

**T.C.  
MİLLÎ EĞİTİM BAKANLIĞI**

**BİLİŞİM TEKNOLOJİLERİ**

**GELİŞMİŞ ANİMASYONLAR  
482BK0145**

**Ankara, 2012**

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- PARA İLE SATILMAZ.

# İÇİNDEKİLER

AÇIKLAMALAR .....	iii
GİRİŞ .....	1
ÖĞRENME FAALİYETİ-1 .....	3
1. MATEMATİK İŞLEMLERİ .....	3
1.1. Matematik Sınıfları .....	3
1.2. Math Sınıfının Sabitleri .....	3
1.2.1. Euler Sabiti (E) .....	4
1.2.2. LN10 .....	4
1.2.3. LN2 .....	4
1.2.4. LOG10E .....	4
1.2.5. LOG2E .....	4
1.2.6. PI .....	5
1.2.7. SQRT1_2 .....	5
1.2.8. SQRT2 .....	5
1.3. Matematik Sınıfının Metotları .....	5
1.3.1. abs( ) .....	5
1.3.2. acos( ) .....	6
1.3.3. asin( ) .....	6
1.3.4. atan( ) .....	6
1.3.5. ceil( ) .....	6
1.3.6. cos( ) .....	6
1.3.7. exp( ) .....	6
1.3.8. floor( ) .....	7
1.3.9. log( ) .....	7
1.3.10. max( ) .....	7
1.3.11. min( ) .....	7
1.3.12. pow( ) .....	7
1.3.13. round( ) .....	7
1.3.14. sqrt( ) .....	8
1.3.15. random( ) .....	8
UYGULAMA FAALİYETİ .....	9
ÖLÇME VE DEĞERLENDİRME .....	11
ÖĞRENME FAALİYETİ-2 .....	12
2. METİNLER .....	12
2.1. Metin Alanları .....	12
2.2. Metin Alanlarını Oluşturma .....	13
2.3. Metin Alanlarını Biçimlendirme .....	14
2.4. HTML Metinleri Görüntülemek .....	15
2.5. Harici Metinleri Görüntülemek .....	17
UYGULAMA FAALİYETİ .....	19
ÖLÇME VE DEĞERLENDİRME .....	21
ÖĞRENME FAALİYETİ-3 .....	22
3. DİZİLER .....	22
3.1. Dizi .....	22
3.2. Dizi Tanımlama .....	22

3.3. Diziler Üzerinde İşlemler.....	23
UYGULAMA FAALİYETİ .....	27
ÖLÇME VE DEĞERLENDİRME .....	29
ÖĞRENME FAALİYETİ-2 .....	30
4. KODLAMA İLE GRAFİK OLUŞTURMAK.....	30
4.1. Düz Çizgiler Çizmek.....	30
4.2. Çizgilerle Şekiller Oluşturmak.....	31
4.3. Şeklin Başlangıç Noktasını Değiştirmek .....	32
4.4. Eğik Çizgiler Çizmek.....	33
4.6. Birden Fazla Şekil Oluşturmak .....	34
4.6. Hazır Geometrik Şekiller Oluşturmak .....	35
4.7. Filtreler.....	37
UYGULAMA FAALİYETİ .....	41
ÖLÇME VE DEĞERLENDİRME .....	43
ÖĞRENME FAALİYETİ-3 .....	44
5. ÇOKLU ORTAM İŞLEMLERİ .....	44
5.1. Harici Resimleri ve Animasyonları Yükleme .....	44
5.2. Harici Sesleri Yükleme .....	46
5.3. Ön Yükleme (Preloading) Oluşturmak .....	46
5.4. Sesleri Kontrol Etme.....	49
5.5. Video Dosyalarını Yükleme.....	51
5.6. Video Oynatımını Kontrol Etme.....	51
UYGULAMA FAALİYETİ .....	53
ÖLÇME VE DEĞERLENDİRME .....	55
ÖĞRENME FAALİYETİ-3 .....	56
6. GELİŞMİŞ ETKİLEŞİM .....	56
6.1. Sürükle –Bırak .....	56
6.2. Çarpışmalar .....	59
6.3. Nesneleri Sahne Üzerinde Rastgele Yerleştirmek .....	62
6.4. Nesneleri Sahne Üzerinde Düzenli Yerleştirme .....	63
UYGULAMA FAALİYETİ .....	65
ÖLÇME VE DEĞERLENDİRME .....	66
MODÜL DEĞERLENDİRME .....	67
CEVAP ANAHTARLARI .....	70
KAYNAKÇA .....	73

# AÇIKLAMALAR

<b>KOD</b>	<b>482BK0145</b>
<b>ALAN</b>	<b>Bilişim Teknolojileri</b>
<b>DAL/MESLEK</b>	<b>Web Programcılığı</b>
<b>MODÜLÜN ADI</b>	<b>Gelişmiş Animasyonlar</b>
<b>MODÜLÜN TANIMI</b>	Bu modül, animasyon yazılımı kullanarak gelişmiş animasyonlar hazırlamak için temel bilgi ve becerilerin kazandırıldığı bir öğrenme materyalidir.
<b>SÜRE</b>	<b>40/32</b>
<b>ÖN KOŞUL</b>	“Etkileşimli Animasyonlar ” modülünü tamamlamış olmak
<b>YETERLİK</b>	Eylem kodları ile gelişmiş animasyonlar hazırlamak
<b>MODÜLÜN AMACI</b>	<b>Genel Amaç</b> Öğrenci, bu modül ile gerekli ortam sağlandığında; eylem kodlarını kullanarak gelişmiş animasyonlar hazırlayabileceksiniz. <b>Amaçlar</b> <ol style="list-style-type: none"><li>1. Matematiksel işlemleri gerçekleştirebileceksiniz.</li><li>2. Metinler ile çalışabileceksiniz.</li><li>3. Dizilerle çalışabileceksiniz.</li><li>4. Animasyon yazılımı ile grafik işlemleri gerçekleştirebileceksiniz.</li><li>5. Çoklu ortam işlemlerini gerçekleştirebileceksiniz.</li><li>6. Etkileşimli gelişmiş uygulamalar hazırlayabileceksiniz.</li></ol>
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	<b>Ortam:</b> Bilişim Teknolojileri laboratuvarı, işletme ortamı <b>Donanım:</b> Bilgisayar
<b>ÖLÇME VE DEĞERLENDİRME</b>	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.



# GİRİŞ

## Sevgili Öğrenci,

Bu modül ile etkileşimli animasyonlar modülünde öğrendiğiniz temel programlama becerilerini animasyon yazılımına nasıl entegre edebileceğinizi daha net olarak göreceksiniz. Eylemler panelini ve script kodları kullanarak *İnternet* sayfaları, *İnternet* sayfaları için banner, logo tasarımı, masaüstü uygulamaları, oyunlar hazırlayabileceksiniz.

Etkileşimli animasyonlar modülünün devamı olarak hazırlanan bu modülde actionscript 3.0 dili ile uygulamalarınız içerisinde matematiksel işlemler yapabilecek, internet sayfalarınız için interaktif metin kutuları oluşturabilecek, dizilere bilgi gönderip diziler içerisinde kullandığınız nesnelere saklayabileceksiniz. Animasyon yazılımı ve fare kullanarak yaptığınız her çizimi script kodları ile oluşturabilecek oluşturduğunuz nesnelere script kodları yardımıyla hareket ettirebileceksiniz.

*İnternet* sayfalarınıza, uygulamalarınıza, oyunlarınıza ses ve video ekleyebileceksiniz. Hazırladığınız uygulamalarda nesnelere birbirleri ile etkileşimlerini kontrol edebilecek, bu etkileşime göre uygulamaya yön verebileceksiniz.





# ÖĞRENME FAALİYETİ-1

## AMAÇ

Bu faaliyet sonunda actionscript kodları ile matematiksel işlemleri gerçekleştirebileceksiniz.

## ARAŞTIRMA

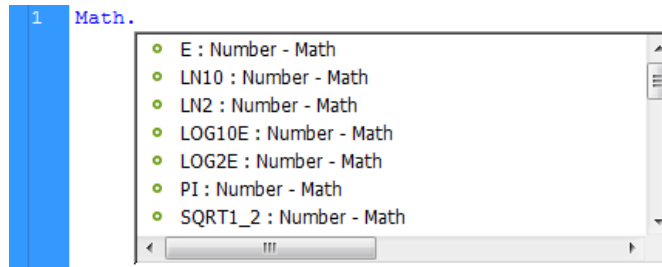
- Matematiksel sabitleri araştırınız.
- Trigonometrik fonksiyonları araştırınız.
- Radyan ve derece arasındaki ilişkiyi araştırınız.

## 1. MATEMATİK İŞLEMLERİ

### 1.1. Matematik Sınıfları

Animasyon yazılımının belli matematiksel, trigonometrik işlemleri yapabilmesi için kullanılan fonksiyonları ve matematiksel sabitleri içerisinde barındıran sınıftır. Matematik sınıfları ile mutlak değer, logaritma, yuvarlama, rastgele sayı üretme vb. işlemler rahat bir şekilde yapılabilmektedir. Ayrıca matematikte kullanılan pi sayısı, Eular sabiti vb. sabitler uygulama içerisinde rahatlıkla kullanılabilir.

Math sınıfı metotlarına eylemler panelinden erişmek için sınıf ismini yazıp nokta koymak yeterlidir. Math sınıfı animasyon yazılımı içerisinde sabit olarak bulunan bir sınıf olduğu için direkt ismi yazılarak ulaşılabilir. Sınıfın sahneye import edilmesine gerek yoktur. Math sınıfı içerisindeki kullanılacak metot ve özellikler karşımıza çıkacaktır.



Şekil 1.1: Math sınıfı

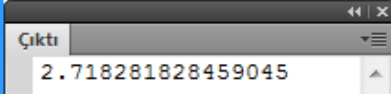
### 1.2. Math Sınıfının Sabitleri

Animasyon yazılımı içerisinde kullanabileceğimiz 8 adet math sınıfı sabiti bulunmaktadır.

### 1.2.1. Euler Sabiti (E)

Kullanıldığında e sayısı değeri olan 2.718281828459045 sayısını verecektir.

```
1 trace(Math.E);
```

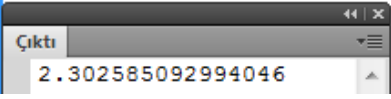


The screenshot shows a command prompt window with a blue background. The command '1 trace(Math.E);' is entered in the prompt. The output is displayed in a separate window titled 'Çıktı' (Output) with the value '2.718281828459045'.

### 1.2.2. LN10

Kullanıldığı yerde Logaritma 10 değeri olan 2.302585092994046 sayısını verir.

```
1 trace(Math.LN10);
```

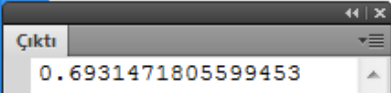


The screenshot shows a command prompt window with a blue background. The command '1 trace(Math.LN10);' is entered in the prompt. The output is displayed in a separate window titled 'Çıktı' (Output) with the value '2.302585092994046'.

### 1.2.3. LN2

Kullanıldığı yerde Logaritma 2 değeri olan 0.6931471805599453 sayısını verir.

```
1 trace(Math.LN2);
```

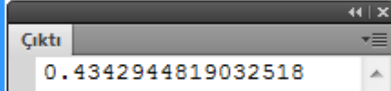


The screenshot shows a command prompt window with a blue background. The command '1 trace(Math.LN2);' is entered in the prompt. The output is displayed in a separate window titled 'Çıktı' (Output) with the value '0.6931471805599453'.

### 1.2.4. LOG10E

E sayısının 10 tabanına göre logaritma değeri olan 0.4342944819032518 sayısını verir.

```
1 trace(Math.LOG10E);
```

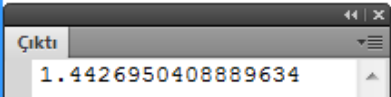


The screenshot shows a command prompt window with a blue background. The command '1 trace(Math.LOG10E);' is entered in the prompt. The output is displayed in a separate window titled 'Çıktı' (Output) with the value '0.4342944819032518'.

### 1.2.5. LOG2E

E sayısının 2 tabanına göre logaritma değeri olan 1.44269504088896387 sayısını verir.

```
1 trace(Math.LOG2E);
```

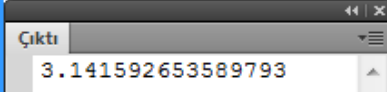


The screenshot shows a command prompt window with a blue background. The command '1 trace(Math.LOG2E);' is entered in the prompt. The output is displayed in a separate window titled 'Çıktı' (Output) with the value '1.4426950408889634'.

### 1.2.6. PI

Pi sayısını verecektir.

```
1 trace (Math.PI);
```

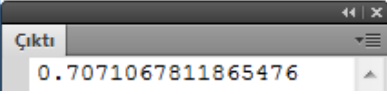


The screenshot shows a code editor with a blue line number '1' next to the code `trace (Math.PI);`. Below the code, a console window titled 'Çıktı' (Output) displays the value `3.141592653589793`.

### 1.2.7. SQRT1\_2

$\frac{1}{2}$  sayısının karekökü olan 0.7071067811865476 sayısını verecektir.

```
1 trace (Math.SQRT1_2);
```



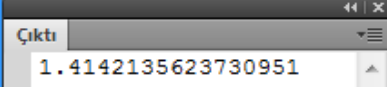
The screenshot shows a code editor with a blue line number '1' next to the code `trace (Math.SQRT1_2);`. Below the code, a console window titled 'Çıktı' (Output) displays the value `0.7071067811865476`.

### 1.2.8. SQRT2

2 sayısının karekökünü verecektir.

```
1 trace (Math.SQRT2);
```

```
2
```



The screenshot shows a code editor with blue line numbers '1' and '2'. Line 1 contains the code `trace (Math.SQRT2);`. Below the code, a console window titled 'Çıktı' (Output) displays the value `1.4142135623730951`.

## 1.3. Matematik Sınıfının Metotları

Animasyon yazılımı içerisinde kullanılabilen 18 adet math sınıfı metodu bulunmaktadır. Hesaplanan değerler trigonometrik metotlar için radyan cinsindedir.

Örneğin 30 derece, radyan cinsine çevrildiğinde 0.5235987755982988 değeri elde edilir. Birim çember üzerinde 30 derecenin sinüs değeri 0.5 iken animasyon yazılımında hesaplatılmaya çalışıldığında `Math.sin (30)`, -0.9880316240928618 değerini alacaktır.

İşlemlerin doğru yapılması için radyan olarak değer verilmesi gerekir, yani `Math.sin(0.5235987755982988)` kullanılmalıdır. Çıkan sonuç 0.5 değeri olacaktır.

Derece ve radyan dönüşümü için;

$$\frac{\text{Radyan}}{\text{Math.pi}} = \frac{\text{derece}}{180}$$

formülü kullanılabilir.

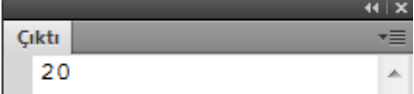
### 1.3.1. abs( )

Parantez içerisine girilen sayının mutlak değerini verir.

```
1 var sayi:Number=Math.abs (-20);
```

```
2 trace (sayi);
```

```
3
```



The screenshot shows a code editor with blue line numbers '1', '2', and '3'. Line 1 contains `var sayi:Number=Math.abs (-20);`, line 2 contains `trace (sayi);`. Below the code, a console window titled 'Çıktı' (Output) displays the value `20`.

