. Algoritma programcılık açısından önemli bir olaydır. Scratch programında da farkında olmadan algoritmalar kuracaksınız.

Algoritmaya örneği;

Problem: Elazığ'dan İstanbul'a gitmek için bir algoritma oluşturun.

Burada öyle bir algoritma oluşturacaksınız ki algoritmanın sonunda istanbul' da olacaksınız.

1.YOL:

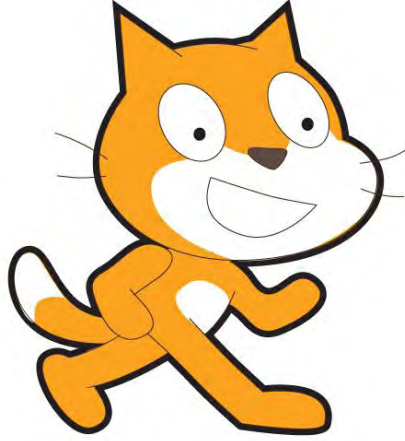
- 1.adım: Evden çıkarım.
- 2.adım: Otopara giderim.
- 3.adım: İstanbul'a bilet alırım.
- 4.adım: Otobüse binerim.
- 5.adım: 18-19 saat yolculuk yaparım.
- 6.adım: İstanbul'da inerim.

2.YOL:

- 1.adım: Evden çıkarım.
- 2.adım: Hava alanına giderim.
- 3.adım: İstanbul'a bilet alırım.
- 4.adım: Uçağa binerim.
- 5.adım: 2 saat yolculuk yaparım.
- 6.adım: İstanbul'da inerim.



SCRATCH



versiyon 1.4

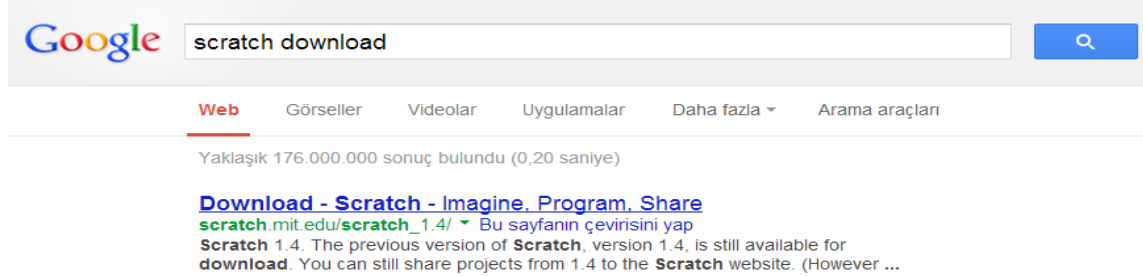
1. GİRİŞ

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>' linkine tıklayarak Scratch ana sayfasına ulaşabilirsiniz.



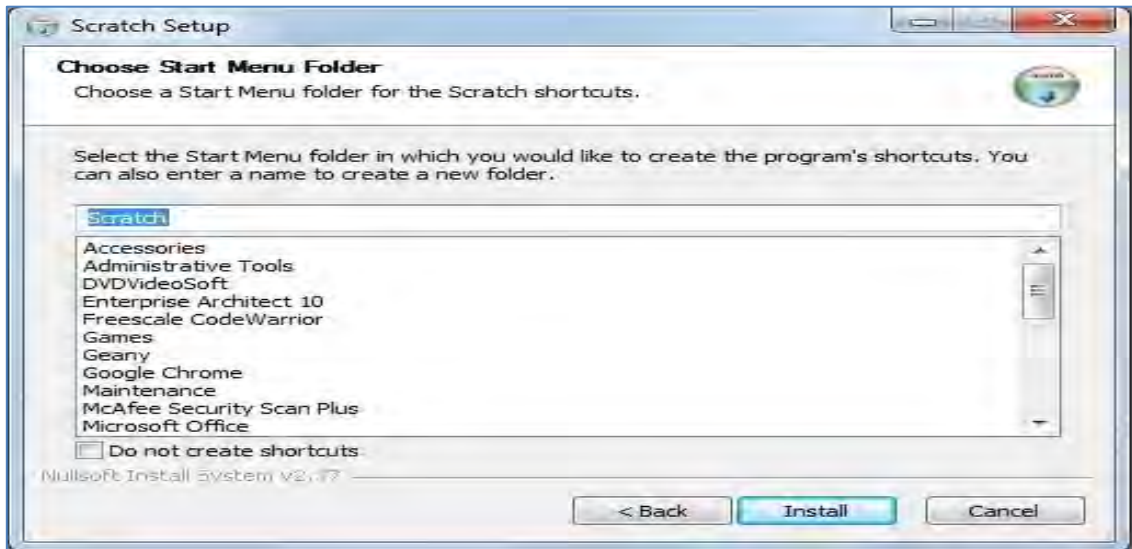
Google Arama Motoruna: **scratch download** yazıyoruz **mit'** nin sitesine giriyoruz.



Açılan sayfada işletim sistemi tercihi sunulmuştur. Kendi işletim sistemimize uygun olanı seçip indiriyoruz.



İndirilen ScratchInstaller1.4 dosyasını sağa tıklayıp çalıştırıyoruz. Açılan pencerede Next-Next- Install kısımlarına sırayla tıklayarak program kurulumunu tamamlamı oluyoruz.



KARAKTER:

Scratch projeleri karakterler olarak adlandırılan objelerden oluşur. Seçtiğimiz bir karaktere farklı kostümler giydirebilir karakterin görünümünü değiştirebiliriz. Karakterimizi hayvana, metine, insana, araca vs. benzetebiliriz. Yeni bir karakter oluştururken ‘Yeni Karakter’ bölümünden Scratch programının bizlere sunduğu hazır karakterlerden seçebilir, bilgisayarımızda yüklü olan bir öğeyi açabilir veya kendimiz yeni bir imaj çizebilir.

Scratch programı ilk açıldığında karşımıza çıkan kedi bir karakterdir.



KOD BLOKLARI:

Kod blokları karakterimize istediğimizi yaptırabilmemiz için kolay tasarımıyla bizlere sunulmuş araçlardır. Kod bloklarında bulunan kodları alt alta sıralayıp komut dizisi oluşturarak karakterimizin hareket etmesini, müzik çalmasını, işlemler yapabilmelerini sağlayabiliriz.

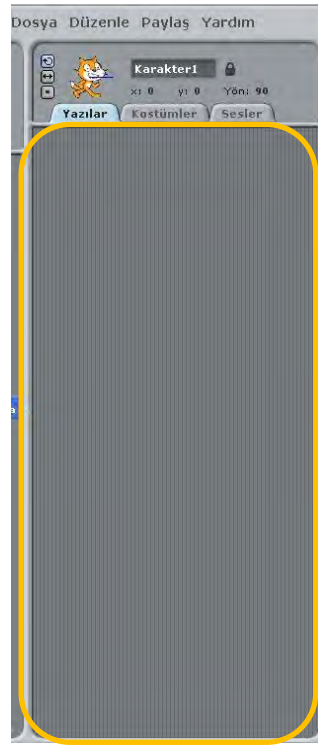
2. SCRATCH ARAYÜZÜ

Scratch'ın gelişmiş ve kullanması kolay bir arayüzü vardır.

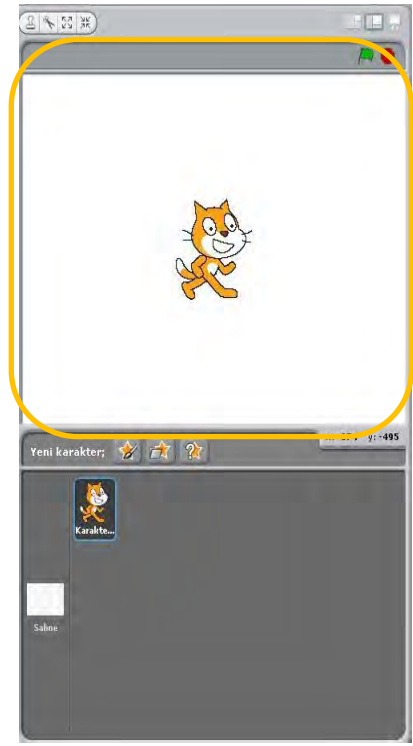
Scratch arayüzü 3 ana bölmeden oluşur.



1. BLOK PALETİ:
Karakterimizi programlamak için kullanabileceğimiz blokların bulunduğu kısımdır.



2. KODLAMA ALANI:
Blokları sürükleyerek komut dizileri oluşturacağımız alandır.



3. PROJE EKRANI:
Bizim sahnemizdir. Projemizi çalıştırdığımızda tasarladığımız herşey burada hayat bulur



Dil ayarla: Bu ikonu kullanarak kullandığımız blok dilini değiştirebilir farklı dillerde kullanabiliriz.

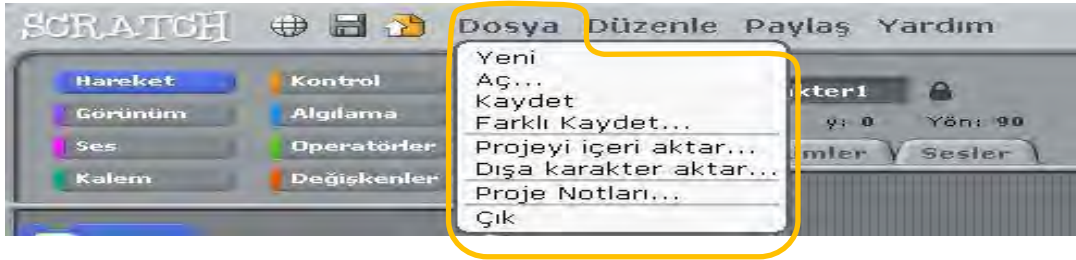


Bu projeyi Kaydet: Bu ikon ile hazırladığımız projelerimizi kaydedebiliriz.



Bu Projeyi Paylaş: Bu ikonu kullanarak Projemizi Scratch'ın web sayfasına yükleyebiliriz.

DOSYA MENÜSÜ



Dosya menüsünde 8 seçenek bulunur. Bu seçenekler ile yapabileceklerimiz şunlardır;

Kaydet: Bu seçenek ile oluşturduğumuz projeyi diske kaydederiz. Kaydet seçeneğine tıkladıktan sonra açılan pencerede projeye ilgili bilgileri girdikten sonra Tamam' a tıkladığımızda projemiz kaydolmuş olur.

Yeni: Projemizi kaydettikten sonra yeni proje oluşturmak için Dosya menüsünden Yeni' ye tıklarız.

Aç: Daha önceden hazırladığımız bir projeyi veya Scratch programını kurarken program ile birlikte yüklenen örnek projeleri açmak, üzeri

nde çalışabilmek için Dosya menüsündeki Aç' a tıklarız.

Farklı Kaydet: Daha önceden kaydettiğimiz bir projeyi farklı bir isimle kaydetmek istiyorsak Farklı Kaydet' e tıklarız.

Projeyi içe aktar: Daha önceden kaydettiğimiz bir projenin karakterlerini ve kodlarını şu anda açık olan projeye aktarmamızı sağlar.

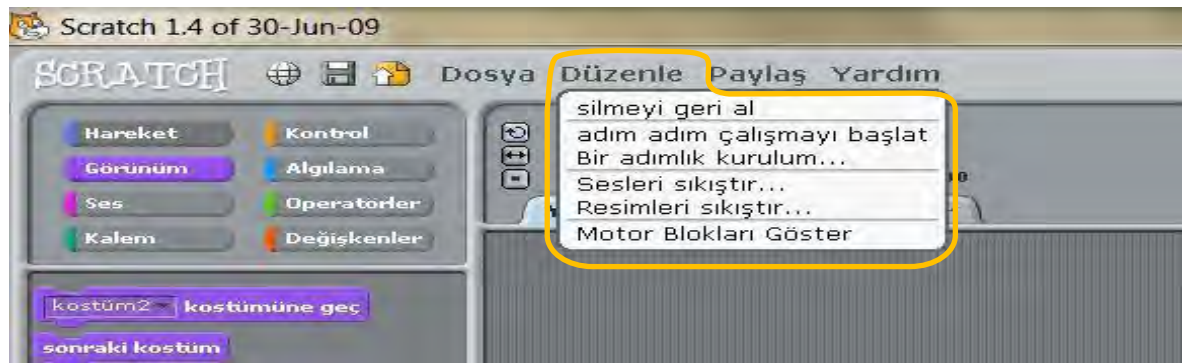


Dışa karakter aktar: Projemizde şunda seçili olan karakterimizi daha sonra başka projelerde kullanabilmek için, Dışa aktar seçeneği ile kaydedebiliriz.

Proje notları: Proje ile ilgili notlar alıp daha sonra bu notlara bakabilir veya notları düzenleyebiliriz.

Çık: Scratch programından çıkış yapmamızı sağlar.

DÜZENLE MENÜSÜ



Düzenle menüsünde 6 seçenek bulunur. Bu seçenekler işimize daha çok yarayacak 5 tanesi ile neler yapabileceğimize bakalım.

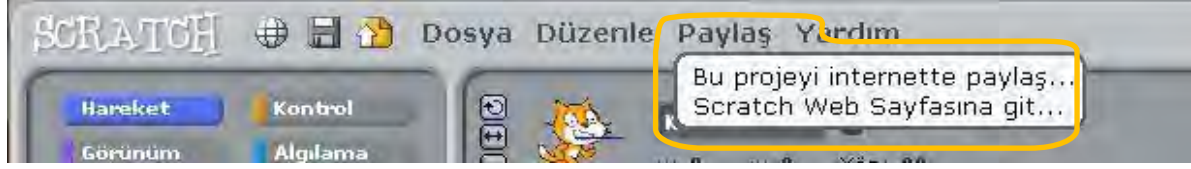
Silmeyi geri al: Scratch' te çalışırken sildiğimiz bir şeyi o anda geri alabilmek için düzenle menüsünde yer alan Silmeyi geri al' a tıklayarak bunu sağlayabiliriz.

Adım adım çalışmayı başlat: Bu seçenek hazırlamış olduğumuz kodların her birini çalışma anında ayrı ayrı görebilmemizi sağlar. Projenin çalışma esnasında programın akışını görebiliriz aynı zamanda bir hata ile karşılaşıldığında bu düzen ile hatayı bulmamız kolaylaşır.

Bir adımlık kurulum: Bu seçeneğimiz Adım adım çalışma mantığıyla aynı olup, farkının bir sefer çalıştırılacak olup bizlere farklı çalışma hızı seçeneği sunmasıdır. (Yüksek hız, Falsh blokları (hızlı) ...)

Sesleri/Resimleri sıkıştır: Proje içerisinde kullandığımız sesleri/resimleri sıkıştırarak kapladıkları yer miktarını azaltabiliriz.

PAYLAŞ MENÜSÜ



Bu Projeyi İnternette Paylaş: Bu seçenek daha önceden bahsettiğimiz Projeyi Paylaş ikonu ile aynı görevi üstlenmiştir. Hazırladığımız projemizi Scratch web sayfası üzerine yüklememizi sağlar.

Scratch Web Sayfasına git: Bu seçeneğe tıkladığımızda web tarayıcınızda Scratch ana sayfasına gidilir.

YARDIM MENÜSÜ

Yardım Menüsü anlayamadığımız ya da merak ettiğimiz konulara göz atabilmemizi sağlar.

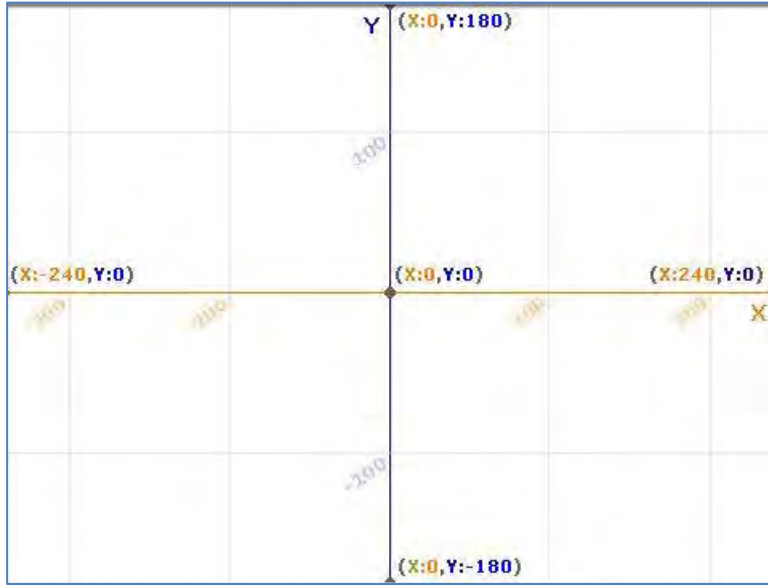




EKRAN

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.





1.



Damga Simgesi ile var olan karakterimizi çoğaltabiliriz.



Makas Simgesi ile istediğimiz bir karakteri silebiliriz.



Bu simgeyi seçerek karakterimizin üzerine tıkladığımızda karakterimizi istediğimiz kadar büyültebiliriz.



Bu simgeyi seçerek karakterimizin üzerine tıkladığımızda karakterimizi istediğimiz kadar küçültebiliriz.

2.



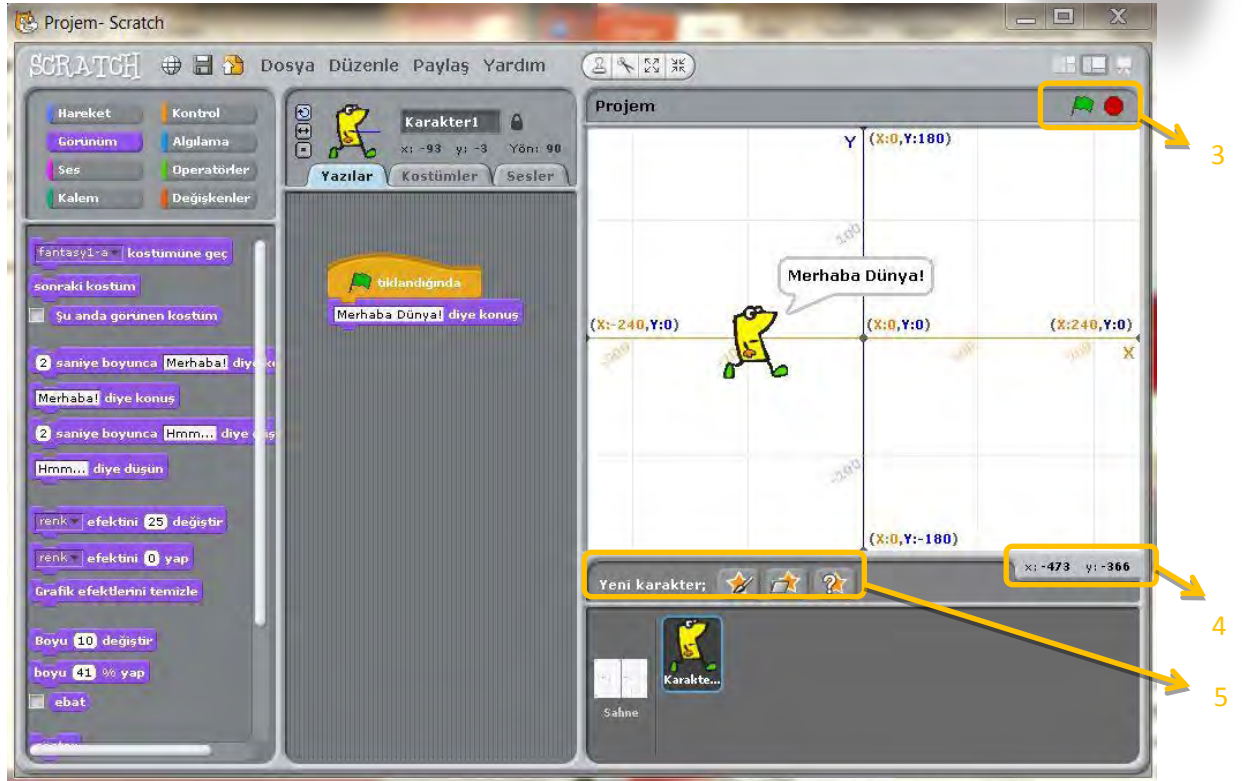
Bu seçeneği Yazı alanımızı daha büyük Sahne alanımızı daha küçük görmek için seçebiliriz.



Bu seçeneği Sahne alanımızı daha büyük Yazı alanımızı daha küçük görmek için seçebiliriz.



Bu simgeyi seçtiğimiz andan itibaren ekranımız bilgisayar ekranını kaplar projemiz sunum durumuna geçer. Bu seçeneği projemiz tamamlandığında olayları görebilmek için seçmek daha uygun olur.



3.

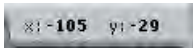


Yeşil bayrak ile projemizi çalıştırabiliriz



Kırmızı daire ile projemizde çalışan her şeyi durdurarak ilk hale getirebiliriz.

4.



Bu simge mause'umuzun x ve y konumunu gösterir.

5.



Yeni karakter seç simgesine tıklayarak kendimiz karakter çizerek projemize yeni bir karakter ekleyebiliriz.



Dosyadan yeni karakter seç simgesi ile var olan örnek bir karakteri seçip projemize ekleyebiliriz.



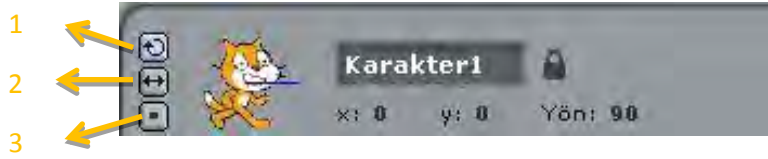
Sürpriz karakter getir seçeneği ile Scratch programı bize rastgele bir karakter seçerek projemize ekler.

ŞİMDİKİ KARAKTER BİLGİSİ

Bu kısımda karakterimizin adı, x-y pozisyonu ve yön bilgisi yer almaktadır. Karakterimizin isminin yazdığı yere tıklayarak karakter adını değiştirebiliriz. Karakter adının yanında yer alan kilide tıklanarak karakterin web üzerine taşınabilme izni açılır veya kapatılır. Karakterin yönü, proje çalışmaya başladığında karakterin hangi yöne gideceğini belirtir.

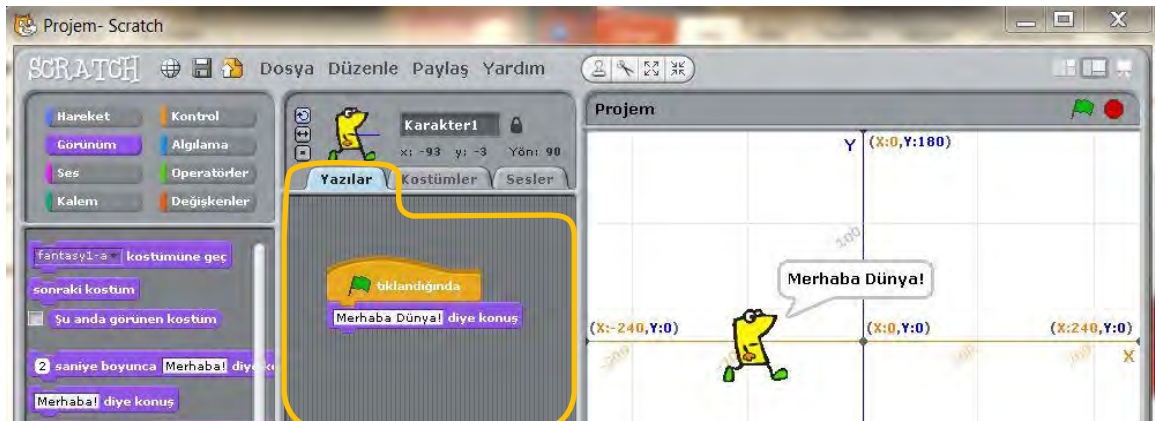
1. Tıklanırsa karakter kostümü 360° dönebilme özelliği kazanır.
2. Tıklanırsa karakter kostümü sağa-sola dönebilme
3. Tıklanırsa karakter kostümü hiç bir şekilde dönemez.

Karakterin üzerine çift tıkladığında x, y ve yön değerleri kabul değerlerine (x=0, y=0, Yön= 90) döner.



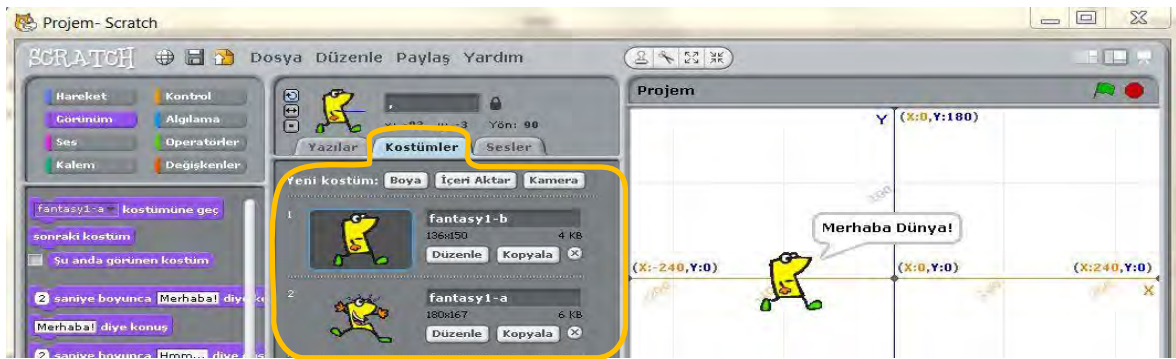
YAZILAR

Scratch ile bilgisayar programı yazmak için kodların bloklar halinde üs üste ve yan yana bir araya getirilmesi gerekir. Scratch programının Çek-Bırak özelliği sayesinde karakterin arzu edilen şeyleri yapabilmesi için yazılar bölümünde kod blok dizileri oluşturulur.



KOSTÜMLER

Karakterimizin başka görünülerinin olmasını istiyorsak kamera ile çekilebilir kendimiz çizebilir ya da içe aktarabiliriz. Var olan kostümleri 'X' butonuna basarak silebiliriz, 'Düzenle' butonu ile kostümü düzenleyebiliriz veya 'Kopyala' butonu ile kostümün aynısından bir tane daha ekleyebiliriz.



SESLER

Karakter için var olan sesleri görebilmek için bu bölüme tıklarız. Bu bölümden yeni sesler ekleyebilir (Kayıt/İçeri Aktar), var olan sesi dinleyebilir veya silebiliriz.





3. BLOK PALETİ

Blok Paletinde karakterleri programlayabilmek için kullanılan bloklar vardır. 6 ana kategoride toplanmış blok grubu bulunmaktadır.



HAREKET:

Hareket bloğunda yer alan blokların işlevleri tablodaki gibidir.

