

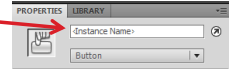
Övning 1

Grunderna för programmering

Design av interaktiv multimedia

Kom ihåg från förra kursen!

- ▶ Använd olika lager.
- ▶ Döp lagren!
- ▶ Organisera era bibliotek!
- ▶ Ge era symboler instansnamn för att hitta dem med AS.
- ▶ Nytt för denna kurs:
 - Ingen ActionScript-kod i .fla-filen!



Förbered er!

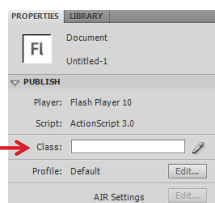
- ▶ **Läs i förväg** om det som övningarna kommer att beröra.
- ▶ Träna hemma både före och efter övningarna.
- ▶ Hitta på egna saker att programmera för att öva, öva, öva.
- ▶ Hinner ni inte färdigt, gör klart övningen hemma!

Introduktion till AS 3.0

- ▶ Filtyper:
 - FLA
 - Vanligt flashfil med alla visuella objekt.
 - Kan innehålla inbäddade objekt (animeringar, ljud, video, grafik).
 - AS
 - Textfil som innehåller all ActionScript-kod.
 - Är beroende av FLA-filen för att kunna tolka och exekvera koden.
 - Slipper använda lager och keyframes = mer strukturerad programmering.
 - SWF
 - Den "färdiga" Flash-filen som innehåller din applikation.

Använda .AS

- ▶ Skapa en ny ActionScript 3.0 FLA-fil, som vanligt.
- ▶ Skapa en ny fil, välj **ActionScript 3.0 Class** för att skapa en AS-fil.
- ▶ I properties, ange class (namnet på din AS-fil).



Variabler

- ▶ En variabel är en "behållare" för ett värde.
- ▶ Variabler i AS 3.0:
 - **String** (text)
 - **Boolean** (sant eller falskt)
 - **Number** (nummer)
 - **Int** (heltal, ej decimaler)
 - **Uint** (positiva heltal, ej decimaler)
 - **Object**
 - och några till...

Deklarera variabler

- ▶ För att använda variabler måste de först deklaras (skapas).

```
var myNumber:int;
var myText:String = "Hello World";
var myBool:Boolean = true;
```

- ▶ Skriv först **var**.
- ▶ Sedan ger du variabeln ett namn.
- ▶ Sedan : och variabeltyp.
- ▶ Om du vill kan du ge variabeln ett värde direkt.

Naming conventions (1)

- ▶ Det finns best practice för hur man namnger variabler.
- ▶ Inom programmering används s.k. **camel case**. Exempelvis:
 - helloWorld
 - myVariable
 - playerOne
 - veryEvilCharacter

Naming conventions (2)

- ▶ Varabler, metoder och instanser börjar med små bokstäver:
 - myVariable
- ▶ Klasser och constructors börjar med STORA bokstäver (PascalCase):
 - MyClass
- ▶ Stora och små bokstäver **spelar roll!**

Kodstruktur (1)

1. Separera koden i "paket":

```
package {
}
```

2. Definiera klassen:

```
package {
    public class Main {
    }
}
```

- **Public**: innebär att informationen kan delas med andra filer/klasser (*encapsulation*).
- Best practice att inleda klassnamn med versaler.

Kodstruktur (2)

håller samman koden i "paket"

encapsulation

```
package {
    public class Main {
    }
}
```

indikerar att det är en klass

klassnamn

Kodstruktur (3)

3. Constructor method:

- Körs automatiskt varje gång en ny instans av klassen skapas.

```
package {
    public class Main {
        public function Main() {
        }
    }
}
```

avsluta funktionsdefinitioner med ()
- kan innehålla parametrar

indikerar att det är en funktion (metod)

- Måste ha samma namn som klassen.

