Fundamentals of Multimedia Laboratory

2020-2021

المرحلة الرابعة / الفصل الدراسي الاول

اساتذة المادة:

أ.م.د. ناصر حسين سلمان

م.د. سهيلة نجم محمد

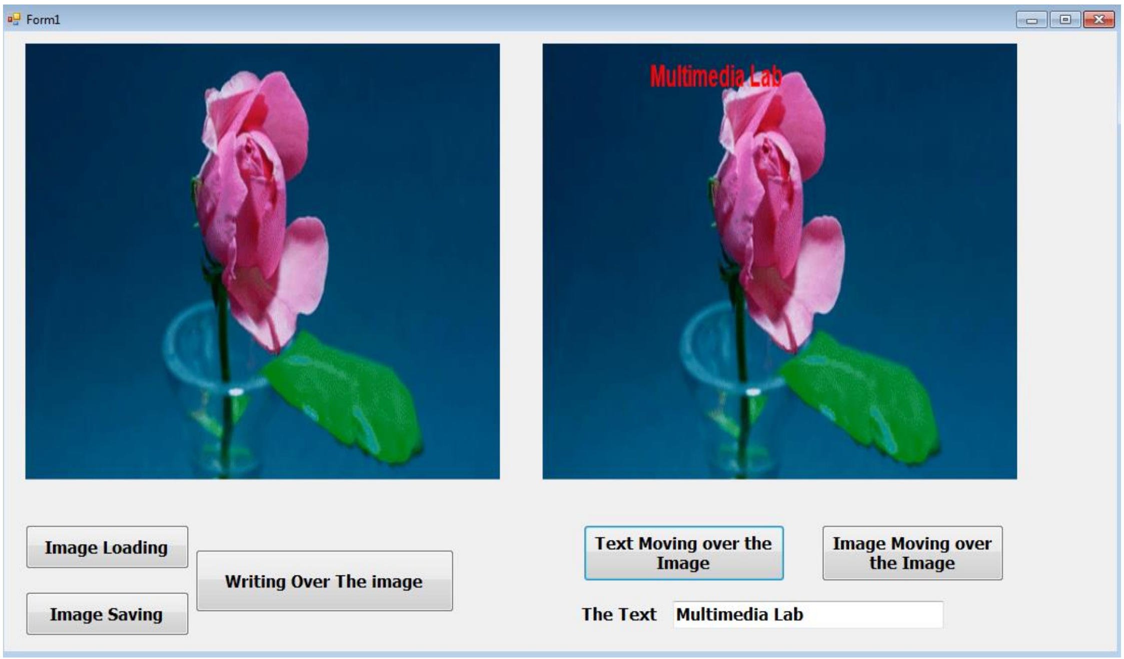
م.م. هدى مصطفى رضا

م.م. شيماء ضياء الدين

Multimedia Lab. 2020-2021 / 4th Class/ Morning Study

Lab. 1 : Image Loading, Saving and Manipulations

Design



Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        Bitmap bmp, bmp2;
        int x = 0, y = 0;

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //Image Loading
            openFileDialog1.ShowDialog();
            bmp = new Bitmap(openFileDialog1.FileName);
            pictureBox1.Image = bmp;
        }
        private void button2_Click(object sender, EventArgs e)
        {
            // Image Saving
            saveFileDialog1.Filter = "Bitmap Image| .bmp";
            saveFileDialog1.ShowDialog();
            bmp2.Save(saveFileDialog1.FileName);
        }

        private void button3_Click(object sender, EventArgs e)
        {
            //Writing a text over the center of the image
            bmp2 = new Bitmap(bmp);
            Graphics G = Graphics.FromImage(bmp2);
            G.DrawString(textBox1.Text, new Font("Arial", 8, FontStyle.Bold),
Brushes.Red, bmp2.Width / 2, bmp2.Height / 2);
            pictureBox2.Image = bmp2;
        }

        private void button4_Click(object sender, EventArgs e)
        {
            //Starting text moving over the image
            timer1.Enabled = true;
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            // Moving the text over the image
            bmp2 = new Bitmap(bmp);
            Graphics G = Graphics.FromImage(bmp2);
            G.DrawString(textBox1.Text, new Font("Arial", 14, FontStyle.Bold),
Brushes.Red, x, y);
            pictureBox2.Image = bmp2;
            pictureBox2.Refresh();
            x = x + 10;
            if (x >= bmp.Width) { y = y + 10; x = 0; }
            if (y >= bmp.Height) y = 0;
        }

        private void timer2_Tick(object sender, EventArgs e)
        {
            // Moving an image over the image
            bmp2 = new Bitmap(bmp);
            Graphics G = Graphics.FromImage(bmp2);

            G.DrawImage(new Bitmap("Bird.bmp"), x, y);
            pictureBox2.Image = bmp2;
            pictureBox2.Refresh();
            x = x + 10;
            if (x >= bmp.Width) { y = y + 10; x = 0; }
            if (y >= bmp.Height) y = 0;
        }

        private void button5_Click(object sender, EventArgs e)
        {
            //Starting image moving over the image
            timer2.Enabled = true;
        }

    }
}
```

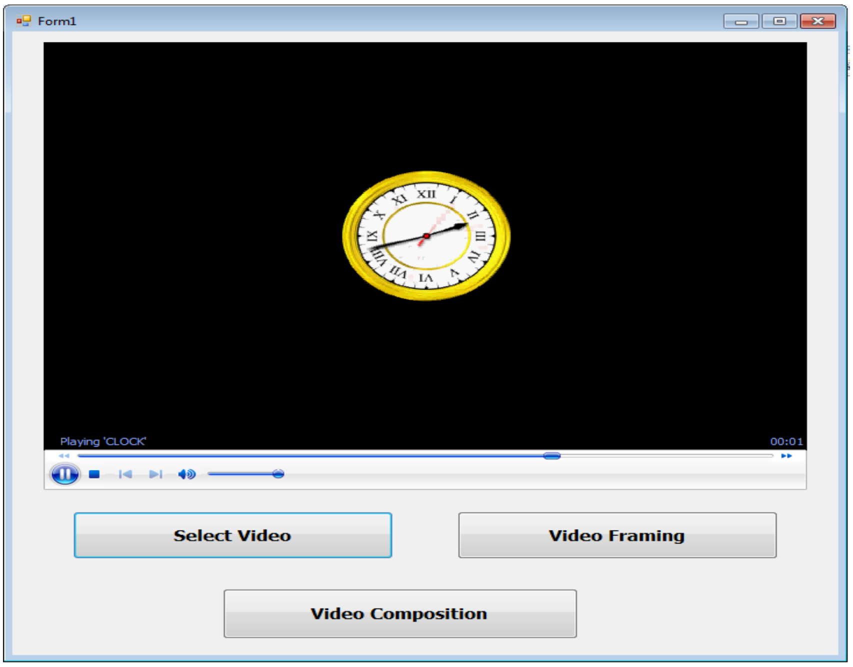
Homework:

*Write C# program to **zoom in** a text gradually form the center of the image.*

Multimedia Lab. 2020-2021 / 4th Class/ Morning Study

Lab. 2 : Video Framing and Composition

Design:



Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.Structure;


namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        Capture capture;
        List<Bitmap> FramList;
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //Video Loading
            if (openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                axWindowsMediaPlayer1.URL = openFileDialog1.FileName;
            }
        }
        private void button2_Click(object sender, EventArgs e)
        {
            // Video Framing
            FramList = new List<Bitmap>();

            capture = new Capture(openFileDialog1.FileName);

            Image<Bgr, Byte> frame = capture.QueryFrame();

            while (frame != null)
            {
                FramList.Add(frame.ToBitmap());
                frame = capture.QueryFrame();
            }

            for (int i = 0; i < FramList.Count; i++)
            {
                FramList[i].Save("Frame" + i.ToString() + ".bmp");
            }

        }

        private void button3_Click(object sender, EventArgs e)
        {
            //Video Composition
            Bitmap bmp = new Bitmap("Frame0.bmp");
            VideoWriter Writer = new VideoWriter("Test.avi", 30, bmp.Width, bmp.Height,
true);
            for (int i = 0; i < FramList.Count; i++)
            {
                Image<Bgr, Byte> temp = new Image<Bgr, byte>("Frame" + i.ToString() +
".bmp");
                Writer.WriteFrame(temp);
            }

            Writer.Dispose();
        }
    }
}
```

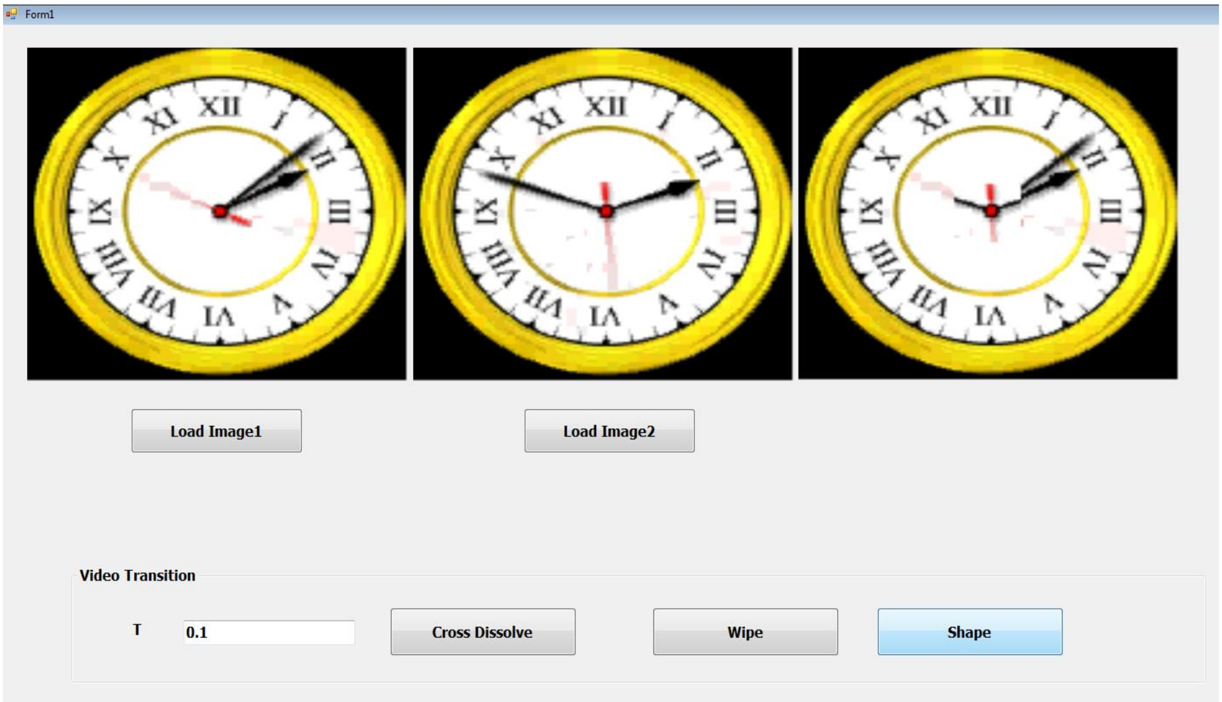
Homework:

Write C# program to convert video frames into grey scale format and then saving them to a new video in a reverse order.