10 adım git	Karakteri öne ya da arkaya doğru hareket ettirir.
15 ↻ derece dön	Karakterin istenilen derece kadar saat yönünde dönmesini sağlar.
15 ↺ derece dön	Karakterin istenilen derece kadar saat yönü tersine dönmesini sağlar.
90 yönüne dön	Karakterin hangi yöne döneceğini belirler. (0= yukarı, 90= sağ, 180= aşağı, -90= sol)
▼ doğru dön	Karakteri 'mause işaretçisi' veya başka karakterler varsa istenilen 'karakter' e doğru döndürülür.
x: 2 y: 0 konumuna git	Karakteri istenilen x ve y konumuna getirir.
▼ ile aynı konuma git	Karakter 'mause işaretçisi' veya başka karakterler varsa istenilen 'karakter' ile aynı konuma gönderilir.
1 saniyede x: 2 y: 0 konumuna git	Karakter belli bir süre içerisinde belirlenen x ve y konumuna hareket eder.
x'i 10 değiştir	Karakterin bulunduğu konumun x değeri istenilen değer kadar artırılır veya azaltılır.
x'i 0 yap	Karakterin x konumunu istenilen değere çeker.
y'yi 10 değiştir	Karakterin bulunduğu konumun y değeri istenilen değer kadar artırılır veya azaltılır.
y'yi 0 yap	Karakterin y konumunu istenilen değere çeker.
kenara geldiğinde geri dön	Karakter ekranın kenarına değdiği an karakteri ters yöne döndürür.
■ x koordinatı	Karakterin x pozisyonu bilgisini verir. Bu seçenek işaretlendiğinde x değerini ekranda görebiliriz.
■ y koordinatı	Karakterin y pozisyonu bilgisini verir. Bu seçenek işaretlendiğinde y değerini ekranda görebiliriz.
■ Yön	Karakterin yön bilgisini verir. Bu seçenek işaretlendiğinde yön bilgisini ekranda görebiliriz.



GÖRÜNÜM:

Görünüm bloğunda yer alan blokların işlevleri tablodaki gibidir.

cat1-a kostümüne geç	Karakteri istenilen kostüme geçmesini sağlar.
sonraki kostüm	Karakterin o anki kostümünden bir sonraki kostümüne geçmesini sağlar.
<input type="checkbox"/> Şu anda görünen kostüm	Karakterin o anki kostüm bilgisini verir. Yan tarafındaki kutu işaretlendiğinde ekranda kostüm bilgisi görülür.
2 saniye boyunca Merhaba! diye konuş	Karakter istenilen süre boyunca 'Merhaba' yazan kutudaki değeri ekranda konuşma balonu içerisinde gösterir.
Merhaba! diye konuş	Karakter 'Merhaba' yazan kutudaki değeri ekranda balon içerisinde gösterir. Kutu içerisi boş bırakılırsa konuşma balonu oluşmaz.
2 saniye boyunca Hmm... diye düşün	Karakter istenilen süre boyunca 'Hmm...' yazan kutudaki değeri ekranda düşünme balonu içerisinde gösterir.
Hmm... diye düşün	Karakter 'Hmm...' yazan kutudaki değeri ekranda balon içerisinde gösterir.
renk efektini 25 değiştir	Karakterin rengi, balıkgözü, Hızlı dön, Piksellere böl, mozaik, parlaklık, hayalet efekti sayı yazan yerdeki değer kadar değişir.
renk efektini 0 yap	Karakterin rengi, balıkgözü, Hızlı dön, Piksellere böl, mozaik, parlaklık, hayalet efekti sayı yazan yerdeki değer olur.
Grafik efektlerini temizle	Karakter üzerinde uygulanmış bütün efektleri geri alır.
Boy 10 değiştir	Karakterin boyunu istenilen değer kadar değiştirir.
boyu 100 % yap	Karakterin boyu istenilen %' ye getirilir.
<input type="checkbox"/> ebat	Karakterin ebat değerini % olarak verir. Yan tarafındaki kutu işaretlendiğinde ebat değeri ekranda görünür.
göster	Karakteri ekranda görünür hale getirir.
gizle	Karakteri Ekranda görünmez yapar.
üste çık	Karakterin diğer bir üst katmana çıkmasını sağlar. Böylece karakter diğer karakterlerin önünde görünebilir.
1 katman alta geç	Karakterin istenilen katman değeri kadar geri gitmesini sağlar. Böylece karakter diğer karakterlerin arkasında gizlenebilir.



SES:

Ses bloğunda yer alan blokların işlevleri tablodaki gibidir.

<input type="text" value="miyav"/> sesini çal	Seçili olan müzik çalmaya başlar. Program müziğin bitmesini beklemeden diğer blokları çalıştırmaya devam eder.
<input type="text" value="miyav"/> sesini bitene kadar çal	Seçili olan müzik çalmaya başlar ve bitene kadar diğer bloğa geçilmez.
tüm sesleri durdur	Çalmaya devam eden müzikler bu blok çalıştığında durur.
<input type="text" value="48"/> sesini <input type="text" value="0.2"/> vuruş çal	48 (davul) sesini istenilen istenen sürede bir çalar. (48 sesinin bulunduğu yer açılır menüdür ve ses buradan değiştirilebilir.)
<input type="text" value="0.2"/> vuruşunu bekle	İstenilen vuruşun bitmesini bekler.
<input type="text" value="60"/> notasını <input type="text" value="0.5"/> vuruş çal	İstenilen notayı istenilen sürede bir çalar.
Enstrümanı <input type="text" value="1"/> yap	Enstrümanı değiştirmeyi sağlar.
Sesi <input type="text" value="-10"/> değiştir	Sesi azaltır ya da artırır.
ses yüksekliğini % <input type="text" value="100"/> yap	Ses yüksekliğini belli bir % ye getirir.
<input type="checkbox"/> ses seviyesi	O anki ses seviyesi bilgisini verir. Yan taraftaki kutu işaretlendiğinde ekranda ses bilgisi % olarak görünür.
Tempoyu <input type="text" value="20"/> değiştir	Tempo değeri istenilen değere olarak belirlenir.
tempoyu saniyede <input type="text" value="60"/> vuruş yap	Tempo değeri saniyede istenilen vuruş değeri yapılır.
<input type="checkbox"/> tempo	O anki tempo değeri bilgisini verir. Yan taraftaki kutu işaretlendiğinde ekranda tempo değeri görünür. (20-500)












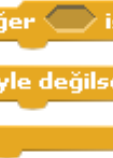




**KALEM:**

Kalem bloğunda yer alan blokların işlevleri tablodaki gibidir.

Kalemle çizilenleri temizle	Ekrana kalem ile çizilenlerin hepsini siler.
Kalemi bastır	Karakterin hareket ettiği doğrultuda kalem ile ekranı çizer.
Kalemi kaldır	Karakterin hareket ettiği yerler çizilmez.
Kalem rengini [renk] yap	Kalem rengi istenilen renk değerine geçer.
Kalem rengini [değer] değiştir	Kalem rengi bu blok çalıştığında istenilen değer kadar değişir.
Kalem rengini [0] yap	Kalem renginin istenilen değer olması sağlanır.
Kalem gölgesini [değer] değiştir	Kalem gölgesi istenilen değer kadar değiştirilir. (kalem gölgesi 0' dan 100' e doğru açılır, 100' den 200'e doğru koyulaşır.)
Kalem gölgesini [50] yap	Kalem gölgesi istenen değer olur.
Kalem boyunu [değer] değiştir	Kalem boyu istenen değer kadar değişir.
Kalem boyunu [1] yap	Kalem boyu istenen değer olur.
damgala	Karakterin o anki konumunda ekrana kendi imajını yapıştırır.

KONTROL:

Kontrol bloğunda yer alan blokların işlevleri tablodaki gibidir.

 tıkladığında	Proje içerisinde yeşil bayrağa tıkladığında bu blok altında sıralanmış kod kümesini çalıştırır.
 boşluk tuşuna basıldığında	Seçilmiş olan tuşa (burada boşluk tuşu seçilmiş) basıldığında bu blok altında sıralanmış kod kümesini çalıştırır.
 Karakter1 tıkladığında	Karaktere tıkladığında bu blok altında sıralanmış kod kümesini çalıştırır.
 1 saniye bekle	Projeyi istenilen süre kadar bekletir, süre tamamlanınca sıradaki bloktan çalışmaya devam eder.
 sürekli	Bu blok içerisinde bulunan bloklar sırasıyla program durdurulana kadar tekrar tekrar çalıştırılır (sonsuz döngü).
 10 kez tekrarla	Bu blok içerisinde bulunan bloklar sırasıyla istenilen değer kadar tekrar tekrar çalıştırılır.
 duyurusu yap	Bu blok ile yeni duyurular yazarak komut kümelerini çalıştırmak için yeni bir blok oluşturmuş oluruz. Bu blok çalıştığında yapılan duyurunun ardından kendi icrası devam ederken bu duyuruya ait kod kümesi de icraya başlar.
 duyurusu yap ve bekle	Yapılan duyurunun ardından bu bloğun altında kod kümesi var ise bu duyuruya ait kod kümesinin icrasını bekler, bitiminden sonra kendi icrasına devam eder.
 duyurusu yapıldığında	Yapılan duyurunun ardından bu blok altındaki kod kümesi çalışır.
 eğer ise sürekli	Eğer' in ardından gelen koşulun doğru olması durumunda program bu blok içerisine girer ve içerideki blokları sürekli olarak (sonsuz döngü) çalıştırır. Koşul doğru değil ise bu blok içerisine girilmez.
 eğer ise	Eğer' den sonra gelen koşul doğru ise program bu blok içerisine girer ve bu blokları çalıştırır, ardından bu blok içerisinden çıkarak sonraki bloktan çalışmaya devam eder.
 eğer ise öyle değilse	Eğer koşul doğru ise hemen altında yer alan kod blokları icra edilir ve bu kod bloğundan çıkılarak program icraya devam eder. Eğer koşul yanlış ise 'öyle değil ise' kısmında yer alan bloklar icra edilir, icradan sonra bu bloktan çıkılarak program akışı kaldığı yerden devam eder.
 olana kadar bekle	Koşul doğru olana kadar beklenir, koşul doğru olunca aşağıdaki bloklar çalışmaya başlar.
 olana kadar tekrarla	Koşul doğru olunca blok içerisine girilerek kodlar icra edilir. Koşulun her döngüde doğru olması halinde icra devam eder. Koşul doğrulanmadığı halde altındaki kodlar çalıştırılır.
 bu blokları durdur	Bulduğu blok kümesini durdurur.
 bütün blokları durdur	Proje içerisindeki bütün kod kümelerini durdurur.



ALGILAMA:

Algılama bloğunda yer alan blokların işlevleri tablodaki gibidir.

	Karakterimizin karaktere/ Mause işaretçisine/ kenara değip değmediği kontrolünü yapar. (seçim için açılır menüye tıklayın.)
	Karakterin seçilen renge değip değmediği kontrolünü yapar. (Rengi değiştirmek için renk üzerine tıklayıp açılan pencereden renk seçin.)
	Karakterin üzerindeki seçtiğiniz rengin seçtiğiniz bir başka renge değip değmediği kontrolünü yapar.
	Sormak istediğiniz soruyu 'Adın ne?' kısmına yazarak Karakterin konuşma balonunda bu soruyu sormasını sağlarsın. Sorunun ardından klavyeden yanıt girmen beklenir.
	Sor ve bekle komutu ardından verdiği yanıt bu 'yanıt' buluşunda tutulur. Blok yanındaki kutuya tıkladığında vermiş olduğu yanıt proje ekranında görünür.
	Farenin x koordinatını verir.
	Farenin y koordinatını verir.
	'Fare basılı' mı kontrolü yapar.
	Seçili olan tuşa basılı olma durumunu kontrol eder.
	Karakterin açılır menüden seçtiğimiz başka bir karakter ya da mause işaretçisiyle arasında olan mesafe bilgisini verir.
	Sayacı sıfırlar.
	Program içerisinde proje açılmasından itibaren geçen süre bilgisini tutar. Yan taraftaki kutuya tıkladığında proje ekranında bu süre bilgisi görünür.
	Seçili olan karakterin/sahnenin seçenekler dahilindeki (x koordinatı/ y koordinatı/ yön/ şu anda görünen kostüm/ ebat/ ses seviyesi) bilgisini kullanabilmemizi sağlar.
	Ses yüksekliği bilgisini barındırır.
	Bu blok Scratch tahtası ile birlikte kullanılır. Tahtada bulunan düğmeye basıldığı, A girişi/ B girişi/ C girişi/ D girişi bağlandı kontrolünü yapar.
	Bu blok Scratch tahtası ile birlikte kullanılır. Öğrenmek istenilen (sürgü, ışık, ses, direnç-A, direnç-B, direnç-C, direnç-D, eğim, mesafe) sensör değerini barındırır



OPERATÖRLER:

Operatörler bloğunda yer alan blokların işlevleri tablodaki gibidir.