### 1.3.2. acos()

Girilen sayının ters kosinüs değerini verir.

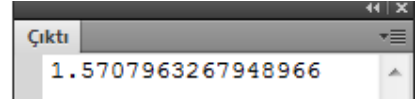
```
1 var terskos:Number=Math.acos(1);
2 trace(terskos);
3
```



### 1.3.3. asin()

Girilen sayının ters sinüs değerini verir.

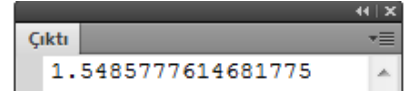
```
1 var teressin:Number=Math.asin(1);
2 trace(teressin);
3
```



### 1.3.4. atan()

Girilen sayının ters tanjant değerini verir.

```
1 var terstanj:Number=Math.atan(45);
2 trace(terstanj);
3
```



### 1.3.5. ceil()

Ondalıklı sayıyı yukarı yuvarlar.

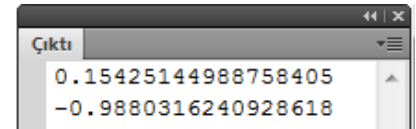
```
1 var yuvarla:Number=Math.ceil(59.0001);
2 trace(yuvarla);
3
```



### 1.3.6. cos()

Parantez içerisine girilen sayının kosinüs değerini verir. Aynı şekilde kullanılan sin() metodu sinüs değerini verir.

```
1 var kos:Number=Math.cos(30);
2 var sin:Number=Math.sin(30);
3 trace(kos);
4 trace(sin);
```

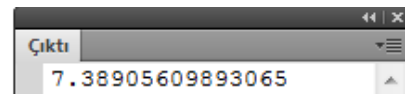


**Not:** Burada ki çıkan sonuçların radyan değerine karşılık olarak verildiğini unutmayın.

### 1.3.7. exp()

e sayısının üssünü almak için kullanılır. Parantez içerisine verilen değer kadar e sayısının üssünü alır.

```
1 var eSayisiKaresi:Number=Math.exp(2);
2 trace(eSayisiKaresi);
3
```



### 1.3.8. floor()

Parantezler arasına girilen sayının tam sayı kısmını verir.

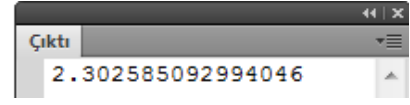
```
1 var tamSayiYap:Number=Math.floor(215.8545);
2 trace(tamSayiYap);
3
4
```



### 1.3.9. log()

Girilen sayının logaritmasını hesaplar.

```
1 var logaritma:Number=Math.log(10);
2 trace(logaritma);
3
```



### 1.3.10. max()

Virgülle ayrılmış iki sayıdan büyük olanını geriye değer olarak verir.

```
1 var buyukSayi:Number=Math.max(25,40);
2 trace(buyukSayi);
3
4
```



### 1.3.11. min()

Virgülle ayrılmış iki sayıdan küçük olanını geriye değer olarak verir.

```
1 var kucukSayi:Number=Math.min(25,40);
2 trace(kucukSayi);
3
```



### 1.3.12. pow()

Üs alma işlemini yapar. Birinci sayı taban, ikinci sayı üs konumunda olur.

```
1 var ust:Number=Math.pow(5,2);
2 trace(ust);
3
```



### 1.3.13. round()

Ondalıklı olarak verilen tam sayıyı kendisine en yakın olan tam sayıya yuvarlar.

```
1 var yuvarla1:Number=Math.round(5.49999);
2 var yuvarla2:Number=Math.round(5.50000);
3 trace(yuvarla1);
4 trace(yuvarla2);
```



### 1.3.14. sqrt()

Girilen sayının karekök değerini verir.

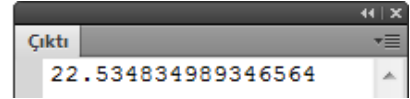
```
1 var karekok:Number=Math.sqrt(4);  
2 trace(karekok);  
3
```



### 1.3.15. random()

Random metodu 0-1 arasında rastgele sayılar üretir. Random metodu ile 0-50 arasında sayı üretmek istenirse çıkan sonucun 50 ile çarpılması gerekir.

```
1 var rastgele:Number=Math.random()*50;  
2 trace(rastgele);  
3
```



## UYGULAMA FAALİYETİ

Matematiksel sabitler ve fonksiyonları kullanmak, üretilen değerleri değişkenlere atamak ve çıkış ekranında görmek için aşağıdaki uygulamayı yapınız.

İşlem Basamakları	Öneriler
➤ Yeni bir animasyon belgesi oluşturun.	➤ Dosya > Yeni (Ctrl + N) komutunu kullanabilirsiniz.
➤ Birinci frame sağ tıklayıp eylemleri seçin.	➤ F9 kısa yolunu kullanabilirsiniz.
➤ Math sınıfındaki metotların neler olduğunu hatırlayın.	➤ Eylemler paneline Math. yazarak metot penceresini görüntüleyebilirsiniz.
➤ Euler sabiti için bir değişken tanımlayıp sabit değerini görüntüleyin.	➤ Math.E komutunu kullanabilirsiniz.
➤ Logaritma 2 değerini görüntüleyin.	➤ LN2 metodunu kullanabilirsiniz.
➤ Pi değerini dairenin alan ve çevresini hesaplayan bir uygulamada kullanın.	➤ Alan için $\pi * r * r$ ,çevre için $2 * \pi * r$ formülünü kullanabilirsiniz.
➤ 30 derece kaç radyan hesaplayan uygulamayı hazırlayın.	➤ 180 derecenin pi değerine eşit olduğu orantısını kullanabilirsiniz.
➤ 1.3445 sayısını yukarı yuvarlayan uygulamayı hazırlayınız.	➤ Ceil( ) fonksiyonunu yuvarlama işlemi için kullanabilirsiniz.
➤ 23,54 sayılarından büyük olanı bulduran kodları yazınız.	➤ Max( ) fonksiyonunu kullanabilirsiniz.
➤ 34.45 sayısının tam sayı kısmını görüntüleyin.	➤ Math.floor( ) fonksiyonunu kullanabilirsiniz.
➤ Pisagor bağıntısını kullanarak üçgenin hipotenüsünü hesaplayan uygulamayı hazırlayın.	➤ Sqrt ve Pow fonksiyonlarını kullanabilirsiniz.

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Math sınıfının tüm metotlarını görüntüleyebildiniz mi?		
2. Yuvarlama işlemini yapabildiniz mi?		
3. Üs alma işlemini yapabildiniz mi?		
4. Karekök alma işlemini yapabildiniz mi ?		
5. Rastgele sayılar üretilen uygulama içerisinde kullanabildiniz mi?		
6. İki sayıdan büyük ve küçük olanını görüntüleyebildiniz mi?		
7. Herhangi bir sayının basamaklarını floor fonksiyonu yardımı ile bulabilir misiniz?		
8. Matematiksel sabitleri uygulama içerisinde kullanabilir misiniz?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise Ölçme ve Değerlendirme 'ye geçiniz.



## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) Math.E değeri değiştirilebilir bir değerdir.
2. ( ) Rastgele sayılar üretmek için Pow() metodu kullanılır.
3. ( ) floor() ile sayının tam sayı kısmını bulabiliriz.
4. ( ) Max() ile üç sayıdan en büyüğü bulunabilir.
5. ( ) Math sınıfı içerisinde matematiksel sabit ve fonksiyonlar bulunur.
6. ( ) Math sınıfını kullanmadan önce import etmemiz gerekmektedir.
7. ( ) Math sınıfındaki trigonometrik hesaplamalarda derece kullanılır.
8. ( ) SQRT2 ile 4 sayısının karekökünü alabiliriz.
9. ( ) SQRT1\_2 ile 4 sayısının karekökü alınabilir.
10. ( ) Round ve floor aynı işlemi yapar.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-2

## AMAÇ

Animasyon yazılımını kullanarak metin alanları ile çalışabileceksiniz.

## ARAŞTIRMA

- Html etiketlerini araştırın.

## 2. METİNLER

### 2.1. Metin Alanları

Animasyon yazılımı içerisinde kullanılacak üç tür metin alanı bulunur. Metin alanları özelliklerine göre statik, dinamik ve giriş metin alanı olarak isimlendirirler. Statik metin alanları içeriği değiştirilmeyen sabit olan herhangi bir müdahaleye açık olmayan yazı alanlarıdır.

Animasyon yazılımı içerisindeki nesnelere müdahale etmek için nesnelere örnek isimleri olmalıdır. Dinamik ve giriş metin alanlarına örnek isim verilebiliyorken, statik metin alanlarına özellikler panelinden örnek isim verilemez. Statik alanlar *İnternet* sayfalarında, site introlarında, kullanıcı bilgilendirilmesi ve değişmeyen alanlarda kullanılır.

Dinamik metin alanını içeriği script kodları ile uygulamanın ya da internet sitesinin akışına göre değiştirilip düzenlenebilir. Örneğin oyunlarda oyuncunun skor bilgisinin oyun içerisinde değiştirilmesi gerekir ya da internet sayfalarda kullanılan bir banner sadece içerisinde kullanılan resim dosyasının ya da yazının değiştirilmesi sonucu değiştirmek istenebilir. Dinamik metin alanları bunlar gibi durumlarda kullanıcıya büyük kolaylıklar sağlar.

Giriş metin alanları dinamik metin alanlarının tüm özelliklerine sahiptir. Dinamik alanlardan farklı olarak metin alanının içeriğini kullanıcı da değiştirebilir. Bu özelliği sayesinde üye girişi, başvuru ve mail formları, interaktif uygulamalarda sık sık kullanılmaktadır.

Metin alanlarını test etmek için;

- Animasyon yazılımını çalıştırıp yeni bir çalışma sayfası açın.
- Araç çubuğundan metin aracını seçin.
- Sahneye boş bir metin kutusu oluşturup özelliklerden statik metin özelliğini seçin.

- Metin alanı içerisine mtegm yazın.
- Aynı işlemi dinamik metin ve giriş metin alanı seçerek de gerçekleştirin.
- Ctrl+enter tuşları ile sahnemizi çalıştırıp fare ile yazıları seçmeye çalışın.

Statik metin alanı içerisine yazılan yazımız seçilemezken, dinamik metin alanı seçilir, giriş metin alanı seçilip değiştirilebilir şekilde olacaktır.

## 2.2. Metin Alanlarını Oluşturma

Metin alanlarını animasyon yazılımında fare yardımı ile sahneye ekleyebilirsiniz. Diğer bir yöntem ise script yazarak sahneye metin alanını eklemek ve özelliklerini değiştirmektir.

Metin alanını script yazarak eklemek için;

- Yeni bir çalışma sahnesi açın.
- F9 ya da Pencere → eylemler menüsüne tıklayarak eylemler panelini açın.
- *yazi* isminde bir nesne oluşturun. Oluşturacağınız nesne TextField sınıfına dahil olduğu için panel bu sınıfı otomatik olarak sahneye dahil edecektir.

```
1 import flash.text.TextField;
2 var yazi:TextField=new TextField;
```

- Oluşturduğumuz nesnemizi sahneye eklemek için *addChild* komutundan faydalanabilirsiniz.

```
1 import flash.text.TextField;
2 var yazi:TextField=new TextField;
3 addChild(yazi);
```

- *CTRL+Enter* tuşları ile sahnemizi test ettiğimizde karşımıza herhangi bir şey çıkmayacaktır. Bunun sebebi nesnemizi sahneye eklemiş olsa da kenarlıkları olmadığı için nesnenin görünmemesidir. Kenarlık eklemek için nesnemizin *border* özelliğini kullanabilirsiniz.

```
1 import flash.text.TextField;
2 var yazi:TextField=new TextField;
3 addChild(yazi);
4 yazi.border=true;
```

Şimdi sahnemizi test ettiğimizde metin alanımızı görebiliriz.

- Kenarlık rengini kırmızı yapın.

```
6 yazi.borderColor=(0xFF0000);
```

- Nesnemizin yükseklik ve genişlik ayarlarını değiştirin.

```
8 yazi.height=100;  
9 yazi.width=100;
```

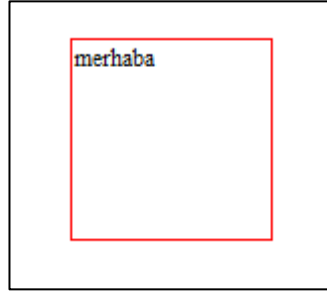
- Nesnemizi sayfa üzerinde x=50, y=50 konumuna getirin.

```
12 yazi.x=50;  
13 yazi.y=50;
```

- Artık *yazi* isimli nesne sahnede istenilen özelliklerde oluştuğuna göre içeriği oluşturabilirsiniz. String ifadelerle kullanım, trace komutu ile aynı şekildedir.

```
11 yazi.text="merhaba";
```

- Sahnemizi *CTRL+Enter* tuşları ile test edin.



Şekil 2.1: Test sahnesi

## 2.3. Metin Alanlarını Biçimlendirme

Web siteleri için sayfa içerikleri hazırlanırken yazılar üzerinde daha fazla değişiklik yapılması kaçınılmazdır. Örneğin, yazının hızlanması, fontu, boyutu, rengi gibi özellikler değiştirilmek istenebilir. Bu gibi durumlarda animasyon yazılımı gerekli tüm ayarları özellikler panelinden yapılması için geliştiricilere fırsat sunmaktadır. Bazı durumlarda ise bu ilemlerin script kullanarak kodla yapılması gerekebilir.

Sahnedeki metin alanının içeriğini biçimlendirmek için;

- Bir önceki örnek çalışmayı açın.
- Biçimlendirme ayarlarını tutmak için *TextFormat* nesnesini kullanacağız. Öncelikle nesnenin *yeniBicim* isminde bir kopyasını oluşturuyoruz.

```
13 var yeniBicim:TextFormat = new TextFormat();
```

- Biçim nesnemizin renk ve font büyüklük değerlerini belirleyin.

```
16 yeniBicim.color = 0xAA0000;  
17 yeniBicim.size = 24;
```

- Yazıyı italik yapın.

```
20 yeniBicim.italic = true;
```

- Yazıyı kalın yapın.

```
22 yeniBicim.bold = true;
```

- Yazının fontunu *Arial* olarak belirleyin. *Arial* ismini tırnak içinde yazmayı unutmayın.

```
24 yeniBicim.font = "Arial";
```

- Yapılan ayarları *yazi* nesnesine dâhil edin.

```
26 yazi.setTextFormat(yeniBicim);
```

- Sahneyi test edin.



Şekil 2.2: Yazının biçimli şekli

Yazının sağa sola dayalı ya da ortalanmış olarak metin kutusunda yazdırmak istenirse aşağıdaki parametreler kullanılarak metin alanı içeriğini hizalanabilir.

```
TextFormatAlign.CENTER  
TextFormatAlign.JUSTIFY  
TextFormatAlign.LEFT  
TextFormatAlign.RIGHT
```

Hizalamak için kullanılacak komut aşağıdaki şekilde olacaktır.

```
26 yeniBicim.align= TextFormatAlign.CENTER
```

## 2.4. HTML Metinleri Görüntülemek

Metin kutuları içerikleri istenirse HTML etiketleri ile düzenlenebilir. HTML metninin en büyük avantajı metin alanı içerisinde image dosyaları kullanılabilmesidir.

