

6. SINIF BİLİŞİM TEKNOLOJİLERİ VE YAZILIM DERSİ

2.DÖNEM 1.SINAV DERS NOTLARI

VERİ TİPLERİ



KARAKTER VERİ TİPİ

Tüm tek haneli sayıları, harfleri ve özel karakterleri kapsar. Tırnak içinde belirtilir. Bilgisayar sözcüğündeki "B" harfi bir karakteri ifade eder.

MANTIKSAL VERİ TİPİ

Evet ya da hayır şeklindeki karar verme süreçlerinde kullanılır. Örneğin; arabası var mı, lise mezunu mu?

SAYISAL VERİ TİPİ

Hesaplama işlemlerinde kullanılır. Tüm sayı çeşitlerini içerir. Pozitif ya da negatif tam sayılar kullanılabilir. Örneğin; açılar, uzaklık, nüfus, ücret, yarıçap...

ÖZEL VERİ TİPİ

Tarih, saat, adres, banka hesap numarası gibi verileri temsil eder.

KARAKTER DİZİSİ VERİ TİPİ

Birden fazla karakterin bir araya gelmesiyle karakter dizisi oluşur. Örneğin "Bilgisayar" sözcüğü bir diziyi ifade eder.

HAYDİ, VERİ TOPLAMAYA!

İlk bilgisayar olarak abaküs kabul edilir. Çünkü abaküs sayma yapmamıza yardımcı olur.

Günümüzdeki bilgisayarlar da temelde sayılarla çalışır. Bu sayıların temelinde 1 ve 0'lar vardır.

Bu 1 ve 0'ları kullanarak diğer veri tiplerini bilgisayara tanıtırız.

Bilgisayarda kullanılan veri türleri şunlardır:

1- **Sayısal Veri Tipi:** Hesaplama işlemlerinde kullanılır. Tüm sayı çeşitlerini içerir. Pozitif ya da negatif tam sayılar kullanılabilir. **Örneğin;** açı=45°, uzaklık=50 km, nüfus=100.000, ücret=2,5 tl vb.....

2- **Karakter Veri Tipi:** Tüm tek haneli sayıları, harfleri ve özel karakterleri kapsar. Tırnak içinde belirtilir. Bilgisayar sözcüğündeki **"B"** harfi bir karakteri ifade eder.

3- **Karakter Dizisi Veri Tipi:** Birden fazla karakter bir araya geldiğinde bu veri tipi ortaya çıkar. Örneğin **"Bilgisayar"** sözcüğündeki karakterlerin tümü bu veri tipindedir.

4- **Mantıksal Veri Tipi:** Yalnızca iki kelime barındırır; doğru ve yanlış. Evet ya da hayır şeklindeki karar verme süreçlerinde kullanılır. Örneğin; Arabası var mı? (**EVET/HAYIR**)

5- **Özel Veri Tipi:** Tarih, saat, adres, banka hesap numarası gibi verileri temsil eder.

SABİT Mİ DEĞİŞKEN Mİ?

Sabit: İlk biçimiyle kalan, değişmeyen ifade ya da nesnelere.

Örnek: suyun kaynama sıcaklığı 100 derecedir.

Değişken: İlk biçimiyle kalmayıp yeni değerler ya da biçimler alabilen ifade ya da nesnelere.

Örnek: Şu an saat 12:45 tir.

Scratch programında değişken tanımlamak için Veri menüsünü kullanırız.

MANTIKSAL OPERATÖRLER

İşlem	Komut	Matematiksel Sembol	Anlamı
VE	AND	.	Koşulların hepsi doğru ise sonuç doğrudur
VEYA	OR	+	Koşullardan en az biri doğru ise sonuç doğrudur
DEĞİL	NOT	'	Sonuç koşulun tersidir . 1 ise 0 dır

MATEMATİKSEL OPERATÖRLER

İşlem	Matematik	Bilgisayar
Toplama	$a + b$	$a + b$
Çıkarma	$a - b$	$a - b$
Çarpma	$a . b$	$a * b$
Bölme	$a \div b$	a / b
Üs alma	a^b	$a ^ b$

ALGORİTMA

Algoritma sözcüğü nereden gelir?

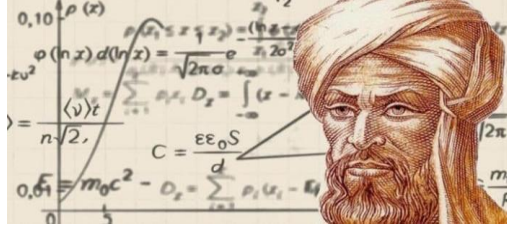
Algoritma sözcüğü, Özbekistan'ın Harezmi bugünkü Türkmenistan'ın Khiva kentinde doğmuş olan Ebu Abdullah Muhammed İbn Musa elHarezmi'den kaynaklanır. Bu alim 9.yüzyılda cebir alanındaki algoritmik çalışmalarını kitaba dökerek matematiğe çok büyük bir katkı sağlamıştır.