Multimedia Lab. 2020-2021 / 4th Class/ Morning Study

Lab. 3 : Video Transitions

Design:



Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        Bitmap bmp1, bmp2, bmp3;
        double tvalue, t;
        int w, h, col,row, xc,yc;
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            openFileDialog1.InitialDirectory = AppDomain.CurrentDomain.BaseDirectory;
        }


        private void button1_Click(object sender, EventArgs e)
        {
            //Image1 Loading
            openFileDialog1.ShowDialog();
            bmp1 = new Bitmap(openFileDialog1.FileName);
            pictureBox1.Image = bmp1;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            //Image2 Loading
            openFileDialog1.ShowDialog();
            bmp2 = new Bitmap(openFileDialog1.FileName);
            pictureBox2.Image = bmp2;
        }

        private void button3_Click(object sender, EventArgs e)
        {
            //Cross Dissolve Method
            w = bmp1.Width;
            h = bmp1.Height;
            bmp3 = new Bitmap(pictureBox1.Image);
            t = Convert.ToDouble(textBox1.Text);
            tvalue = 0;
            timer1.Start();
        }
        private void timer1_Tick(object sender, EventArgs e)
        {
            for (int i = 0; i < w; i++)
            {
                for (int j = 0; j < h; j++)
                {
                    int A = bmp1.GetPixel(i, j).ToArgb();
                    int B = bmp2.GetPixel(i, j).ToArgb();
                    int AB = Convert.ToInt32(((1 - tvalue) * A) + (tvalue * B));
                    bmp3.SetPixel(i, j, Color.FromArgb(AB));
                }
            }
            pictureBox3.Image = bmp3;
            pictureBox3.Refresh();
            tvalue = tvalue + t;
            if (tvalue > 1) timer1.Stop();
        }


        private void button4_Click(object sender, EventArgs e)
        {
            //Wipe Method
            w = bmp1.Width;
            h = bmp1.Height;
            bmp3 = new Bitmap(pictureBox1.Image);
            col = 0;
            timer2.Start();
        }

        private void timer2_Tick(object sender, EventArgs e)
        {
            for (int j = 0; j < h; j++)
            {
                Color A = bmp2.GetPixel(col, j);
                bmp3.SetPixel(col, j, A);
            }

            pictureBox3.Image = bmp3;
            pictureBox3.Refresh();
            col++;
            if (col >= w) timer2.Stop();
        }
        private void button5_Click(object sender, EventArgs e)
        {
            //Shape Method
            w = bmp1.Width;
            h = bmp1.Height;
            xc = w / 2;
            yc = h / 2;
            bmp3 = new Bitmap(pictureBox1.Image);
            col = 0;
            row = 0;
            timer3.Start();
        }

        private void timer3_Tick(object sender, EventArgs e)
        {
            for (int i = xc-col ; i < xc+col; i++)
            {
                for (int j = yc-row  ; j < yc+row  ; j++)
                {
                    Color A = bmp2.GetPixel(i, j);
                    bmp3.SetPixel(i, j, A);
                }
            }
            pictureBox3.Image = bmp3;
            pictureBox3.Refresh();
           if ((col <(w/2))&&(row <(h/2))) {
            row++;
            col++;
           }
           else if ((col < (w / 2)) && (row >= (h / 2)))
           {
               col++;
               row=(h / 2)-1;

           }
           else if ((col >=(w / 2)) && (row <(h / 2)))
           {
               row ++;
               col = (w / 2)-1;
           }
            else    timer3.Stop();

        }

    }

}
```

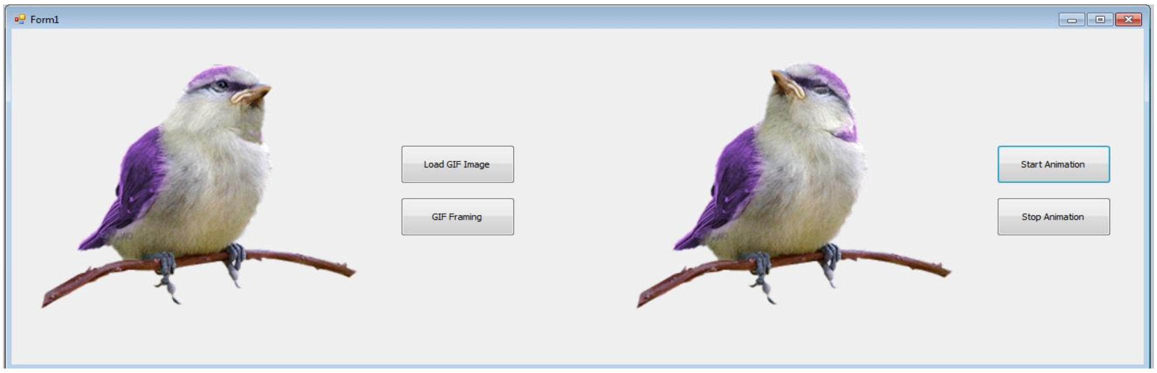
Homework:

*Write C# program to implement **(1)Spilt** and **(2)Clock** video transition methods.*

Multimedia Lab. 2020-2021 / 4th Class/ Morning Study

Lab. 4 : Image Animation

Design:



Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Drawing.Imaging;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        Bitmap gif;
        List<Bitmap> Frames;
        int fcount;
        int FrameNum = 0;
        public Form1()
        {
            InitializeComponent();
        }

        private void Button4_Click(object sender, EventArgs e)
        {
            //GIF image loading
            OpenFileDialog1.ShowDialog();
            gif = new Bitmap(OpenFileDialog1.FileName);
            PictureBox1.Image = gif;
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            OpenFileDialog1.InitialDirectory = AppDomain.CurrentDomain.BaseDirectory;
            Timer1.Interval = 100;
        }

        private void Button5_Click(object sender, EventArgs e)
        {
            //GIF image framing
            fcount = gif.GetFrameCount(FrameDimension.Time);

            for (int i = 0; i < fcount; i++)
            {
                gif.SelectActiveFrame(FrameDimension.Time, i);
                PictureBox1.Invalidate();
                Bitmap bmp = new Bitmap(PictureBox1.Image);
                bmp.Save("GIFFrame" + i.ToString() + ".bmp");
                Application.DoEvents();
                System.Threading.Thread.Sleep(1000);
            }
            MessageBox.Show("Done");
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            //Starting image animation
            Frames = new List<Bitmap>();
            for (int i = 0; i < fcount; i++)
            {
                Frames.Add(new Bitmap("GIFFrame" + i.ToString() + ".bmp"));
            }
            Timer1.Enabled = true;

        }

        private void Timer1_Tick(object sender, EventArgs e)
        {
            FrameNum = (FrameNum + 1) % fcount;
            pictureBox2.Image = Frames[FrameNum];
        }

        private void Button2_Click(object sender, EventArgs e)
        {
            //Stoping image animation
            Timer1.Enabled = false;
        }

    }
}
```

Homework:

1. Try to save a sequence of image frames as a *gif* image file format.

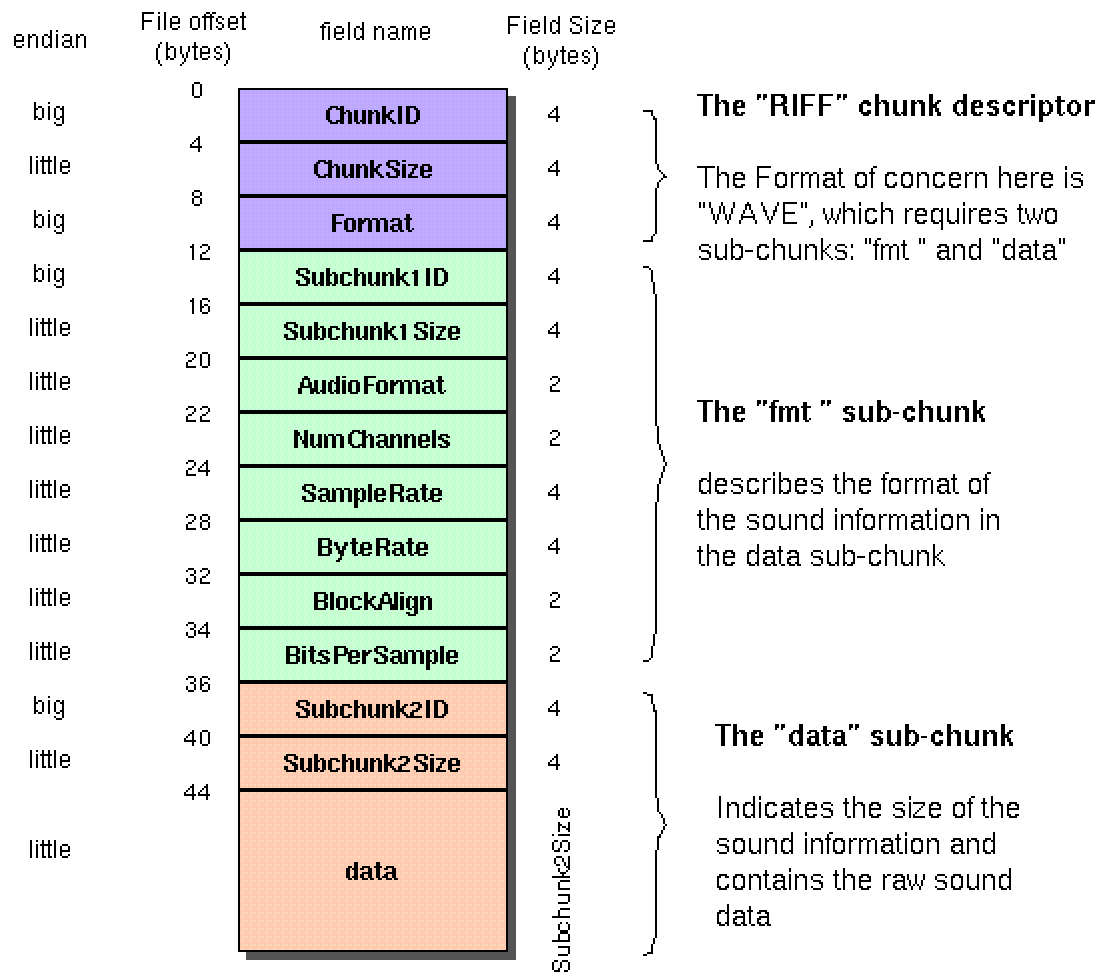
2. Try to control the animation of the loaded *gif* image (stop animation and resume animation controls).