	Boş kutulara yazılan iki değeri toplar.
	İlk kutudaki değerden ikinci kutudaki değeri çıkarır.
	İki değeri çarpar.
	İlk kutudaki değeri ikinci kutudaki değere böler.
	Belirtilen iki değer arasında rastgele bir sayı üretir.
	İlk kutudaki değerın ikinci kutudaki değerden küçük durumu kontrolü yapılır.
	İki değerin bir biri iler eşitliği durumu kontrolü yapılır.
	İlk kutudaki değerın ikinci kutudaki değerden büyük durumu kontrolü yapılır.
	Belirtilen iki koşulunda doğruluk kontrolünü yapar.
	Belirtilen iki koşuldan en az birinin doğru olması durumunu arar.
	Koşulda verilen durumun olmaması istendiği durumlarda bu blok kullanılır.
	İstenilen iki değeri metin olarak birleştirilir.
	İstenilen değerın hangi sırasındaki harfi isteniyor ise bu blok ile bu bilgi elde edilir.
	Değerın uzunluk bilgisini bu blok ile elde edilir.
	Bu blok ilk kutuya yazılan değerın ikinci kutuya yazılan değere göre modu bulunurken elde edilen kalan bilgisini verir.
	İstenilen değerın en yakın tam sayıya yuvarlanmasını sağlar.
	Matematiksel terimleri bulabileceğimiz (mutlak değer, karekök, sin, cos, tan, asin, acos, atan, ln, log, e^, 10^) bu blok ile kutuya yazılan değerın karşılığını bilebilir ve bu değeri sayısal işlemlerde kullanabiliriz.

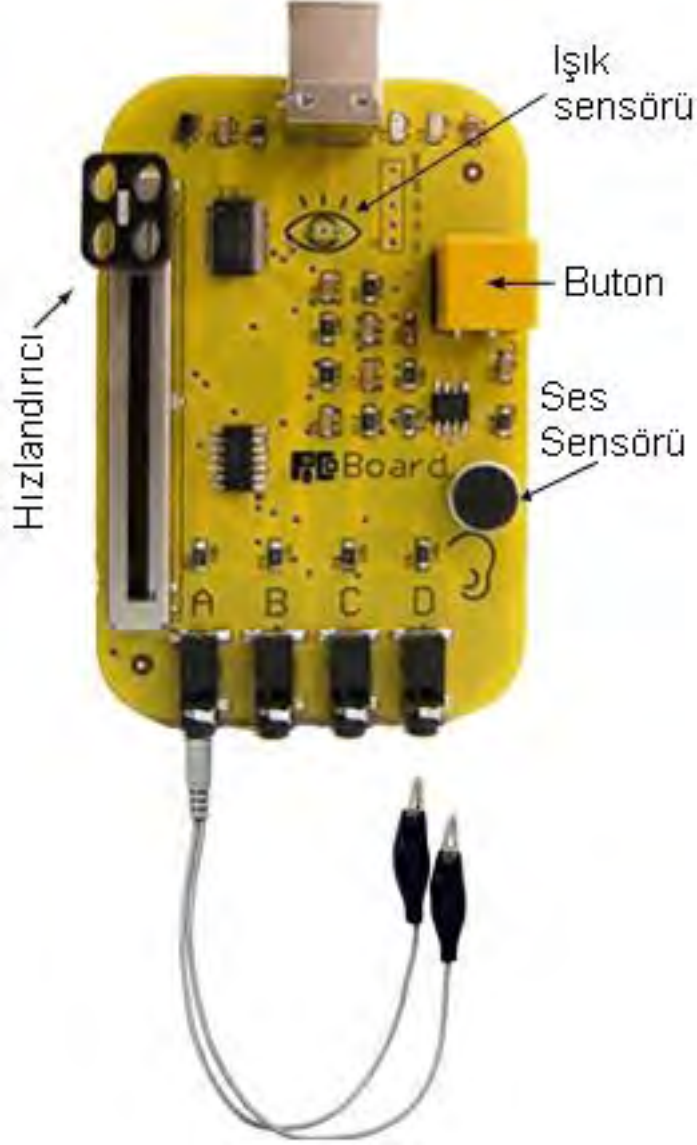


DEĞİŞKENLER:

Değişkenler bloğunda yer alan blokların işlevleri tablodaki gibidir.

Değişken Oluştur	Bu blok ile yeni bir değişken oluştururuz.
Bir değişken sil	Bu blok ile var olan istediğimiz bir değişkeni sileriz.
<input checked="" type="checkbox"/> değişken	Oluşturulan 'değişken' isimli değişken verisi. (Kuru işaretlendiğinde ekranda değişken değerinin ne olduğu görünür.)
değişken değişkenini 0 yap	Var olan değişkenlerden istenilen seçilerek istenilen değer yapılır.
değişken değişkenini 1 değiştir	Var olan değişkenlerden istenilen değişken değeri arzu edilen rakam kadar değiştirilir.
değişken değişkenini göster	Var olan değişkenlerden istenilen değişkeni ekranda gösterir.
Liste Oluştur	Değişkene sahipsem 'Liste oluştur' seçeneği açılır. Bu blok ile bir liste oluşturulur.
Bir liste sil	Var olan bir listeyi buradan silebilirim.
<input checked="" type="checkbox"/> liste	Liste bilgisini barındırır. Kutu işaretlendiğinde liste içeriği ekranda görünür.
liste listesine nesne ekle	İstenen listeye yeni bir nesne eklemeyi sağlar.
liste listesinden 1 elemanı çıkartır	İstenen listeden ilk/son yada bütün elemanları çıkartılır.
liste listesinin 1 sirasına nesne elemanını ekle	İstenen listenin ilk/son veya herhangi bir sırasına yeni bir nesne eklenir. (Eski bilgi silinmez, bu bilgi bir sonraki eleman olarak kaydırılır.)
liste listesinin 1 sıradaki elemanını nesne ile değiştir	İstenen listenin ilk/son veya herhangi bir sıradaki elemanı yerine yeni nesnemiz eklenir.
liste listesinin 1 sıradaki elemanı	İstenen listenin ilk/son veya herhangi bir sıradaki elemanı bilgisini barındırır.
liste listesinin eleman sayısı	İstenen listenin eleman sayısı bilgisini barındırır.
nesne , liste listesinin bir elemanıdır	İstenilen değer seçilen listede olma durumunu kontrol eder.

SENSOR BOARDS



Picobords ile scratch arasında bağlantı kurarak tasarladığımız scratch projelerinin gerçek dünyadaki etkilerini görme fırsatı buluruz.

Ayrıntılı bilgi için aşağıdaki linki ziyaret edebilirsiniz.

http://info.scratch.mit.edu/Sensor_Boards

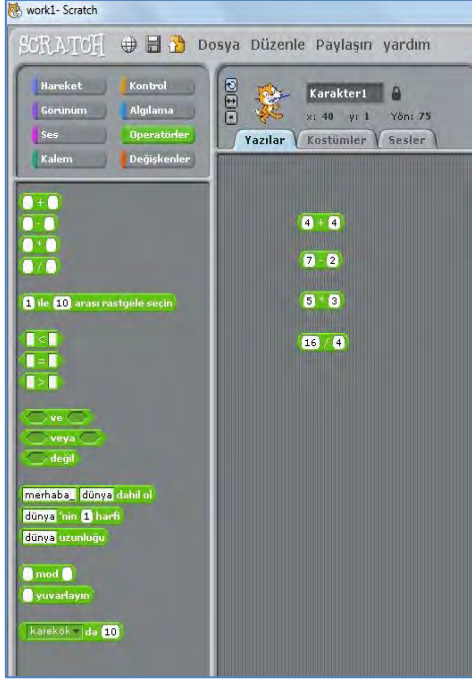
- Bir değişken oluşturalım.
- «**Değişken oluşturun**» kısmına tıklayıp sabit isimli bir değişken oluşturalım.



- Ekranın sol kısmında «**sabit**» değişkenimiz oluştu. Sağ taraftaki beyaz ekranın sol üst köşesinde sabitimiz yerleşti.



OPERATORLER



Operatorler:

Dört işlem yapmak için $+$ $-$ $*$ $/$ operatorlerini kullanacağız.

Karşılaştırma yapmak için

$<$, $=$, $>$ operatorlerini kullanacağız.

Soldaki menüden

« $+$ $-$ $*$ $/$ » işlemleri tutup yazılar kısmına sürüklüyoruz.

Kutucukların içine sayılar yazabiliriz.

Örnek:

- $4 + 4$
- $7 - 2$
- $5 * 3$
- $146 / 4$




Kutucukların üzerine tıkladığımızda işlem sonucunu “baloncuk” içerisinde görebiliriz.




Operatörler menüsündeki diğer işlemleri inceleyelim.

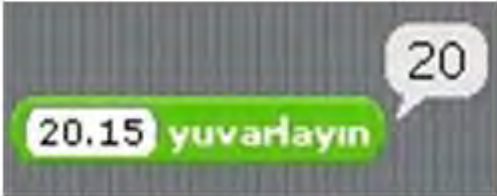
Mod




Karekök hesaplama



Ondalık sayıyı doğal sayıya yuvarlama

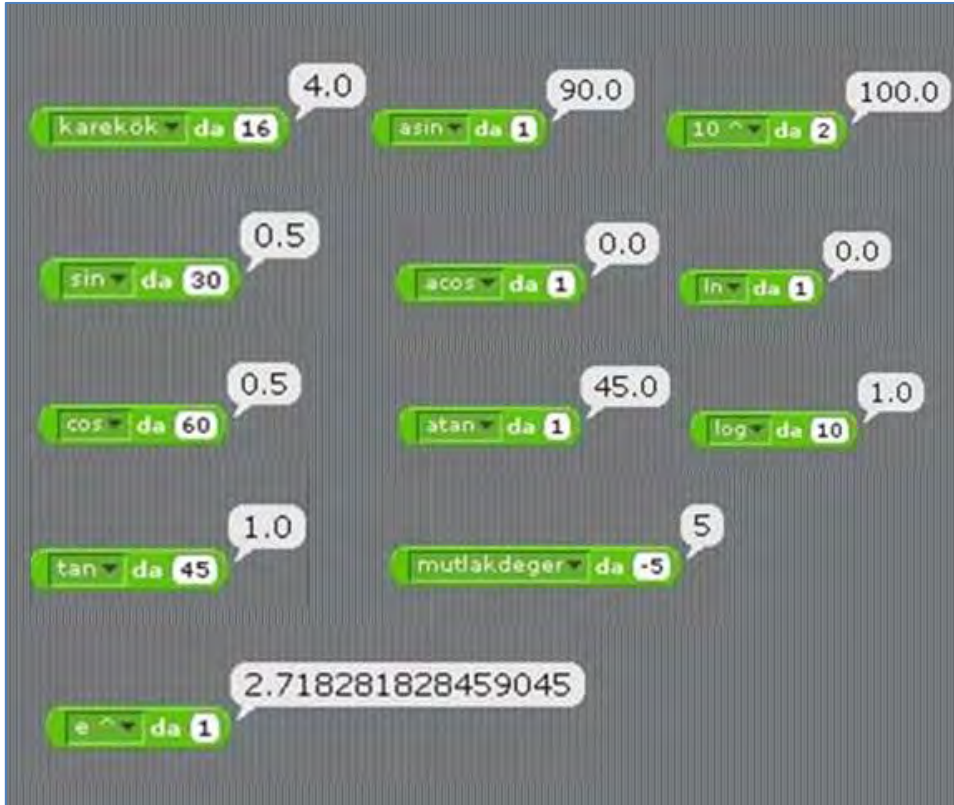




- Karekök yazan kod parçasında ok işaretine tıklarsak başka işlemleri de seçebiliriz.



- Mutlak değer
- Karekök
- Sinüs, cosinüs, tanjant
- Arcsinüs, arccosinüs, arctanjant
- Logaritma, ln
- e üzeri sayı
- 10 üzeri sayı



Rasgele sayı üretme işlemi

Kod parçasının üzerine her tıkladığımızda 1 ile 10 arası farklı sayı üretir.

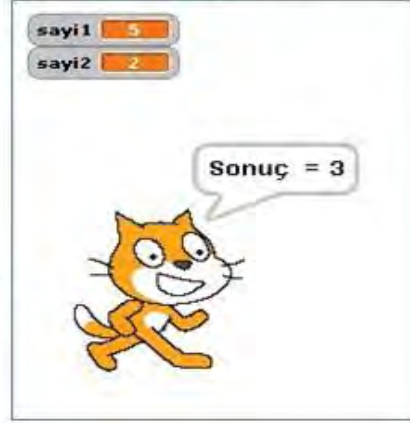


İŞLEM PROGRAMI

- İki tane değişken oluşturalım. İki sayı arasında işlemler yapalım. **sayi1** ve **sayi2** değişkenleri oluşturup aralarında çıkarma işlemi yaptırdık.

```

sayi1 = 5 yapın
sayi2 = 2 yapın
Söyle: Sonuç = sayi1 - sayi2 dahil ol süre: 2 saniye
  
```



BİRÇOK DEĞİŞKENLE İŞLEM

- $(2 * 5) + (16 / 4) + (3 - 1)$ işlemini yapalım.
- Her sayı için bir **değişken** oluşturalım.
- Değişken isimlerimiz :
- **Sayi1 = 2** **Sayi2 = 5**
- **Sayi3 = 16** **Sayi4 = 4**
- **Sayi5 = 3** **Sayi6 = 1**

```

tıklandığında
sayi1 = 2 yapın
sayi2 = 5 yapın
sayi3 = 16 yapın
sayi4 = 4 yapın
sayi5 = 3 yapın
sayi6 = 1 yapın
  