Algoritma Nedir?

Algoritma, bir problemi çözmek için kullanıma yol ya da yöntemlere verilen isimdir. Algoritmalar hazırlanırken mutlaka başla komutu ile başlanır ve bitir komutu ile sonlanır. Başla ve bitir olmayan komutlar teoride yok sayılır. Algoritmada problemi çözerken; çözüm adımları açık ve net bir şekilde belirtilmesi gerekir.

Algoritma Ne İşe Yarar?

Severek oynadığımız bilgisayar oyunları, severek girdiğimiz sosyal medya platformları olsun bunların hepsinde bir algoritma vardır. Ve kullanılan programlar bir algoritma çerçevesi içerisinde çalışırlar.



Algoritma ve Kodlama İlişkisi

Şu an için bilgisayarda aklınıza gelen tüm yazılımlar belirli bir algoritmaya göre hazırlanıp, ondan sonra kodlanmaktadır. Yani ilk olarak mutlaka algoritma yazılır ve daha sonra kod yazmaya geçilir. Algoritma olmadan maalesef kodlama yapmak neredeyse imkansızdır. Algoritmada aslında temel mantık hiç bir iş şansa bırakılmadan hareket edilmeli ve tüm ihtimaller denenmelidir. Diyelim ki bir oyun yapmaya karar verdiniz. Bu oyunun algoritması içerisinde kullanıcının yani oyunu oynayan kişilerin hangi tuşlara basınca ne olacağını net olarak belirtmeniz gerekiyor eğer siz net olmazsanız algoritmanız hatalı olur, algoritma hatalı olursa yazmış olduğumuz kodlarda sorun olur kodlarda da sorun olursa zaten yazdığımız hazırladığımız oyun da hatalar ile karşılaşırız.

Akış Şeması Nedir?

Akış şeması, algoritması hazırlanan bir programın(problemin) çeşitli şekillerle birbirine bağlanarak ortaya çıkan, neden sonuç ilişkisini gösteren şema tipidir. Akış şeması da aynı algoritma gibi adımlarla ifade edilir ve başla komutu ile başlanır ve bitir komutu ile sonlandırılır. Akış şeması çizerken dikkat edilmesi gereken en önemli nokta mutlaka algoritmanıza göre akış şeması çizenizdir. Kısaca akış şemaları bizim yazmış olduğumuz algoritmaların şekillerle daha anlaşılır bir şekilde gösterilmesi denebilir

Akış Şemalarında Kullanılan Şekiller (Semboller)

Akış şeması sembolleri



ELİPS

Akışı başlatır ve bitirir.



DİKDÖRTGEN

Eylemi/işlemi belirtir.



PARALELKENAR

Dışarıdan bilgi/veri girişini belirtir.



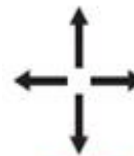
EŞKENAR DÖRTGEN

Karar verme merkezidir.



DALGALI DÖRTGEN

Ekran / yazıcı çıktısı.



OKLAR

Akış yönünü gösterir ve sembolleri birbirine bağlarlar.



ALTİGEN

Tekrar eden komutları belirtir.