Html etiketlerini metin alanı içinde kullanmak için;

- Yeni bir çalışma sayfası açın.
- 100x200 lük metin kutusu oluşturup sahneye ekleyin.

```
1 import flash.text.TextField;
2 var metinAlani:TextField=new TextField;
3 addChild(metinAlani);
4 metinAlani.width=100;
5 metinAlani.height=200;
6 metinAlani.border=true;
```

- Metin alanı için zemin rengi atayın.

```
9 metinAlani.background=true;
10 metinAlani.backgroundColor=0xff6600;
```

- İçerik adında string türünde bir değişken tanımlayın. HTML etiketleri kullanarak içeriği oluşturun.

```
15 var icerik:String="<p>Metin kutuları içeriklerini isterseniz"+
16 "<b>html</b>etiketleri ile düzenleyebilirsiniz. </p><p>"+
17 "<i>Html</i> metnin <b>en büyük avantajı</b> <u>metin alanı"+
18 "</u> içerisinde <b>image</b>dosyaları kullanabilmenizdir</p>";
19
```

- Oluşturduğunuz içeriği htmlText komutu ile metin kutusunun içine atın.

```
20 metinAlani.htmlText=icerik;
```

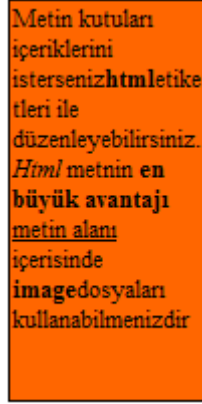
- Yazdığımız kodları *CTRL+Enter* tuşları ile test edin.
- Sahnede metin kutusu oluşacak; fakat yazılan yazı kutunun dışına taşacaktır.



Şekil 2.3: Satır kaymasız metin kutusu

- Satır kaydırma yapmak için *wordWrap* özelliğinin aktif edilmesi gerekmektedir.

```
22 metinAlani.wordWrap=true;
```



Şekil 2.4: Satır kayma yapılan metin kutusu

- İçeriğimize resim ekleyin.

```
19 "<img src='gulumse.png' width='25' height='25'>";
```

İçerik değişkenine yukarıdaki kod eklendiğinde artık metin alanı içerisinde *gulumse* isimli dosya gözükecektir. Yukarıdaki *gulumse* isimli resim dosyası çalışan swf dosyası ile aynı yerde olmalıdır.



Şekil 2.5: Resim eklenmiş metin kutusu

## 2.5. Harici Metinleri Görüntülemek

*Internet* sayfaları tasarlanırken içeriğin yoğun olması içeriğin değiştirilmesinde problemlere sebep olacaktır. Hazırlanan sayfa içeriği basit bir text dosyasından düzenlenip sayfa üzerinde değişmesini sağlamak kolay bir yoldur. Text dosyalarının değiştirilmesi ile sayfalar otomatik olarak güncellenmiş olur.

Harici metinleri sayfaya dâhil etmek için;

- Masaüstü ekranında metin isimli bir *txt* dosyası oluşturup içeriğini kendi isteğiniz şekilde hazırlayın.
- Yeni bir çalışma sayfası açıp masaüstünde istediğiniz isimde kaydedin.
- Metin kutusunu sahneye ekleyin.

```
1 import flash.text.TextField;  
2 var metinAlani:TextField=new TextField;  
3 addChild(metinAlani);
```

- *link* isminde bir değişken tanımlayıp yüklenecek dosyanın yolunu tanımlayın.

```
5 var link:URLRequest = new URLRequest("metin.txt");
```

- *yukle* isimli bir değişken oluşturun.

```
7 var yukle:URLLoader = new URLLoader();
```

- *link* ismindeki değişkenin içeriğini *yukle* isimli değişkene yükleyin.

```
9 yukle.load(link);
```

- *yukle* nesnesine bir olay dinleyicisi ekleyin. Bu dinleyici dosyanın tam olarak yüklenmesini kontrol etsin. Tam olarak yüklendiğinde bir fonksiyon tetiklesin.

```
11 yukle.addEventListener(Event.COMPLETE, calistir)
```

- Artık fonksiyonu yazabilirsiniz.

```
14 function calistir(e:Event)
15 {
16     metinAlani.text=yukle.data;
17 }
18 metinAlani.wordWrap=true;
```

Fonksiyon çalıştığında sahnedeki *metinAlani* isimli metin kutusuna “*yukle*”nin içindeki bilgiyi atın.

- Metin alanının satır kaydırmasını aktif edin. Artık masaüstündeki *metin.txt* dosyasının içeriği, sahnedeki *metinAlani* isimli *metin* kutusu içerisine aktarılmış oldu.



## UYGULAMA FAALİYETİ

Eylemler panelini kullanarak 0-500 arasında rastgele sayı oluşturup metin kutusunda gösteren uygulamayı hazırlayınız.

İşlem Basamakları	Öneriler
➤ Yeni bir animasyon belgesi oluşturun.	➤ Dosya > Yeni (Ctrl + N) komutunu kullanabilirsiniz.
➤ Birinci frame sağ tıklayıp eylemleri seçin.	➤ F9 kısa yolunu kullanabilirsiniz.
➤ Yazı isminde textField nesnesi oluşturun.	➤ textField nesnesi metin kutuları oluşturur.
➤ Metin kutusunun kenarlığını aktif edin.	➤ Border metodun kullanabilirsiniz.
➤ Sahneye metin kutusunu ekleyin.	➤ addChild komutu nesneyi sahneye ekler.
➤ Rastgele sayı oluşturup bir değişkene atayın.	➤ random fonksiyonu ile rastgele sayılar oluşturulabilir.
➤ Rastgele sayıyı tam sayıya çevirin.	➤ Math sınıfının floor metodunu kullanarak tam sayı yapabilirsiniz.
➤ Tam sayıyı string ifadeye çevirin.	➤ Metin kutusunda yazdırmak için string ifade olması gereklidir.
➤ Metin kutusunda, bulunan sayıyı yazdırın.	➤ Text komutunu kullanarak kutunun içerisinde sayıyı görüntüleyebilirsiniz.
➤ Sahneyi test edin.	➤ CTRL+Enter tuşlarını kullanabilirsiniz.

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. <i>TextField</i> nesnesi oluşturabildiniz mi?		
2. Metin alanlarını sahneye ekleyebildiniz mi?		
3. Metin alanları içerisinde string ifadeler görüntüleyebildiniz mi?		
4. Metin rengini ve büyüklüğünü değiştirdiniz mi?		
5. Metin stilini değiştirdiniz mi?		
6. Metin fontunu değiştirdiniz mi?		
7. HTML içerik oluşturduunuz mu?		
8. Harici metinleri metin kutusunda görüntüleyebildiniz mi?		
9. <i>URLLoader</i> nesnesi oluşturabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise **D**, yanlış ise **Y** yazınız.

1. ( ) Animasyon yazılımı içerisinde kullanılabilecek iki tür metin alanı bulunur.
2. ( ) Statik metin alanlar içeriğini script kodları ile değiştirebiliriz.
3. ( ) Metin alanları *Graphics* sınıfı içerisinde bulunur.
4. ( ) *border* parametresi ile metin alanına kenarlık ekleyebilirsiniz.
5. ( ) *wordWrap* komutu satır kaydırmak için kullanılır.
6. ( ) Metin alanları içerisinde resim dosyaları görüntülenemez.
7. ( ) *wordWrap* komutu ile kelimeleri eğip bükebiliriz.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-3

## AMAÇ

Animasyon yazılımını kullanarak diziler ile çalışabileceksiniz.

## ARAŞTIRMA

- Öğrendiğiniz diğer programlama dillerindeki diziler konusunu araştırarak tekrar edin.

## 3. DİZİLER

### 3.1. Dizi

Diziler içinde birden fazla değer tutabilen değişkenler gibi düşünülebilir. Örneğin, bir sınıftaki öğrencilerin adres bilgisi tutulmak istenirse; öğrenci sayısı kadar ad, soyad, adres bilgisi tutulması gerekir. Bunun için 30 kişilik sınıfta 90 tane değişken tanımlanmalıdır. Bu uygulama dizilerle yapılacak olursa 3 tane dizi tanımlanması yeterli olacaktır.

### 3.2. Dizi Tanımlama

Dizi tanımlamanın birden fazla yolu bulunmaktadır.

- Boş bir dizi oluşturup içeriğini sonradan ekleyebilirsiniz.

```
1 var ad:Array=new Array();  
2 ad[0]="volkan";  
3 ad[1]="melek";  
4 ad[2]="mehmet";
```

- Dizi tanımlamak için “var” kelimesini daha sonra dizinin ismini yazmamız gereklidir.

Kodlar incelendiğinde birinci satırda içeriği boş olarak diziyi tanımlanmıştır. Sonrasında ad değişkenine 0, 1, 2 index numaralarına göre değer verilmiştir.

Diziler 0. değerden başladığı unutulmamalıdır.

- Dizi oluşturup değer vermenin diğer bir yolu, diziyi oluştururken değeri aynı satırda vermektir.

```
1 var ad:Array=["volkan","melek","mehmet"];
```

- Değerler virgül ile birbirinden ayrılır. İsimler string değer olduğu için tırnak içerisine alınarak yazılır. Diziye string, number, boolean vb. veri türlerini

karışık olarak değer verilebilir. Dizi tanımlarken ve değer verirken tür sınırlaması yoktur.

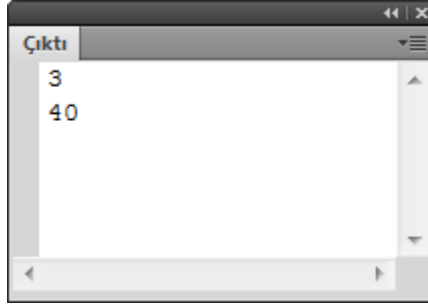
```
1 var ad:Array=new Array("volkan","melek","mehmet");
```

Dizi tanımladıktan sonra trace komutu ile dizi kontrol edilebilir.

```
3 trace(ad);
```

Dizilerin eleman sayısını bulmak için *length* komutunu kullanılabilir. *length* komutu string ifadelerinde karakter sayısını bulmak için kullanılabilir.

```
1 var ad:Array=new Array("volkan","melek","mehmet");
2 var isim:String="MESLEKİ VE TEKNİK EĞİTİM GENEL MÜDÜRLÜĞÜ";
3 trace(ad.length);;
4 trace(isim.length);
```



Şekil 3.1: Çıktı ekranı

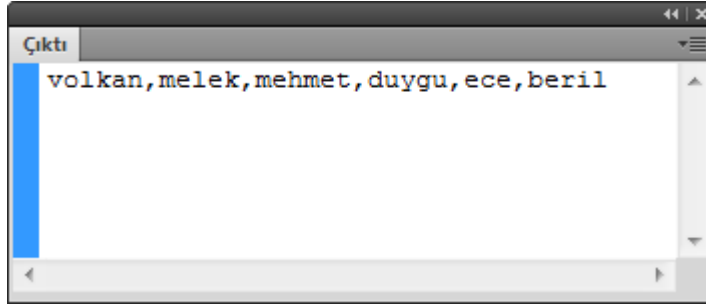
### 3.3. Diziler Üzerinde İşlemler

#### ➤ push()

Dizinin sonuna yeni bir dizi elemanı eklemek için kullanılır.

```
1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 ad.push("ece","beril");
3 trace(ad);
```

Yukarıdaki kodlar incelendiğinde Ad dizisine ece ve beril isimleri eklendiği görülecektir. Trace ile kontrol edildiğinde dizinin artık 6 elemanlı bütün isimlerin bulunduğu bir dizi olduğunu görülebilir.



Şekil 3.2: Çıktı ekranı

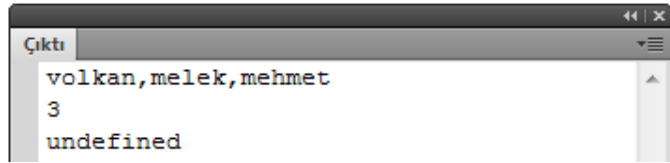
### ➤ **pop()**

Dizinin sonundaki elemanı kaldırmak için kullanılır.

```
1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 ad.pop();
3 trace(ad);
```

Komutlar test edildiğinde dizinin son ismi olan duygu yok olacaktır. Dizinin eleman sayısını kontrol edildiğinde dizinin üç elemanlı bir dizi haline geldiği görülecektir.

```
1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 ad.pop();
3 trace(ad);
4 trace(ad.length);
5 trace(ad[3]);
6
```



Ad dizisinin 3. Elemanını çıkış ekranına yazdırmak istersek tanımsız (undefined) hatasını alırız. Pop komutu dizinin elemanını hiç olmamışçasına yok edecektir.

### ➤ **unshift()**

Dizinin başına eleman ekler. Birden fazla eleman bir kerede eklenebilir.

```
1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 ad.unshift("ece","beril");
3 trace(ad);
4 trace(ad.length);
```

Komut çalıştırıldığında ece ve beril isimleri dizinin 0. ve 1. elemanları olarak karşımıza çıkacaktır.

Dizinin uzunluğu kontrol edildiğinde dizide 6 adet eleman olduğunu görürüz.

### ➤ **shift()**

Dizinin başındaki elemanı kaldırır.

```

1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 ad.shift();
3 trace(ad);
4 trace(ad.length);
5
6

```

```

Çıktı
melek, mehmet, duygu
3

```

### ➤ Delete

Dizide istenilen elemanı silmek için kullanılır. *pop* ve *shift* komutları dizi elemanını tamamen kaldırırken Delete komutu ile dizi elemanını siler fakat yeri dizi içerisinde kalır.

```

1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 delete ad[1];
3 trace(ad);
4 trace(ad.length);
5

```

```

Çıktı
volkan, , mehmet, duygu
4

```

Dizinin 1 index numaralı elemanı için silme komutu verilmiştir. bu komutla dizinin 1 index numaralı bölümünün içeri boşaltmıştır.. Dizi elemanı kontrol edildiğinde dizinin yine 4 elemanlı olduğunu görülecektir.

### ➤ Splice()

Dizide istenilen bölüme eleman eklenmesini ya da eleman silinmesini sağlar.

- Dizi içerisinde eleman silmek için

```

1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 ad.splice(1,1);
3 trace(ad);
4

```

```

Çıktı
volkan, mehmet, duygu

```

komutları kullanılır. *Splice* komutunda parantez içerisinde yazılan ilk rakam hangi dizi elemanından sonra silineceği, ikinci rakam ise kaç eleman silineceğini belirtir. Yukarıdaki kullanımda ad dizisinde 1. elemanından sonra bir eleman siler. Eğer ikinci rakam kullanılmazsa birinci rakamın gösterdiği dizi elemanından sonraki tüm elemanlar silinir.

```

1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 ad.splice(1);
3 trace(ad);
4

```

```

Çıktı
volkan

```

- Dizi içerisine eleman eklemek için;

```

1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 ad.splice(1,0,"ece","beril");
3 trace(ad);
4
5

```

```

Çıktı
volkan, ece, beril, melek, mehmet, duygu

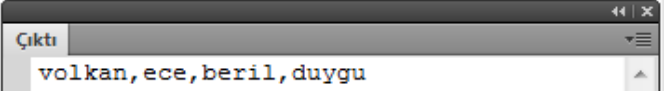
```

komutları kullanılır. Yukarıdaki kullanımda *splice* komutu birinci elemandan sonra ece ve beril kelimelerini eklemek üzere çalışacaktır. Eklemek ve silmek arasındaki fark

ikinci rakamın 0 olarak belirlenmesidir. İkinci rakam silinecek eleman sayısını belirlediği için *splice* kelimeleri eklemek için çalışır.

