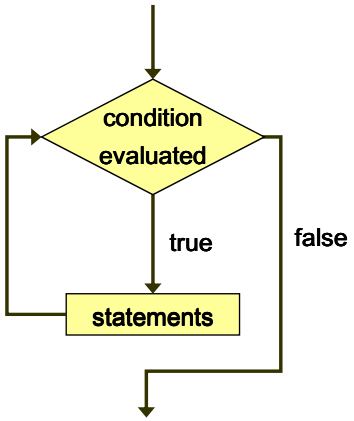
Loops

Chapter 9 - Lecture Slides

Loop

- A loop is when
- A loop stops when the condition in the expression evaluates to
- Three kinds of loops:
  -
  -
  -
- Although we could convert any loop into another loop, there is a reason why there are 3 different loops – the programmer should use the correct loop for a given situation

while




```
while( booleanExpression )
{
   statements;
}
```

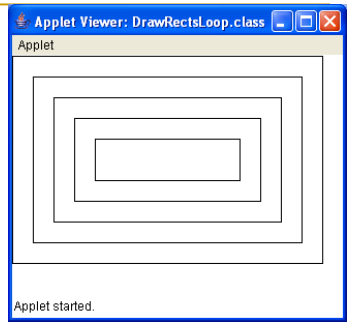
- continually executes the statements until the booleanExpression becomes
- the body of a `while` loop will execute

while Example

```
int count = 1;
while (count <= 5)
{
   System.out.println (count);
   count = count + 1;
}
```

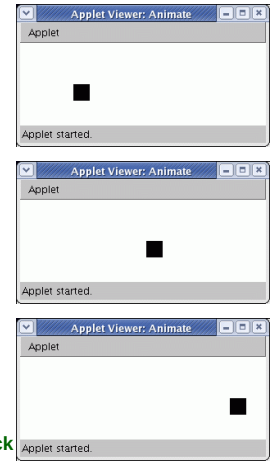
```
int count = 1;
while (count <= 5)
{
   count = count + 1;
   System.out.println (count);
}
```

while Example




```
import java.awt.*;
import javax.swing.*;
 public class DrawRectsLoop extends JApplet
 {
   int x=0;
   int y=0;
   int width=300;
   int height=200;
   int spacing = 20;

   public void paint ( Graphics g )
   {
      while( width>spacing && height>spacing )
      {
            g.drawRect( x,y, width, height );
            x = x + spacing;
            y = y + spacing;
            width = width - 2*spacing;
            height = height - 2*spacing;
      }
   }
 }
```


while Example with Animation




```
/* Demonstrate loops with animation */
import java.awt.*;
import javax.swing.*;
public class Animate extends JApplet
{   int x, y, size, move, speed, count;
   public void paint ( Graphics g )
   {       x = 0;       y = 50;
           size = 20;      // width and height same
           move = 1; //num pixels to move each time thru loop
          count = 0;       speed = 10000000;  // slow down loop
          while ( count < 5000 )
          {
                  g.clearRect ( x, y, size, size);  // erase space
                   x = x + move;   // new x coodinate, y stays same
                   g.fillRect ( x, y, size, size );
                     // did we hit the edge of the applet?
                   if ( x + size > getWidth( ) )
                                  move = -move;            // if so, go back
                    for ( int i=0; i<speed; i++ )
                            ;  // slow down painting so we can see it
                   count = count + 1;
          }
   }
}
```

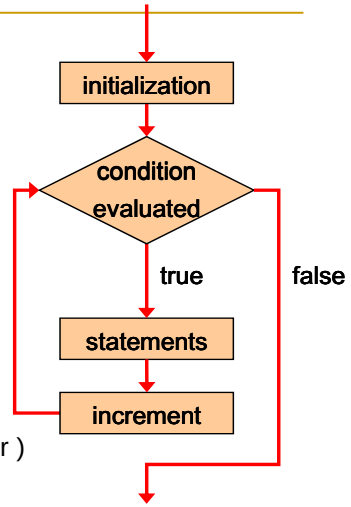

for

For loop

- Used when you know
- The body of a for loop will execute

```
for( initialization ; booleanExpression ; incrementer )
{
   statements;
}
```


for



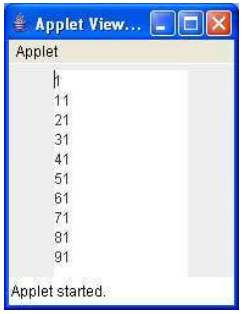

```
for( initialization ; booleanExpression ; incrementer )
{
    statements;
}
```

for Example

- Write a loop that sums the values between 1 and 100

```
int sum = 0;




sumLabel.setText( sum );
```