ÖRNEK

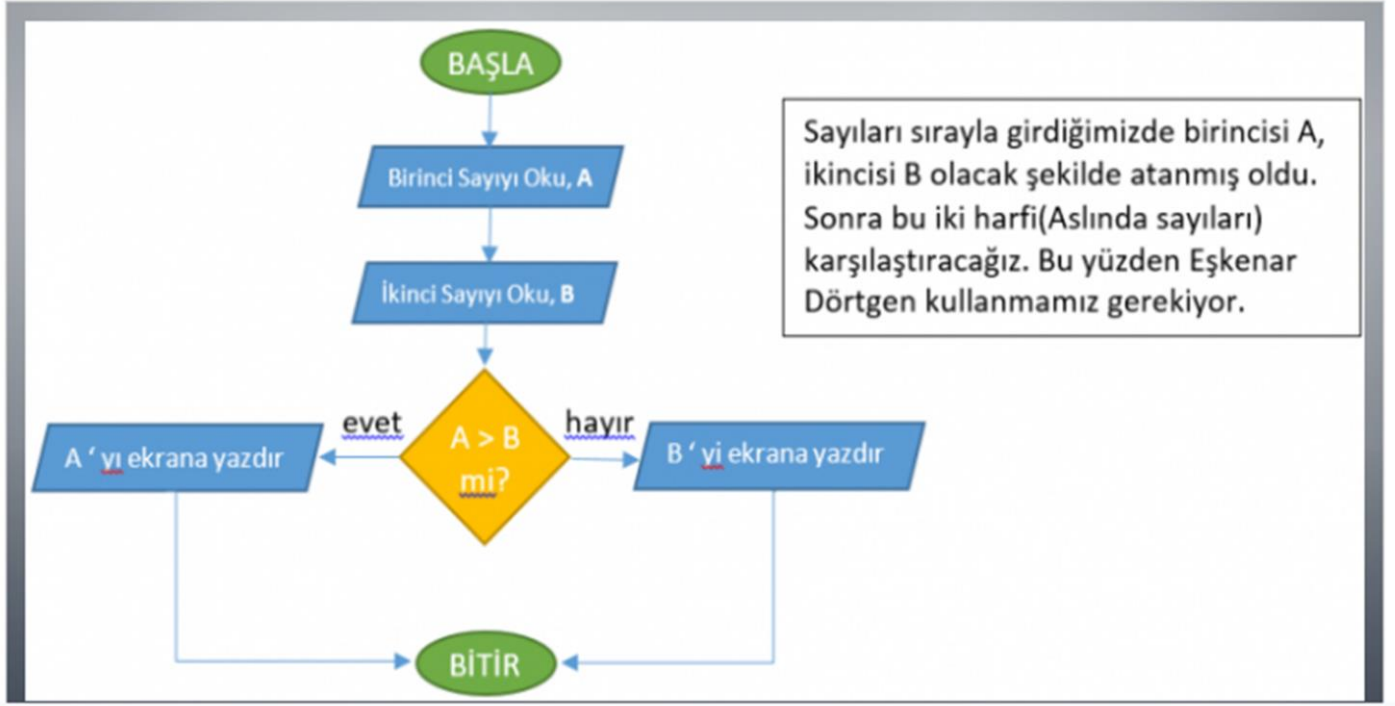
Klavyeden girilen iki sayıdan büyük olanını ekrana yazan program.

ADIM 1: BAŞLA

ADIM 2: A ve B sayısını oku

ADIM 3: Eğer $A > B$ ise A'yı ekrana yaz değilse B'yi ekrana yaz

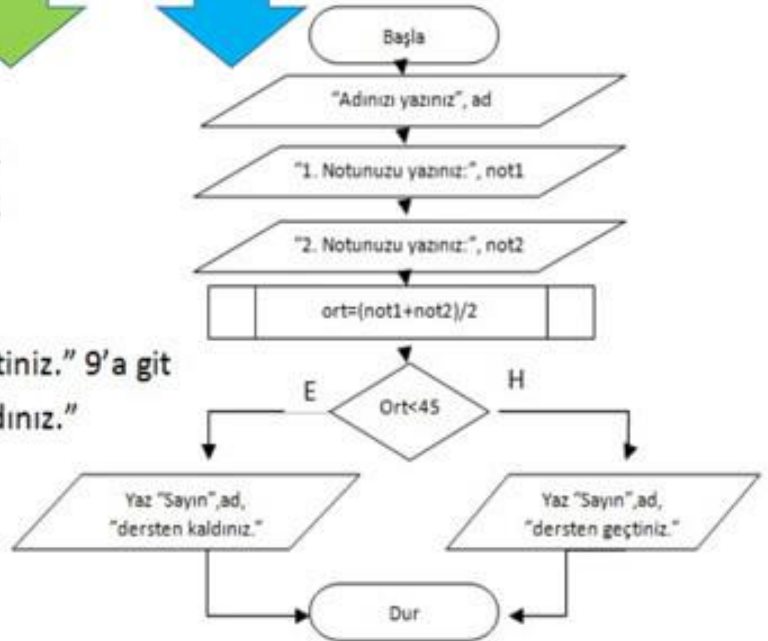
ADIM 4: BİTİR



ALGORİTMA

1. Başla
2. "Adınızı yazınız:",ad
3. "1. Notunuzu yazınız:", not1
4. "2. Notunuzu yazınız:", not2
5. $ort=(not1+not2)/2$
6. Eğer $ort < 45$ ise 8'e git
7. Yaz "Sayın",ad,"dersten geçtiniz." 9'a git
8. Yaz "Sayın",ad,"dersten kaldınız."
9. Dur

AKIŞ ŞEMASI (AKIŞ DİYAGRAMI)



DÖNGÜLER



Döngü kullanarak bütün nektarı topla.