```


İÇİÇE OPERATÖRLER

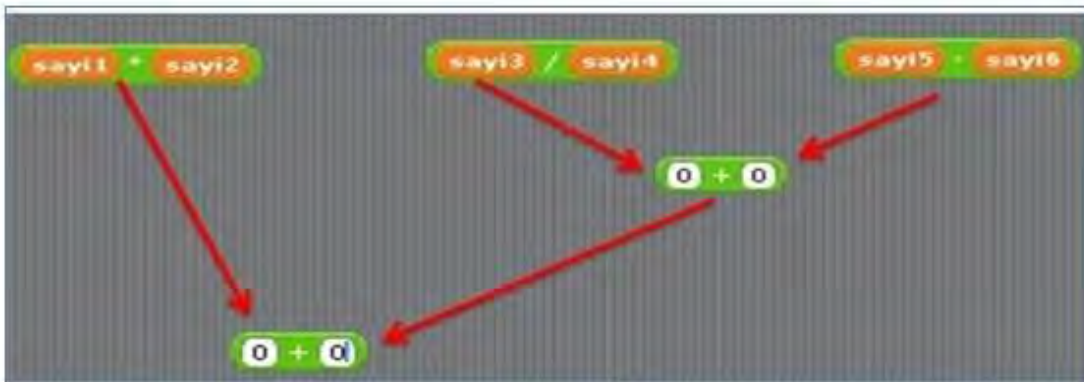
- $(2 * 5) + (16 / 4) + (3 - 1)$ işlemini yapalım.
- Her sayı yerine oluşturduğumuz değişkenleri koyuyoruz:
 $(\text{sayi1} * \text{sayi2}) + (\text{sayi3} / \text{sayi4}) + (\text{sayi5} - \text{sayi6})$



İşlemler iki sayı arasında yapılıyor. Çoklu işlem yapmak için operatörleri birleştireceğiz. 😊

$$(2 * 5) + (16 / 4) + (3 - 1)$$

$$(\text{sayi1} * \text{sayi2}) + (\text{sayi3} / \text{sayi4}) + (\text{sayi5} - \text{sayi6})$$



Ok işareti ile gösterdiklerimizi, 0 yazan yerlere mouse ile sürüklüyoruz. 😊😊

ÇOKLU İŞLEM OLUŞTURMA



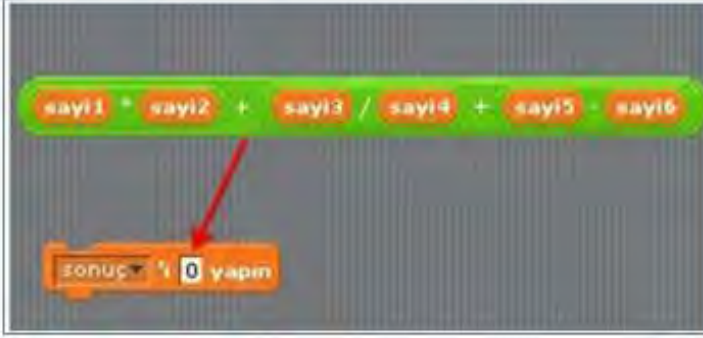
İşlemimizin son hali ve işlem sonucu: 16



- $(2 * 5) + (16 / 4) + (3 - 1)$ işlemimizin sonucunu ekranda yazdıralım.



İşlem sonucumuzu ekranda göstermek için önce **sonuç** diye bir değişken oluşturalım. **Sonuç değişkenine**, işlem sonucumuzu kaydedelim.



İşlemimizin bulunduğu program parçasını **0** yazılan yere mouse ile sürükleyelim.



İşlem sonucunu ekrana yazdırmak için

Operatorler kısmında :



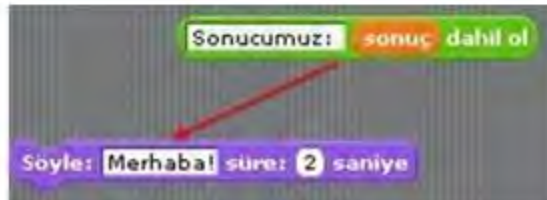
bu kod parçasını **Yazılar** kısmına sürüklüyoruz.

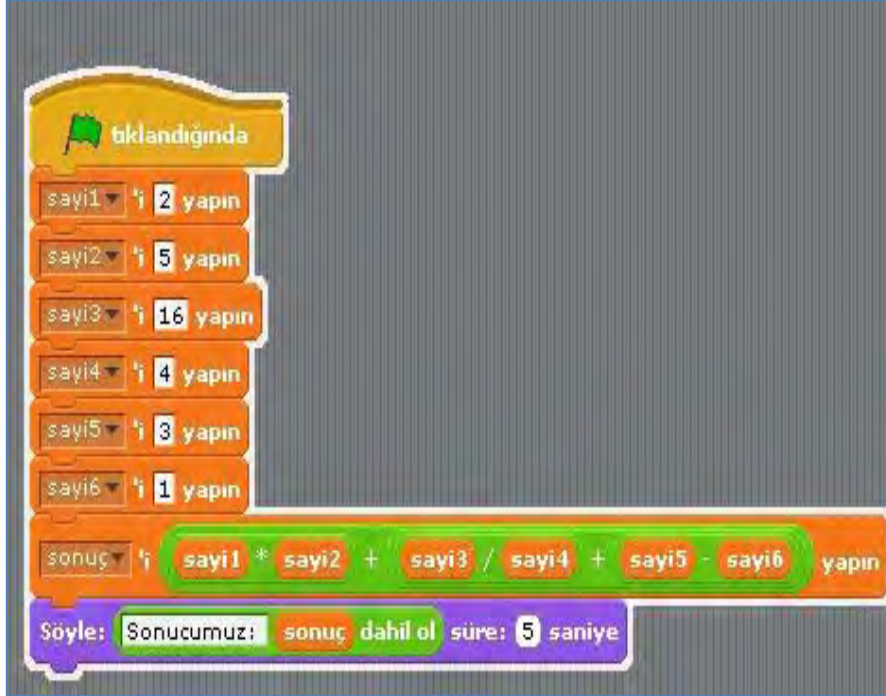
merhaba_ yazan yere: **Sonucumuz:** yazalım.

dünya yazan yere : **Değişkenler** kısmında **sonuç** değişkenini koyalım.



Ekrana yazdırmak için bu kod parçasını Görünüm kısmındaki kod parçası ile birleştiriyoruz.



Programımız:**Ekran Görüntüsü:**

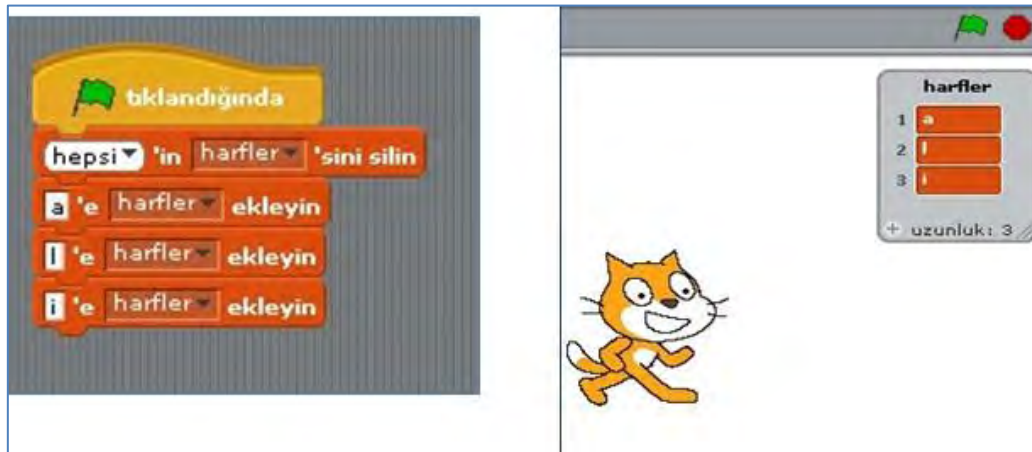
LİSTE OLUŞTURMA

- **Değişenler -> Listele** kısmında bir liste oluşturacağız. Liste dediğimiz, birçok karakteri bir arada tutan **karakter dizisidir**.
- **Listele**'ye tıklayıp: **harfler** isimli, bir liste oluşturalım.



LİSTE DOLDURMA

- Oluşturduğumuz harfler listesine harf ekleyelim. Değişkenler kısmına gelip bir kod parçası ile listemize harf ekleyebiliriz.
- **harfler** listemize **a,l,i** harflerini ekledik sırasıyla.



LİSTEDE HARF ARAMA

- Oluşturduğumuz listede herhangi bir harfi arayan program yazalım. Bir listede harf aramak için şunu kullanırız:
- **Harfler** listemizde **'i'** harfinin olup olmadığını kontrol edelim.



doğru

Üzerine tıkladığımızda doğru veya yanlış yazar.

LİSTE UZUNLUĞU

- Bir listeye istediğimiz kadar harf, sayı... kaydedebiliriz. Listemizde kaç eleman bulunduğunu şu kod parçası ile bulabiliriz:



3

- Harf uzunluğunu ekrana yazdırmak için :



Harfler listesinin uzunluğu = 3

KONTROL

- Menü -> **Kontrol** kısmında koşullu işlemler için kodlar yer almaktadır.

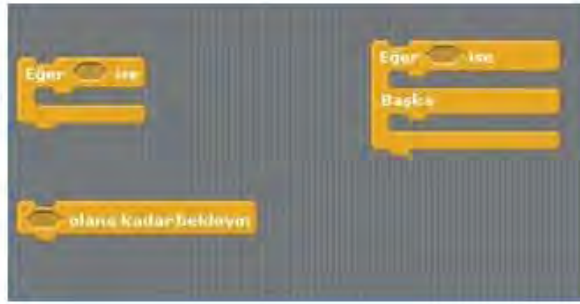
Programımızı başlatmak için koşul koyabiliriz.

Sağ üstte **yeşil bayrak'a** tıkladığı zaman programın başlaması için:









- **Eğer ise & Başka**

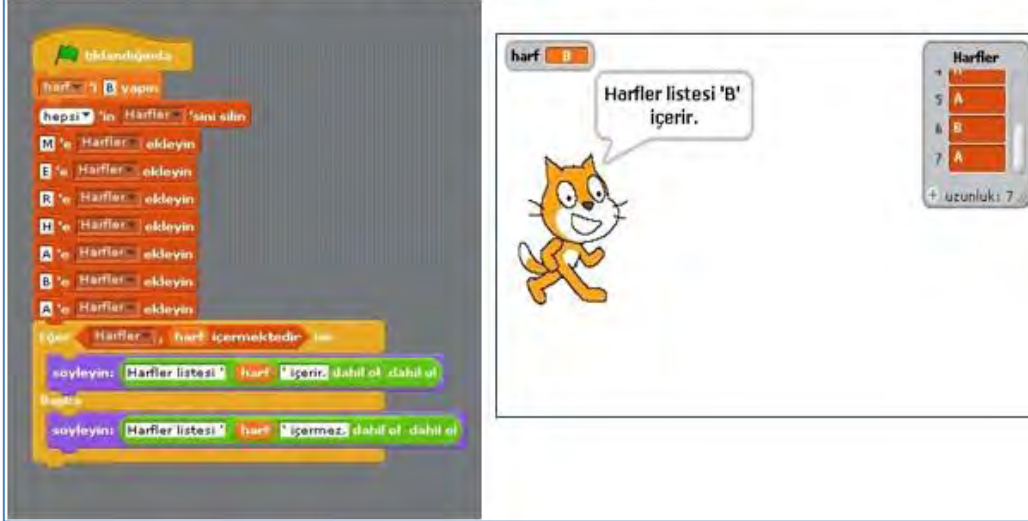
Sağdaki kod parçaları da koşullu işlem yapmak için kullandığımız kodlardır.



Koşullu işlemlerle örnekler

- Bir listede 'B' harfinin olup olmadığını kontrol edelim.



The image shows a Scratch script and its execution. The script starts with a 'when green flag clicked' block, followed by a 'say B yapın' block. Then, a 'say hepsi in Harfler sırası alın' block is used to create a list. The list is populated with 'M', 'E', 'R', 'H', 'A', 'B', 'A' characters. An 'if' block checks if the character 'B' is in the list. If true, it says 'Harfler listesi 'B' içerir.'; if false, it says 'Harfler listesi 'B' içermez.' The interface shows the list 'Harfler' with indices 1-7 and the character 'B' at index 6. The cat character says 'Harfler listesi 'B' içerir.'

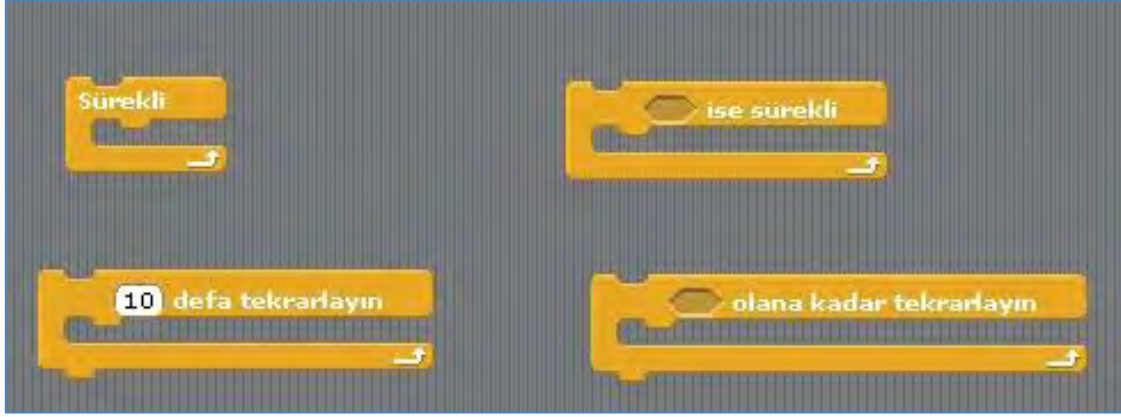
- Harfler listesinde 'K' harfinin olup olmadığını kontrol edelim.