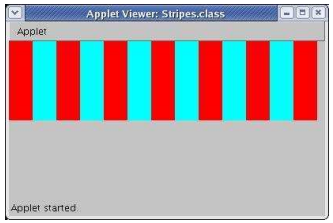

for Example




```
import java.awt.*;
import javax.swing.*;
public class forCountBy10 extends JApplet
{
   JTextArea ta_display;
   public void init()
   {
       setLayout( new FlowLayout( ) );
       ta_display = new JTextArea( 15, 40 );
       for( int value=1; value <= 100; value = value + 10 )
              ta_display.append( "" + value + "\n" );
       add( ta_display );
   }
}
```


for Example in paint method

```
public void paint( Graphics g )
   {
   int stripes = 0;
   int width = 30;
   int height = 100;
   for (int x=0; stripes < 13; x=x+width)
   {
        if (stripes%2 == 0)
        {
                g.setColor(Color.RED);
        }
        else
        {
                g.setColor(Color.CYAN);
        }
        g.fillRect(x, 0, width, height);
        stripes = stripes + 1;
        }   // end while loop
   }   // end paint method
}
```



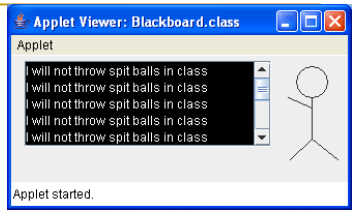
for Example on a list:

- To print out all the items in a list, use a for loop and the method **getItem** on the list object

```
import java.awt.*;
import javax.swing.*;
public class List2TA extends JApplet
{
   JList list;
   DefaultListModel model;
   JTextArea textarea;
   public void init( )
   {
     setLayout( new FlowLayout( ) );
     setupList( );

     textarea = new JTextArea( 5,10 );
     add(textarea);

     addListItemsToTextarea( );
   }
```

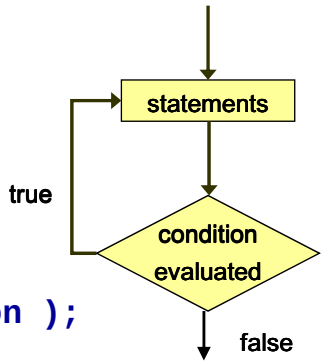
```
public void setupList( )
{
   model = new DefaultListModel( );
   list = new JList(model);
   model.addElement( "Milk" );
   model.addElement( "Cookies" );
   model.addElement( "Eggs" );
   add(list);
}   // end init method

public void addListItemsToTextarea( )
{
   // getSize returns # of items in list
    for( int i=0; i<model.getSize( ); i++ )
    {
      // grab the item at index i
      textarea.append((String )model.get(i));
      textarea.append( "\n" );
    }
  }
}
```

for example




```
for ( int i=0; i<10; i++ )
   bboard.append( "I will not throw spit balls in class\n" );
```


do...while Loop




```
do
{
  statements;
} while( booleanExpression );
```

- continually executes the statements until the `booleanExpression` becomes
- the body of a `do…while` loop will execute

Which loop to use?

if (you know the # of iterations)
- use a

else if (statements should be done at least once)
- use a

else
- use a

Careful!

- Off-by-one error
- Missing squigglys
- Semi-colon syntax
- Infinite loops

Careful!

- **Off-by-one error**

Want to add values 1 through 10 to textarea:

```
for( int i=1; i<10; i=i+1 )
  textarea.append( String.valueOf(i) );
```

What gets added to the textarea?

Careful!

- **Missing squigglys**

```
int x = 10;
while( x < 100 )
   textarea.append( String.valueOf(x) );
   textarea.append("\n" );
   x = x + 10;
```

- should be:

```
int x = 10;
while( x < 100 )

   textarea.append( String.valueOf(x) );
   textarea.append("\n" );
   x = x + 10;
```


Careful!

- **Semi-colon syntax**
- You won't get an error message for putting a ; at the end of your while or for loop – but you probably won't get the results you expect!

```
for( int x =1; x<=10; x=x+1 )  ;
{
   textarea.append( "Hello" );
}
```

- The semi-colon declares an empty body of the loop!

Infinite Loop

- **Infinite Loops**
  -

```
while( true )
   drinkCoffee();

for( int i=1; i>0; i++ )
   walkOn( );

x = 1;
while( x < 10 );
     x = x + 5;

y = 1;
while( y < 10 )
   ta.append( y );
   y++;
```

Summary

- `while` loop
- `for` loop
- `do… while` loop
- When to use each loop
- Common Errors