Conditionals: When your program needs to make a choice?

Read Chpt. 5 in Shiffman

Should I take the left or right path?

- Make a *choice* of what to do (generally) based on *data*
- e.g. Student feedback program
  - Student got an A, say “Great job!”
  - Student got an F, say “More time at the library for you!”
- Grade is data, response is the action
- Branching
  - Code *branches* (does something different) based on the decision made

Checking Conditions

- Condition must evaluate to true or false
  - e.g. grade > 90
  - grade < 40
  - Often, but not necessarily based on data
- Use command *if* to check the condition

```
if(<condition>)
{
   //code to run if condition evaluated to true
}
```

Conditions

- Must evaluate to true or false
- Boolean variables are this by definition
- Checks

==	Equal to
>	Greater than
>=	Greater than or equal true
<	Less than
<=	Less than or equal to

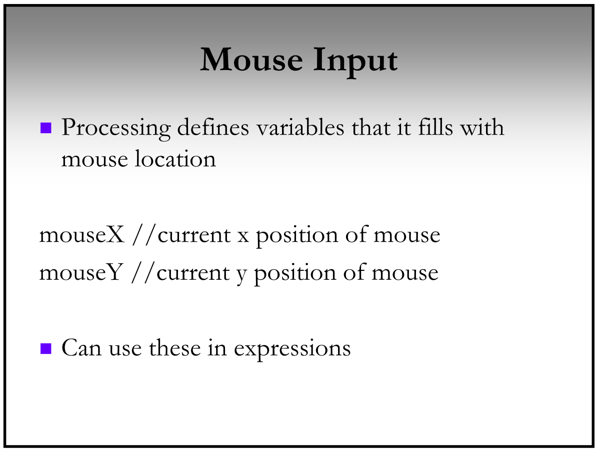
Outputing Text to Processing Window

- Use println(<text>);
- Examples:

```
int f = 4; float y = 47.2;
println(“some text”); //display text string.  Must
                            //use quotation marks
println(f);//print the value stored in f
println(“The value of f is ” + f + “and y is “ + y );
//+ concatenates.  Prints:
The value of f is 4 and y is 47.2
```

Conditional Text Output

```
int grade = 75;
if( grade >= 90)
{
   println("Awesome!  You got an A!");
}
```

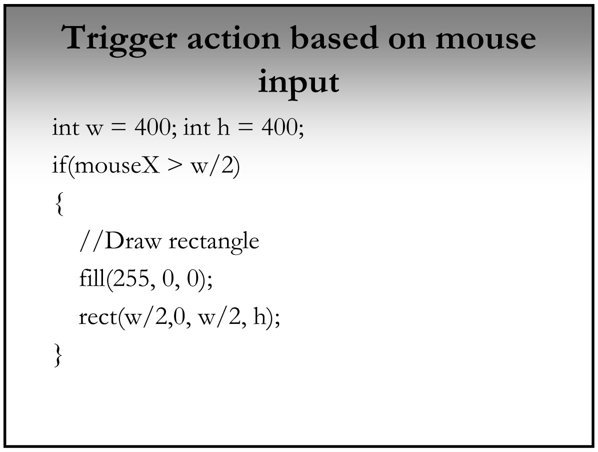


Mouse Input

- Processing defines variables that it fills with mouse location

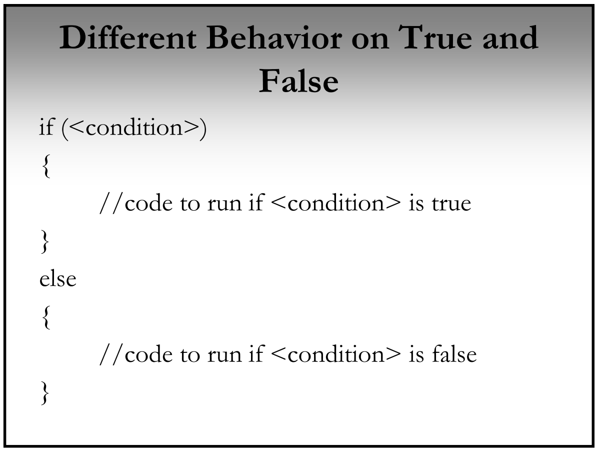
```
mouseX //current x position of mouse
mouseY //current y position of mouse
```