3. Convert *gif* image format into *avi* video format.

Lab. 5: WAVE File Description

The WAVE file format is a subset of Microsoft's RIFF specification for the storage of multimedia files. A RIFF file starts out with a file header followed by a sequence of data chunks. The header is used to provide specifications on the file type, sample rate, sample size and bit size of the file, as well as its overall length. The header of a WAV (RIFF) file is 44 bytes long. A WAVE file is often just a RIFF file with a single "WAVE" chunk which consists of two sub-chunks -- a "fmt " chunk specifying the data format and a "data" chunk containing the actual sample data. Call this form the "Canonical form".

The Canonical WAVE file format



endian	File offset (bytes)	field name	Field Size (bytes)
big	0	ChunkID	4
little	4	ChunkSize	4
big	8	Format	4
big	12	Subchunk1ID	4
little	16	Subchunk1 Size	4
little	20	AudioFormat	2
little	22	NumChannels	2
little	24	SampleRate	4
little	28	ByteRate	4
little	32	BlockAlign	2
little	34	BitsPerSample	2
big	36	Subchunk2ID	4
little	40	Subchunk2Size	4
little	44	data	Subchunk2Size

The "RIFF" chunk descriptor

The Format of concern here is "WAVE", which requires two sub-chunks: "fmt " and "data"

The "fmt " sub-chunk

describes the format of the sound information in the data sub-chunk

The "data" sub-chunk

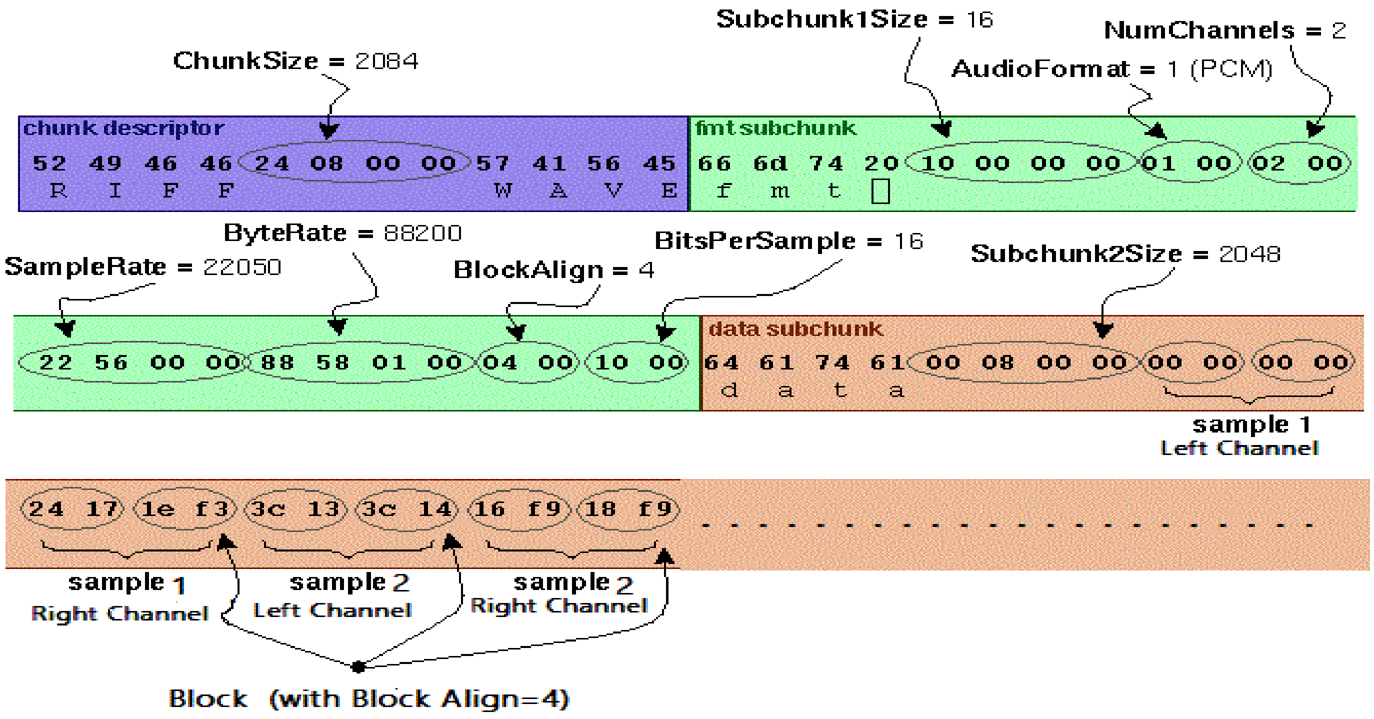
Indicates the size of the sound information and contains the raw sound data

The canonical WAVE format starts with the RIFF header:

Offset	Size	Name	Description
0	4	**ChunkID**	Contains the letters "RIFF" in ASCII form (0x52494646 big-endian form).
4	4	**ChunkSize**	36 + SubChunk2Size, or more precisely: 4 + (8 + SubChunk1Size) + (8 + SubChunk2Size) This is the size of the rest of the chunk following this number. This is the size of the

```
                               entire file in bytes minus 8 bytes for the
                               two fields not included in this count:
                               ChunkID and ChunkSize.
8         4   Format           Contains the letters "WAVE"
                               (0x57415645 big-endian form).

The "WAVE" format consists of two subchunks: "fmt " and "data":
The "fmt " subchunk describes the sound data's format:

12        4   Subchunk1ID      Contains the letters "fmt "
                               (0x666d7420 big-endian form).
16        4   Subchunk1Size    16 for PCM.  This is the size of the
                               rest of the Subchunk which follows this number.
20        2   AudioFormat      PCM = 1 (i.e. Linear quantization)
                               Values other than 1 indicate some
                               form of compression.
22        2   NumChannels      Mono = 1, Stereo = 2, etc.
24        4   SampleRate       8000, 44100, etc.
28        4   BitRate         == SampleRate * NumChannels * BitsPerSample
32        2   BlockAlign       == NumChannels * BitsPerSample/8
                               The number of bytes for one sample including
                               all channels.
34        2   BitsPerSample    8 bits = 8, 16 bits = 16, etc.


The "data" subchunk contains the size of the data and the actual sound:

36        4   Subchunk2ID      Contains the letters "data"
                               (0x64617461 big-endian form).
40        4   Subchunk2Size    == NumSamples * NumChannels * BitsPerSample/8
                               This is the number of bytes in the data.
                               You can also think of this as the size
                               of the read of the subchunk following this
                               number.
44        *   Data             The actual sound data.
```

As an example, here are the bytes of the opened WAVE file shown as hexadecimal numbers:



Important Definitions

- ## Sampling Rate

Sampling rate or sampling frequency defines the number of samples per second taken from a continuous signal to make a discrete or digital signal. Frequencies are measured in in hertz (Hz) per second. The Nyquist–Shannon sampling theorem (Nyquist principle) states that perfect reconstruction of a signal is possible when the sampling frequency is greater than twice the maximum frequency of the signal being sampled. For example, if an audio signal has an upper limit of 20,000 Hz (the approximate upper limit of human hearing), a sampling frequency greater than 40,000 Hz (40 kHz) will avoid aliasing and allow theoretically perfect reconstruction.

- ## Bits per Sample

Bits per Sample (Bit depth or sample size) is the number of bits of information in each sample, and it directly corresponds to the resolution of each sample. Bit size determines how much information can be stored in a file. For most of today's purposes, bit size should be 16 bit. 8 bit files are smaller (1/2 the size), but have less resolution. Bit size deals with amplitude. In 8 bit recordings, a total of 256 (0 to 255) amplitude levels are available. In 16 bit, a total of 65,536 (-32768 to 32767) amplitude levels are available.

Bit Rate

Bite rate is a measure of the speed of data processing usually calculated as the number of bits per second (the average number of bits required for one second of sound).

- ## Channels

Channels are the number of separate recording elements in the data. For a real quick example, one channel is mono and two channels are stereo.

- ## Wave File Data

The data is the individual samples. An individual sample is the bit size times the number of channels. For example, a monaural (single channel), eight bit recording has an individual sample size of 8 bits. A monaural sixteen-bit recording has an individual sample size of 16 bits. A stereo sixteen-bit recording has an individual sample size of 32 bits. Samples are placed end-to-end to form the data. So, for example, if you have four samples (s1, s2, s3, s4) then the data would look like: s1s2s3s4.