bloklar

ilerle

sola dön U

sağa dön U

nektarı al

bal yap

Tekrarla ??? kez tekrarla

Çalıştığı zaman

Tekrarla 5 kez tekrarla

yap

ilerle

nektarı al

KOŞULLANDIRICI



Mor çiçeğin ne kadar nektar içerdiğini bilmiyoruz 1, 2
Çözümü bütün nektarları toplayacak şekilde tamamla

bloklar

ilerle

sağa dön U

sola dön U

nektarı al

bal yap

Çalıştığı zaman

ilerle

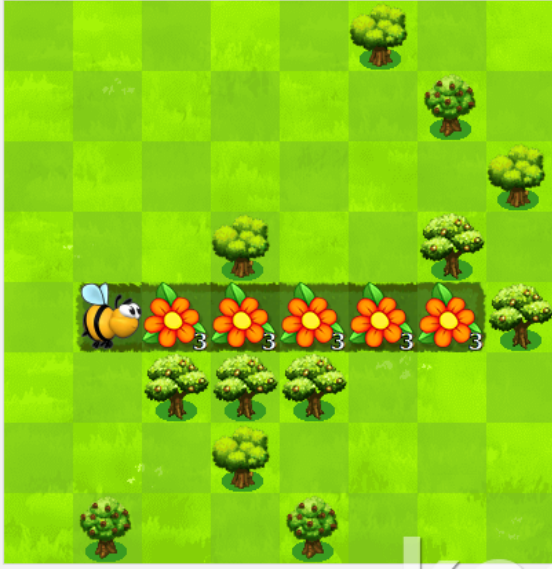
Tekrarla 3 kez tekrarla

yap

eğer nektar > 0

yap nektarı al

İÇ İÇE DÖNGÜLER



Her çiçekteki bütün nektarı topl. İç içe döngü kullan.

bloklar

ilerle

sola dön 5

sağa dön 5

nektarı al

bal yap

Tekrarla ??? kez tekrarla

Çalıştığı zaman

Tekrarla 5 kez tekrarla

yap

ilerle

Tekrarla 3 kez tekrarla

yap

nektarı al



Tüm çiçeklerdeki nektarları topl ve peteklere bal ya

bloklar

ilerle

sola dön 5

sağa dön 5

nektarı al

bal yap

Tekrarla ??? kez tekrarla

Çalıştığı zaman

Tekrarla 5 kez tekrarla

yap

ilerle

Tekrarla 4 kez tekrarla

yap

nektarı al

ilerle

Tekrarla 4 kez tekrarla

yap

bal yap

İÇ İÇE DÖNGÜLER



Bu döngüyü başka bir döngü içine koyarak iç içe döngü olu

bloklar

ilerle

sola dön ↶

sağa dön ↷

nektarı al

bal yap

Tekrarla ??? kez tekrarla

Çalıştığı zaman

Tekrarla 3 kez tekrarla

yap

ilerle

ilerle

sürece nektar > 0

yap nektarı al

İÇ İÇE DÖNGÜLER



"Tekrarla" döngüsü içine "sürece yap" döngüsü yerleştirerek

bloklar

ilerle

sola dön ↶

sağa dön ↷

nektarı al

bal yap

Tekrarla ??? kez tekrarla

Çalıştığı zaman

Tekrarla 3 kez tekrarla

yap

ilerle

sola dön ↶

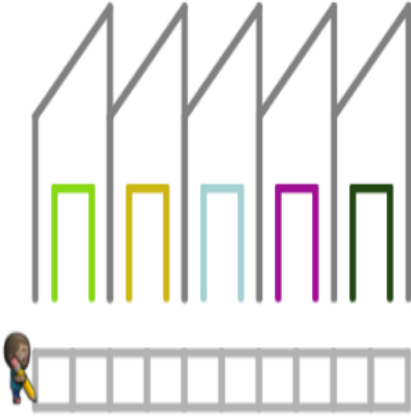
ilerle

sürece nektar > 0

yap nektarı al

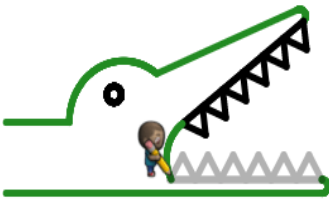
sağa dön ↷

İÇ İÇE DÖNGÜLER



```
bloklar
ileriye taşı 100 piksel
kadar sağa dön 90 derece
kadar sola dön 90 derece
ileri atla 100 piksel
Tekrarla ??? kez tekrarla
yap
Çalıştığı zaman
Tekrarla 10 kez tekrarla
yap
  Tekrarla 4 kez tekrarla
  yap
    ileriye taşı 30 piksel
    kadar sola dön 90 derece
  ileri atla 30 piksel
```

İÇ İÇE DÖNGÜLER



```
bloklar
ileriye taşı 100 piksel
kadar sağa dön 90 derece
kadar sola dön 90 derece
ileri atla 100 piksel
Tekrarla ??? kez tekrarla
yap
Çalıştığı zaman
Tekrarla 6 kez tekrarla
yap
  Tekrarla 3 kez tekrarla
  yap
    ileriye taşı 20 piksel
    kadar sola dön 120 derece
  ileriye taşı 20 piksel
```


FONKSİYONLAR

Fonksiyonlar, belirli işlemleri yürüten ve sonuçları döndüren bir işlem kümesidir.