- Can use these in expressions



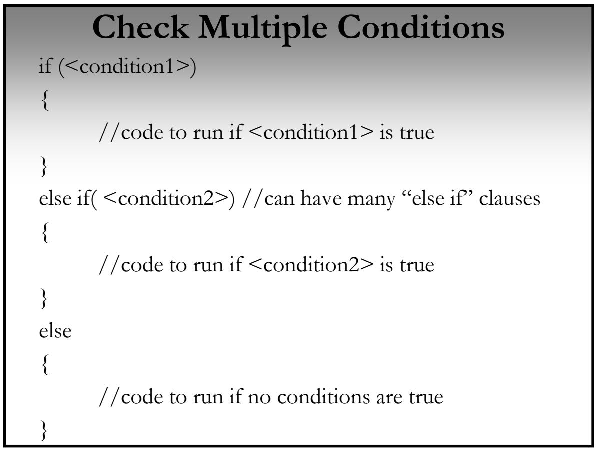
Trigger action based on mouse input

```
int w = 400; int h = 400;
if(mouseX > w/2)
{
   //Draw rectangle
   fill(255, 0, 0);
   rect(w/2,0, w/2, h);
}
```



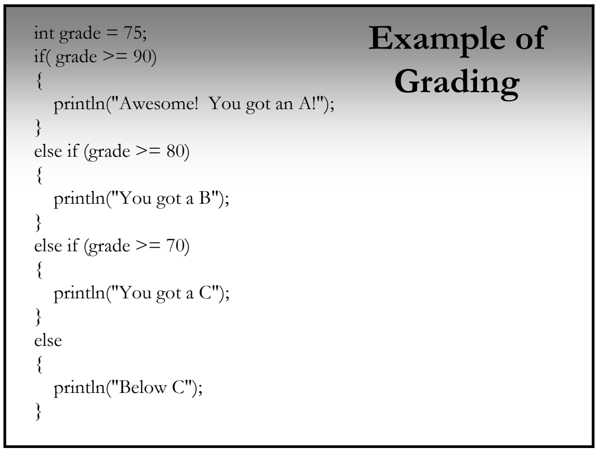
Different Behavior on True and False

```
if (<condition>)
{
        //code to run if <condition> is true
}
else
{
        //code to run if <condition> is false
}
```



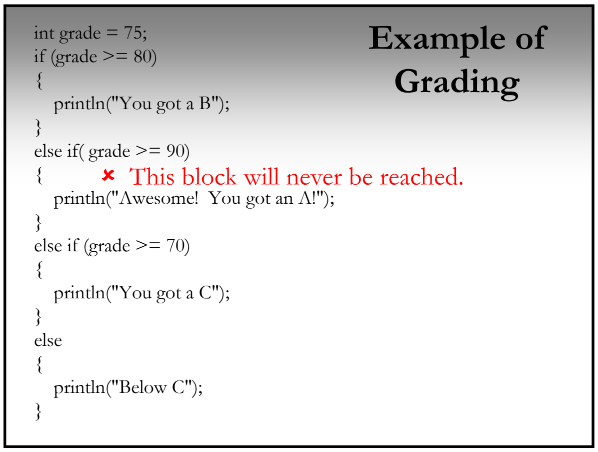
Check Multiple Conditions

```
if (<condition1>)
{
        //code to run if <condition1> is true
}
else if( <condition2>) //can have many “else if” clauses
{
        //code to run if <condition2> is true
}
else
{
        //code to run if no conditions are true
}
```



Example of Grading

```
int grade = 75;
if( grade >= 90)
{
   println("Awesome!  You got an A!");
}
else if (grade >= 80)
{
   println("You got a B");
}
else if (grade >= 70)
{
   println("You got a C");
}
else
{
   println("Below C");
}
```



Example of Grading

```
int grade = 75;
if (grade >= 80)
{
   println("You got a B");
}
else if( grade >= 90)
{        × This block will never be reached.
   println("Awesome!  You got an A!");
}
else if (grade >= 70)
{
   println("You got a C");
}
else
{
   println("Below C");
}
```

Checking Multiple Conditions

- In else if clause, only proceed to the next else if in the case that the previous if condition failed (i.e. was false)
- Order of conditions matters
- First true condition will be executed
- The final else is executed if all conditions are false

Logical Operators

- Can combine individual conditions to make more complex logical conditions

||	**OR**	**Either the left or right condition is true (or both are true)**
&&	AND	Both the left and right must be true
!	NOT	Flips meaning of condition. True if condition was false.

Logical Operators

```
if( a > b  && a > c)
{
        //a is greater than b and c
}
if( a > b  || a > c)
{
        //a is greater than b OR a is greater than c
        //OR both can be true
}
```

Logical Operators

```
if( !(a > b))
{
        //a is less than or equal to b
}
```

More Complex Mouse Example

Nesting

- Can have if's within if's

```
if( a > b)
{
   if(a > 100)
   {
      println("a is bigger than b and bigger than 100");
   }
}
```

Animation

Animation

- Creates the illusion of movement
  - Rapidly show slightly changing still frames
- Relies on "Persistence of Vision"
- Can animate any parameter
  - Color
  - Size
  - Position
  - Rotation
  - etc.