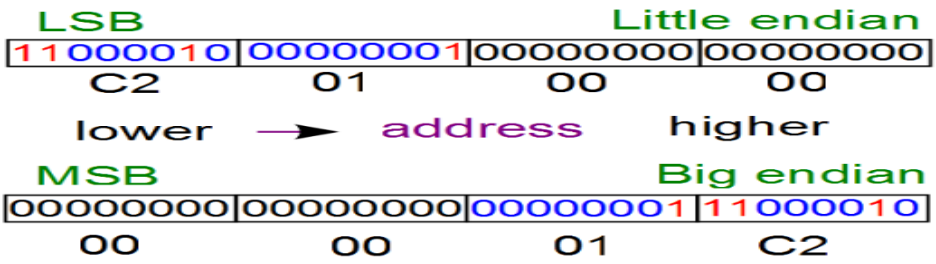
• *RIFF (Resource Interchange File Format) Files*

Multimedia applications require the storage and management of a wide variety of data, including bitmaps, audio data, video data, and peripheral device control information. RIFF provides a way to store all these varied types of data. The type of data a RIFF file contains is indicated by the file extension. Examples of data that may be stored in RIFF files are:

- Audio/visual interleaved data (.AVI)
- Waveform data (.WAV)
- Bitmapped data (.RDI)
- MIDI information (.RMI)
- Color palette (.PAL)
- Multimedia movie (.RMN)
- Animated cursor (.ANI)
- A bundle of other RIFF files (.BND)

• *Big-endian and little-endian*

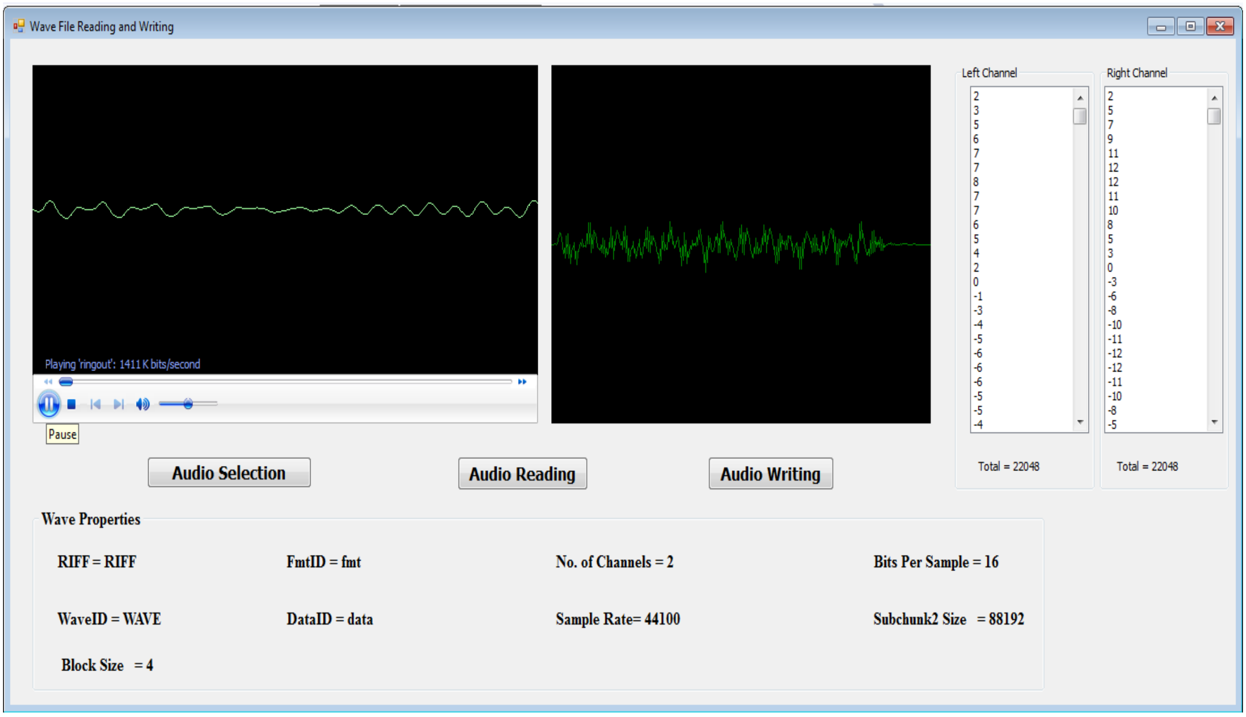
Big-endian and little-endian are terms that describe the order in which a sequence of bytes are stored in computer memory. Big-endian is an order in which the "big end" (most significant value in the sequence) is stored first (at the lowest storage address). Little-endian is an order in which the "little end" (least significant value in the sequence) is stored first. For example, the two bytes required by the number 450 (in the hexadecimal = x000001C2) will be stored in a little-endian and big-endian systems as following:



Multimedia Lab. 2020-2021 / 4th Class/ Morning Study

Lab. 5 & 6: Audio

Design:



Wave File Reading and Writing

Playing 'ringout': 1411 K bits/second

Pause

Audio Selection

Audio Reading

Audio Writing

Left Channel	Right Channel
2	2
3	5
5	7
6	9
7	11
7	12
8	12
7	11
7	10
6	8
5	5
4	3
2	0
0	-3
-1	-6
-3	-8
-4	-10
-5	-11
-6	-12
-6	-12
-6	-11
-5	-10
-5	-8
-4	-5
Total = 22048	Total = 22048

Wave Properties

RIFF = RIFF   FmtID = fmt   No. of Channels = 2   Bits Per Sample = 16

WaveID = WAVE   DataID = data   Sample Rate= 44100   Subchunk2 Size = 88192

Block Size = 4

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        struct WavHeader
        {
            public char[] riffID; // Chunk ID "riff" (4 bytes)
            public uint size;  // Chunk Size (4 bytes)
            public char[] wavID;  //Format  "WAVE" (4 byts)
            public char[] fmtID;  // Subchunk1ID   "fmt " (4bytes)
            public uint fmtSize; // Subchunk1Size (4 bytes)
            public ushort format; // AudioFormat  (2 bytes)
            public ushort channels; // NumChannels  (2 bytes)
            public uint sampleRate; //  SampleRate  (4 byts)
            public uint bytePerSec; // ByteRate     (4 bytes)
            public ushort blockSize; //BlockAlign      (2 bytes)
            public ushort bit;  // BitsPerSample   (2 bytes)
            public char[] dataID; // Subchunk2ID "data"  (4 bytes)
            public uint dataSize; // Subchunk2Size    (4 bytes)
        }
        WavHeader Header = new WavHeader();
        List<object > lDataList = new List<object >();
        List<object > rDataList = new List<object >();

        public Form1()
        {
            InitializeComponent();
        }

        private void button3_Click(object sender, EventArgs e)
        {
             //Audio File Selection

            openFileDialog1.InitialDirectory = AppDomain.CurrentDomain.BaseDirectory;
            openFileDialog1.Filter = "(wav)|*.wav";

            if (openFileDialog1.ShowDialog() == DialogResult.OK)

                axWindowsMediaPlayer1.URL = openFileDialog1.FileName;
        }


        private void button1_Click(object sender, EventArgs e)
        {

             //Audio File Reading
            listBox1.Items.Clear();
            listBox2.Items.Clear();
            FileStream fs = new FileStream(openFileDialog1.FileName, FileMode.Open,
FileAccess.Read);
            BinaryReader br = new BinaryReader(fs);

            Header.riffID = br.ReadChars(4);
            Header.size = br.ReadUInt32();
            Header.wavID = br.ReadChars(4);
            Header.fmtID = br.ReadChars(4);
            Header.fmtSize = br.ReadUInt32();
            Header.format = br.ReadUInt16();
            Header.channels = br.ReadUInt16();
            Header.sampleRate = br.ReadUInt32();
            Header.bytePerSec = br.ReadUInt32();
            Header.blockSize = br.ReadUInt16();
            Header.bit = br.ReadUInt16();
            Header.dataID = br.ReadChars(4);
            Header.dataSize = br.ReadUInt32();

            label1.Text = "RIFF = ";
            label2.Text = "WaveID = ";
            label3.Text = "FmtID = ";
            label4.Text = "DataID = ";
            for (int i = 0; i < 4; i++)
            {
                label1.Text = label1.Text + Header.riffID[i].ToString();
                label2.Text = label2.Text + Header.wavID[i].ToString();
                label3.Text = label3.Text + Header.fmtID[i].ToString();
                label4.Text = label4.Text + Header.dataID[i].ToString();

            }
            label5.Text = "No. of Channels = " + Header.channels;
            label6.Text = "Sample Rate= " + Header.sampleRate;
            label7.Text = "Bits Per Sample = " + Header.bit ;
            label8.Text = "Subchunk2 Size    = " + Header.dataSize ;
            label11.Text = " Block Size    = " + Header.blockSize ;

            lDataList.Clear();
            rDataList.Clear();

            if (Header.channels == 1)
            {
                if (Header.bit == 8)
                {
                    for (int i = 0; i < Header.dataSize / Header.blockSize; i++)
                    {
                        lDataList.Add(br.ReadByte());
                    }
                }
                else if (Header.bit == 16)
                    {
                        for (int i = 0; i < Header.dataSize/Header.blockSize; i++)
                        {
                            lDataList.Add(br.ReadInt16 ());
                        }
                    }


                for (int i = 0; i < lDataList.Count; i++)
                {
                    listBox1.Items.Add(lDataList[i]);
                }


            }
            else
            {
                if (Header.bit == 8)
                {

                    for (int i = 0; i < Header.dataSize/Header.blockSize ; i ++)
                    {
                        lDataList.Add(br.ReadByte());
                        rDataList.Add(br.ReadByte());
                    }
                }
                else if (Header.bit == 16)
                {

                    for (int i = 0; i < Header.dataSize / Header.blockSize; i ++)
                    {
                        lDataList.Add(br.ReadInt16());
                        rDataList.Add(br.ReadInt16());
                    }
                }
                for (int i = 0; i < lDataList.Count; i++)
                {
                    listBox1.Items.Add(lDataList[i]);
                }
                for (int i = 0; i < rDataList.Count; i++)
                {
                    listBox2.Items.Add(rDataList[i]);
                }

            }

            label9.Text = "Total = " + lDataList.Count;
            label10.Text = "Total = " + rDataList.Count;


            // Set up for drawing
            pictureBox1.BackColor = Color.Black;
            Bitmap canvas = new Bitmap(pictureBox1.Width, pictureBox1.Height);
            Graphics offScreenDC = Graphics.FromImage(canvas);
            Pen pen = new System.Drawing.Pen(Color.WhiteSmoke);

            // Determine channnel boundries
            int width = canvas.Width;
            int height = canvas.Height;
            double center = height / 2;

            // Draw left channel
            double scale = 0.5 * height / 32768;  // a 16 bit sample has values from -
32768 to 32767
            int xPrev = 0, yPrev = 0;
            for (int x = 0; x < width; x++)
            {
                int y = (int)(center + (Convert.ToInt16(lDataList[lDataList.Count /
width * x]) * scale));
                if (x == 0)
                {
                    xPrev = 0;
                    yPrev = y;
                }
                else
                {
                    pen.Color = Color.Green;
                    offScreenDC.DrawLine(pen, xPrev, yPrev, x, y);
                    xPrev = x;
                    yPrev = y;
                }
            }

            // Clean up
            pictureBox1.Image = canvas;
            offScreenDC.Dispose();

        }


        private void button2_Click(object sender, EventArgs e)
        {
             //Audio Writing

            List<object > lNewDataList = lDataList;
            List<object > rNewDataList = rDataList;