Sayfalarca kod yazacağınıza fonksiyonları kullanarak program satırlarınızı çok daha az satırlı hale getirebilirsiniz.

```
Çalıştığı zaman  
bu işlemleri 2 kez tekrarla  
yap  
ilerle  
nektarı al  
nektarı al
```



```
Çalıştığı zaman  
bu işlemleri 2 kez tekrarla  
yap  
ilerle  
2 nektarı al
```

```
Fonksiyon  
2 nektarı al  
nektarı al  
nektarı al
```

FONKSİYONLAR



```
Çalıştığı zaman  
5 al
```

```
Fonksiyon  
5 al  
bu işlemleri 5 kez tekrarla  
yap nektarı al
```

FONKSİYONLAR



Çalıştığı zaman
ilerle ▾
7 al
sağa dön ↻ ▾
ilerle ▾
ilerle ▾
7 al

Fonksiyon
7 al
bu işlemleri 7 kez tekrarla
yap nektarı al

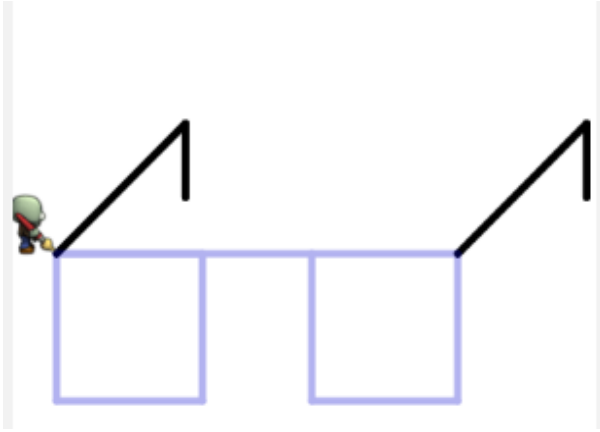
FONKSİYONLAR



Çalıştığı zaman
hareket et ve 3 al
bu işlemleri 3 kez tekrarla
yap ilerle ▾
hareket et ve 3 al
bu işlemleri 2 kez tekrarla
yap ilerle ▾
hareket et ve 3 al

Fonksiyon
hareket et ve 3 al
sağa dön ↻ ▾
ilerle ▾
bu işlemleri 3 kez tekrarla
yap bal yap
geriye git ▾
sola dön ↻ ▾

