

Events

Flash is event driven. That means that as the movie plays, Flash listens for various events to happen such as a mouse moving or clicked, a key pressed, or even just going to a new frame. When one of these events occurs Flash performs the functions that you have told it to perform for that event.

Moving a Car on Entering a Frame

Open the car program that you created in the lesson on Properties. In this lesson we are going to make the car move by increasing the value of `car.x` each time we enter the frame. Our movie only has one frame, but that frame keeps reentering each time we enter frame 1.

We must do this in two steps. First we will create a function that we want to execute each time we enter a frame,; then we will tell Flash to “listen” for the EnterFrame event and to executed that function.

```
function frames(e:Event):void  
{ car.x=car.x+5;  
}
```

In the first line, the keyword **function** is followed by the name that we make up. This name must start with a letter and is followed by letters, digits and `_`. It can NOT have any spaces or other punctuation marks. We have named the function **frames**. Next we have a pair of parenthesis with the name of an argument, followed by a colon and the type of the argument. We can make up any name instead of **e** (using the rules above for creating names), but we must use the keyword **Event**. An event is also an object with properties. (*Just an Event, like we have here doesn't have very many interesting properties, but other events, such as MouseEvent have properties that tell you whether the control key was down, the and the position of the mouse when the MouseEvent occurred.*) After the closing parenthese around the arguments, there is a colon followed by the type of value that the function returns. This function is **void**, it doesn't return anything.

The body of the function is enclosed in a pair of curly braces. These is just one statement in this function. We say to give `car.x` a new value that is 5 more than it's current value. This is called an assignment statement. The property or variable that is getting a new value is on the left followed by the equal sign and the new value. Flash will compute the value of the expression on the right, then assign that as the new value. All statements must end with a semicolon.

If you run this program now, nothing will happen. In addition to writing the function, we must also tell Flash when to execute the function. Add the following line under the function:

```
this.addEventListener(Event.ENTER_FRAME, frames);
```

The keyword **this** refers to the Movie itself, it has a method called **addEventListener**. This method requires two arguments: the event, and the function to

execute when that event occurs. In this case, when the movie enters a new frame, we want to execute the function we wrote called **frames**.

If you run the program now, the car will move across that stage and disappear on the right. It isn't coming back! Unlike motion tweening, where a movie plays over and over, we only have one frame and the only way that car will come back is if we add code to start it back on the left. We would like to do this when the car disappears on the right.

Modify function frames as shown below:

```
function frames(e:Event):void
{
    car.x=car.x+5;
    if(car.x>stage.stageWidth) car.x=-1*car.width;
}
```

If you run the program now, the car will move across that stage and disappear on the right, then reappear on the left and move across the stage again.

The if statement has the keyword if followed by a Boolean expression inside parenthesis. A Boolean expression is something that will be true or false. After the Boolean expression, write the statement that you want to execute when the Boolean expression is true. If you want to execute more than one statement, you must enclose all of the statement inside another pair of curly braces.

Comments

We could end up with lots of code and lots of pairs of curly braces. Experienced programmers leave lots of comments to remind themselves why they add a piece of code. You can add a comment by putting // after a statement, then write anything you want. Flash ignores anything on the line after // Flash will also ignore anything between /* and */ You can use this pair to create a comment that is on several lines. Notice the code above with comments added:

```
/* Written by Janet Joy
A car moves across the stage. When it leaves on the
right, it reappears on the left. */
```

```
function frames(e:Event):void
{
    car.x=car.x+5; //move 5 spaces to the right
    if(car.x>stage.stageWidth) car.x=-1*car.width;
    if(car.x<-1*car.width) car.x=stage.stageWidth;
} //frames
this.addEventListener(Event.ENTER_FRAME, frames);
```

Experiment: Create a movie where the car moves from right to left; try making a rocket ship that moves from the bottom to the top.