            FileStream fs = new FileStream("NWave.wav", FileMode.Create,
FileAccess.Write);
            BinaryWriter bw = new BinaryWriter(fs);

                    bw.Write(Header.riffID);
                    bw.Write(Header.size);
                    bw.Write(Header.wavID);
                    bw.Write(Header.fmtID);
                    bw.Write(Header.fmtSize);
                    bw.Write(Header.format);
                    bw.Write(Header.channels);
                    bw.Write(Header.sampleRate);
                    bw.Write(Header.bytePerSec);
                    bw.Write(Header.blockSize);
                    bw.Write(Header.bit);
                    bw.Write(Header.dataID);
                    bw.Write(Header.dataSize);

                    if (Header.channels == 1)
                    {
                        if (Header.bit == 8)
                        {
                          for (int i = 0; i < Header.dataSize / Header.blockSize; i++)
                            {
                                bw.Write(Convert.ToByte(lNewDataList[i]));
                            }
                        }
                        else if (Header.bit == 16)
                        {
                            for (int i = 0; i < Header.dataSize / Header.blockSize;
i++)
                            {
                                bw.Write(Convert.ToInt16(lNewDataList[i]));
                            }
                        }

                    }
                    else
                    {
                        if (Header.bit == 8)
                        {
                           for (int i = 0; i < Header.dataSize / Header.blockSize; i++)
                            {
                                bw.Write(Convert.ToByte (lNewDataList[i]));
                                bw.Write(Convert.ToByte(rNewDataList[i]));

                            }
                        }
                        else if (Header.bit == 16)
                        {
                          for (int i = 0; i < Header.dataSize / Header.blockSize; i++)
                            {
                                bw.Write(Convert.ToInt16(lNewDataList[i]));
                                bw.Write(Convert.ToInt16(rNewDataList[i]));

                            }
                        }

                    }

                    bw.Close();

        }

    }
}
```

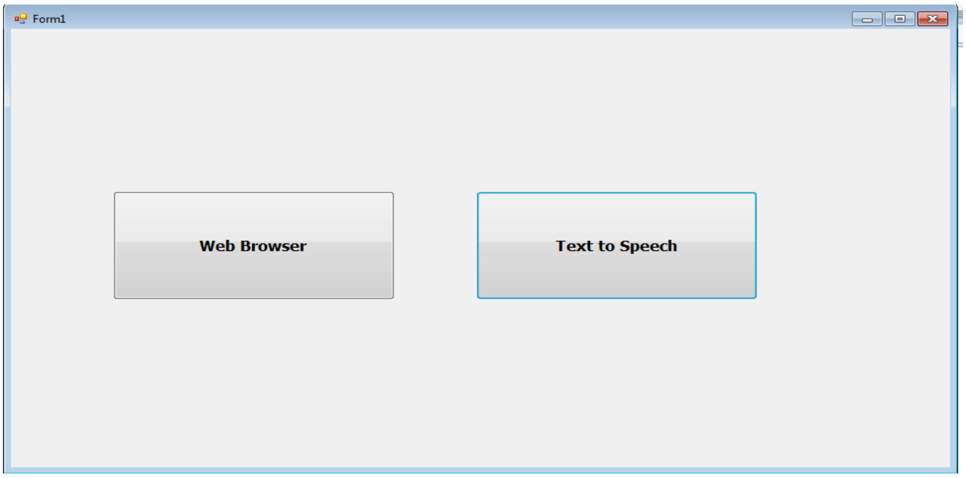
Homework

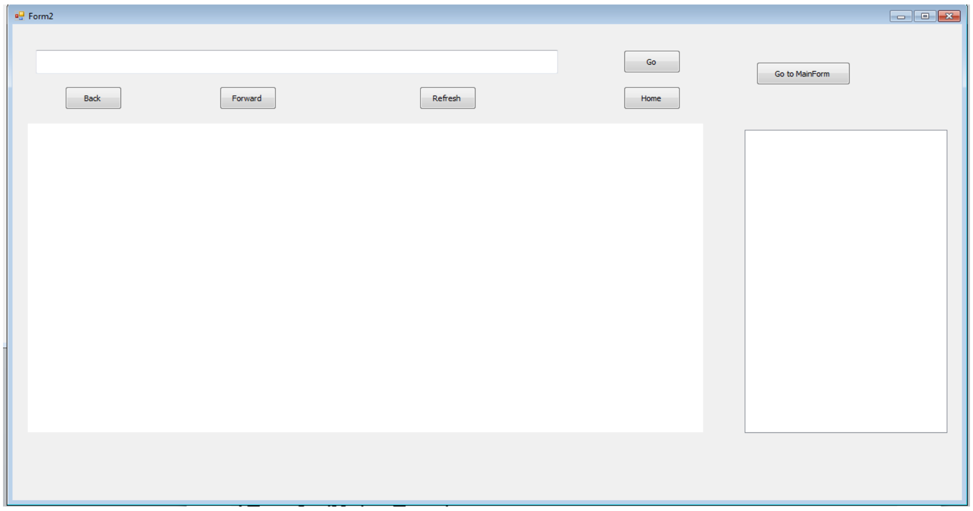
- *Try to design **audio cutter** software.*
- *Using Microsoft C#, load a stereo wave file and then convert it to a mono wave.*

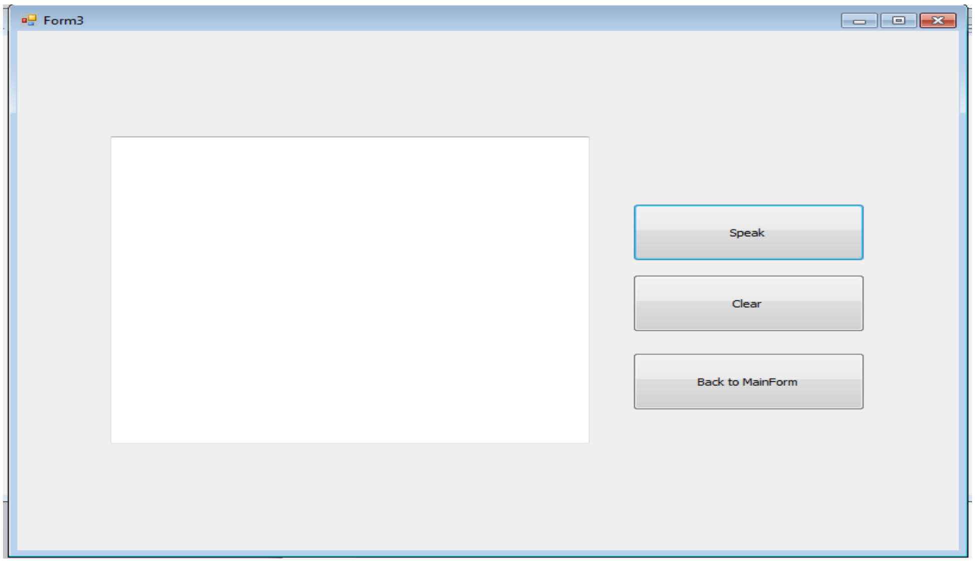
Multimedia Lab. 2020-2021 / 4th Class/ Morning Study

Lab.7: Multi-Form (Text to Speech+ Web Browser)

Design:







Code:

//Form1 (Main Form)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form3 obj = new Form3();
            obj.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {

            Form2 obj = new Form2();
            obj.Show();
        }
    }
}
```

//Form2 (Simple Web Browser Creation)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form2 : Form
    {
        int i = 0;
        public Form2()
        {
            InitializeComponent();
        }
        private void Button1_Click(object sender, EventArgs e)
        {
            WebBrowser1.Navigate(TextBox1.Text);
            ListBox1.Items.Add(TextBox1.Text);
            i = i + 1;
        }

        private void Button2_Click(object sender, EventArgs e)
        {
            WebBrowser1.GoBack();
            i = i - 1;
            TextBox1.Text = ListBox1.Items[i].ToString();
        }

        private void Button3_Click(object sender, EventArgs e)
        {
            WebBrowser1.GoForward();
        }

        private void Button4_Click(object sender, EventArgs e)
        {
            WebBrowser1.Refresh();
        }

        private void Button5_Click(object sender, EventArgs e)
        {
            WebBrowser1.GoHome();
        }

        private void Button6_Click(object sender, EventArgs e)
        {
            this.Hide();
        }
    }
}
```

//Form3 (Text to Speech Conversion)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using SpeechLib;

namespace WindowsFormsApplication1
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            SpVoice sp = new SpVoice();
            sp.Volume = 100;
            if (TextBox1.Text == "")
            {
                MessageBox.Show("Please Type Text To Speak!");
                TextBox1.Focus();
            }
            else
                sp.Speak(TextBox1.Text,
SpeechVoiceSpeakFlags.SVSFDefault);

        }

        private void Button2_Click(object sender, EventArgs e)
        {
            TextBox1.Clear();
        }

        private void Button3_Click(object sender, EventArgs e)
        {
            this.Hide();
        }
    }
}
```

Lab. 8 & 9: Adobe Photoshop

Aim of the experiment

Adobe Photoshop is the industry-standard tool for digital imaging, which makes Photoshop expertise a valuable commodity in the workplace. Learning Photoshop is also a good way to learn imaging concepts. Concepts you learn from working with Photoshop apply to other imaging tools as well (including Photoshop Elements, which has a very similar user interface and features).

Theory

Adobe Photoshop is a graphics editing program developed and published by Adobe Systems.

Adobe's 2003 "Creative Suite" rebranding led to Adobe Photoshop 8's renaming to Adobe Photoshop CS. Thus, Adobe Photoshop CS6 is the 13th major release of Adobe Photoshop. The CS rebranding also resulted in Adobe offering numerous software packages containing multiple Adobe programs for a reduced price. Adobe Photoshop is released in two editions: **Adobe Photoshop**, and **Adobe Photoshop Extended**, with the Extended having extra 3D image creation, motion graphics editing, and advanced image analysis features. Adobe Photoshop Extended is included in all of Adobe's Creative Suite offerings except Design Standard, which includes the Adobe Photoshop edition.