FONKSİYONLAR



Çalıştığı zaman

bir kare çiz

ileriye ▼ taşı 175 pikseller

bir kare çiz

Fonksiyon

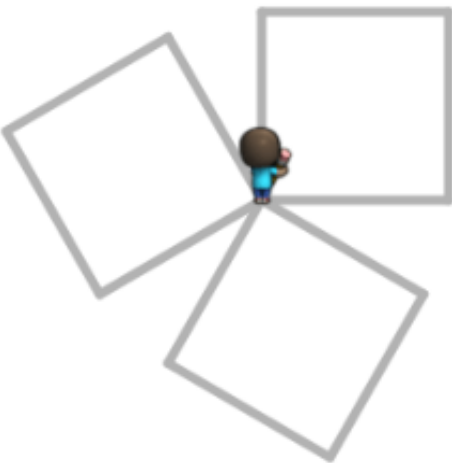
bir kare çiz

bu işlemleri 4 kez tekrarla

yap ileriye ▼ taşı 100 pikseller

kadar sağa ▼ dön 90 derece

FONKSİYONLAR



Çalıştığı zaman

bu işlemleri 3 kez tekrarla

yap bir kare çiz

kadar sağa ▼ dön 120 derece

Fonksiyon

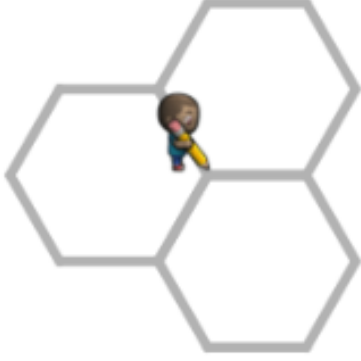
bir kare çiz

bu işlemleri 4 kez tekrarla

yap ileriye ▼ taşı 100 pikseller

kadar sağa ▼ dön 90 derece

FONKSİYONLAR



Çalıştığı zaman

bu işlemleri 3 kez tekrarla

yap altıgen çiz

kadar sağa dön 120 derece

Fonksiyon

altıgen çiz

bu işlemleri 6 kez tekrarla

yap ileriye taşı 50 pikseller

kadar sağa dön 60 derece

FONKSİYON SORULARI

SORU:1



Çalıştığı zaman

ilerle

7 nektar al 7 bal yap

ilerle

ilerle

7 nektar al 7 bal yap

Fonksiyon

7 nektar al 7 bal yap

FONKSİYON SORULARI

SORU:2



Çalıştığı zaman

ilerle ▼

7 al 7 yap

bu işlemleri 3 kez tekrarla
yap

ilerle ▼

7 al 7 yap

Fonksiyon

7 al 7 yap

FONKSİYON SORULARI

SORU:3



Çalıştığı zaman

Fonksiyon

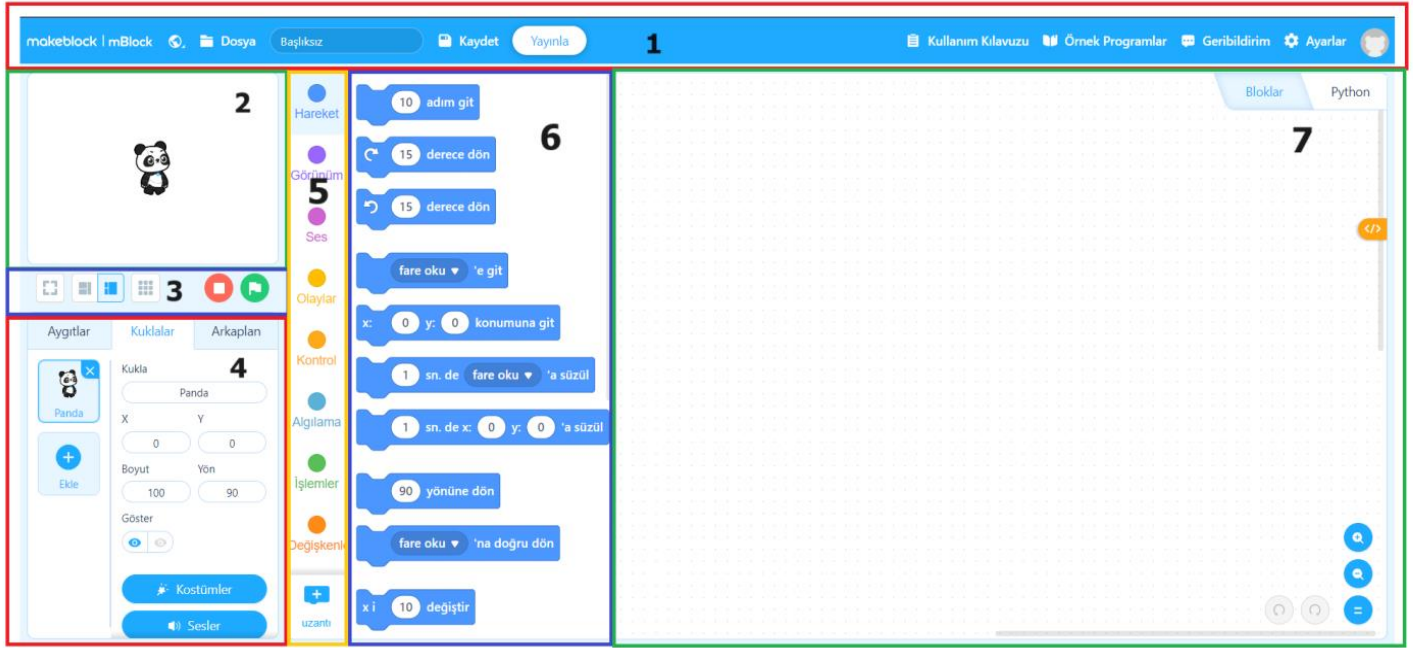
5 al

bu işlemleri 5 kez tekrarla
yap

nektarı al

mBlock

mBlock, MIT(Massachusetts Institute of Technology) tarafından geliştirilen ve çocuklar için kodlama eğitimini blok tabanlı programlama mantığıyla kazandırmaya çalışan SCRATCH adı verilen grafiksel programlama dilinin üzerine inşa edilmiş, donanım ve yazılım entegrasyonu ile kodlama öğrenmeyi eğlenceli hale getiren bir yazılımdır.



Bölüm 1 - Menü Bar

Programın en üst kısmında yer alan menülerle başlayalım.



Bu simge ile programın dil seçeneği ayarlanabilmektedir.

Dosya

Bu menü içinde Yeni Proje, Aç, Farklı Kaydet, Bilgisayarınızdan Açın, Bilgisayarınıza Kaydet, Paylaş gibi alt menüler yer alır.

Başlıksız

Başlıksız yazan kısma çalışmanızla ilgili bir isim vererek **Kaydet** seçeneği ile projenizi yeni bir isimle kaydetmenizi sağlar.

Kaydet

Yayınla

Projenize verdiğiniz yeni ismi kaydetmenizi sağlar. Yayınla, oluşturduğunuz

çalışmayı mBlock'un projeler sayfasında paylaşır. Bu işlemi gerçekleştirebilmek için oturum açmak gereklidir.