Diziden bazı değerleri silip yeni değerler eklenmek istenirse;

```
1 var ad:Array=new Array("volkan","melek","mehmet","duygu");
2 ad.splice(1,2,"ece","beril");
3 trace(ad);
4
5
```




şeklinde bir kullanım olacaktır.

### ➤ indexOf

Dizi içinde geçen bir değerın aranması için kullanılır. . Uzun dizilerde kelimenin dizinin kaçınıcı elemanı olduğunu gösterir.

```
1 var ad:Array=new Array("volkan","melek","mehmet","duygu","ece","beril");
2 trace(ad.indexOf("volkan"));
3 trace(ad.indexOf("mehmet"));
4 trace(ad.indexOf("beril"));
5
6
```

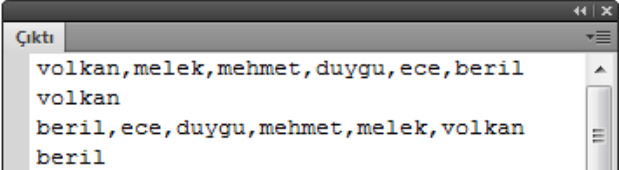


Aranılan kelime dizi içerisinde yok ise -1 değerini döndürecektir.

### ➤ Reverse()

Dizi elemanlarını ters çevirmemizi sağlar.

```
1 var ad:Array=new Array("volkan","melek","mehmet","duygu","ece","beril");
2 trace(ad);
3 trace(ad[0]);
4 ad.reverse();
5 trace(ad);
6 trace(ad[0]);
7
```




Dizi reverse komutundan önce tanımlandığı gibi ekrana yazdırılıyor. Reverse komutu ile birlikte dizi elemanları ters çevrilip yeniden listeleniyor. Dizinin sıfırıncı elemanın nasıl değiştiğini yukarıdaki kod ile rahat bir şekilde görebiliyoruz.

### ➤ Sort()

Dizi elemanlarını sıralamak için kullanılır. Komut kullanıldığında a-z, 0-9 aralıklarında sıralama işlemini yerine getirir.

```
1 var ad:Array=new Array("volkan","melek","mehmet","duygu","ece","beril");
2 var sayilar:Array=[3,4,6,8,7,1,0]
3 ad.sort();
4 sayilar.sort();
5 trace(ad);
6 trace(sayilar);
7
```





## UYGULAMA FAALİYETİ

Klavyeden boşluk tuşuna basarak sahnedeki metin kutusunun içeriğine yazılan isimleri isimler dizisine ekleyen bir uygulama hazırlayın. Boşluk tuşuna basıldığında metin kutusu içeriği otomatik olarak boşaltılsın.

İşlem Basamakları	Öneriler
➤ Yeni bir animasyon belgesi oluşturun.	➤ Dosya > Yeni (Ctrl + N) komutunu kullanabilirsiniz.
➤ Eylemler panelini çalıştırın.	➤ F9 kısa yolunu kullanabilirsiniz.
➤ isimler isminde bir dizi tanımlayın.	➤ Array sınıfını kullanarak oluşturabilirsiniz.
➤ Yazı isminde bir metin alanı oluşturun.	➤ textField sınıfını kullanabilirsiniz.
➤ Metin alanı türünü input yapın.	➤ type parametresini “input” ayarlayın.
➤ Metin alanını sahneye ekleyin.	➤ addChild metodunu kullanın.
➤ Sahneye klavye olaylarını dinleyen bir olay yöneticisi tanımlayın ekle isimli fonksiyonu tetiklesin.	➤ Key_down özelliğini kullanabilirsiniz..
➤ Ekle isimli fonksiyon yazın. Klavyeden boşluk tuşuna basıldığında diziye eleman eklesin.	➤ Boşluk tuşu için ascii kodu 32 olarak kullanabilirsiniz..
➤ Eğer boşluğa basılırsa metin kutusunda yazılan ismi diziye ekleyecek kodları yazın.	➤ Push metodunu kullanarak diziye eleman ekleyebilirsiniz.
➤ Metin kutusunu boşaltın.	➤ text="" şeklinde kullanabilirsiniz.
➤ Çalışmayı test ederek görüntüleyin.	➤ Kontrol et > Filmi test Et > Test Et (Ctrl + Enter) komutunu kullanabilirsiniz.

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Dizi tanımlaması yapabildiniz mi?		
2. Diziye eleman ekleyebildiniz mi?		
3. Dizi uzunluğunu bulabildiniz mi?		
4. <i>Push</i> metodunu kullanabildiniz mi?		
5. <i>Splice</i> metodunu kullanabildiniz mi?		
6. <i>Unshift</i> metodunu kullanabildiniz mi?		
7. <i>indexOf</i> metodu ile dizi index numarasını bulabildiniz mi?		
8. Diziyi ters çevirebildiniz mi?		
9. Diziyi sıralayabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınızı “**Evet**” ise “Ölçme ve Değerlendirme” ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise **D**, yanlış ise **Y** yazınız.

1. ( ) Diziler içerisine birden fazla eleman eklenemez.
2. ( ) Dizi tanımlarken *array* sınıfını kullanılır.
3. ( ) *push* metodu ile diziye eleman eklenebilir.
4. ( ) *Sort* metodu ile dizi içerisinde elemanlar sıralanabilir.
5. ( ) *reverse* metodu ile diziler ters çevrilebilir.
6. ( ) *index of* metodu ile elemanın dizi içerisindeki sırası bulunabilir.
7. ( ) *trace* komutu ile dizi içeriği görüntülenemez.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-4

## AMAÇ

Animasyon yazılımı ile grafik işlemleri gerçekleştirebileceksiniz.

## ARAŞTIRMA

- Animasyon yazılımında şekil oluştururken kullanılan parametreleri araştırın.

## 4. KODLAMA İLE GRAFİK OLUŞTURMAK

### 4.1. Düz Çizgiler Çizmek

Animasyon yazılımında sahne üzerinde fare ile çizilen tüm vektörler, script kodları ile çizilebilir. Çizim api leri kullanarak düz ve eğik çizgiler, hazır geometrik şekilleri rahatlıkla çizilebilir *Shape, movieClip, Sprite* sınıflarının *graphics* özelliklerini kullanarak hızlı bir şekilde çizimler yapabiliriz.

Düz çizgi çizmek için;

- Yeni bir çalışma sahnesi açın eylemler panelini kodlamaya hazır hale getirin.
- İlk olarak *cizgi* isiminde bir *movieClip* oluşturun. *MovieClip* sınıfını editör otomatik olarak ekleyecektir.

```
1 import flash.display.MovieClip;
2 var çizgi:MovieClip=new MovieClip();
```

- Çizgi için stil ayarlarını yapın. Başlangıçta kullanılması gereken üç adet parametre bulunmaktadır. Bunlar sırasıyla çizgi kalınlığı, çizgi rengi ve çizgi görünürlüğüdür.

```
4 çizgi.graphics.lineStyle(
```

komutunu yazıldığında editör kodu yazmak için kolaylık sağlayacak ipucu penceresini çıkartacaktır.

```
lineStyle(thickness:Number=null, color:uint=0, alpha:Number=1.0,
```

4 px kalınlığında 00cc00 renginde ve yarı saydam bir çizgi elde etmek için

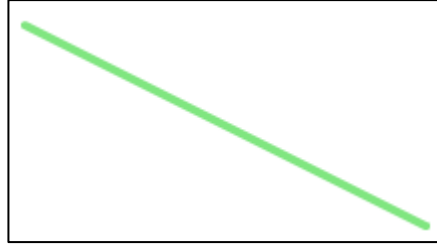
```
4 çizgi.graphics.lineStyle(4,0x00CC00,0.5);
```

komutu kullanılmalıdır. Alpha değeri 0 ile1 arasında bir değerdir. 0 görünmez iken 1 tam görünürdür.

- *lineTo* komutu ile bitiş noktasını belirlenmelidir. Başlangıç noktası standart olarak 0,0 noktası olacaktır. Bitiş noktası olarak x=200, y=100 olarak verildiğinde 0, 0 noktasından 200, 100 noktasına kadar çizgi oluşturmak için gereken kodları yazılmış olur.

```
7 cizgi.graphics.lineTo(200,100);
```

- Sahnemizi *CTRL+Enter* ile çalışmayı test edin.



Şekil 4.1: Oluşturulan çizgi

## 4.2. Çizgilerle Şekiller Oluşturmak

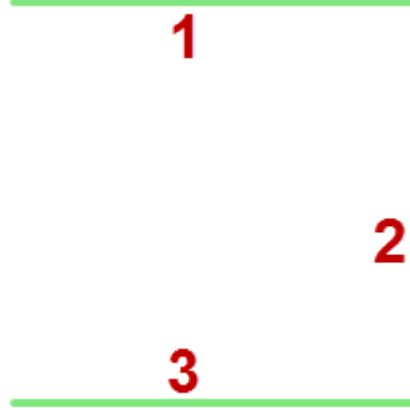
Birden fazla çizgiyi kullanarak şekiller oluşturabiliriz. Kare çizmek için;

- *MovieClip*'i oluşturulup çizgi stillerini belirleyin.

Karenin 0, 0 noktasından başladığı düşünülerek sırayla çizgiler için noktalar belirlenmelidir.

```
1 import flash.display.MovieClip;
2 var cizgi:MovieClip=new MovieClip();
3 cizgi.graphics.lineStyle(4,0x00CC00,0.5);
4
5 cizgi.graphics.lineTo(200,0);
6 cizgi.graphics.lineTo(200,200);
7 cizgi.graphics.lineTo(0,200);
8
9 addChild(cizgi);
```

Yukarıdaki komutlar sonucunda üç adet çizgi oluşacaktır. Şekle herhangi bir kaplama verilmediği için şekil otomatik olarak kapanmayacaktır.



Şekil 4.2: Kapatılmamış şekil

```
5 çizgi.graphics.beginFill(0x225544,1);
```

- *beginFill* metodu hazırlanan şekle renk dolgusu verecektir. Bu metot kullanıldığında şekli otomatik olarak kendisi kapatır ve renk dolgusunu oluşturur. Sol alt nokta belirlendikten sonra 0, 0 noktasına dönülmesine gerek yoktur.



Şekil 4.3: Dolgu verilmiş şekil

- *AddChild* metodu ile *movieClip*'i sahneyimize ekleyin.
- CTRL+Enter tuşları ile sahneyimizi test ediyoruz.

### 4.3. Şeklin Başlangıç Noktasını Değiştirmek

Standart olarak çizilen şekil sahne üzerinde 0,0 koordinatlarından başlayarak çizilir. Şeklin başlangıç noktasını değiştirmek için *moveTo* komutu kullanılır. Başlangıç noktası belirlendikten sonra diğer işlemler aynıdır.

Sahnemizde 50,50 noktasından başlayan bir kare çizmek için ;

- Yeni bir çalışma sahnesi açın.
- *MovieClip* oluşturup çizgi stilini ve dolgu rengini belirleyin.

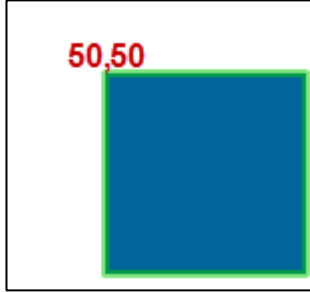
```
1 import flash.display.MovieClip;
2 var cizgi:MovieClip=new MovieClip();
3 cizgi.graphics.lineStyle(4,0x00CC00,0.5);
4 cizgi.graphics.beginFill(0x006699,1);
```

- Şeklimiz için başlangıç noktası belirliyoruz. Çizgileri oluşturmak için gerekli diğer noktaları kodlarda yazıyoruz. Doğru koordinat noktalarını bulmak için öncelikle defterinizde çizerek noktaların değerlerini tespit edebilirsiniz.

```
6 cizgi.graphics.moveTo(50,50);
7 cizgi.graphics.lineTo(150,50);
8 cizgi.graphics.lineTo(150,150);
9 cizgi.graphics.lineTo(50,150);
```

- Şekli sahnemize ekleyip test ediniz.

```
11 addChild(cizgi);
```

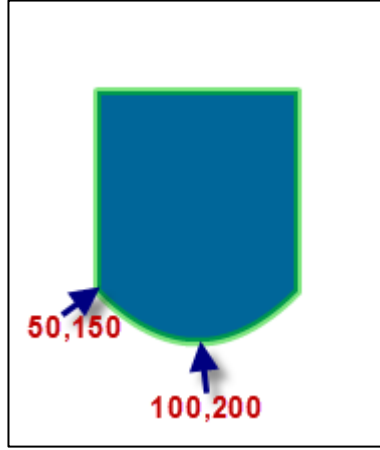


Şekil 4.4: Başlangıç noktası 50,50 olan şekil

## 4.4. Eğik Çizgiler Çizmek

Düz çizgi yerine istenirse çizgiler eğik olarak çizilebilir. Düz çizgi çizerken oluşturulurken çizginin bitiş koordinatlarının girilmesi yeterli olmaktadır. Eğik çizgi çizerken bitiş koordinatlarının yanında eğikliği belirtecek *controlX* ve *controlY* parametreleri de yazılmış olmalıdır.

Aşağıdaki şekli çizmek için;



Şekil 4.5: Eğik çizgili şekil

- Yeni bir çalışma sahnesi açarak *cizgi* isiminde bir *movieClip* oluşturunuz.
- Çizgi stilini ve dolgu rengini belirleyiniz.

```
1 import flash.display.MovieClip;
2 var çizgi:MovieClip=new MovieClip();
3 çizgi.graphics.lineStyle(4,0x00CC00,0.5);
4 çizgi.graphics.beginFill(0x006699,1);
```

- Şekli oluşturmak için iki düz çizgi, bir eğik çizgi kodu yazıp şeklin otomatik kapanmasını sağlıyoruz. Eğik çizgideki kontrol noktası çizginin eğikliğini ayarlamaktadır. Noktanın yerini değiştirerek eğikliği değiştirebilirsiniz.

```
6 çizgi.graphics.moveTo(50,50);
7 çizgi.graphics.lineTo(150,50);
8 çizgi.graphics.lineTo(150,150);
9 çizgi.graphics.curveTo(100,200,50,150);
10 addChild(çizgi);
```

## 4.6. Birden Fazla Şekil Oluşturmak

Sahne üzerinde birden fazla şekil çizilmek istenebilir. Düz çizgiler oluştururken *lineTo* komutu noktaları birleştirerek şekilleri meydana getiriyordu. Bu yüzden şekil bitip diğer şekle bağlandığında ActionScript birinci şeklin son noktası ve ikinci şeklin ilk noktasını birbirine bağlamaya çalışacak ve şekli bozacaktır.

Bu olayı engellemek için *endfill* komutu ile çizilen şeklin bittiğini belirtmek gereklidir.

