

This is my document class

```
package {

    import com.brewerthompson.textclasses.*;
    //imports the text classes I have written

    import flash.display.Sprite;

    public class InheritanceEx extends Sprite{

        public function InheritanceEx():void{

            //instantiates a new TextExtend class and sets font, size and color
            var newTxt:TextExtend= new TextExtend("Copperplate Std 33 BC",24,"0x999999");
            addChild(newTxt);

            //calls the makeNewText function in the textExtend class.
            newTxt.makeNewText(100,100, "called from the doc class", true)

        }

    }

}
```

```
package com.brewerthompson.textclasses{

    import flash.display.MovieClip;
    import flash.text.*;

    public class SuperText extends MovieClip{

        var newText:TextField = new TextField();
```

This is an external class called "SuperText.as" Note that this extends the MovieClip class. This means that it "inherits" all the properties of a MovieClip. This is called INHERITANCE

```

public function SuperText(){
    testThis();

}

function makeText(xPos:int=0, yPos:int=0, str:String=" ", border:Boolean=fi
    /*the above arguments set default values--- which means this will be used unless
    parameters are passed in the subclass.
    */

    trace("supertext here");
    newText.type=TextFieldType.DYNAMIC;
    newText.selectable=false;
    newText.background= true;
    newText.autoSize = TextFieldAutoSize.LEFT;
    newText.border=border;
    newText.maxChars=30;
    newText.text= str;
    newText.x=xPos;
    newText.y=yPos;
    addChild(newText);
}

function testThis():void{
    trace("SUPERTEXT: running the testThis() function");
}

function runThis():void{
    trace("this was called from the subclass"); }

```

```
package com.brewerthompson.textclasses{
```

In addition to “extending” classes like MovieClips, you can extend your own custom classes.

```
import flash.text.*;
```

```
public class TextExtend extends SuperText{
```

```
    /*this extends the SuperText class I've already written
```

```
        this means that the new (TextExtend) class INHERITS all variables and functions in the “SUPER” class  
        this makes TextExtend a “Subclass”. (FYI: SuperText extends the MovieClip class which allows it to do  
        anything a MovieClip can. )
```

```
        Why write a subclass? Well, here I have a pretty bare-bones Superclass that I want to add to.
```

```
        Writing SUBCLASSES allows me to use what I've already written and add to it--- without having to re-write  
        the whole thing!
```

```
    */
```

```
    public var myFormat:TextFormat = new TextFormat();
```

```
    public function TextExtend(fnt:String, fntSize:int, fntColor:String):void{
```

```
        trace ("this extends the SuperText class");
```

```
        /*note that when running this-- it does what is specified in this clas! (above the trace) PLUS what  
        is in the class it extends!
```

```
    */
```

```
super.runThis()
```

```
//this line CALLS a method from the SuperText class
```

```
//I'm using this as an example of how you can access methods from the superclass.
```

```
//these are set in the doc class when instantiating the TextExtend class
```

```
//newTxt.makeNewText(100,100, "called from the doc class", true)
```

```
myFormat.font=fnt;  
myFormat.size=fntSize;  
myFormat.color=fntColor;
```

```
}
```

1/2

The INHERITANCE of these classes looks like this:

