

## Variables

A variable is a named location in memory where you can store values.

**Variable Names** must start with a letter of the alphabet: after the first letter, it can have more letters, digits, and underscores. Blanks are not allowed. Two words can be separated by an underscore, or the second word can begin with a capital letter to make it readable: **HoursWorked** or **Hourly\_rate**, for example. You can not use *keywords* such as **button**, **if** or **stop** that have a special meaning in Flash.

**Type:** Each variable also has a type. There are many types, but we usually use one of the following:

**Number:** Values that we want to do arithmetic operations on.

**String:** for names, words, phone numbers, etc.

**Boolean:** for variables that can be either True or False.

**Note:** *Although a phone number is composed of digits and even has the word number in the name, we do not intend to do arithmetic operation on it, so it would be a string, not a number.*

**Declaration:** A variable **dx** is declared as shown below. Notice that a variable can be given an optional initial value when it is declared.

**var dx: Number=0;**

**var dy: Number;**

**var greeting:String="Hello World";**

**var answer:String;**

**var correct: Boolean=false;**

**Scope:** If you declare a variable inside a function, it is known only inside that function. If you want to use a variable anywhere in the movie, declare it in frame 1, but not inside any function. A variable that is declared inside a function is said to be local to that function.

### Assigning Values to a Variable:

A variable can be assigned a value with the assignment statement.

The format is: **variable = newvalue;**

#### Examples:

The new value is a constant: **x = 5;**

The new value is another variable: **x = y;**

The new value is an expression: **x = y/3;**

It can even use its current value in the expression: **x = x + 1;**

The last statement is called incrementing and can also be written: **x++;**

*The last two statements assign a new value to x that is one more than its current value!*

## Moving the Car

1. Open the car program that you created in the previous lesson. (Or start a new project with a Movie Clip called car.)
2. Modify the code in Frame 1 as shown below:

```
var dx: Number=1;
function frame(e:Event):void
{ car.x=car.x+dx;
  if(car.x>stage.stageWidth) car.x=-1*car.width;
} //frame
this.addEventListener(Event.ENTER_FRAME, frame);
```

Instead of adding 1 to the value of x we are adding the value of the variable dx, which happens to be 1. In the next section, we will use a button to add to the value of dx.

## Buttons

From the menu select Window, Common Libraries, Buttons. Select a red button and a green button. If desired, change or add text to the button. In the properties window name the red button btnSlow. Name the green button btnFast. Rewrite the code as shown below:

```
var dx: Number=1;
function frame(e:Event):void
{ car.x=car.x+dx;
  if(car.x>stage.stageWidth) car.x=-1*car.width;
  if(car.x<-1*car.width) car.x=stage.stageWidth;
}
this.addEventListener(Event.ENTER_FRAME, frame);

function faster(e:MouseEvent)
{ dx++;
} //faster
function slower(e:MouseEvent)
{ dx--;
} //slower
btnFast.addEventListener(MouseEvent.CLICK, faster);
btnSlow.addEventListener(MouseEvent.CLICK, slower);
```

Just as we added code for the ENTER\_FRAME event of the movie, we have added the functions faster and slower to execute on the **MouseEvent.CLICK** for the two buttons. Test the movie and click on the buttons to make the car faster and slower.

## Trace

If you would like to see the value of dx add the following line to function faster:

```
function faster(e:MouseEvent)
{ dx++;
  trace(dx);
} //faster
```

The value of dx will display in the output window whenever you click btnFast.