Sahnede yan yana iki adet üçgen şekli çizmek için;

- Yeni bir çalışma sayfası açın, eylemler panelinde *şekil* isiminde yeni bir *movieclip* oluşturun.



```
1 import flash.display.MovieClip;
2 var sekil:MovieClip=new MovieClip();
```

- Çizgi ve dolgu stillerini belirleyin.

```
6 sekil.graphics.lineStyle(4,0x55ccaa,1);
7 sekil.graphics.beginFill(0x444,1);
```

- Üçgenin başlangıç noktası olarak 200, 200 olarak belirleyin.

```
10 sekil.graphics.moveTo(200,200);
11 sekil.graphics.lineTo(250,250);
12 sekil.graphics.lineTo(150,250);
```

- Diğer Üçgene başlanmadan birinci üçgenin kapatılması gerekmektedir.

```
16 sekil.graphics.endFill();
```

- İkinci üçgeni çizmek için dolgu rengi ve noktaları belirleyin.

```
18 sekil.graphics.beginFill(0xcc,1);
19 sekil.graphics.moveTo(300,200);
20 sekil.graphics.lineTo(350,250);
21 sekil.graphics.lineTo(250,250);
```

- Şekli sahneye ekleyin.

```
20 addChild(sekil);
```



Şekil 4.6: Oluşturulan şekil

## 4.6. Hazır Geometrik Şekiller Oluşturmak

Noktalarla her şekil oluşturulabilir. Bazı geometrik şekiller ön tanımlı olarak *graphics* sınıfının içerisinde bulunmaktadır. Elips, daire, dikdörtgen, köşeleri yuvarlatılmış dikdörtgen şekilleri *draw* komutu ile hızlı bir şekilde ekrana çizilebilir.

Dikdörtgen çizerken *drawRect* komutu kullanılır. Dört adet parametresi vardır. Bunlar sırasıyla çizimin ekrandaki başlama koordinatları *x* ve *y*, dikdörtgenin genişlik değeri, dikdörtgenin yükseklik değerleridir.

Ekrana 200 px genişliğinde 100 px yüksekliğinde 50,50 px koordinatlarından başlayacak şekilde bir dikdörtgen çizmek için;

- Yeni bir çalışma sayfası açın ve eylemler paneline geometrik isminde bir *movieclip* oluşturun.
- Çizgi stilini belirleyin.

```
1 import flash.display.MovieClip;
2 var geometrik:MovieClip=new MovieClip();
3 geometrik.graphics.lineStyle(2,0x002288,1);
```

- *drawRect* komutu ile sahneye dikdörtgeni ekleyin.

```
5 geometrik.graphics.drawRect(50,50,200,100);
6 addChild(geometrik);
```

Daire çizerken *drawCircle* komutu kullanılır. Üç adet parametresi vardır. Şeklin ekrandaki başlangıç değeri olan x, y ve dairenin yarıçapı olan üçüncü parametre ile daire ekranda oluşturulur.

Ekranda 50,50 koordinatlarından başlayacak 40 px yarıçapında olan bir daire çizmek için;

- Yeni bir çalışma sayfası açın ve eylemler paneline geometrik isiminde bir *movieclip* oluşturun.
- Çizgi stilini belirleyin.

```
1 import flash.display.MovieClip;
2 var geometrik:MovieClip=new MovieClip();
3 geometrik.graphics.lineStyle(2,0x002288,1);
```

- *drawCircle* komutu ile sahneye daire ekleyin.

```
5 geometrik.graphics.drawCircle(50,50,40);
6 addChild(geometrik)
```

Elips çizerken *drawEllipse* komutu kullanılır. Dört adet parametresi vardır. Şeklin ekrandaki başlangıç değeri olan x, y elipsin yatay genişliği ve yükseklik değerleri ile elips ekranda oluşturulur.

Ekranda 50,50 koordinatlarından başlayacak 200 px genişliğinde ve 100px yüksekliğinde olan bir elips çizmek için;

- Yeni bir çalışma sayfası açın ve eylemler paneline geometrik isiminde bir *movieclip* oluşturun.
- Çizgi stilini belirleyin.

```
1 import flash.display.MovieClip;
2 var geometrik:MovieClip=new MovieClip();
3 geometrik.graphics.lineStyle(2,0x002288,1);
```

- *DrawEllipse* komutu ile sahneye elips ekleyin.

```
5 geometrik.graphics.drawEllipse(50,50,200,100);
6 addChild(geometrik)
```

Köşeleri yuvarlatılmış dikdörtgen çizerken *drawRoundRect* komutu kullanılır. Altı adet parametresi vardır. Bu parametreler şeklin ekrandaki başlangıç değeri olan x, y dikdörtgenin yatay genişliği ve yükseklik değerleri ile yuvarlatma değerleridir.

Ekranda 50, 50 koordinatlarından başlayacak 200 px genişliğinde, 100px yüksekliğinde ve 50,50 yuvarlatma değerlerinde olan bir dikdörtgen çizmek için;

- Yeni bir çalışma sayfası açın ve eylemler paneline geometrik isminde bir movieClip oluşturun.
- Çizgi stilini belirleyin.

```
1 import flash.display.MovieClip;
2 var geometrik:MovieClip=new MovieClip();
3 geometrik.graphics.lineStyle(2,0x002288,1);
```

- *DrawRoundRect* komutu ile sahneye dikdörtgen ekleyin.

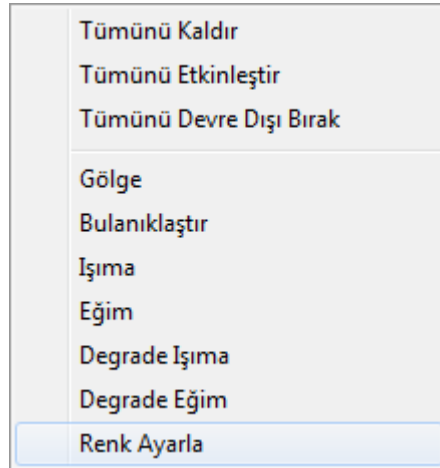
```
5 geometrik.graphics.drawRoundRect(50,50,200,100,50,50);
6 addChild(geometrik)
```



Şekil 4.7: Köşeleri yuvarlatılmış dikdörtgen

## 4.7. Filtreler

Animasyon yazılımında moviecliplere özellikler panelinden *ışırma (glow)*, *bulanıklık(blur)*, *gölge(drop shadow)* ve *eğim(bevel)* filtreleri uygulanabilir ve özellikler panelinden bunların parametreleri değiştirilebilir. Yapılan tüm işlemler eylemler panelinden script kodları ile yapabilirsiniz.



Şekil 4.7: Filtre menüsü

Sahne oluşturulan kare nesnesine sırasıyla bulanıklık, gölge, ışım ve eğim efekti uygulamak için;

- Yeni bir çalışma sahnesi açın ve eylemler panelini F9 ile çalıştırın.
- Sahneye kare isiminde 50.50 px boyutlarında bir kare oluşturun.

```
3 var kare:MovieClip=new MovieClip();
4 kare.graphics.lineStyle(1,0x000,1)
5 kare.graphics.beginFill(0x00ccff,1);
6 kare.graphics.drawRect(100,100,50,50);
7 addChild(kare);
```

Filtreleri kullanmadan önce filtre sınıfının sahneye dâhil edilmesi gerekmektedir.

```
1 import flash.filters.*;
```

- *Bulanik* isiminde bir *blurFilter* nesnesi tanımlayarak kare nesnesine filtreyi uygulayın.

```
9 var bulanik:BlurFilter = new BlurFilter();
10 kare.filters = [bulanik];
```

- *CTRL+Enter* ile animasyonu test edin.



Şekil 4.8: Blur efekti

- Sahnedeki bulanıklık filtre kodlarını silerek gölge efekti vermek için gerekli kodları yazın.

```
15 var ds:DropShadowFilter=new DropShadowFilter();
16 kare.filters=[ds];
```

- *CTRL+Enter* ile animasyonu test edin.



Şekil 4.9: Gölge efekti

- Sahnedeki gölge filtre kodlarını silerek ışım efekti vermek için gerekli kodları yazın

```
12 var glw:GlowFilter=new GlowFilter();
13 kare.filters=[glw];
```

- *CTRL+Enter* ile animasyonu test edin.



Şekil 4.10: Işıma efekti

- Sahnemizdeki ışımaya filtre kodlarını silerek eğim efekti vermek için gerekli kodları yazın.

```
18 var bevelF:BevelFilter=new BevelFilter();
19 kare.filters=[bevelF];
```

- *CTRL+Enter* ile animasyonu test edin.



Şekil 4.11: Eğim efekti

Uyguladığımız filtreleri standart değerlerde uyguladık. İstersek nesne tanımlanırken parantezin içerisinde rehber yardımı ile parametreleri değiştirebiliriz.

```
18 var bevelF:BevelFilter=new BevelFilter(
: Number=1.0, shadowColor:uint=0, shadowAlpha:Number=1.0, blurX:Number=4.0, blurY:Number=4.0, strength:Number=
```

Filtrelerin uygulanış sırası önemlidir. Son uygulanan filtre diğerleri kapatır. Filtreleri birlikte karıştırarak kullanabilirsiniz. Bunun için bir dizi oluşturup nesneye uygulamanız yeterlidir.

Birden fazla filtreyi nesneye uygulamak için;

- Uygulanmak istenen filtreleri tanımlayıp filtre isimli bir dizi içerisine elaman tanımlayın. Nesneye filtre olarak uyguladığımızda tüm filtreler dizi içerisindeki sırasıyla nesne üzerinde uygulanacaktır.

```
9 var bulanik:BlurFilter = new BlurFilter();
10 var glw:GlowFilter=new GlowFilter();
11 var ds:DropShadowFilter=new DropShadowFilter();
12 var bevelF:BevelFilter=new BevelFilter();
13
14 var filtre:Array=new Array(bulanik,glw,ds,bevelF);
15 kare.filters=filtre;
```

- *CTRL+Enter* ile animasyonu test edin.



Şekil 4.12: Tüm efektler

- Filtrelerin uygulanış sıralarını deęiřtirerek daha farklı sonuçlar alabilirsiniz.

```
14 var filtre:Array=new Array(bulanik,bevef,glw,ds);  
15 kare.filters=filtre;
```

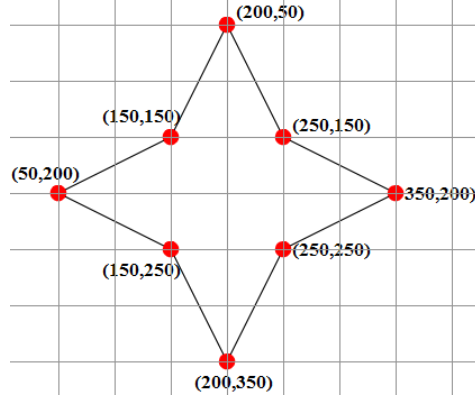


***CTRL+Enter*** ile animasyonu test edin.

řekil 4.13: Tm efektler

## UYGULAMA FAALİYETİ

Aşağıdaki şekli script kodları ile oluşturun.



İşlem Basamakları	Öneriler
➤ Yeni bir belge açınız.	➤ Karşılama ekranını veya Dosya > Yeni (Ctrl + N) komutunu kullanabilirsiniz.
➤ Eylemler panelini açın.	➤ F9 kısa yolunu kullanabilirsiniz.
➤ yıldız isminde movieClip nesnesi tanımlayın.	➤ movieClip nesnesi şekil için kullandığımız nesne türünden biridir.
➤ Başlangıç noktasını 200, 50 olarak belirleyin.	➤ moveTo komutunu kullanabilirsiniz.
➤ Sırasıyla noktaları birbirine bağlayın, saat yönünü kullanın.	➤ LineTo komutunu kullanabilirsiniz.
➤ Nesneyi sahneye ekleyin.	➤ addChild metodunu kullanabilirsiniz.
➤ Animasyonu test edin.	➤ CTRL+Enter tuşları ile test edebilirsiniz.

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Düz çizgiler oluşturabildiniz mi?		
2. Çizginin stilini belirleyebildiniz mi?		
3. Çizgilerden şekiller oluşturabildiniz mi?		
4. Oluşturduğunuz şekillere dolgu verebildiniz mi?		
5. Şeklin başlangıç noktasını değiştirebildiniz mi?		
6. Eğik çizgiler oluşturduğunuz mu?		
7. Eğik çizgiler için kontrol noktalarını tespit edebildiniz mi?		
8. Aynı sahnede birden fazla şekil oluşturabildiniz mi?		
9. <i>endFill</i> metodunun ne iş yaptığını öğrendiniz mi?		
10. Hazır geometrik şekiller çizebildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.



## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) Çizgi ve şekiller *graphics* sınıfı ile oluşturulur.
2. ( ) *LineStyle* ile çizginin eğimini belirleriz.
3. ( ) Şekle dolgu vermek için *LineStyle* kullanılır.
4. ( ) Sahneye çizgiyi eklemek için *addLine* komutu kullanılır.
5. ( ) Şeklin başlangıç noktası standart olarak 0,0 noktasıdır.
6. ( ) Şeklin başlangıç noktası değiştirilemez.
7. ( ) Eğik çizgiler oluşturmak için kontrol noktaları eklenmelidir.
8. ( ) Sahneye üç den fazla şekil eklenemez.
9. ( ) *MovieClip* nesnesi *display* sınıfının bir ögesidir.
10. ( ) Yıldız şeklini hazır bir şekilde *graphics* sınıfı içerisinde çizilebilir.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-5

## AMAÇ

Animasyon yazılımını kullanarak ses ve video ile çalışabilecek çoklu ortam işlemlerini gerçekleştirebileceksiniz.

## ARAŞTIRMA

- Animasyon yazılımı ile hazırlanan hangi uygulamalarda çoklu ortam nesneleri kullanılır? Araştırınız.

## 5. ÇOKLU ORTAM İŞLEMLERİ

### 5.1. Harici Resimleri ve Animasyonları Yükleme

Flash içerisine resim dosyalarını, animasyon (swf) dosyalarını script kodları ile uygulamanın içerisine, uygulamanın akışına göre dahil edilebilir. Dosyaların yüklenip yüklenmediğini, yüzde olarak ne kadar yüklediği gibi bilgileri öğrenebilir, kullanıcı bilgilendirilebilir.

Tüm görsel dosyalar aynı şekilde yüklenir. Dosyaları yüklerken *loader* nesnesi ve *URLRequest* komutunu kullanılır. *URLRequest* komutu dışarıdan alınan dosya bilgilerini *loader* nesnesine yüklemekle sorumludur.

Temel olarak işleyiş aşağıdaki şekildedir:

- İlk önce *loader* nesnesi oluşturulur. *yukleyici* isiminde bir *loader* nesnesi oluşturun.

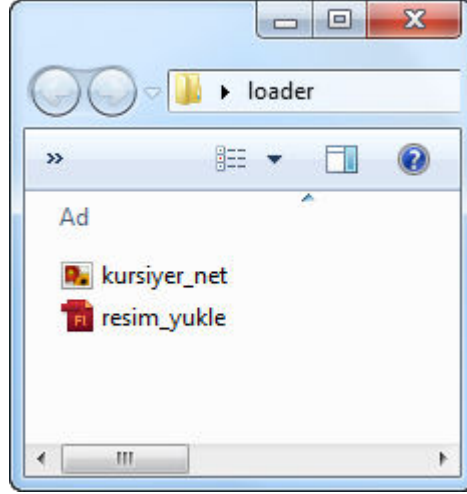
```
1 import flash.display.Loader;
2 var yukleyici:Loader=new Loader();
```

- Dosyayı *loader* nesnesine yükleyecek *istek* isimindeki *URLRequest* komutunu çalıştırın. *URLRequest* dosyanın yerini belirten string ifade yazılmasını bekler. Parantez içerisine dosyanın yolu ve ismi yazılmalıdır. Hazırlanan *swf* dosyası ile yüklenecek dosya aynı klasör içerisindeyse sadece dosyanın ismi yazılmalıdır. Dosya ismi yazılırken Türkçe karakterler kullanılmamalıdır.

```
1 import flash.display.Loader;
2 import flash.net.URLRequest;
3 var yukleyici:Loader=new Loader();
4 var istek:URLRequest=new URLRequest(
5     URLRequest(url:String=null))
```

*Kursiyer\_net.jpg* dosyasını çalışma sahnemizin ortasına ekleyen kodları yazmak için;

- Masaüstünde *loader* isimli bir klasör açıp, actionscript 3.0 dosyasını içerisine kaydedin. *Kursiyer\_net.png* dosyasını *loader* klasörünün içerisine kaydedin.