How do you drive the motion?

- One option: Use a global "clock"
  - Create a global variable that you increment each time you call draw
  - Calculate command parameters based on this

```
float frameCnt =0;
void draw()
{
  //increment the timer
  frameCnt = frameCnt + 1.0;
  //use the value in a command
  rect(frameCnt, 100, 10, 40);
}
```

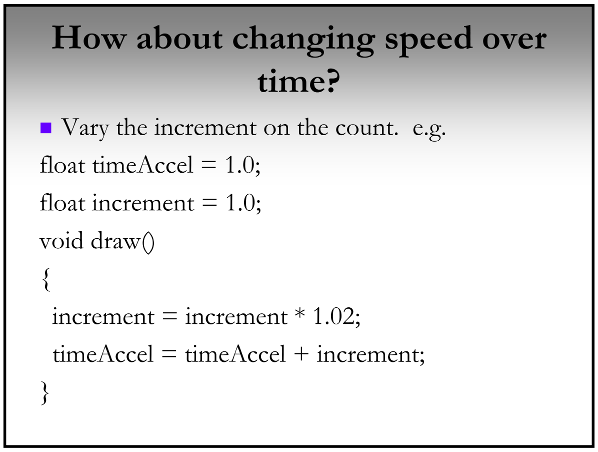
What if you want to vary the speed for different objects?

- Do operations on counter, e.g.:

```
float rot1 = 2* frameCnt;

float rot2 = 5* frameCnt;
```

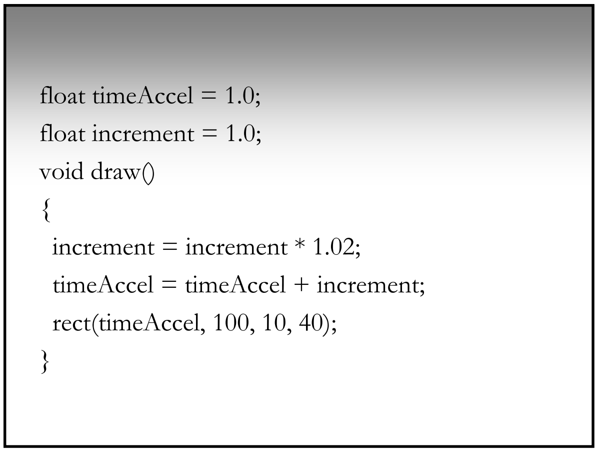
```
float frameCnt =0;
void draw()
{
  //increment the timer
  frameCnt = frameCnt + 1.0;
  //use the value in a command
  rect(frameCnt, 100, 10, 40);
  rect(frameCnt*2, 200, 10, 40);
}
```



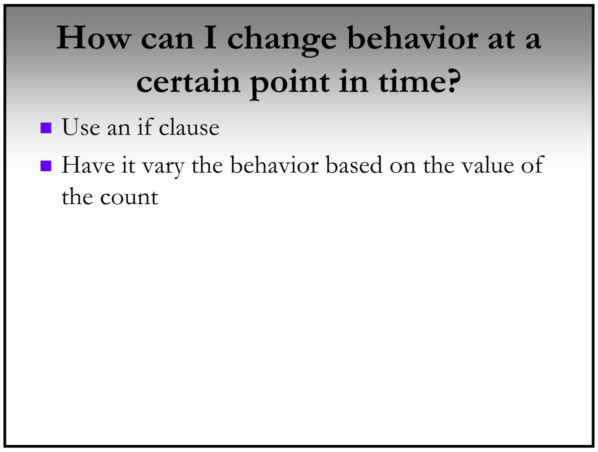
How about changing speed over time?