Kodstruktur (4)

4. Ge funktionen direktiv:

```
package {
    public class Main {
        public function Main() {
            trace("Hello world!");
        }
    }
}
```

avsluta alltid statements med semikolon!

- ▶ trace() är en inbyggd metod som skriver ut en sträng i output-fönstret.

Kodstruktur (5)

- ▶ I vårt fall krävs att klassen MovieClip importeras.

```
package {
    import flash.display.MovieClip;
    public class Main extends MovieClip {
        public function Main() {
            trace("Hello world!");
        }
    }
}
```

sökväg till klassen som ska importeras

indikerar att din klass ska ärvta egenskaper av en annan klass

vilken klass din klass "förväntar"

Kodstruktur (6)

- ▶ Allt inom { } hör till ett s.k. "block".
- ▶ Ett block kan innehålla andra block.

```
package {
    import flash.display.MovieClip;
    public class Main extends MovieClip {
        public function Main() {
            trace("Hello world!");
        }
    }
}
```

Kommentera koden (1)

- ▶ En rad med //

```
//Denna är en kommentar
```

- ▶ Flera rader med /* */

```
/* Detta är också en kommentar,
Men den tar upp flera rader
Utän att du måste kommentera varje rad för sig */
```

Kommentera koden (2)

```
//Add event listener
link.addEventListener(MouseEvent.CLICK, linkClicked);

//-----
//Listens for link being clicked
function linkClicked(e:MouseEvent):void {

    var url:String = "http://www.weheartit.com"; //Link URL
    var request:URLRequest = new URLRequest(url); //URL request for link

    //Navigates to link unless error
    try {
        navigateToURL(request, "_blank");
    } catch (e:Error) {
        trace("An error occurred!"); //Prints error message to console
    }
}
```



Svårläst kod...

```
//Add event listener
link.addEventListener(MouseEvent.CLICK, linkClicked);
//-----
//Listens for link being clicked
function linkClicked(e:MouseEvent):void {
var url:String = "http://www.weheartit.com"; //Link URL
var request:URLRequest = new URLRequest(url); //URL request for link
//Navigates to link unless error
try {
navigateToURL(request, "_blank");
} catch (e:Error) {
trace("An error occurred!"); //Prints error message to console
}
}
```

Lättläst kod!

```
//Add event listener
link.addEventListener(MouseEvent.CLICK, linkClicked);
//-----
//Listens for link being clicked
function linkClicked(e:MouseEvent):void {
    var url:String = "http://www.weheartit.com"; //Link URL
    var request:URLRequest = new URLRequest(url); //URL request for link

    //Navigates to link unless error
    try {
        navigateToURL(request, "_blank");
    } catch (e:Error) {
        trace("An error occurred!"); //Prints error message to console
    }
}
```

Operators (inledning) (1)

- ▶ Operators används för att t.ex.:
 - Ge ett värde till en variabel
 - Öka/minska värdet på en variabel
 - Jämföra variabler
 - Med mera...

Operators (inledning) (2)

- ▶ Operatorer:
 - +
 - -
 - =
 - /
 - *
 - %
 - +=
 - -=
 - ++
 - --
 - &&
 - ||
 - ==
 - ===
 - >
 - <
 - <=
 - >=
 - !
 - !=
 - m.fl.

Operators (inledning) (3)

- ▶ Tilldela ett värde till en variabel med =

```
var myText:String = "Hello World";
```

- ▶ Öka värdet på en variabel med +

```
var myNumber:Int = 2;
myNumber = myNumber + 5;
myNumber += 1;
myNumber++;
```

- ▶ Slå ihop två strängar med +

```
var myText:String = "Hello World!" + " again...";
```

Operators (inledning) (4)

- ▶ För matematiska uträkningar:

- +
- -
- /
- *

Panik?

- ▶ <http://www.theflashstudio.net/>
- ▶ http://livedocs.adobe.com/flash/9.0/main/flash_as3_programming.pdf
- ▶ <http://www.adobe.com/support/flash/>
- ▶ <http://www.actionscript.org/>