Şekil 5.1: Loader klasörü

- *Loader* ve *URLRequest* nesnelarını eylemler panelinde oluşturun.

```
1 import flash.display.Loader;
2 import flash.net.URLRequest;
3 var yukleyici:Loader=new Loader();
4 var istek:URLRequest=new URLRequest("\kursiyer_net.png");
```

- Dosyamızı *yukleyici*'nin içerisine yüklüyoruz ve *yukleyici*'yi sahneye ekleyin.

```
6 yukleyici.load(istek);
7 addChild(yukleyici);
```

- Sahneye eklediğimiz *yukleyici*'nin x ve y değerlerini sahnenin yükseklik ve genişlik değerlerine göre belirleyip sahnenin ortasına ekleyin.

```
10 yukleyici.x = stage.stageWidth/2;
11 yukleyici.y = stage.stageHeight/2;
```



Şekil 5.2: Sahneye yüklenen resim dosyası

*stage.stageWidth* sahnenin genişliğini, *stage.stageHeight* sahnenin yüksekliğini verir. Animasyon ( swf ) dosyalarını eklemek resim dosyaları eklemekten farklıdır.

## 5.2. Harici Sesleri Yükleme

MP3 formatındaki ses dosyalarını swf içerisinde iki şekilde kullanılabilir. Bunlardan birincisi ses dosyasını swf dosyasının içerisine dâhil etmek, diğeri de swf dosyası ile ses dosyasını birbirine bağlamaktır. Ses dosyasını dâhil etme işlemi küçük ses dosyaları için kolay ve hızlı olsa da büyük dosyalar için kullanışlı değildir. Uygulamanın yüklenmesini ve çalıştırılmasını geciktirecektir. Hızın çok önemli olduğu internet sayfalarında veya internet uygulamalarında bu şekilde bir yapı çok gereksiz olacaktır.

Diğer yöntem ses dosyasının yolunu göstererek kullanılacağı zaman dosyayı dışarıdan çağırması ve çalması şeklinde olur. Birçok mp3 dosyasının kullanılacağı bir player için kullanıcının istediği şarkıyı çalan bir sistem, MP3 dosyasının oynatılması sırasında diğer dosyaları yüklemeye çalışmayacağı için hızlı ve kullanışlı olacaktır.

Sesleri çalışma sahnesine dâhil etmek görüntü dosyalarını yüklemeye çok benzerdir. *URLRequest* nesnesi ve *loader* nesnesi yerine *sound* nesnesi kullanılmaktadır.

*Muzik* klasörünün içerisindeki *sarki.mp3* dosyasını çalışma sahnesine eklemek için;

- Yeni bir çalışma sahnesi açıp *muzik* klasörünün yanına *ses fla* olacak şekilde kaydedin.
- *URLRequest* ile ses dosyamızın adresini *Lnk* kelimesine atayalım.

```
1 import flash.net.URLRequest;  
2 import flash.media.Sound;  
3 var Lnk:URLRequest=new URLRequest("muzik/sarki.mp3");
```

- *Sarki* isminde *Sound* nesnesi oluşturun.

```
5 var sarki:Sound=new Sound();
```

- *Lnk* değişkeninin tuttuğu değeri *sarki* nesnesine yükleyin.

```
7 sarki.load(Lnk);
```

- Ses dosyasının çalması için gereken komutu yazıp test için *Ctrl+Enter* tuşlarını kullanın.

```
9 sarki.play();
```

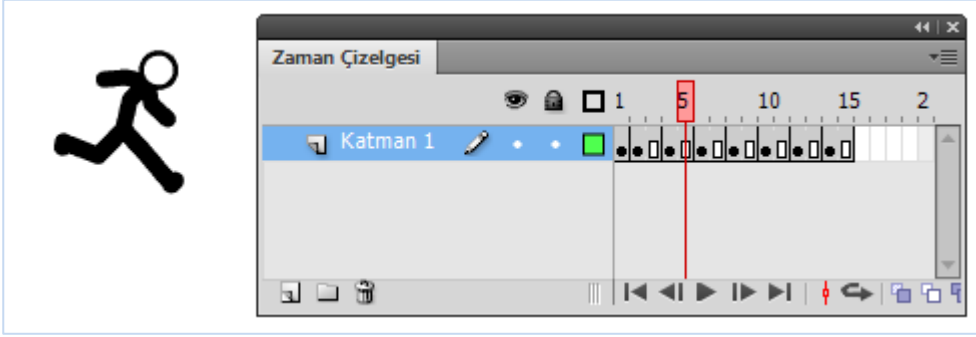
## 5.3. Ön Yükleme (Preloading) Oluşturmak

Dışarıdan yüklenen ses ya da görüntü dosyaları çok büyük boyutlarda olabilir. Hazırlanan film klipleri internette yayınlandığında kullanıcıların dosyalar yüklenene kadar beklemesi gerekir. Bu gibi durumlarda sayfaya ön yükleme koymak kullanıcının sayfanın yüklendiğine dair bilgilendirilmesi için çok önemlidir. Aksi takdirde internet sayfasını açmaya çalışan kullanıcı sayfanın hatalı olduğunu düşünebilir.

Ön yükleme tamamıyla geliştiricilerin hayal gücüne kalmıştır. Yükleme durumu yüzde ile gösterilebileceği gibi yükleme durumu için bir animasyon da hazırlanabilir.

Yükleme durumunu hem animasyon hem de yüzde olarak gösteren ön yükleyici yapmak için;

- Animasyon yazılımımızı açarak yeni bir çalışma dosyası açın.
- Animasyon teknikleri ile koşan adam animasyonu hazırlayınız.



Şekil 5.2: Koşan adam animasyonu

- Dosya menüsünden *Dışa Aktar* → *Filmi Dışarıya Aktar* (*Ctrl+Alt+Shift+S*) komutu ile film klipi haline getirin dosyanın ismi *kosanAdam.swf* olacak şekilde kaydedin.
- Yeni bir çalışma sayfası açarak eylemler panelini çalıştırın. Sahneye koşan adam animasyonunu dâhil edin.

```
6 var yukleyici:Loader=new Loader();
7 var istek:URLRequest=new URLRequest("kosanAdam.swf");
8 yukleyici.load(istek);
9 addChild(yukleyici);
10 yukleyici.scaleX=.5;
11 yukleyici.scaleY=.5;
```

- Animasyon sayfada büyük olursa *ScaleX* ve *ScaleY* değerlerini değiştirebilirsiniz.
- Sahneyi yüzde olarak değeri gösterecek bir metin kutusu ekleyin.

```
13 var yazi:TextField=new TextField();
14 addChild(yazi);
15 yazi.y=50;
```

- Metin kutusu standart olarak 0,0 noktasına eklenecektir. 50 px aşağıya kaydırıp hazırladığınız animasyonla aynı düzleme getirin.
- Sayfada gösterilecek fotoğrafı sahneye ekleyin. Yüksek çözünürlükte boyut olarak büyük bir fotoğraf simülasyon için çok uygun olacaktır.

```
17 var resim:Loader=new Loader();
18 var ftgrf:URLRequest=new URLRequest("fotograf.jpg");
19 resim.load(ftgrf);
```

Önyükleyici, fotoğraf tam olarak film klibi içerisine alındığında göstermek için yapılmaktadır. `addChild` ile sahneye hem yüklenirse önyükleyici boşuna yapılmış olacaktır.

- Sahneye, sahnenin çalışma durumunu gözleyen bir olay dinleyicisi ekleyin..Sahne çalıştığında `calistir` fonksiyonunu tetikleyecek şekilde kodları yazın.

```
25  addEventListener(Event.ENTER_FRAME,calistir)
```

- `Calistir` fonksiyonu, resim tam olarak yüklendiğinde, resmin görünmesini sağlayacaktır. Resim dosyasının yüklenmesini yüzde olarak görmek için, dosyanın toplam boyutunun ve sahneye yüklenen boyutunun bilinmesi gerekiyor. Basit bir yüzde hesabı ile sayısal değerlere ulaşılabilir.

```
27  function calistir(e:Event):void
28  {
29  var tumYuklenecek:Number=resim.contentLoaderInfo.bytesTotal;
```

- Resim dosyasının toplam boyutunu içerik *yükleyici bilgileri* (`contentLoaderInfo`) sınıfından *toplam byte* (`bytesTotal`) metodu ile alınabilir. Bu bilgi `tumYuklenecek` isimli bir değişken içinde tutulmaktadır.

```
31  var yuklenen:Number=resim.contentLoaderInfo.bytesLoaded;
32  var yuzde:Number=Math.floor(yuklenen/tumYuklenecek*100);
```

- Yüklenen bilgisini, *içerik yükleyici bilgileri* (`contentLoaderInfo`) sınıfından yüklenen *byte* (`bytesLoaded`) metodu ile alınabilir.
- En son olarak `yuzde` isminde bir değişken oluşturulup *yuvarla* ( $\frac{\text{yuklenen bilgi}}{\text{toplam bilgi}} * 100$ ) formülü ile `yuzde` değerini hesaplatıyoruz.
- Hesaplanan yüzde değerini metin kutusu içerisine aktarıp ekranda görünmesini sağlayın.

```
34  yazi.text=String(yuzde);
```

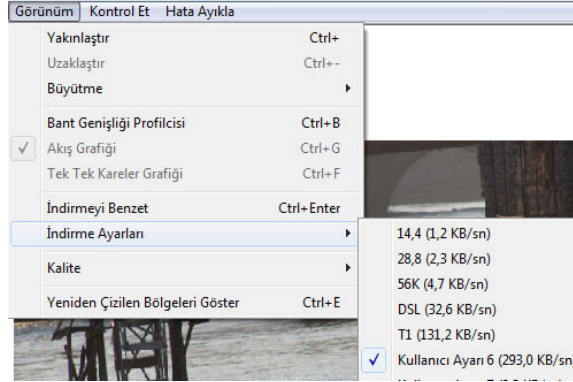
- Yükleme işleminin en önemli noktası *if kontrol* deyimi ile şartın yazılmasıdır.

```
36  if(tumYuklenecek==yuklenen)
37  {
38  addChild(resim);
39  resim.scaleX=.1
40  resim.scaleY=.1
41  resim.x=50;
42  resim.y=100;
43
44  }else
45  {
46  yukleyici.x++;
47  yazi.x++;
48  }
```

- Yukarıda gösterilen şart kodları ile yuklenen bilgi ile toplam bilgi birbirine eşitse ekrana resim nesnesini ekle değilse animasyonu oynat ifadesi yer almaktadır.
- Resim dosyası çok büyük ve ekrana sığmadığı için 39. Ve 40. satırda resmin resmin boyutu 1/10 küçültülmüştür. Sonraki satırlarda x ve y koordinatları değiştirilerek resmin konumu belirlenmiştir. Şart sağlanmadığında ise animasyon ve metin kutusu x ekseninde hareket etmesi sağlanmıştır. Bu sayede koşan adam animasyonunun çalışması sağlanmıştır.
- *CTRL+Enter* tuşları ile sahneyi test edin. Animasyonçalışmadan ve yüzde değeri görülmeden resim dosyası ekran gelmektedir. Bunun sebebi yüklenen dosya, harddisk erişim hızı ile yüklenmekte olmasıdır.

İnternet ortamını taklit etmek için ;

- Hazırladığınız yükleyici örneğini *Ctrl+Enter* ile çalıştırın.
- Çalışan uygulama ekranında, indirme ayarlarından *İnternet* bağlantı hızlarından uygun olan birisi seçilir.



Şekil 5.4: *İnternet* simülasyonu ayarı

- *İndirmeyi benzet* (*Ctrl+Enter*) kısa yolu ile internet ortamı simüle edilir.

## 5.4. Sesleri Kontrol Etme

*İnternet* sayfalarında, uygulamalarda veya oyunlarda ses kullanımı çalışmaya görselliğin yanında farklı bir etki katacaktır. Sahneye eklenen ses dosyalarını istenilen yerden oynatabilir, istenilen şekilde durdurabilir, hatta farklı sesler aynı anda çalınabilir.

Ses dosyasını çalmak için ;

- *URLRequest* ile ses dosyasının bilgilerini *lnk* değişkenine aktarın.

```

1 import flash.net.URLRequest;
2 import flash.media.Sound;
3 var lnk:URLRequest=new URLRequest("muzik/sarki.mp3");

```

- *sarki* isminde *Sound* nesnesi oluşturun.

```
5 var sarki:Sound=new Sound();
```

- *Lnk* değişkeninin tuttuğu değeri *sarki* nesnemize yükleyin ve *play* komutu ile şarkının çalmasını sağlayın.

```
7 sarki.load(Lnk);
```

```
9 sarki.play();
```

- Şarkının istenilen yerden çalması istenirse *play* komutu içerisine milisaniye cinsinden değeri yazılmalıdır. Örneğin 3 saniyeden çalınması isteniyorsa kullanımı aşağıdaki gibi olacaktır.

```
1 sarki.play(3000);
```

- Şarkı bittikten sonra tekrar çalması isteniyorsa tekrar değeri başlama değerinden sonra girilmelidir. Örneğin şarkının 5 defa tekrar etmesi istenirse kullanımı aşağıdaki gibi olacaktır.

```
1 sarki.play(3000,5);
```

- Burada 3000 başlangıç zamanı (*startTime*) değeri, 5 ise tekrar (*loops*) parametresidir.

Ses dosyaları uzun sürelere sahip olabilir. Belli bir yerde bekletip kalman yerden devam ettirilmek istenebilir. Action script içerisinde *sound* nesnesi sadece durdurulabilir. Kaldığı yerden başlatmak için çalışacak direk bir komut yoktur. *soundChannel* sınıfının *position* özelliği sayesinde ses dosyasının hangi saniyede olduğunu öğrenilebilir ve bu noktadan çalmaya başlatılabilir.

Ses dosyasını bekletme ve kaldığı yerden oynatmak için;

- Öncelikle ses dosyamızı *ses* isimli değişkenimize yükleyin.

```
1 var ses:Sound = new Sound(new URLRequest("sarki.mp3"));
```

- Yeni bir kanal oluşturup ses nesnemizi bu kanala atın. Kanallar birden fazla ses dosyası ile çalışmaya olanak sağlar. Ses dosyasının özelliklerine *soundChannel* sınıfı içerisinden ulaşılabilir. Bu yüzden kanallarla çalışmak ekstra kontrol sağlayacaktır.

```
3 var channel:SoundChannel = ses.play();
```

- Ses dosyasının çalarken hangi saniyede olduğu bilgisini *nerede* isimli değişkene atın. *Stop* komutu ile ses dosyasının çalışması durdurulmaktadır.

```
6 var nerede:int = channel.position;
```

```
7 channel.stop();
```

- Artık *nerede* değişkeni çalınan ses dosyasının çalma zamanını *milisaniye* cinsinden tutmaktadır. *nerede* değişkeni *play* özelliğine *startTime* değeri olarak verildiğinde ses dosyası kaldığı yerde devam edecek şekilde çalmaya başlayacaktır.



```
9 channel = snd.play(nerede);
```

## 5.5. Video Dosyalarını Yükleme

Animasyon yazılımı içerisine ses ve animasyon dosyaları eklenmesi gibi video dosyalarını da eklenebilir. Bu işlem için *NetConnection* nesnesinin tanımlanması gerekiyor.