- Vary the increment on the count. e.g.

```
float timeAccel = 1.0;
float increment = 1.0;
void draw()
{
  increment = increment * 1.02;
  timeAccel = timeAccel + increment;
}
```

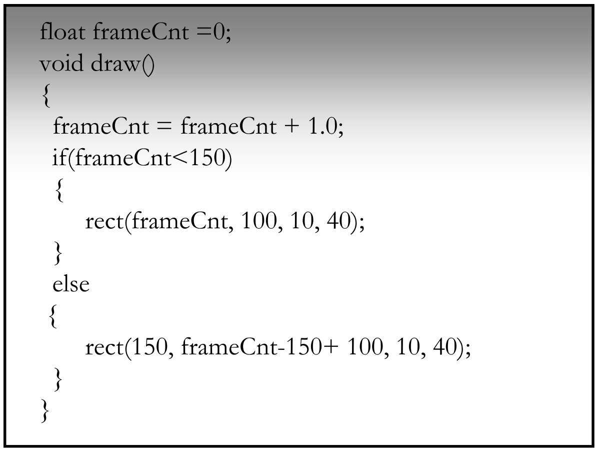


```
float timeAccel = 1.0;
float increment = 1.0;
void draw()
{
  increment = increment * 1.02;
  timeAccel = timeAccel + increment;
  rect(timeAccel, 100, 10, 40);
}
```

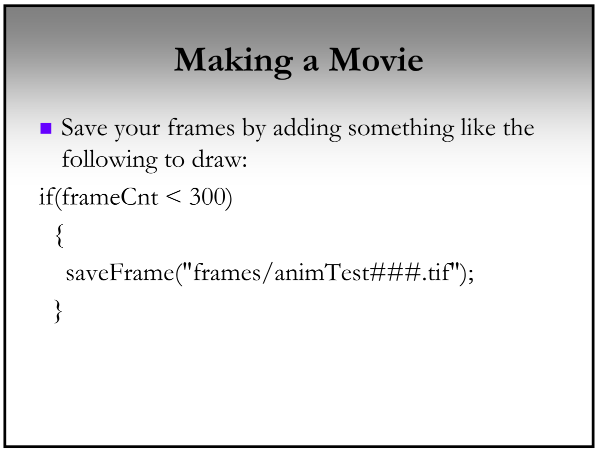


How can I change behavior at a certain point in time?

- Use an if clause
- Have it vary the behavior based on the value of the count



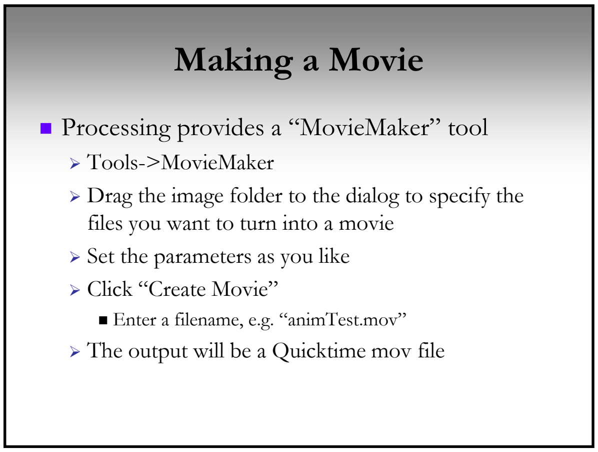
```
float frameCnt =0;
void draw()
{
  frameCnt = frameCnt + 1.0;
  if(frameCnt<150)
  {
      rect(frameCnt, 100, 10, 40);
  }
  else
  {
      rect(150, frameCnt-150+ 100, 10, 40);
  }
}
```



Making a Movie

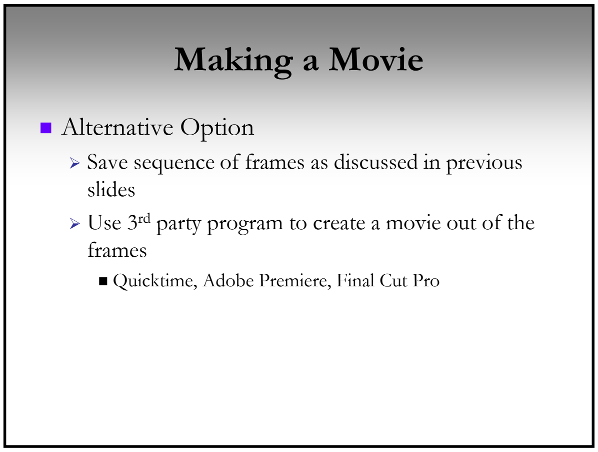
- Save your frames by adding something like the following to draw:

```
if(frameCnt < 300)
  {
    saveFrame("frames/animTest###.tif");
  }
```



Making a Movie

- Processing provides a “MovieMaker” tool
  - Tools->MovieMaker
  - Drag the image folder to the dialog to specify the files you want to turn into a movie
  - Set the parameters as you like
  - Click “Create Movie”
    - Enter a filename, e.g. “animTest.mov”
  - The output will be a Quicktime mov file



Making a Movie

- Alternative Option
  - Save sequence of frames as discussed in previous slides
  - Use 3rd party program to create a movie out of the frames
    - Quicktime, Adobe Premiere, Final Cut Pro