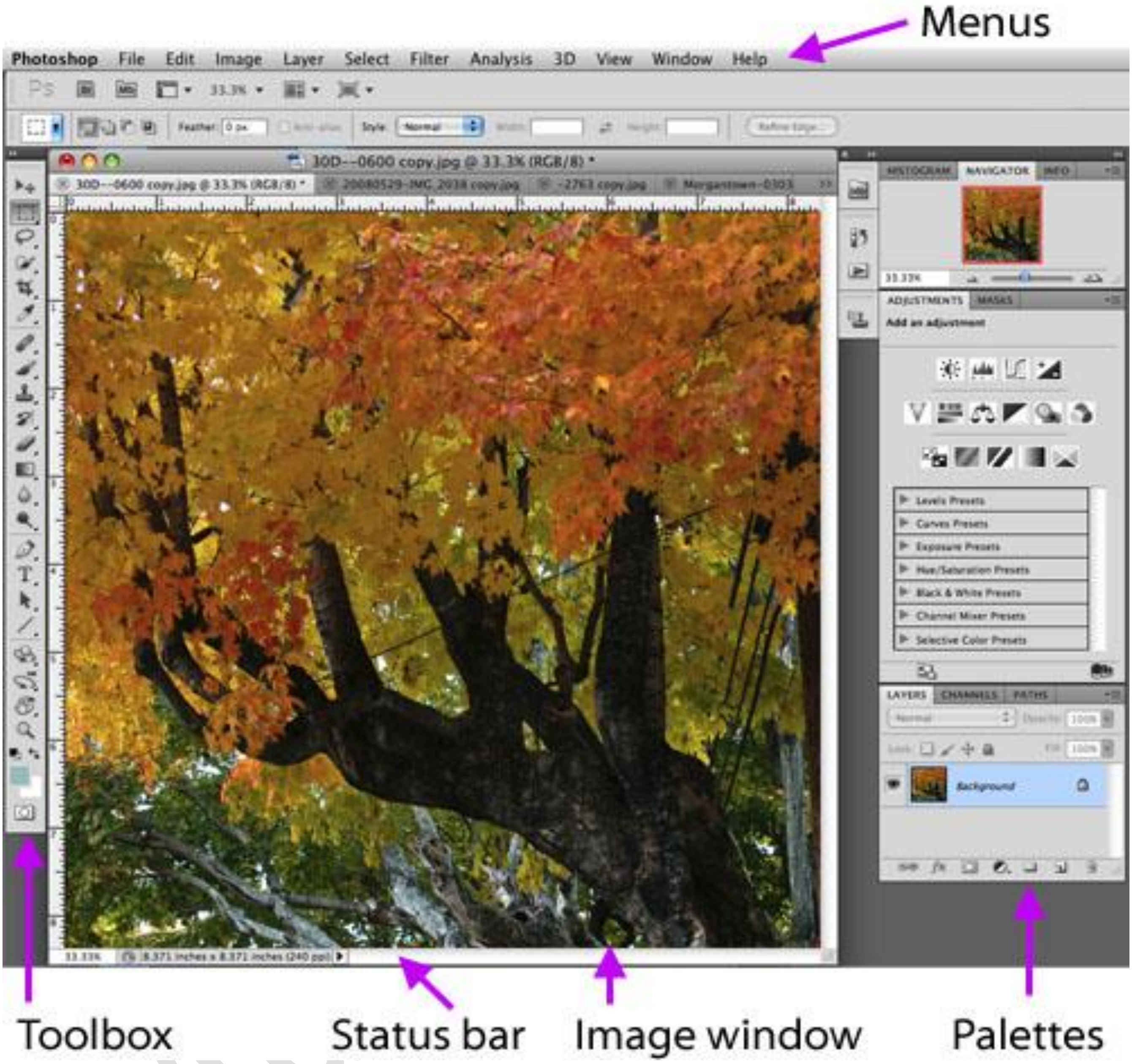
Alongside Photoshop and Photoshop Extended, Adobe also publishes Photoshop Elements and Photoshop Lightroom, collectively called "The Adobe Photoshop Family". In 2008, Adobe released Adobe Photoshop Express, a free web-based image editing tool to edit photos directly on blogs and social networking sites; in 2011 a version was released for the Android operating system and the iOS operating system.

Adobe only supports Windows and Macintosh versions of Photoshop, but using Wine, Photoshop CS5 can run reasonably well on Linux

Photoshop files have default file extension as **.PSD**, which stands for "Photoshop Document." A PSD file stores an image with support for most imaging options available in Photoshop. These include layers with masks, color spaces, ICC profiles, CMYK Mode (used for commercial printing), transparency, text, alpha channels and spot colors, clipping paths, and duotone settings. This is in contrast to many other file formats (e.g. .JPG or .GIF) that restrict content to provide streamlined, predictable functionality. A PSD file has a maximum height and width of 30,000 pixels, and a length limit of 2 Gigabytes.

Photoshop files sometimes have the file extension **.PSB**, which stands for "Photoshop Big" (also known as "large document format"). A PSB file extends the PSD file format, increasing the maximum height and width to 300,000 pixels and the length limit to around 4 Exabytes. The dimension limit was apparently chosen arbitrarily by Adobe, not based on computer arithmetic constraints (it is not close to a power of two, as is 30,000) but for ease of software testing. PSD and PSB formats are documented

Getting Started


Menus
Photoshop File Edit Image Layer Select Filter Analysis 3D View Window Help
30D--0600 copy.jpg @ 33.3% (RGB/8) *
Toolbox
Status bar
Image window
Palettes

Components of the Photoshop desktop include the menu bar, image window, toolbox, status bar, and palettes.

The Menu Bar

The menu bar, at the top of the Photoshop desktop, includes several dropdown menus for choosing commands. Right below the menu bar, options for the currently-selected tool are displayed (see Toolbox tools, below).

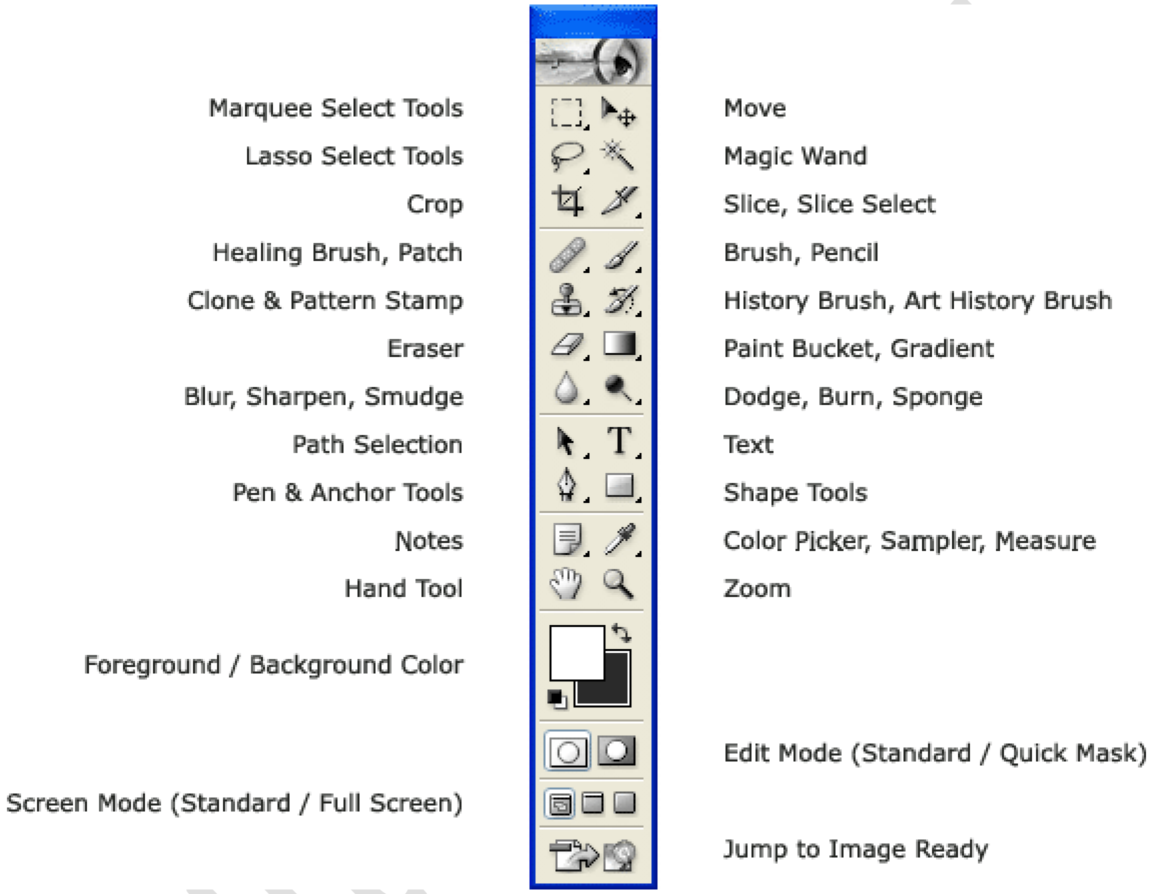
Image Window

The image window displays the current image. The name of the image file appears at the top edge of the image window. If rulers are turned on, they appear at the top and left edges of the image window. To display or hide rulers, choose Ctrl-R or View>Rulers.

About Rulers: Rulers show the size of your image. To change the unit of measurement (such as from pixels to inches), double click on the ruler, or go to Preferences (Edit>Preferences>Units & Rulers). If you are creating an image for the Web, use pixels as the unit of measurement.

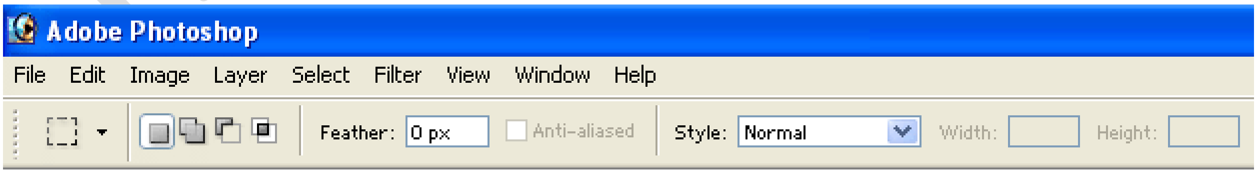
The Toolbox

The toolbox contains the main tools for working on images. Click any tool to select and use it.A small arrow next to a tool in the toolbox indicates that the tool also has additional options available. In Photoshop, click and hold your mouse on a tool to see its options. For example, if you click and hold on the select tool, you'll see select options such as eliptical selction, single row selection, etc.



Tool Options

When you select a tool, additional options appear in the Tool Options bar (by default this is located just below the main menu). The example below shows the tool options bar when the rectangular marquee tool is active.