Kullanım Kılavuzu

mBlock ile ilgili ayrıntılı bilgi için ulaşabileceğimiz sayfaya ulaşmamızı sağlar.

Örnek Programlar

mBlock ile yapılmış örnek projelere ulaşmayı sağlar.

Geribildirim

mBlock'ta yaşadığınız herhangi bir sıkıntıyı iletebileceğiniz bölümdür.

Ayarlar

mBlock'ta değiştirebileceğiniz bazı ayarlar bulunur. Karanlık mod, sahneyi gizle, mBlock hakkında, vb.



mBlock hesabına giriş yaptıktan sonra **profiliniz** ile ilgili düzenlemeler yapabilmeyi, projelerinize ulaşabilmeyi sağlar.

Bölüm 2 - Sahne






Bu alan sahne alanıdır. Kullanıcılar, sahnede farklı karakterler, nesnelere ve arka planlar ekleyerek kendi dijital dünyalarını yaratabilirler. Bu sayede programlama becerilerini eğlenceli bir şekilde geliştirirken hayal güçlerini de kullanabilirler.



Sahnedeki kukla simgesinin bulunduğu bölümdür. Kuklayı mBlock kütüphanesinde bulunan birçok karakterlerden istediğimizi seçerek değiştirebiliriz ya da kendimiz sahneye kukla tasarlayabiliriz. Kukla kostüm ve ses işlemlerini de bu bölümde ayarlayabiliriz. Bu bölümde ayrıca programa bağlanacak cihaz seçimini de yapabiliriz.

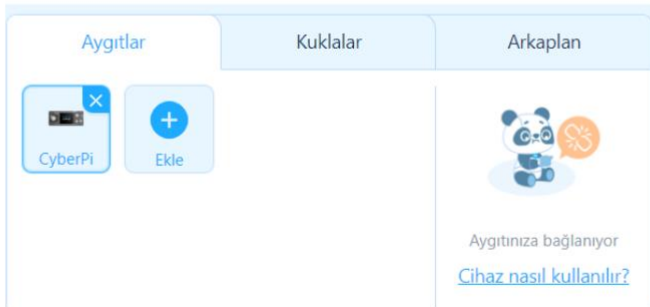
Bölüm 3 - Sahne Ayarları

Bu bölümde sahne düzeni ile ilgili değişiklikler yaparız.

	Sahneyi tam ekran yapmak için kullanılır.
	Sahneyi büyültüp küçültmek için kullanılır.
	Sahne ekranına ızgara eklemek için kullanılır.
	Animasyonu durdurmak için kullanılır.
	Animasyonu başlatmak için kullanılır.

Bölüm 4

Bu bölümde mBlock'ta kullanılacak aygıtlar, kuklalar ve arka planlar ile ilgili eklemelerin, düzenlenmelerin yapıldığı bölümdür.



Aygıtlar: Arduino cihazımızı bilgisayara taktığımızda programın cihaza bağlantısını yapabilir, cihaza kod gönderebilir veya bağlantıyı kesebiliriz.

Kuklalar: Uygulama da kullanılacak kuklalar eklenir, ismi ve görünümü gibi özellikler değiştirilir.

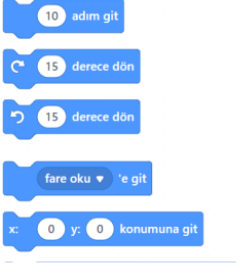
Arkaplan: Uygulama da kullanılacak arka planlar eklenir.



Bölüm 5 - Blok Grupları

Bu bölümde programlama için kullanacağımız bloklar sınıflandırılmıştır. Her bir sınıf içerisinde farklı bloklar yer almaktadır.

Bölüm 6 – Bloklar



Kod bloklarının bulunduğu kısımdır. Bu kod bloklarını fare ile sol tuşuna basılı tutarak sağ alandaki blok tasarım alanına sürükleyebiliriz. Silmek istediğimiz kod bloğunu ise tasarım alanından fare ile sol tuşuna basılı tutarak tekrar bu alana sürükleyerek silebiliriz.

Bölüm 7

Bu alan kod bloklarını tasarladığımız kısımdır.

MBLOCK ÜYE OLMAK VE OTURUM AÇMAK

mBlock programında üye olmak için Profil sekmesine veya Kaydet, Yayınla butonuna bastığınızda Oturum Aç veya Kayıt Ol ekranı gelecektir.

