

# Advanced Multimedia Technologies

Events, Listeners & Handlers in Practice

# Outcomes

- Practice the event notification system using built in events.
- Purge unneeded listeners and make them available for garbage collection.

# Demo of Events

- Mouse Events
- Keyboard Events
- Focus Events
- Progress Events
- Complete Events

```
// Mouse DOWN and OUT Example
```

```
mc.addEventListener(MouseEvent.CLICK, onOverListener)  
mc.addEventListener(MouseEvent.CLICK, onOutListener)
```

```
function onOverListener(e:MouseEvent):void{  
    // adds Glow to mouse over target  
    e.target.filters = [new GlowFilter()]  
}  
function onOutListener(e:MouseEvent):void{  
    // removes all filters from mouse out target  
    e.target.filters = []  
}
```

```
// Listen when the key on the keyboard has been released
text_box.addEventListener(KeyboardEvent.KEY_UP, onKeyUpListener)

function onKeyUpListener(e:KeyboardEvent):void{
    // Output a random number between 0.00 and 500.00
    trace(Math.random()*500)
}
```

```
// Listen when the key is pressed and move the box
box.addEventListener(KeyEvent.KEY_DOWN, onKeyDownListener)

function onKeyDownListener(e:KeyboardEvent):void{
    switch(e.keyCode){
    case Keyboard.LEFT:
        e.target.x+=10;
        break;

    case Keyboard.RIGHT:
        e.target.x-=10;
        break;
    }
}
```

```
// Listen for FOCUS IN and OUT of text boxes
text_box.addEventListener(FocusEvent.FOCUS_IN, onFocusInListener)
text_box.addEventListener(FocusEvent.FOCUS_OUT, onFocusOutListener)
```

```
function onFocusInListener(e:KeyboardEvent):void{
    e.target.filters = [new GlowFilter()]
}
function onFocusOutListener(e:MouseEvent):void{
    // removes all filters from mouse out target
    e.target.filters = []
}
```

```
// Listen for Progress
var img:Loader = new Loader();
img.loader('http://www.google.com/intl/en_ALL/images/srpr/logo1w.png')
img.contentLoaderInfo.addEventListener(ProgressEvent.PROGRESS,
onProgressListener);

function onProgressListener(e:ProgressEvent):void{
    trace(e.bytesLoaded/e.bytesTotal);
}
```

```
// Listen for Complete
var img:Loader = new Loader();
img.loader('http://www.google.com/intl/en_ALL/images/srpr/logo1w.png')
img.contentLoaderInfo.addEventListener(Event.COMPLETE,
onCompleteListener);

function onCompleteListener(e:ProgressEvent):void{
    trace("Completely Loaded")
}
```

# removeEventListener()

Too many Event listeners can bog down your application. It's necessary to purge unneeded listeners.

```
mc.removeEventListener(MouseEvent.CLICK, onClick);
```

```
// THERE'S NO NEED TO CONTINUE TO LISTEN FOR THE
// COMPLETE LISTENER ONCE IT HAS LOADED.
// REMOVE IT THROUGH removeEventListener()

// register the listener
img.contentLoaderInfo.addEventListener(Event.COMPLETE, onCompleteListener)

function onCompleteListener(e:Event):void{
    // remove the listener
    img.contentLoaderInfo.removeEventListener
    (Event.COMPLETE, onCompleteListener)
    trace("Completely Loaded")
}
```

# Garbage Collection

An internal process performed by the Flash Player to remove unused objects, reference, listeners, etc.

To take advantage of garbage collection objects must be set to null and ALL references to listeners must be removed.

```
// From our most recent example...  
  
img.contentLoaderInfo.removeEventListener(Event.COMPLETE, onCompleteListener)  
  
// removes the listener and makes it available for garbage collection  
  
img = null  
  
// sets the img object to null  
// img is now available for garbage collection.
```

# REMEMBER!

Garbage collection is an operation run by the Flash Player. It determines when the best time to remove objects. You can **NOT** force items to be purged. All you can do is make them available for garbage collection.

# Outcomes

- Practice the event notification system using built in events.
- Purge unneeded listeners and make them available for garbage collection.