Feather: 0 px

☐ Anti-aliased

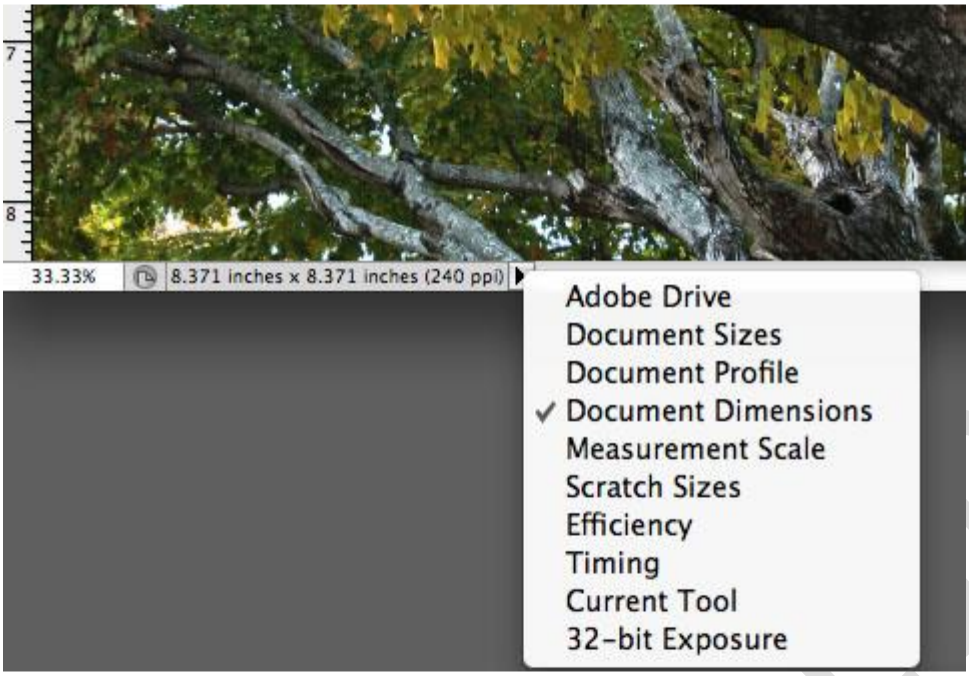
Style: Normal

Width:

Height:

The Status Bar

The status bar at the bottom of the Photoshop desktop provides information about your current view and other information that you can specify by choosing from a dropdown menu (in this example, the view is now 33.33% and the document dimensions are 8.371" x 8.371").



Palettes

Palettes enable you to perform a wide variety of tasks in Photoshop. To display or hide particular palettes, go to the Window menu and choose a palette name.

Palettes are usually organized into groups. In the above example, Layers, Channels, and Paths are a group. You can easily move a palette group by dragging its blue bar (at the top of the group). To shrink a palette group, double-click on its blue bar. To hide a palette group, click its close box (the "X" on the right side of the blue bar).

You can drag the tabs of palettes to move them around, or to combine them into other groups.

To reset palette locations to the default, choose Window>Workspace>Reset Palette Locations

Image Size, Resolution, and Print Size

Digital images are made up of pixels (picture elements), which can be defined as colored squares. Each pixel is only one color.

A good way to learn about image size, resolution, and print size is to experiment with an image using the Image Size dialog box. To do this, open an image in Photoshop. Then choose Image > Image Size, and the Image Size dialog box appears.

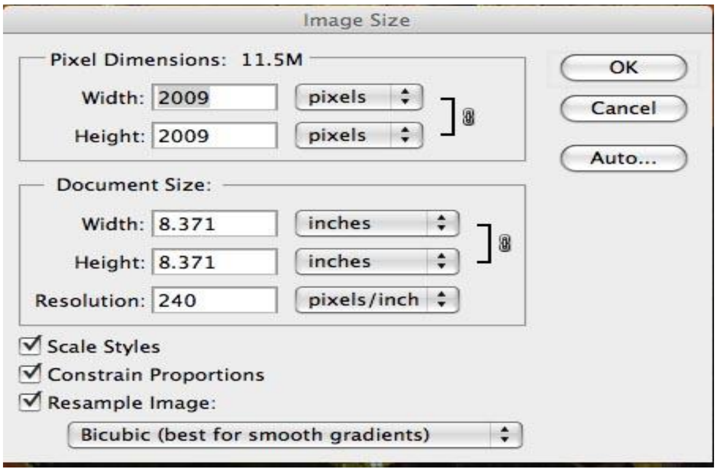


Image Size

Pixel Dimensions: 11.5M

Width: 2009 pixels

Height: 2009 pixels

Document Size:

Width: 8.371 inches

Height: 8.371 inches

Resolution: 240 pixels/inch

☑ Scale Styles

☑ Constrain Proportions

☑ Resample Image:

Bicubic (best for smooth gradients)

Under *Pixel Dimensions*, Width and Height refer to the number of pixels in an image, which has to do with the size of display on screen.

Under *Document Size*, Width and Height refer to the size of the image when printed.

For Online Display

If you are placing an image on the Web, you may want to **resample** the image (add or delete pixels) in order to change the display size of the image. To resample, make sure "Resample Image" is checked. Unless you want to distort your image, also be sure that "Constrain Proportions" is checked. Then type in a new number of pixels in either the width or height field of Pixel Dimensions.

Sampling up (increasing the number of pixels) makes images larger on a display screen. For sampling up, we suggest that you choose the Bicubic or Bicubic Smoother option from the Resample Image dropdown menu.

Sampling down (decreasing the number of pixels) makes images smaller. For sampling down, we suggest that you choose either Bicubic or Bicubic Sharper from the Resample Image dropdown menu.

Note: The resolution at which an image will display on a monitor is not related to image resolution. Typically the display resolution is around 100 pixels per inchi. This is the case regardless of the setting under Resolution in in the Image Size dialog box. You do not need to be concerned with this setting if you are placing the image on the Web; it only applies to printing.

For Printing

For printing purposes, you may not need to resample your image. Sampling down reduces the number of pixels, which degrades the quality of the print (but does have the advantage of lowering file size). Sampling up (adding pixels) is not usually necessary, because to increase the print size a better method is to lower the resolution rather than sampling up. (Sampling up would unnecessarily increase the file size.)

To change the print size of an image without resampling, uncheck the "Resample Image" box. When you turn resampling off, notice that the Pixel Dimensions are no longer editable.

Under *Document Size*, there is a tradeoff between Width and Height and the Resolution. If you increase the width and height, resolution decreases. If you increase the resolution, width and height decrease.

As a general guideline, newsletters and newspapers are likely to require resolutions in the range of 150 to 200 ppi, and magazines are likely to prefer high resolution images of 250 ppi or more.

A resolution of more than 300 ppi may be unnecessary. But keep in mind that if you crop the image and then enlarge the cropped image to the size of the original, the resolution will be lower than the original... so an image that was 300 ppi, then cropped and enlarged, may then be much less than 300 ppi.

Color Modes

In Photoshop, you are always working in a **color mode**. The default mode is RGB (Red-Green-Blue), which is also the default mode for the Web. If you are only working to create color images for the Web or for a desktop inkjet printer, you may not need to use any of the other modes. But it is useful to know about modes because this gives you a better idea of how Photoshop creates images.

Each color mode defines the colors that combine to make up the color of every pixel in the image. To change color modes, choose Image > Mode.

Color modes are made up of channels. Each channel is a grayscale image. A grayscale image is an image made up of shades of brightness (as many as 256 shades, from white to black). To view the channels, choose Window > Channels, and the Channels palette will appear. Modes with fewer channels result in smaller file sizes.

For example, in RGB mode, the red channel is a grayscale image made up of brightness values for the color red; the green channel is made up of brightness values for the color green; and the blue channel is made up of brightness values for the color blue. Each pixel in the full color image is made up of a combination of of brightness values for red, green, and blue.

Note: It is possible to edit an individual channel, which is useful to improve the quality of some images. Detailed techniques for working with channels are beyond the scope of this tutorial. For more information refer to one of the resources listed in the Welcome section.

Some of the most frequently used color modes are:

RGB mode

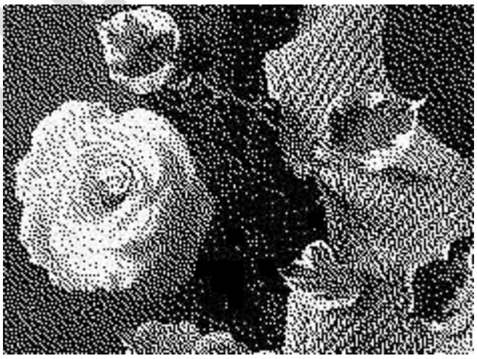
- three channels (Red, Green, and Blue)
- used by all monitors and by the Web
- recommended by Adobe for editing color images

Grayscale mode

One channel, 256 shades of gray

Bitmap mode

- black and white (no shades of gray)
- used for line art



Indexed mode

- one color channel, up to 256 color values
- used for GIF images destined for the Web

CMYK mode

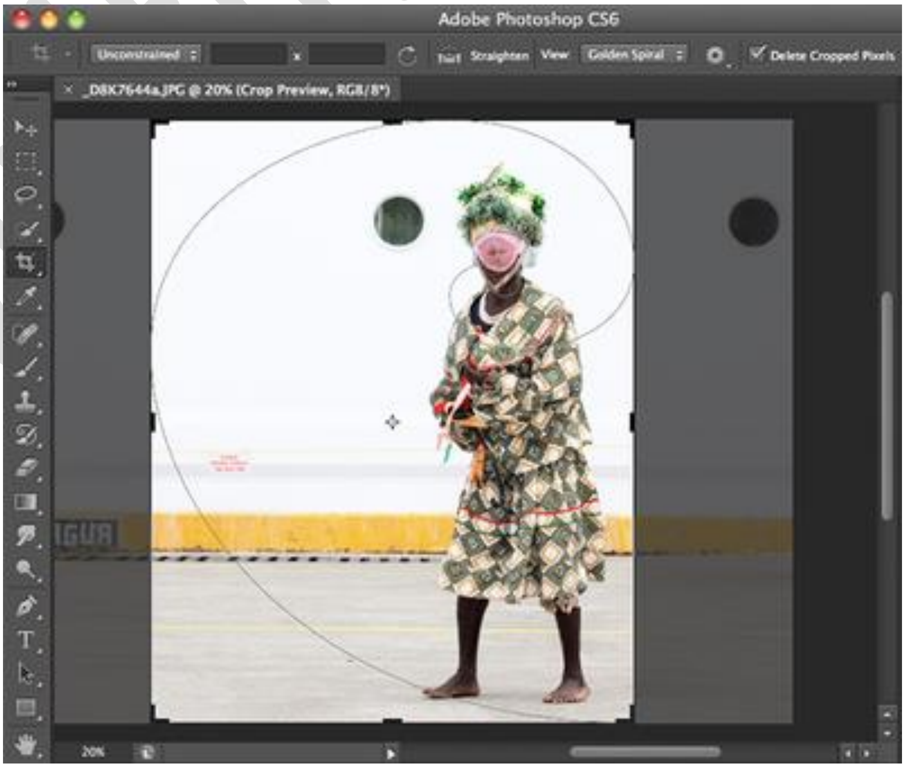
- four channels (Cyan, Magenta, Yellow, Black)
- used by professional printing houses. If you are producing a images for a publication or poster, you may need to save it in CMYK mode.