Sahneye 320\*240 px boyutlarında, *goruntu.flv* dosyasını eklemek istersek

- *baglanti* isminde *NetConnection* nesnesi oluşturun.

```
1 var baglanti:NetConnection = new NetConnection();  
2 baglanti.connect(null);
```

Media Server dışından herhangi bir yerden video dosyası alıncaksa *connect* parametresi olarak *null* verilmelidir.

- *NetConnection* nesnesi parametre olarak alan bir *NetStream* nesnesi oluşturup yüklenecek video nesnesin belirtin.

```
4 var ns:NetStream = new NetStream(baglanti);  
5 ns.play("goruntu.flv");
```

- Yeni bir video nesnesi oluşturup video sınıfının *attachNetStream()* yöntemi ile *NetStream* nesnesini yükleyin. Daha sonra *addChild()* yöntemini kullanarak video nesnesini sahneye ekleyin.

```
8 var vdo:Video = new Video(320,240);  
9 vdo.attachNetStream(ns);  
10 addChild(vdo);
```

- CTRL + Enter ile sahneyi test edin. Video dosyası otomatik olarak çalışmaya başlayacaktır.

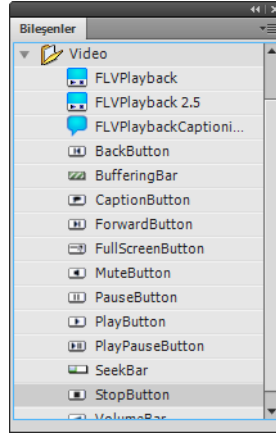
## 5.6. Video Oynatımını Kontrol Etme

Video dosyasını sahneye ekledikten sonra kontrol edilmesi çok kolaydır. Kontrol için gerekli tüm metodlar *netStream* içerisinde hazır bir şekilde bulunmaktadır.

Eklenen videoyu oynatma, başa sarma ve bekletme olayları hızlı bir şekilde hazırlanabilir.

Sahneye eklenen *vdo* isimli video dosyasını kontrol etmek için;

- Pencere menüsünden bileşenleri seçip *play stop*, *pause* butonlarını sahneye sürükleyin.



Şekil 5.5: Bileşenler penceresi



Şekil 5.6: Sahnedeki butonlar

- Sırasıyla düğmelerin örnek adına *playBtn*, *pauseBtn*, *stopBtn* kelimelerini yazın.
- Düğmelere tıklama olayını izleyen olay dinleyicileri ekleyin.

```
13 pauseBtn.addEventListener(MouseEvent.CLICK, beklet);  
14 playBtn.addEventListener(MouseEvent.CLICK, oynat);  
15 stopBtn.addEventListener(MouseEvent.CLICK, dur);
```

- Dinleyiciler *beklet*, *oynat* ve *dur* fonksiyonlarını çalıştırmaktadırlar. Dur fonksiyonu için herhangi bir *stop()* yöntemi yoktur. Bir akışı durdurmak için, oynatmayı duraklatma ve video akışının başını arama gerekir. Seek yönetimi 0 parametresi ile akışı en başa sarar. *play()* yöntemi oynatmayı yeniden başlatır, video dosyalarının yüklenmesi için kullanılır.

```
17 function beklet(event:MouseEvent):void  
18 {  
19     ns.pause();  
20 }  
21 function oynat(event:MouseEvent):void  
22 {  
23     ns.resume();  
24 }  
25 function dur(event:MouseEvent):void  
26 {  
27     ns.pause();  
28     ns.seek(0);  
29 }
```

- *Ctrl+Enter* ile sahneyi test edin.

## UYGULAMA FAALİYETİ

Sahneye dışarıdan dâhil ettiğimiz bir resim yüklendiğinde uyarı.mp3 ses dosyasını çalan bir uygulama hazırlayınız.

İşlem Basamakları	Öneriler
➤ Sahneye eklenecek resim dosyası için loader ve XMLHttpRequest nesnesini tanımlayın.	➤ XMLHttpRequest nesnesi tanımlanırken resim dosyasının adresi yazılmalıdır.
➤ Resim dosyasını sahneye dahil edin.	➤ addChild metodunu kullanabilirsiniz.
➤ Resim dosyamızın tam olarak yüklendiğini kontrol için olay dinleyicisi ekleyin.	➤ resim.contentLoaderInfo.addEventListener ➤ (Event.Complete,yuklendi) satırını kullanabilirsiniz.
➤ yuklendi fonksiyonu tetiklendiğinde uyarı.mp3 dosyasını çalsın.	➤ Uyarı fonksiyonu içerisinde ses nesnesini tanımlayıp sahneye dahil edebilirsiniz.
➤ Uygulamayı test edin.	➤ Ctrl+Enter tuşları ile test işlemi yapabilirsiniz. Simülasyon ayarlarını yapıp test işlemi tekrarlayabilirsiniz.

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Sahneye resim ve animasyon yükleyebildiniz mi?		
2. Harici ses dosyalarını yükleyebildiniz mi?		
3. Önyükleme hazırlayabildiniz mi?		
4. Sesleri çaldırıp, durdurabildiniz mi?		
5. Video dosyalarını kontrol edebildiniz mi?		
6. Video dosyalarını kaldığı yerden başlatabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) Resim yüklemek için *Loader* ve *URLRequest* kullanılır.
2. ( ) *URLRequest* içerisinde dosya bilgileri bulunur.
3. ( ) Dosya bilgilerini *loader* nesnesine *load* komutu ile yüklenir.
4. ( ) *Preloading* yüksek boyutlu dosyaları sayfaya dâhil ederken kullanıcının bilgilendirilmesini sağlar.
5. ( ) Ses dosyaları istenilen yerden animasyon içinde yeniden başlatılamaz.
6. ( ) *SoundChannel* nesnesi ile birden fazla ses dosyasını çalışma sayfamıza dâhil edilebilir.
7. ( ) Video dosyaları sayfaya dâhil edilirken *NetConnection* nesnesi kullanılır.
8. ( ) *Connect* parametresi daima *null* olmak zorundadır.
9. ( ) *Pause* ile video dosyaları kaldığı yerden oynatılabilir.
10. ( ) *Resume* ile video dosyaları oynatmaya ara verebilir.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-6

## AMAÇ

Animasyon yazılımını kullanarak etkileşimli gelişmiş uygulamalar hazırlayabileceksiniz.

## ARAŞTIRMA

- *İnternet* sitelerindeki sürükle bırak ile yapılmış menüleri, oyunlardaki nesnelerin birbiri ile temasını gözlemleyerek araştırın.

## 6. GELİŞMİŞ ETKİLEŞİM

### 6.1. Sürükle –Bırak

Hazırlanan uygulamalarda çalışma sahnesindeki nesnelere fare yardımıyla sürüklenip yerleri değiştirilebilir. İnternet sayfalarımıza hareketli menüler, fotoğraf galerileri yerleştirebilir, fare etkileşimli oyunlar hazırlayabiliriz.

Nesne fare ile tutulduğunda sürüklemeye başlanmış, bırakıldığında ise bırakma işlemi yapılmış olur.

Sahnemizde bulunan 50\*50 px lik bir kareye sürüklemeye bırakma işlemi yapmak için;

- Yeni bir çalışma sahnesi açarak, eylemler panelini çalıştırın. Ekranı bir kare oluşturacak kodları yazın.

```
1 import flash.display.MovieClip;
2 var geometrik:MovieClip=new MovieClip();
3 geometrik.graphics.lineStyle(2,0x000,1);
4 geometrik.graphics.beginFill(0x000,1);
5 geometrik.graphics.drawRect(100,100,50,50);
6 addChild(geometrik);
```

- Sürüklemeye ve bırakma işlemi için nesneye olay dinleyicisi ekleyin. Dinleyici nesnenin sürüklenme olayını gözleyecektir.

```
8 geometrik.addEventListener(MouseEvent.CLICK, surukle);
```

- *Mouse\_click* olayı ile dinleyici *surukle* fonksiyonunu çalıştıracak. *Surukle* fonksiyonu *startDrag* komutu ile geometrik isimli nesnemizi sürüklemeye başlayacak.

```
10 function surukle(e:MouseEvent) {
11
12     geometrik.startDrag();
13 }
```

- *Mouse up* olayını gözleyen bir dinleyici daha yerleştirip bırakılma olayını gerçekleştiren bir fonksiyon yazın.

```
15 geometrik.addEventListener(MouseEvent.CLICK, birak);
16
17 function birak(e:MouseEvent) {
18
19     geometrik.stopDrag();
20 }
```

- Sahneyi *CTRL+Enter* ile test edin.

Başka bir uygulamada basit menüyü tasarlayıp oluşturulan düğmeye link verilsin ve menü sürüklenip istenilen yere taşınabilsin.



Şekil 6.1: Menü

- Çalışma sahneleyicisini açıp menü için ekrana dikdörtgen şeklinizi ekleyin.

```
1 import flash.display.MovieClip;
2 import flash.text.TextField;
3
4 var menu:MovieClip=new MovieClip();
5 menu.graphics.lineStyle(3,0x556644,1);
6 menu.graphics.beginFill(0x006699,1);
7 menu.graphics.drawRect(100,100,95,200);
8 addChild(menu);
```

- Menü içerisindeki düğme şeklini hazırlayın. Şekli *addChild* ile menü içerisine ekleyin. Sahneye eklenirse düğme menü ile birlikte hareket etmeyecektir.

```

10 var meb:MovieClip=new MovieClip();
11 meb.graphics.beginFill(0x00ccff,1);
12 meb.graphics.drawRect(110,110,75,20);
13 menu.addChild(meb);

```

- Düğmenin içerisindeki metin alanını oluşturup *M.E.B.* yazısını yazdırın. *selectable=false* özelliği sayesinde yazıyı seçilemez hale getirin.

```

15 var mebYazi:TextField=new TextField();
16 mebYazi.text="M.E.B."
17 mebYazi.x=130;
18 mebYazi.y=110;
19 mebYazi.selectable=false;
20 meb.addChild(mebYazi);

```

- Düğme için bir bağlantı oluşturun.

```

23 meb.addEventListener(MouseEvent.CLICK,lnk)
24 function lnk(e:MouseEvent)
25 {
26 var istek:URLRequest=new URLRequest("http://www.meb.gov.tr")
27     navigateToURL(istek,"_blank");
28 }
29

```

- Düğmeye olay dinleyicisi ekleyin. Tıklama olayı gerçekleştiğinde *lnk* isimli fonksiyonu tetiklenecektir. Fonksiyon içerisinde *istek* isimli bir *URLRequest* nesnesi tanımlayıp bağlantı adresini değer olarak atayın. *navigateToURL* ile bağlanılacak sayfanın açılmasını sağlayın.
- Hazırlanan şekle yer değiştirme özelliği vermek için *startDrag* ve *stopDrag* komutları kullanılmalıdır. Menü nesnesine bir olay dinleyicisi ekleyip fonksiyonu yazın.

```

30 kare.addEventListener(MouseEvent.MOUSE_DOWN, surukle);
31
32 function surukle(e:MouseEvent) {
33
34     kare.startDrag();
35
36 }

```

- Bu fonksiyonda *kare.startDrag()*; komutunun altına *kare.alpha=.5* komutu eklenerek menü taşınırken transparan yapılabilir ve güzel bir görüntü elde edilebilir.
- Bırakma eylemi için yine olay dinleyicisi ekleyip fonksiyonu yazın.



```

38 kare.addEventListener(MouseEvent.CLICK, birak);
39
40 function birak(e:MouseEvent) {
41
42     kare.stopDrag();
43
44 }

```

- Bırakırken menünün *alpha* değerini 1 yapılırsa yeniden eski haline dönecektir.
- **CTRL+Enter** tuşları ile sahneyi test edin ve çalışmasını gözlemleyin.

## 6.2. Çarpışmalar

Nesnelerin birbiri ile temas durumlarının bilinmesi etkileşimi arttırmak için gereklidir. Örneğin basit bir puzzle oyununda yanlış yere götürülen parçanın tekrar puzzle dışına çıkması istenebilir ya da karakterin sahnedeki duvarlar ile teması sonucu durması istenebilir. Kısacası bir nesnenin başka bir nesne ile teması gözlenip temas durumuna göre uygulamanın akışını değiştirilir.

*hitTestObject( )* ve *hitTestPoint( )* fonksiyonları nesnenin nesneye ya da noktanın nesneye olan temasını kontrol eder.

*hitTestObject( )* fonksiyonu *nesne1.hitTestObject(nesne2)* şeklinde; *hitTestPoint( )* fonksiyonu ise *nesne1.hitTestPoint(x,y,true/false)* şeklinde kullanılır.

Sahnedeki iki nesnenin birbiri ile temasını *hitTestObject* ile kontrol eden bir uygulama için;

- Yeni bir çalışma sayfası açarak eylemler panelini çalıştırın.
- Sahneye bir kare ve bir daire nesnesi oluşturun.

```

1 import flash.display.MovieClip;
2 import flash.events.Event;
3
4 var kare:MovieClip=new MovieClip();
5 kare.graphics.lineStyle(3,0x556644,1);
6 kare.graphics.beginFill(0x006699,1);
7 kare.graphics.drawRect(100,100,50,50);
8 addChild(kare);

```

- Kare nesnesini *drawRect* metodu ile oluşturun ve *addChild* ile sahneye ekleyin.

```

10 var daire:MovieClip=new MovieClip();
11 daire.graphics.lineStyle(3,0x556644,1);
12 daire.graphics.beginFill(0x006699,1);
13 daire.graphics.drawCircle(50,100,25);
14 addChild(daire);

```

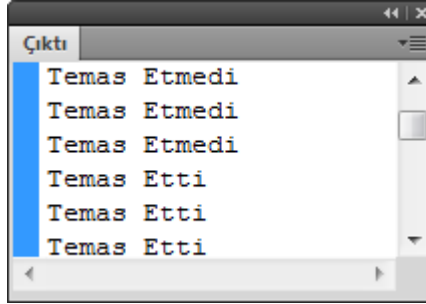
- Sahnenin çalışma durumunu gözleyen bir olay dinleyicisi ekleyin. Sahne çalıştığında *degdimi* isimli fonksiyonu çalıştırın.

```
16 stage.addEventListener(Event.ENTER_FRAME, degdimi);
```

- *degdimi* fonksiyonu nesnelerin birbiri ile temasını kontrol ederek kullanıcıyı bilgilendirsin.

```
18 function degdimi(e:Event)
19 {
20     if(kare.hitTestObject(daire))
21     {
22         trace("Temas Etti");
23     }else
24     {trace("Temas Etmedi");}
25 }
```

- Nesnelerin birbiri ile teması halinde çıkış ekranında *Temas Etti*, temas etmediği zamanlarda *Temas Etmedi* şeklinde uyarıyı sahne çalışma hızına göre sürekli olarak verecektir. Sahne olayını gözleyen dinleyicimiz *frameRate* değeri kadar *degdimi* fonksiyonunu tetikleyecektir.