The image shows a Scratch script and its execution. The script starts with a 'when green flag clicked' block, followed by a 'say K yapın' block. Then, a 'say hepsi in Harfler sırası alın' block is used to create a list. The list is populated with 'M', 'E', 'R', 'H', 'A', 'B', 'A' characters. An 'if' block checks if the character 'K' is in the list. If true, it says 'Harfler listesi 'K' içerir.'; if false, it says 'Harfler listesi 'K' içermez.' The interface shows the list 'Harfler' with indices 1-7 and the character 'A' at index 7. The cat character says 'Harfler listesi 'K' içermez.'

TEKRARLI İŞLEMLER

Aynı işlemi istediğimiz sayıda yaptırmak için şu kod parçalarını kullanırız



Ekrana sürekli Merhaba yazdırmak için :



“Sürekli” kontrol yapısını kullanırsak ekranda “Merhaba” yazısı hiç kaybolmaz.

Bir işlemi istediğimiz sayıda yaptırmak için:

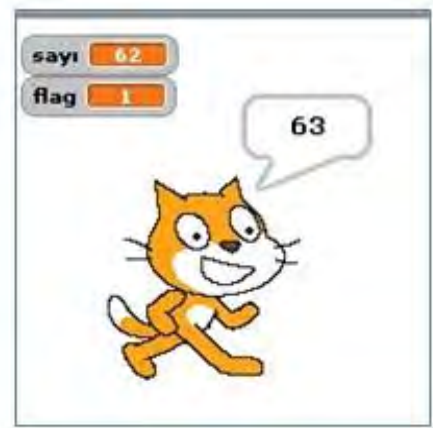


Bir değişken oluştururuz. Değişken ismi “sayı”dır. Sayı değişkeninin değerini önce 0 yaparız. Sonra 5 defa sayıyı 1 ile toplarsak sayı değerimiz 5 olur. Toplama işlemi bittikten sonra sayıyı ekrana yazdırırız.



Bir işlemi, koşula bağlı olarak tekrarlı yaptırmak için:

İki değişken oluşturalım. Değişken isimleri **flag** ve **sayı** olsun. **flag**'i 1 yapalım, **sayı**'yı 0 yapalım.



flag = 1 ise sürekli sayı değişkenine 1 ekleyip ekrana yazdıralım.

Ekrana baktığımızda sayı değeri sürekli değişmektedir. Kodu yazıp deneyelim. **Yeşil Bayrak**'a tıklayıp ekrana bakalım.☺

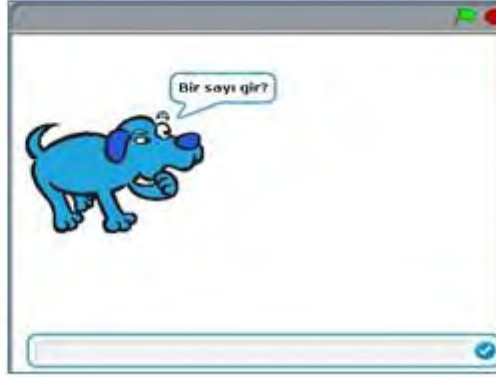


sayı isimli bir değişken oluşturalım. **sayı**'ya ilk olarak 0 atayalım.

sayı'yı 15 olana kadar 1 ile toplayalım ve **sayı**'yı ekrana yazdıralım

Verilen bir sayının karesini bulan program yazalım.

Önce kullanıcıdan bir sayı girmesini isteyelim.

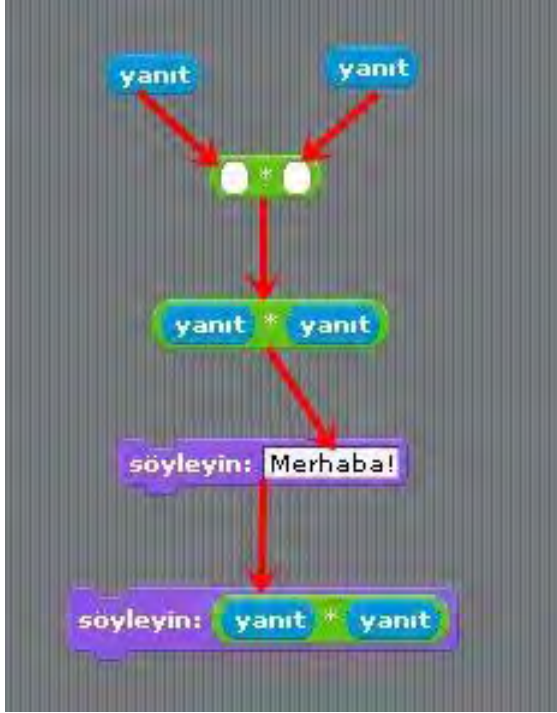


Kullanıcının girdiği sayı **yanıt** ' a kaydolur. Ekranda sol tarafta yanıt'ın değerini görebiliriz. **yanıt = 4**



Menüden **Algılama** kısmında **yanıt** değişkeni bulunur. **Yanıt**'ın yanındaki boşluğa ✓ **tik** koyarsak ekranda görebiliriz.





4' ün karesini hesaplamak için; **yanıt** değerini kendisiyle çarparız.

Ve sonucu ekrana yazdırırız.

Kullanıcı 4 girdi ve 4'ün karesini hesaplayıp ekrana yazdırmış olduk.



Faktöriyel hesaplayan program yazalım.

Örnek

$$4! = 24 \quad 5! = 120 \quad 1! = 1 \quad 0! = 1$$

Kullanıcının verdiği sayının faktöriyelini hesaplayalım. Önce kullanıcıdan bir sayı girmesini isteyelim.

İki tane değişken oluşturalım.

i ve *result* değişkenleri. Değerleri 1 olsun.



Bir sayının faktöriyelini, 1' den başlayarak kullanıcının girdiği sayı'ya kadar, tüm sayıları çarparak buluruz.

Örneğin, kullanıcı 5 girdiyse, 1' den itibaren 5' e kadar

1,2,3,4,5 sayılarını çarparak buluruz.



Programımız yanda:

- *i* ve *result* değişkenleri oluşturup değerlerini 1 yapmıştık.
- Tekrarlı işlem yapacağımız için, döngü kullandık.
- *i* değerini döngü içinde 1'er artırıyoruz, her seferinde.
- *result* ile *i* değerini her seferinde çarparak faktöriyeli hesaplıyoruz.
- Döngü bittiğinde faktöriyel sonucunu ekrana yazdırıyoruz.

```

result = 1 yapın
i = 1 yapın
i > yanıt olana kadar tekrarlayın
  result = result * i yapın
  i = i + 1 kadar değiştirin
    
```

- **yanıt = 5**
 - **result = 1**
 - **i = 1**
-
- **Döngü kontrolü**
 - **i > yanıt** , **1 > 5** doğru olmadığı için döngüye giriyoruz. (1)
 - **result = result * i = 1 * 1 = 1**
 - **i = i + 1 = 1 + 1 = 2**
-
- **i > yanıt** , **2 > 5** doğru olmadığı için döngüye giriyoruz. (2)
 - **result = result * i = 1 * 2 = 2**
 - **i = i + 1 = 2 + 1 = 3**

- **i > yanıt** , **3 > 5** doğru olmadığı için döngüye giriyoruz. (3)
 - **result = result * i = 2 * 3 = 6**
 - **i = i + 1 = 3 + 1 = 4**
-
- **i > yanıt** , **4 > 5** doğru olmadığı için döngüye giriyoruz. (4)
 - **result = result * i = 6 * 4 = 24**
 - **i = i + 1 = 4 + 1 = 5**
-
- **i > yanıt** , **5 > 5** doğru olmadığı için döngüye giriyoruz. (5)
 - **result = result * i = 24 * 5 = 120**
 - **i = i + 1 = 5 + 1 = 6**

```

bklandığında
  Bir sayı girer misin? sor ve bekle
  result = 1 yapın
  i = 1 yapın
  i > yanıt olana kadar tekrarlayın
    result = result * i yapın
    i = i + 1 kadar değiştirin
  Söyle: yanıt i dahil ol result dahil ol süre: 5 saniye
    
```

- **i = 6**
- **i > yanıt** , **6 > 5** doğru olduğu için döngüye **girmiyoruz.**
- **result = 120** en son değerimiz.
- Sonucu ekrana yazdıralım.

```

Söyle: yanıt i dahil ol result dahil ol süre: 5 saniye
    
```


Faktöriyel Programımız :

```

    tıkladığında
    result = 1 yapın
    i = 1 yapın
    Soyle: "Seninle faktöriyel hesaplayalım!" süre: 2 saniye
    Bir sayı girer misin? sor ve bekle
    i > yanıt olana kadar tekrarla
    result = result * i yapın
    i = i + 1 kadar değiştirin
    Soyle: "yanıt " + result + " dahil ol" süre: 5 saniye
    
```

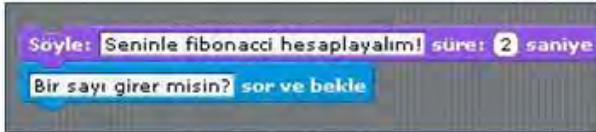




Fibonacci Sayılar

- **Fibonacci sayılarını** bulalım.
- Kendisinden önce gelen iki sayının toplamı olan sayılara **fibonacci sayıları** diyoruz.
- 1 1 2 3 5 8 13 21 34 55 ...
- Elimizde iki değişken olmalı. Fibonacci sayıların hesaplanmasında ilk başta iki tane 1'e ihtiyaç var.
- $1 + 1 = 2$
- $1 + 2 = 3$ $2 + 3 = 5$
- $3 + 5 = 8$ $5 + 8 = 13$
- $8 + 13 = 21$ $13 + 21 = 34$

- Fibonacci Sayıların hesaplanmasında kullanılacak değişkenleri tanımlayalım.
- İlk önce iki tane 1'e ihtiyacımız olduğu için **öncekiSayı** ve **yeniSayı** değişkenlerini **1** yapıyoruz.
- **i** ve **temp** değişkenleri de oluşturalım.



- Kullanıcıya mesaj verelim ve bir sayı isteyelim.
- Kullanıcının istediği fibonacci sayısını söyleyelim.
- **1 1 2 3 5 8 13 21 34**
- Kullanıcı 6 girerse, cevabımız 6.fibonacci sayısı olmalı, yani **8** olmalı.
- Kullanıcı 10 girerse, cevabımız **21** olmalı.
- Döngü oluştururuz. Her defasında:

- $temp = yeniSayı + öncekiSayı$
- $öncekiSayı = yeniSayı$
- $yeniSayı = temp$

- işlemlerini yaparız.



Döngü koşulumuz, kullanıcın istediği fibonacci sayısına ulaşılan kadar döngü tekrarlanır.

$i > \text{yanıt} - 2$

2 çıkarma nedenimiz; Fibonacci sayıların başındaki iki tane 1'i, yani öncekiSayı ve yeniSayı değişkenlerimize kaydetmiş olmamız. Döngü içinde en son satırda i 'yi 1 artırırız.

```

i > yanıt - 2 olana kadar tekrarlayın
temp 'i yeniSayı + öncekiSayı yapın
öncekiSayı 'i yeniSayı yapın
yeniSayı 'i temp yapın
i 1 kadar değiştirin
    
```

```

bklandığında
öncekiSayı 'i 1 yapın
yeniSayı 'i 1 yapın
temp 'i 0 yapın
i 'i 1 yapın

i > yanıt - 2 olana kadar tekrarlayın
temp 'i yeniSayı + öncekiSayı yapın
öncekiSayı 'i yeniSayı yapın
yeniSayı 'i temp yapın
i 1 kadar değiştirin
    
```

öncekiSayı = 1
yeniSayı = 1
temp = 0
i = 1
yanıt = 4 ise (kullanıcı 4 girdiyse)

$i > \text{yanıt} - 2 \rightarrow 1 > 4 - 2 \rightarrow 1 > 2$ doğru olmadığı için, döngüye girilir
temp = $1 + 1 = 2$
öncekiSayı = 1
yeniSayı = 2
 $i = i + 1 = 1 + 1 = 2$

$i > \text{yanıt} - 2 \rightarrow 2 > 4 - 2 \rightarrow 2 > 2$ doğru olmadığı için, döngüye girilir
temp = $2 + 1 = 3$
öncekiSayı = 2
yeniSayı = 3
 $i = i + 1 = 2 + 1 = 3$

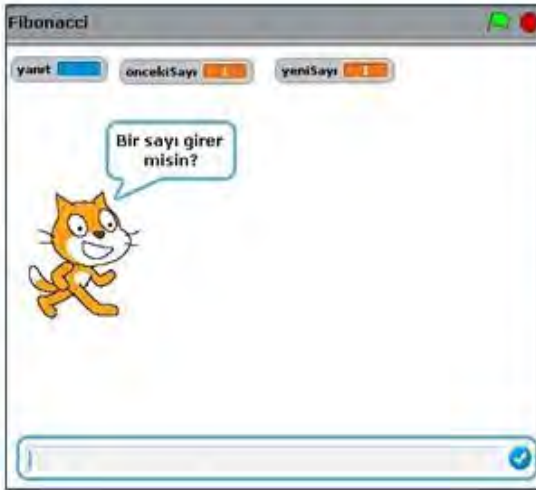
$i > \text{yanıt} - 2 \rightarrow 3 > 4 - 2 \rightarrow 3 > 2$ doğru olduğu için, **döngüye girilmez.**

yeniSayı = 3'tür. Fibonacci sayılarından 4. sayıyı bulmuş olduk. (1 1 2 3 5 8 13...)