- Daha önceden hesabınız yok ise Kayıt Ol sekmesine basarak bir eposta adresi giriniz. Yaş
- seçeneğinizi işaretleyerek bir sonraki adıma geçiniz. 2. Adımda karşınıza sözleşme ekranı
- gelecektir. Kabul Ediyorum ve Devam butonuna basın. 3. Adımda eposta adresinize bir kod gelecektir. Gelen kodu "Doğrulama Kodu" satırına yazıp, Parola kısmına da en az 6 karakterden oluşan bir parola yazıp Kaydol butonuna basarak kayıt işlemi tamamlayabilirsiniz.
- Artık sağ üst köşede yer alan Profil simgesi değişmiş ve sisteme giriş yapılmış olarak gözükecektir.



SLIENT TEACHER

Aşağıda verilen kodları uygulandığında ekrana yazılacak sonucu bulunuz.

1

```
var a = 1  
a + 2
```

Sonuç:

2

```
var a = 4  
var b = 2  
a + b
```

Sonuç:

3

```
var a = 'vv'  
var b = 'jd'  
a + b
```

Sonuç:

4

```
var a = '0'  
var b = '1'  
a + b
```

Sonuç:

5

```
var a = 4  
a = 5  
a + 0
```

Sonuç:

6

```
var a = 4  
a = a + 3  
a + 3
```

Sonuç:

7

```
function hello (a, b) {  
  return a + b  
}  
  
hello(0, 2)
```

Sonuç:

8

```
function hello (a, b) {  
  return a * b  
}  
  
hello(1, 2)
```

Sonuç:

9

```
function hello (a) {  
  return a * a  
}  
  
hello(4)
```

Sonuç:

10

```
function hello () {  
  return 0  
}  
  
hello() + 2
```

Sonuç:

11

```
var a = 'xtzvzyt'  
a.length
```

Sonuç:

12

```
var a = [9, 9, 6, 5, 'n']  
a.length
```

Sonuç:

13

```
var a = ['i', 'y', 4]  
a[0]
```

Sonuç:

14

```
var a = ['d', 'x', 8, 4, 'r', 6, 3]  
a[5]
```

Sonuç:

15

```
var a = [7, 't', 't', 6, 9]  
a[3]
```

Sonuç:

16

```
var a = ['u', 'k', 'd', 'm', 'v']
a.indexOf('u')
```

Sonuç:

17

```
23 === 7
```

Sonuç:

18

```
14 === 14
```

Sonuç:

19

```
21 !== 21
```

Sonuç:

20

```
19 === '19'
```

Sonuç:

21

```
5 < 2
```

Sonuç:

22

```
var a = 25
a !== 25
```

Sonuç:

23

```
var a = 15
a === 24
```

Sonuç:

24

```
var a = 3
a === 9
a + 2
```

Sonuç:

25

```
var a = 1
if (2 < 3) {
  a = 3
}
a + 2
```

Sonuç:

26

```
var a = 1
if (4 < 1) {
  a = 4
}
a + 2
```

Sonuç:

27

```
if (1 < 2) {
  var a = 1
} else {
  var a = 3
}
a + 2
```

Sonuç:

28

```
if (5 < 2) {
  var a = 1
} else {
  var a = 3
}
a + 1
```

Sonuç:

29

```
if (3 === 9) {
  var a = 4
} else {
  var a = 2
}
a + 2
```

Sonuç:

30

```
if (2 < 5) {
  var a = 5
} else if (4 < 6) {
  var a = 2
} else {
  var a = 3
}
a + 3
```

Sonuç:

31

```
if (7 < 8) {
  var a = 4
} else if (8 < 7) {
  var a = 5
} else {
  var a = 6
}
a + 4
```

Sonuç:

32

```
if (6 < 10) {
  var a = 4
} else if (7 < 5) {
  var a = 6
} else {
  var a = 1
}
a + 1
```

Sonuç:

33

```
function hi (a, b) {
  return a * b
}
hi(0, 1)
```

Sonuç:

34

```
function hi (a, b) {
  if (a < b) {
    return a + b
  } else {
    return a * b
  }
}
```

```
hi(2, 2)
```

Sonuç:

35

```
function hi (a, b) {
  return a * b
}

function hello (a, b) {
  return a + b
}
```

```
hi(1, 2) + hello(1, 1)
```

Sonuç:

36

```
function hi (a, b) {
  return a * b
}

function hello (a, b) {
  return a + b
}

var a = hi(3, 2)
hello(a, 4)
```

Sonuç:

37

```
function hi (a, b) {
  return a * b
}

function hello (a, b) {
  return a + b
}
```

```
hello(hi(1, 2), 2)
```

Sonuç:

38

```
function hi (a, b) {
  return a * b
}

function hello (a, b) {
  return a + b
}
```

```
hello(hi(2, 2), 2)
```

Sonuç:

39

```
function hi (a, b) {
  return a * b
}

function hello (a, b) {
  return hi(a, b + 2)
}
```

```
hello(4, 3)
```

Sonuç:



Başarılar Dilerim



Nurdane Öner