

Boolean Properties

If a variable is declared as Boolean, it can only have values of true or false. Example:

```
var ready: Boolean=false;
```

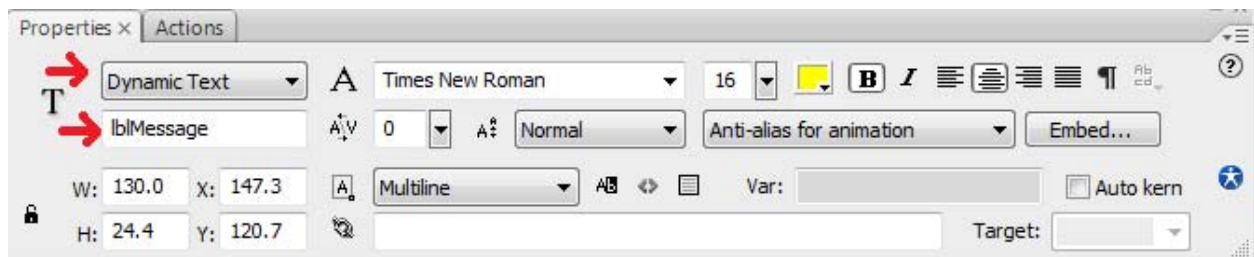
We can make a movie clip or button visible or invisible by assigning a value of true or false to the visible property. (There is no invisible property):

```
btnStart.visible=false; //button is invisible, and cannot be seen or clicked
```

In this lesson we will use the visible property of a movie clip to create a holiday greeting. This is for Halloween, but any holiday will do. The movie shows a greeting with a message to click the greeting. When the user moves the mouse over the movie clip it disappears and a different one appears.

Happy Halloween

1. Start a new Flash ActionScript3 project.
2. Save as holiday.fla
3. Type a greeting, then break it apart into letters, and then break the letters into drawing objects.
4. Select all of the text and make it a movie clip.
5. Create 2 or 3 more movie clips with different pictures.
6. Make sure you name the instances. In the example the instances of the movie clip are happy (the greeting), bat, witch, and pumpkin.
7. Select the text tool **T**, click anywhere on the stage and type the words "Click the witch"
8. Open the properties window and select Dynamic Text, name the instance lblMessage.



The code starts off by making the pumpkin invisible by setting its visible property to false. When either the witch or the pumpkin is moved over with the mouse the function goClick is called. In goClick the visible property of both the witch and pumpkin are “toggled” by setting the value of visible to the opposite value. That is, if visible is true it becomes false; if visible is false it becomes true.


When you test the movie the witch is visible but the pumpkin is not. The message says “Click the witch.” When you move the mouse over to the witch to click it, the witch disappears and the pumpkin becomes visible. The message now says “Click the pumpkin.” When you move the mouse over to the pumpkin to click it, the pumpkin disappears and the witch becomes visible with the message “Click the witch”: a little Halloween trick.

```
pumpkin.visible=false; //hide the pumpkin by setting its visible property to false.
```

```
//when either the pumpkin or witch is clicked call function goTrick
pumpkin.addEventListener(MouseEvent.CLICK, goTrick);
witch.addEventListener(MouseEvent.CLICK, goTrick);
```

```
function goTrick(e:MouseEvent): void
{ witch.visible=!witch.visible; //if visible make it not visible; if not visible make it visible.
  pumpkin.visible=!pumpkin.visible;
  if(pumpkin.visible) //if the pumpkin is visible the message will say click the pumpkin.
    lblMessage.text="Click the pumpkin";
  else //if the pumpkin is not visible, the witch is visible: the message will say click the witch.
    lblMessage.text="Click the witch";
} //goTrick
```

Debugging:

- Make sure that every { has a matching }; every (has a matching).
- The strings, “Click the witch” and “Click the pumpkin” must be enclosed in double quotes.
- Every statement ends in a semicolon, but there is NO semicolon after the if and else lines because if(pumpkin.visible) lblMessage.text="Click the pumpkin"; is considered one statement; else lblMessage.text="Click the witch"; is also considered one statement.
- Before you test the movie, click  in the Action window to check for syntax errors.
- Make sure that all of the names used in the ActionScript match the names of the instances on the stage: including case!

Be sure to look at the variations: one moves the message, the other rotates between the witch, bat, pumpkin and message.