Fibonacci Programımız:

```

when green flag clicked
  set previous number to 1
  set new number to 1
  set temp to 0
  set i to 1
  say "Seninle fibonacci hesaplayalım!" for 2 seconds
  ask "Bir sayı girer misin?" and wait
  loop until i > answer - 2
    set temp to new number + previous number
    set previous number to new number
    set new number to temp
    set i to i + 1
  say "yanıt : Fibonacci = dahil ol yeniSayı dahil ol" for 5 seconds
  stop program
  
```





TRIANGULAR SAYILAR

$$1 + 2 = 3$$

$$1 + 2 + 3 = 6$$

$$1 + 2 + 3 + 4 = 10$$

$$1 + 2 + 3 + 4 + 5 = 15$$

$$1 + 2 + 3 + 4 + 5 + 6 = 21$$

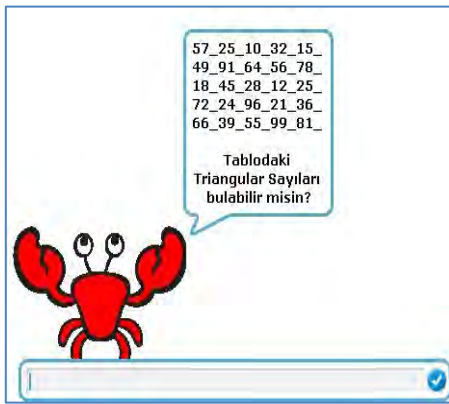
3 , 6 , 10 ,15, 21 ...

gibi sayılara Triangular Sayılar diyoruz.

1'den başlayarak ardışık sayıların

toplamının sonucunda Triangular Sayıları

elde ediyoruz.



Bir Scratch Programı yapalım.

Bize birçok sayıdan oluşan

bir tablo verilsin.

Tablodaki Triangular sayıları

bulalım. Doğru bulursak

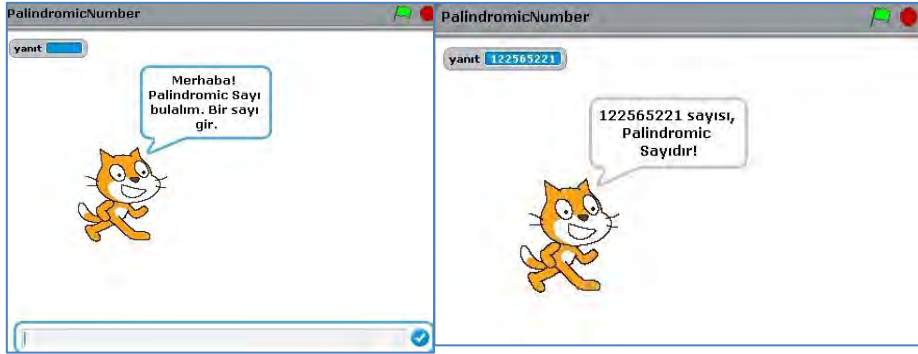
“Tebrikler!Doğru bildin!” mesajı versin, yanlış

Bulursak “Üzgünüm!Yanlış bildin!” mesajı versin.

Palindromic Sayılar

121 12421 56865
50605 1478741 1234321

Soldan ve sağdan bakıldığında aynı olan sayılara Palindromic sayılar deriz.



Kullanıcı 122565221 sayısını girince program o sayıyı kontrol ediyor.

Palindromic Sayı olup olmadığını söylüyor.

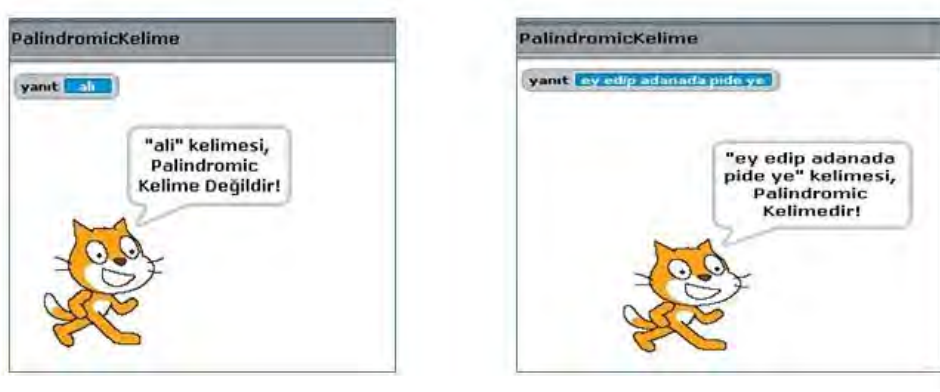
Palindromic Kelimeler

ey edip adanada pide ye
küllük kılık neden
kabak milim niçin

Soldan ve sağdan bakıldığında aynı olan kelimelerdir.

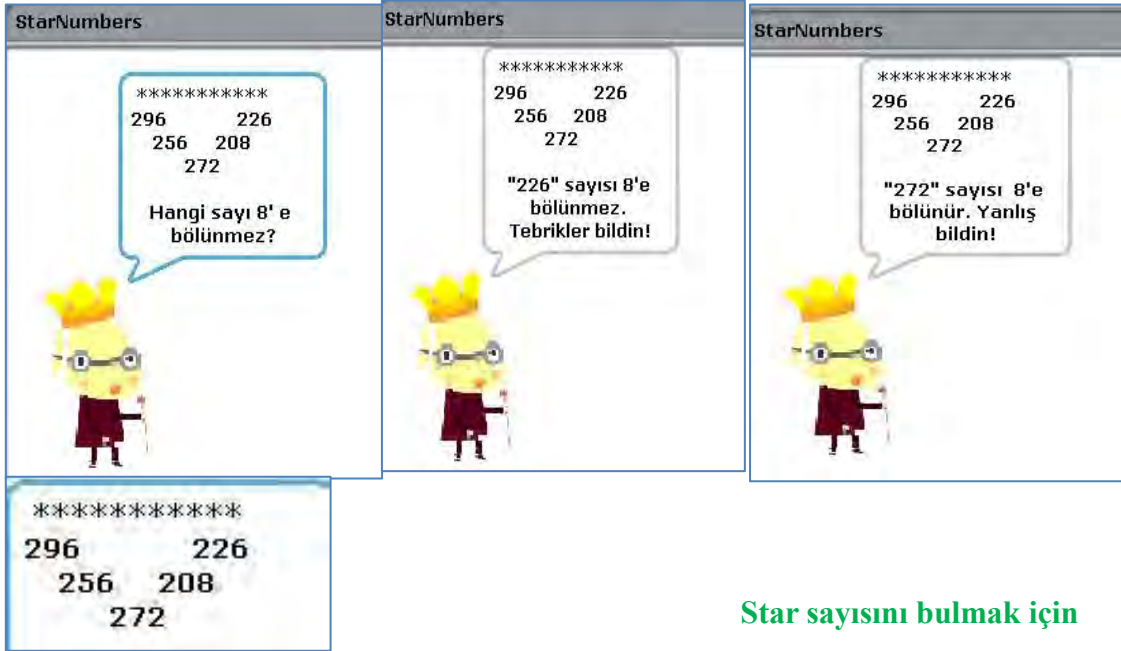
Palindromic Kelimleri bulmak için bir Scratch Programı yazalım.

Kullanıcı bir kelime girsin. Program da girilen kelimenin Palindromic kelime olup olmadığını kontrol etsin.



Star Sayılar

Bir program yazalım. Verilen beş sayıdan Hangi sayının 8'e bölünmediğini bulalım.



Star sayısını bulmak için yazdığımız programı inceleyelim.

Tablodaki sayıları tutmak için bir liste oluşturulmuş.

Ve listeye sayıları ekleyelim. Ekran sayıları düzenli yerleştirmek için boşlukları da listeye ekliyoruz.

Sayılar isimli bir liste oluşturuldu ve tablodaki sayılar eklendi.



Star sayısını bulmak için yazdığımız programı inceleyelim.

Tablodaki sayıları tutmak için bir liste oluşturalım. Ve listeye sayıları ekleyelim. Ekranı sayıları düzenli yerleştirmek için boşlukları da listeye ekliyoruz.

- **Sayılar** isimli bir liste oluşturuldu ve tablodaki sayılar eklendi.
- **Sayı** değişkeni oluşturduk ve değerini **8** yaptık.

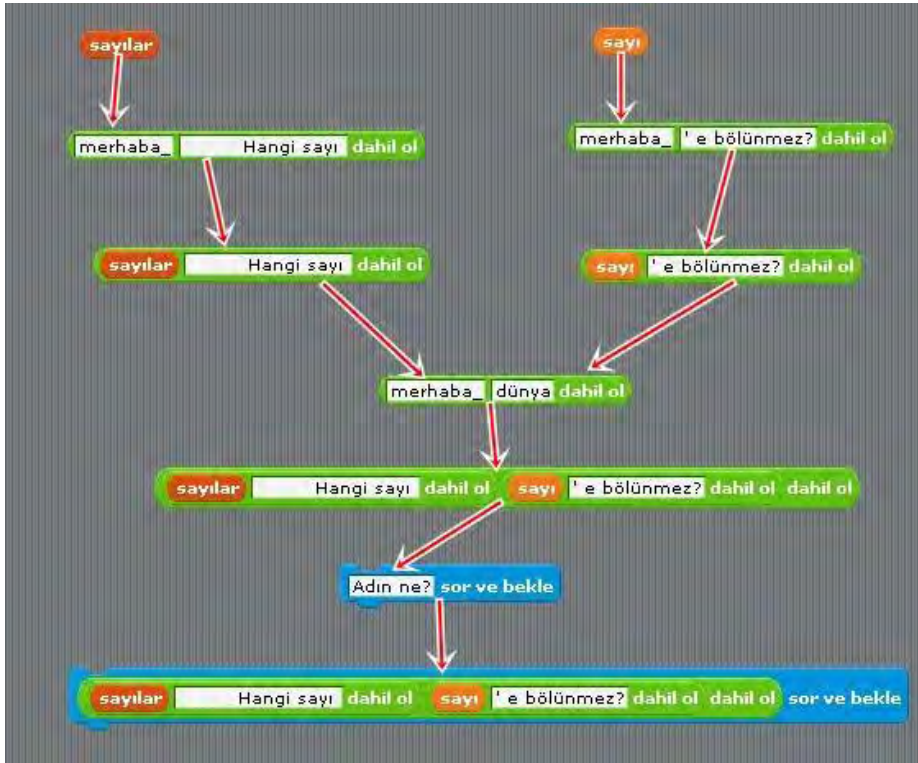


dogruCevap isimli bir değişken daha oluşturduk.

Ve değerini 226 yaptık.
226 8'e bölünmez.



- Sayılar listemizi ekrana yazdırmayı bu şekilde yapıyoruz.



Programımız :

```

    tıkladığında
    hepsi'in sayılar' sini silin
    sayı' i 8 yapın
    *****' e sayılar ekleyin
    296' e sayılar ekleyin
    ' e sayılar ekleyin
    226' e sayılar ekleyin
    ' e sayılar ekleyin
    256' e sayılar ekleyin
    ' e sayılar ekleyin
    208' e sayılar ekleyin
    ' e sayılar ekleyin
    272' e sayılar ekleyin
    ' e sayılar ekleyin
    doğruCevap' i 226 yapın
    sayılar Hangi sayı dahil ol sayı 'e bölünmez? dahil ol dahil ol sor ve bekle
    Eğer yanıt = doğruCevap ise
    Soyle: sayılar " yanıt " sayısı 8'e bölünmez. Tebrikler bildin! dahil ol dahil ol dahil ol süre: 5 san
    Başka
    Soyle: sayılar " yanıt " sayısı 8'e bölünür. Yanlış bildin! dahil ol dahil ol dahil ol süre: 5 saniye
    
```



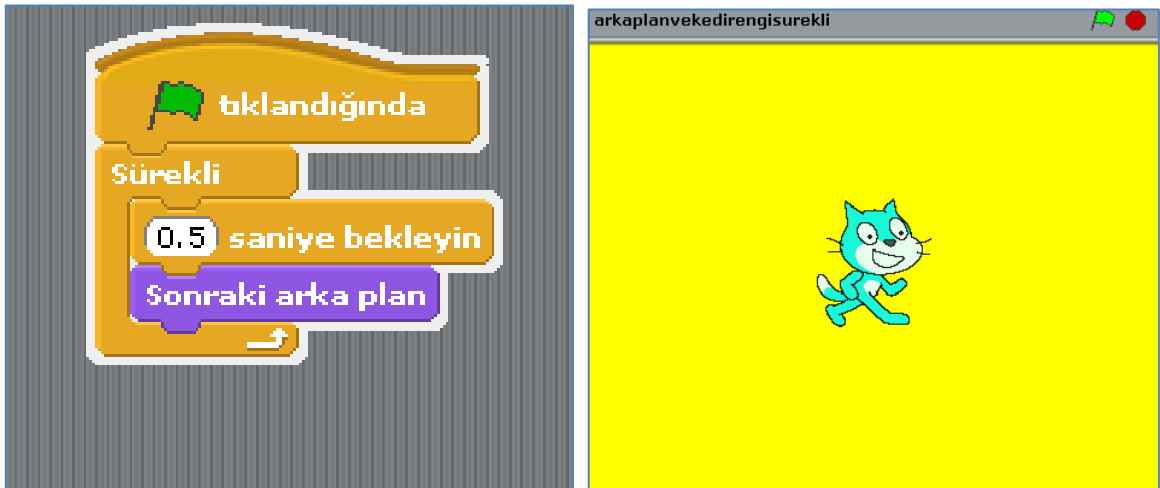
Renk Değiştiren Kedi

Bir Scratch uygulaması yapalım.

Yeşil bayrağa tıklandığında sürekli rengi değişen bir kedi ve sürekli rengi değişen bir arka planımız olsun. Bunun için kolay anlaşılır bir algoritma yaparız. Tıkladığımızda sürekli 0.5 saniye aralıklarla rengini değiştiren bir karakter yapmamız aşağıdaki gibi mümkündür.



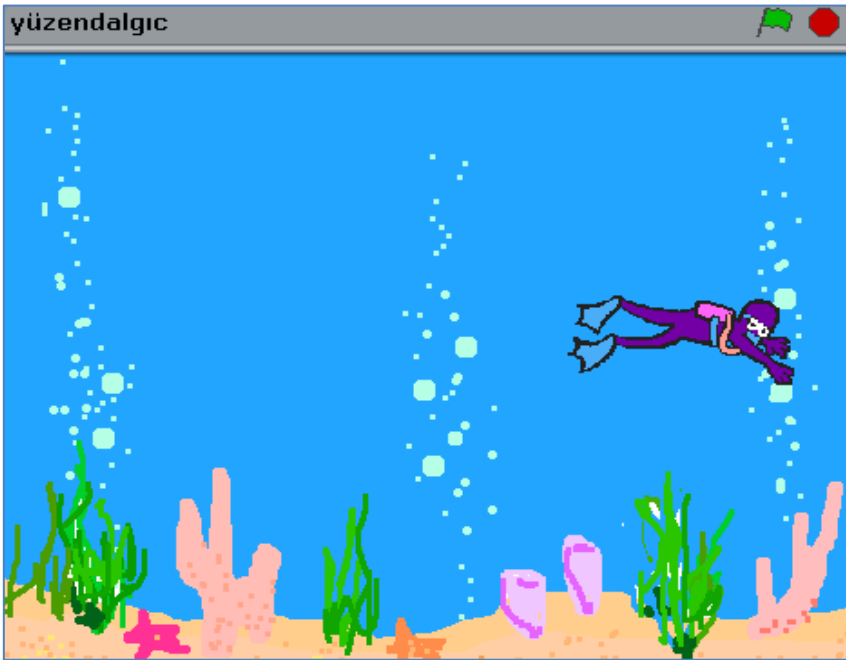
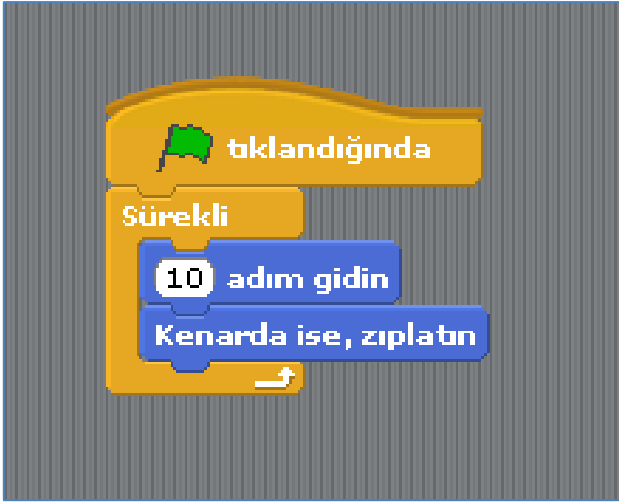
Arka plan için kendi oluşturduğumuz renklerdeki sahneleri kullanarak bu sahnelerin sürekli değişmesini aşağıdaki gibi sağlarız.



Yüzen Adam

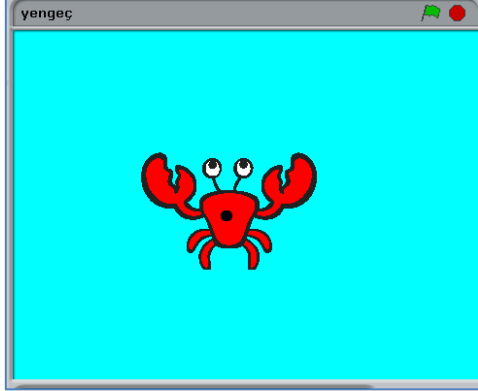
Karakter olarak bir dalgıç seçtik. Sualtında yüzen bir dalgıç için acaba nasıl bir algoritma tasarlamalıyız ?

Şekildeki gibi bir kod bunun için yeterli olacaktır. Sahnemizi ise kütüphanemizde hazır bulunan sualtı sahnesini kullanarak yaptık.





Dans Eden Yengeç



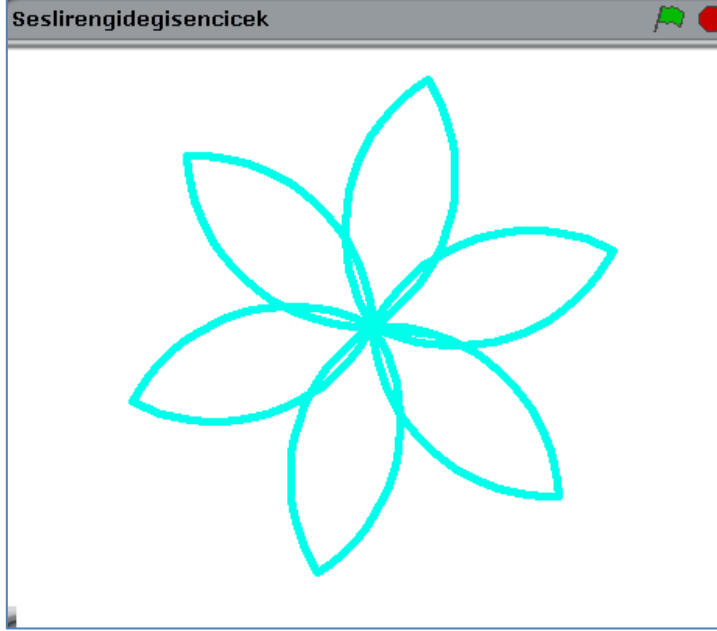
Karakter olarak seçtiğimiz yengecin dans etmesi için yeşil bayrağa tıkladığımızda yengeç verdiğimiz komutlarla hareket etmeye başlıyor.

Ek olarak programımıza ses özelliği kattık.



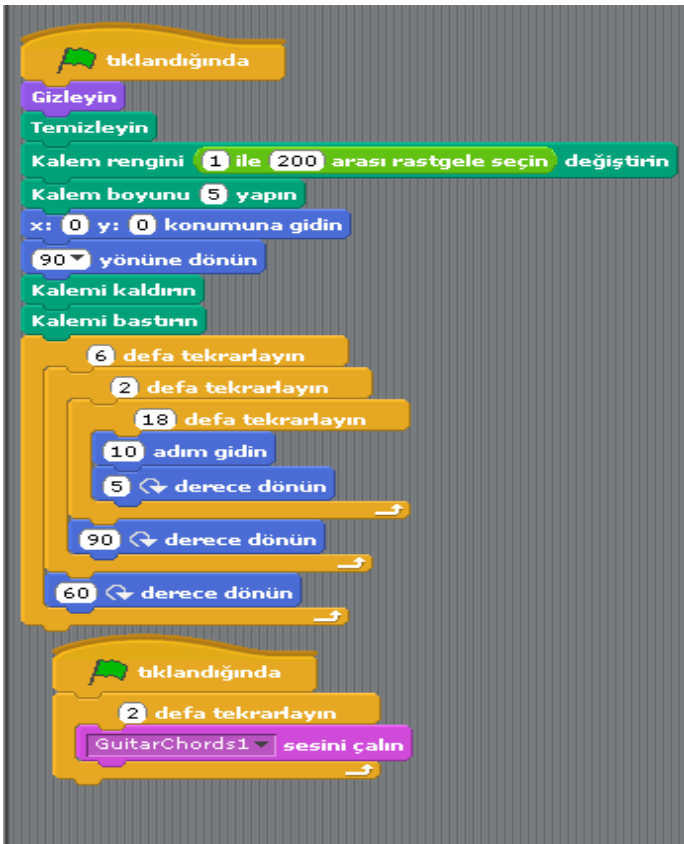


Çiçek Çizimi

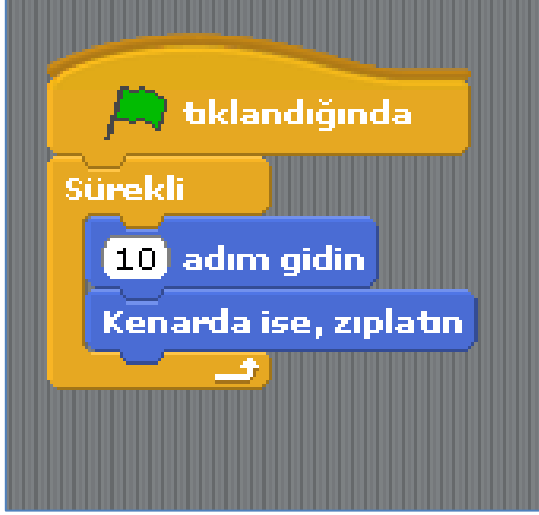


Sürekli rengi değişen bir çiçek çizmek ister misiniz?

Kalem rengini ve açığı sürekli değiştirerek bu mümkündür.



Köpekbalığı ve kaçan balıklar



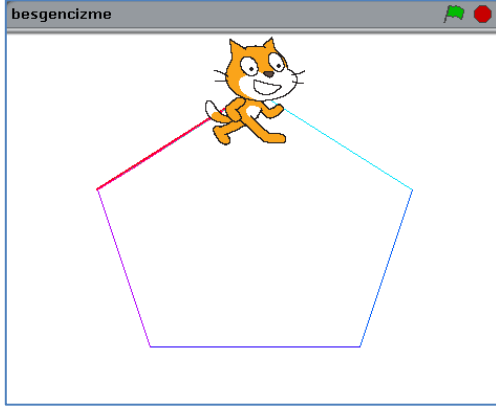
Bir denizde yüzen köpek balığı ve ondan kaçan altı küçük balığın macerasına hazır mısınız?

Karakterlerimizin hepsi için alttaki kodu kullanırız. Yeşil bayrağa tıkladığında kaçış başlamış olur.



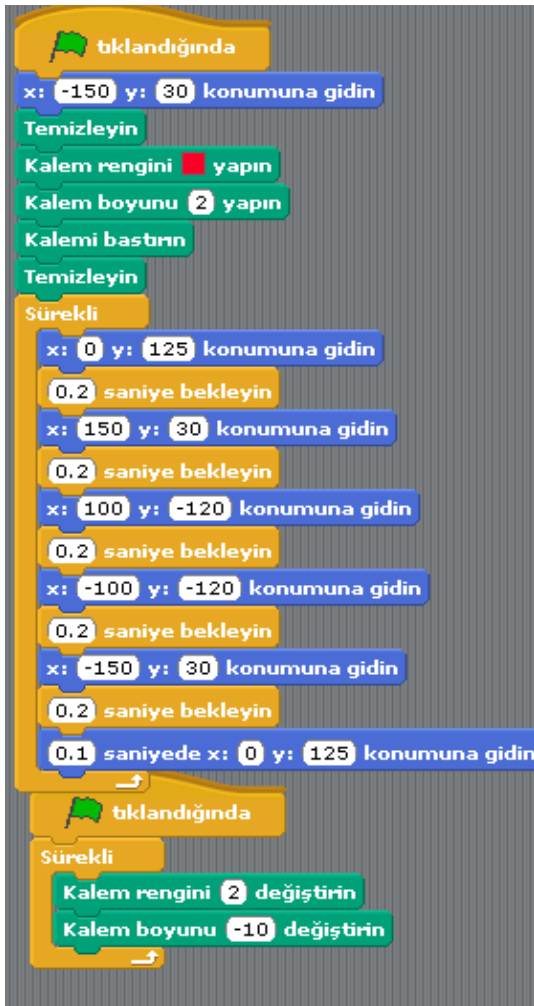


Beşgen Çizme



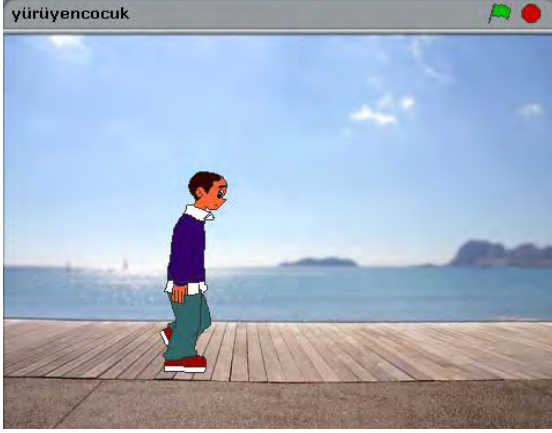
Her kenarı farklı renkte olan bir beşgen çizmek istiyorsak ne yapmalıyız ?

Bunu bir inceleyelim.





Yürüyen Çocuk



Bu programımızda karakterimize adım attırmak ve ona yürüyüş yaptırmak istiyoruz.

Karakter 0.2 saniye aralıklarla bir adım atıp kostüm değiştirsin .

Bunu nasıl yapmalıyız?



Dans Eden Üç Arkadaş



Bir sahnede dans eden üç arkadaş adlı programımızda kütüphanemizden 3 karakter seçeriz ve onların belirli komutlarla dans etmelerini sağlarız.



Müziğimizi eklemeyi unutmayalım.

2013