Şekil 6.2: Çıkış ekranı

- Nesnelerin temas etmesi için hareket etmeleri gerekmektedir. Hareket etmeleri için sürükle bırak tekniği kullanın.
- Kare nesnesi için *startDrag* ve *stopDrag* komutlarını yazın.

```

30 kare.addEventListener(MouseEvent.CLICK, surukle);
31
32 function surukle(e:MouseEvent) {
33
34     kare.startDrag();
35
36 }
37
38 kare.addEventListener(MouseEvent.CLICK, birak);
39
40 function birak(e:MouseEvent) {
41
42     kare.stopDrag();
43
44 }

```

Sahnedeki fare ile nesnenin temasını *hitTestPoint* ile kontrol eden bir uygulama için;

- Yeni bir çalışma sayfası açarak eylemler panelini çalıştırın.
- Sahneye 400 px genişliğinde 80 px yüksekliğinde 404, 0 noktasından itibaren bir elips oluşturun.

```

1 import flash.display.MovieClip;
2 import flash.events.Event;
3
4 var elips:MovieClip=new MovieClip();
5 elips.graphics.beginFill(0x6699,1);
6 elips.graphics.drawEllipse(40,40,400,80);
7 addChild(elips);

```

- Daire isminde bir nesne oluşturun ama sahneye eklemeyin.

```

12 var daire:MovieClip=new MovieClip();
13 daire.graphics.lineStyle(2,0xff,1);
14 daire.graphics.drawCircle(0,0,5)

```

- Sahneye bir olay dinleyicisi ekleyin. Bu dinleyici sahne çalıştığında kontrol fonksiyonunu çalıştırmalıdır.

```

17 stage.addEventListener(Event.ENTER_FRAME,kontrol)
18 function kontrol(e:Event)
19 {
20     if(elips.hitTestPoint(mouseX,mouseY,false))
21     {
22         addChild(daire);
23         daire.x=mouseX;
24         daire.y=mouseY;
25     }
26 }

```

- Yukarıdaki kodlar incelenecek olursa;
  - 20 nolu satırda *kontrol* fonksiyonu fare işaretçisinin elipsin sınırına temasını kontrol etmektedir.
  - Temas oluşursa 22 nolu satırda daire sahneye eklenecektir.

- 23 ve 24 nolu satırlarda dairenin koordinatlarına imleci koordinatları atanmıştır ve imlecin etrafında daire oluşmuştur.
- 20 nolu satırdaki 3. Parametre *true* olursa noktanın nesneye teması, *false* olursa noktanın sınıra teması kontrol edilmektedir.



Şekil 6.3: Nesne-nokta teması

### 6.3. Nesneleri Sahne Üzerinde Rastgele Yerleştirmek

Sahneye *for döngüsü* kullanılarak istenilen sayıda nesne eklenebilir. Bu nesnelere *random* fonksiyonu ile rastgele yerlere yerleştirilebilir.

Sahneye eklenen elli adet daireyi rastgele yerlere yerleştiren, görünürlük, boyut ve derinlik bilgilerini rastgele belirleyen kodları yazmak için;

- Yeni bir çalışma sahnesi açın ve eylemler panelini çalıştırın.
- Sahneye 50 nesne için bir *for döngüsü* oluşturup nesnelere rastgele koordinatlarını belirleyecek değişkenleri tanımlayın.

```
1 for (var i:Number=0;i<=50;i++)
2 {
3
4 var xKoordinat:Number=Math.random()*200;
5 var yKoordinat:Number=Math.random()*200;
6 var yariCap:Number=Math.random()*20;
```

- x ve y koordinatlar 0-200 aralığına kadar herhangi bir sayı üretecek, *yariCap* ise 0-20 arasında bir sayı üretecek şekilde oluşturulmuştur.
- *drawCircle* parametresi olarak rastgele üretilen sayıları tutan değişkenler vererek, daireyi sahneye ekleyin.

```
8 var daire:MovieClip=new MovieClip();
9 daire.graphics.beginFill(0x006699,1);
10 daire.graphics.drawCircle(xKoordinat,yKoordinat,yariCap)
11 addChild(daire);
```

- Dairenin özelliklerini değiştirerek rastgele büyütüp z ekseninde 180 derece rastgele döndürün.

```
14 daire.scaleX=Math.random()*2;
15 daire.scaleY=Math.random()*2;
16 daire.z=Math.random()*180;
17 }
```



Şekil 6.4: Rastgele oluşan ve yerleşen daireler

## 6.4. Nesneleri Sahne Üzerinde Düzenli Yerleştirme

İnternet sayfalarında çok sayıda nesnenin, ekranda düzenli şekilde sıralanması istenebilir. Örneğin, çok sayıdaki resmi tek tek yerleştirmek yerine bir seferde düzenlemek mümkündür.

Sahneye eklediğimiz 100 adet kare şeklini onluk gruplarla sahneye dizmek için;

- Eylemler panelini açarak for döngüsü oluşturun. Döngü içerisinde 100 adet kare nesnesi oluşturup sahneye eklenecektir.

```
1 for (var i:Number=0; i<=99;i++)
2 {
3     var kare:MovieClip=new MovieClip();
4     kare.graphics.beginFill(0x006699,1);
5     kare.graphics.drawRect(0,0,25,25);
6     addChild(kare);
```

- *CTRL+Enter* ile test edildiğinde sahnede 0,0 noktasında 25\*25 px boyutlarında bir adet kare görülecektir. Sahnede 100 adet kare oluşmuş fakat hepsi üst üste binmiş durumdadır. Karelerin x ve y koordinatlarını değiştirerek istenilen şekilde kareler Sahneye yerleştirilebilir.

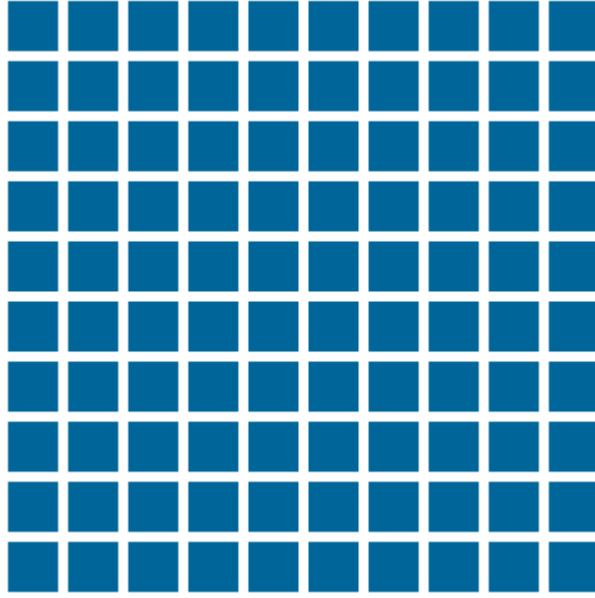
- X eksenindeki koordinatı 10'luk gruplar halinde olacağı için mod olarak karenin genişliği kadar ekrana yan yana sıralanacaktır. 5 ekleyerek nesnelere arasındaki boşluğu ayarlanmıştır.

```
8 kare.x=i%10*(kare.width+5);
```

- Y eksenindeki koordinatları ise bölme işlemi ile yapıyoruz. *i* değişkenini 10'a bölerek kare nesnesinin yüksekliği ile çarpıyoruz. Buradaki 5 değerini dikey boşluk bırakmak için ekliyoruz.

```
12 kare.y=Math.floor(i/10)*(kare.height+5); }
```

- Uygulamamızı test ettiğimizde aşağıdaki ekran görüntüsü ile karşılaşırız.



**Şekil 6.5: Düzenli şekilde yerleşen kare nesneleri**

## UYGULAMA FAALİYETİ

Kontrol deyimleri ve *hitTestObject* kullanarak kaçan-kovalanır oyunu hazırlayalım.

İşlem Basamakları	Öneriler
➤ Yeni bir belge açınız.	➤ Karşılama ekranını veya Dosya > Yeni (Ctrl + N) komutunu kullanabilirsiniz.
➤ Eylemler panelini açarak bir adet kare ve bir adet daire nesnesi oluşturun. Kare nesnesi daireyi kovalayacak daire ise kaçacaktır.	➤ <code>drawRect</code> ve <code>drawCircle</code> metotlarını kullanabilirsiniz.
➤ kare nesnesini hareket ettirmek için sahneye klavye olaylarını gözleyen bir klavye dinleyicisi yerleştirin. ➤ Dinleyici hareket fonksiyonunu çalıştırın.	➤ Klavye dinleyicileri klavyeye basılan tuşların asci değerlerini bulur.
➤ 4 adet if deyimi kullanarak yukarı aşağı sağa ve sola basılan tuşlar için kare nesnemizi sahnede 1 px olacak şekilde hareket etmesini sağlayın.	➤ Sağa ve sola hareket için <code>kare.x++</code> , <code>kare.x--</code> , yukarı ve aşağı için <code>kare.y++</code> ve <code>kare.y--</code> kodlarını kullanabilirsiniz.
➤ if kullanarak kare nesneyi daire ile temas ettiğinde daire nesnesinin başka bir noktaya hareket etmesini sağlayın.	➤ Temas için <code>HitTestPoint</code> kullanılabilir. Dairenin rastgele hareketi için <code>daire.x=Math.random()*500</code> <code>daire.y=Math.random()*500</code> kullanabilirsiniz.
➤ Her temas ettiğinde tutulan skor isimli değişkenin değerini 1 arttırın	➤ <code>Skor++</code> ya da <code>skor=skor+1</code> ile gerçekleştirebilirsiniz.
➤ Skor değişkenini sahnede göstereyin	➤ <code>Skortut</code> isimli <code>textfield</code> elemanı tanımlayabilirsiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) Nesneler sürkleme işleminden sonra bırakılamaz.
2. ( ) Sahnedeki nesnelerin birbiri ile temasını kontrol edilebilir.
3. ( ) Sahnede Nesne-nesne ya da nesne- nokta teması kontrolleri gerçekleştirilebilir.
4. ( ) *random* fonksiyonu ile nesnelerin x ve y koordinatları rastgele değerler verilebilir
5. ( ) Birden fazla nesne aynı anda sahneye düzgün bir şekilde yerleştirilemez.
6. ( ) *startDrag* ile sürüklenme işlemini tamamlanır.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.



# MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden hangisi Math sınıfının sabiti değildir?  
A) Ln10  
B) Ln20  
C) E  
D) Pi
2. Aşağıdakilerden hangisi Math sınıfının metodu değildir?  
A) Abs()  
B) Atan()  
C) Ceil()  
D) Rand()
3. Aşağıdakilerden hangisi metin alanı değildir?  
A) Statik metin alanı  
B) Dinamik metin alanı  
C) Giriş metin alanı  
D) Çıkış metin alanı
4. Aşağıdakilerden hangisi Metin alanı oluşturur?  
A) Textfield  
B) Text  
C) TextBox  
D) Textinput
5. Aşağıdakilerden hangisi Metin alanlarını biçimlendirmek için kullanılan nesnedir?  
A) Format  
B) TextfieldFormat  
C) textFormat  
D) textformatcolor
6. Aşağıdakilerden hangisi dizilerin uzunluğunu gösterir?  
A) Array  
B) Lenght  
C) Push  
D) Pop
7. Aşağıdakilerden hangisi dizinin son elemanını kaldırır?  
A) Array  
B) Lenght  
C) Push  
D) Pop

8. Aşağıdakilerden hangisi dizinin başındaki elemanını kaldırır?  
A) Delete  
B) shift  
C) Push  
D) Pop
9. Aşağıdakilerden hangisi 4 piksel kalınlığında 00cc00 renginde yarı saydam bir çizgi oluşturur?  
A) Graphics.linestyle(0.5,00cc00,4)  
B) Graphics.linestyle(4,00cc00,0.5)  
C) Graphics.linestyle(4,0x00cc00,0.5)  
D) Graphics.linestyle(4,00cc00,0.5)
10. Aşağıdakilerden hangisi şekle gölge efekti verir?  
A) Glow  
B) dropShadow  
C) BevelFilter  
D) Blur
11. ContentLoaderInfo sınıfında resim içeriğinin toplam büyüklüğü aşağıdakilerden hangisi ile bulunabilir?  
A) Total  
B) Byte  
C) ByteTotal  
D) TotalByte
12. Aşağıdakilerden hangisi ses dosyasını 3. Saniyeden başlatıp 3 tekrar yaptıran komuttur?  
A) Play(3,3)  
B) Play(30,3)  
C) Play(33)  
D) Play(3000,3)
13. Aşağıdakilerden hangisi sürüklenme işlemini başlatır?  
A) StartDrag();  
B) Drag()  
C) stopDrag(9)  
D) Start
14. Aşağıdakilerden hangisi nesnenin noktaya temasını kontrol eder?  
A) hitTestPoint  
B) hitTestObject  
C) hitTest  
D) TestPoint

15. Video dosyasının oynatımını bekletmek için aşağıdakilerden hangisi kullanılır?
- A) pause
  - B) resume
  - C) Play
  - D) stop

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	Yanlış
2	Yanlış
3	Doğru
4	Yanlış
5	Doğru
6	Yanlış
7	Yanlış
8	Yanlış
9	Yanlış
10	Yanlış

## ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	Yanlış
2	Yanlış
3	Yanlış
4	Doğru
5	Doğru
6	Yanlış
7	Yanlış

## ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Doğru
4	Doğru
5	Doğru
6	Doğru
7	Yanlış

### ÖĞRENME FAALİYETİ-4'ÜN CEVAP ANAHTARI

1	Doğru
2	Yanlış
3	Yanlış
4	Yanlış
5	Doğru
6	Yanlış
7	Doğru
8	Yanlış
9	Doğru
10	Yanlış

### ÖĞRENME FAALİYETİ-5'İN CEVAP ANAHTARI

1	Doğru
2	Doğru
3	Doğru
4	Doğru
5	Yanlış
6	Doğru
7	Doğru
8	Yanlış
9	Yanlış
10	Yanlış

### ÖĞRENME FAALİYETİ-6'NIN CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Doğru
4	Doğru
5	Yanlış
6	Yanlış

## MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

<b>1</b>	<b>B</b>
<b>2</b>	<b>D</b>
<b>3</b>	<b>D</b>
<b>4</b>	<b>A</b>
<b>5</b>	<b>C</b>
<b>6</b>	<b>B</b>
<b>7</b>	<b>C</b>
<b>8</b>	<b>B</b>
<b>9</b>	<b>C</b>
<b>10</b>	<b>B</b>
<b>11</b>	<b>C</b>
<b>12</b>	<b>D</b>
<b>13</b>	<b>A</b>
<b>14</b>	<b>A</b>
<b>15</b>	<b>A</b>

## KAYNAKÇA

- [http://livedocs.adobe.com/flash/9.0/main/flash\\_as3\\_programming.pdf](http://livedocs.adobe.com/flash/9.0/main/flash_as3_programming.pdf) (20-25 Mart 2012 )
- [http://help.adobe.com/en\\_US/FlashPlatform/reference/actionsript/3/index.html](http://help.adobe.com/en_US/FlashPlatform/reference/actionsript/3/index.html) (20-25 Mart 2012)