Image Editing

Photoshop provides numerous powerful tools for working with images. This section describes key tools for basic image editing.

Cropping Images

1. Choose the Crop tool ( ) in the toolbox. Then click and drag across the image. An outline will appear that shows you the area to be cropped.



2. Adjust the area to be cropped by using handles on the edges of the image.

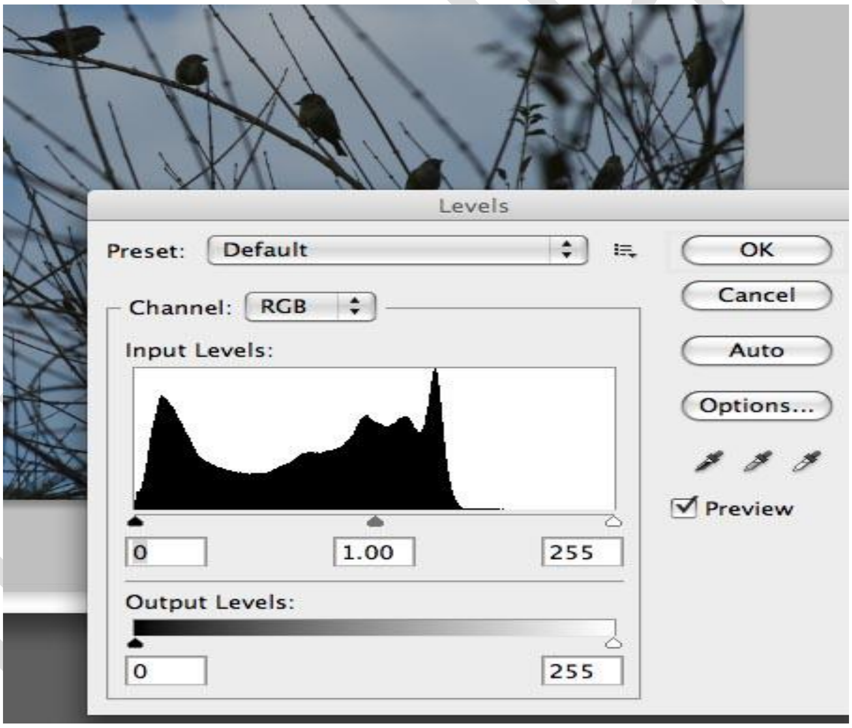
Note: If you move the cursor just outside of one of the corner handles, it turns into a curved cursor, which then allows you change the alignment of the selected area. This is useful, for example, to correct a scan of a picture that was in a crooked position on the scanner.

3. Press Enter to crop or Escape to cancel

Adjusting Contrast

The Levels command (Image>Adjust>Levels) is an excellent tool for adjusting contrast. The Levels dialog box displays a histogram of the brightness values of the pixels in your image. There are two sets of sliders (triangles): input levels and output levels. For most editing tasks, use the input levels sliders (the three triangles just below the histogram).

- To increase contrast (create more dark and light pixels): Move the left and right sliders toward the middle.
- To adjust the midtones: Move the middle input levels slider to the left to lighten midtones, and move the middle slider to the right to darken the midtones.
- Keep the Preview box checked in order to see the effects of moving the sliders on your image.
- To make the adjustment, click OK. To cancel it, click Cancel.



Preset: Default

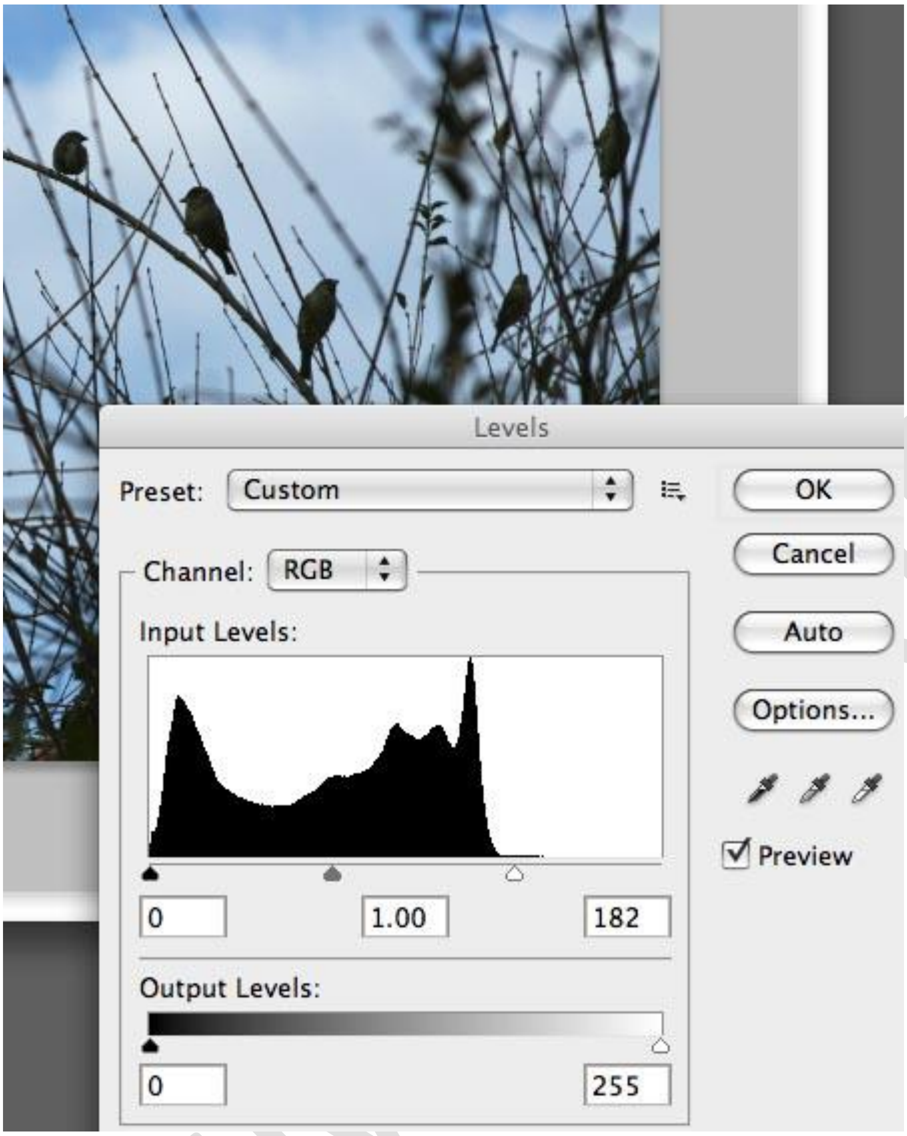
Channel: RGB

Input Levels: 0 1.00 255

Output Levels: 0 255

☑ Preview

In the above example, there are few (if any) pixels that are white or even light gray. By moving the right slider towards the middle, we can lighten image to correct the exposure. Because the Preview box in the layers palette is checked, we can see the result on our image as we adjust the sliders.



Levels

Preset: Custom

Channel: RGB

Input Levels: 0 1.00 182

Output Levels: 0 255

☑ Preview

Note: In the above example, the Levels dialog box Channel popup menu is set to the default of RGB, which means the histogram represents a combination of all of the channels (Red, Green, and Blue light) in the image. You can use Levels on an individual channel, but this is likely to change the color balance of the image, so in most cases is not likely to be a good idea.

Note: The Levels command is often a good way to get a more even distribution of brightness of pixels. But this is not always desirable. For example, a properly exposed photograph of something dark such as a coal mine would be dark, and an image of a snow scene would be white. In such cases the histogram would be lopsided. In such a case there may be no need to make an adjustment in the Levels dialog box.

Color Correction

Often you may need or want to change the balance of color in an image. For example, a photo take outdoors may look too blue, or a photo taken indoors may look yellow or orange.

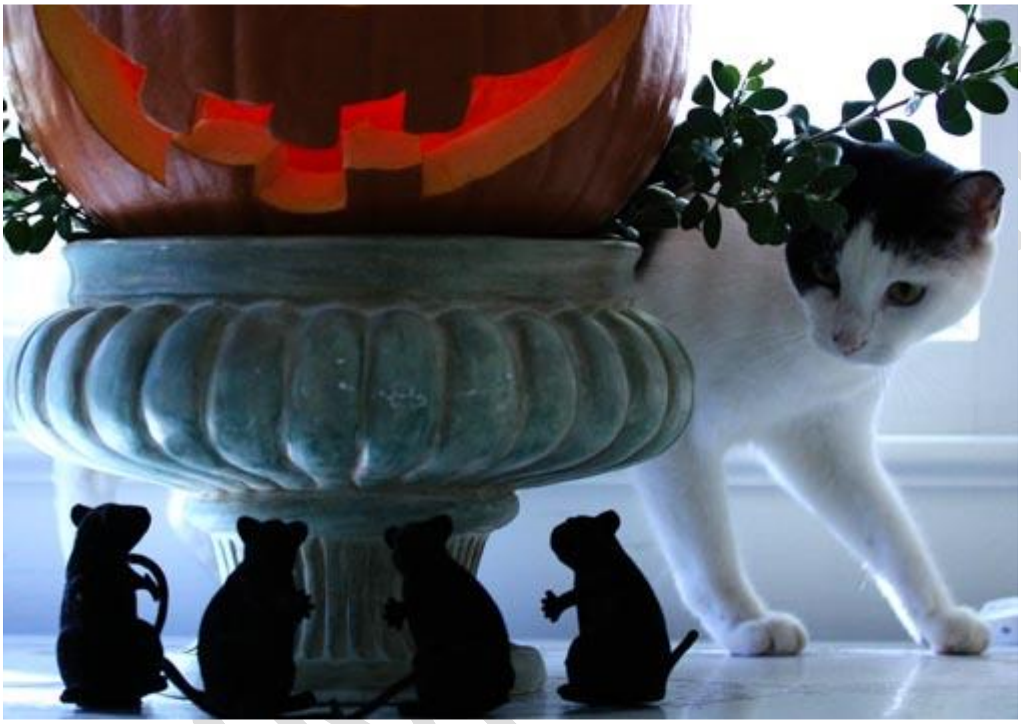
There are several ways to adjust color, many of which are under Adjustments in the Image menu, including Color Balance, Color Mixer, and Photo filters. These tools are easy to use and the best way to learn about them is to open a practice image and choose one of these commands, and experiment with the settings in the dialog box that appears for the command.

Auto Color

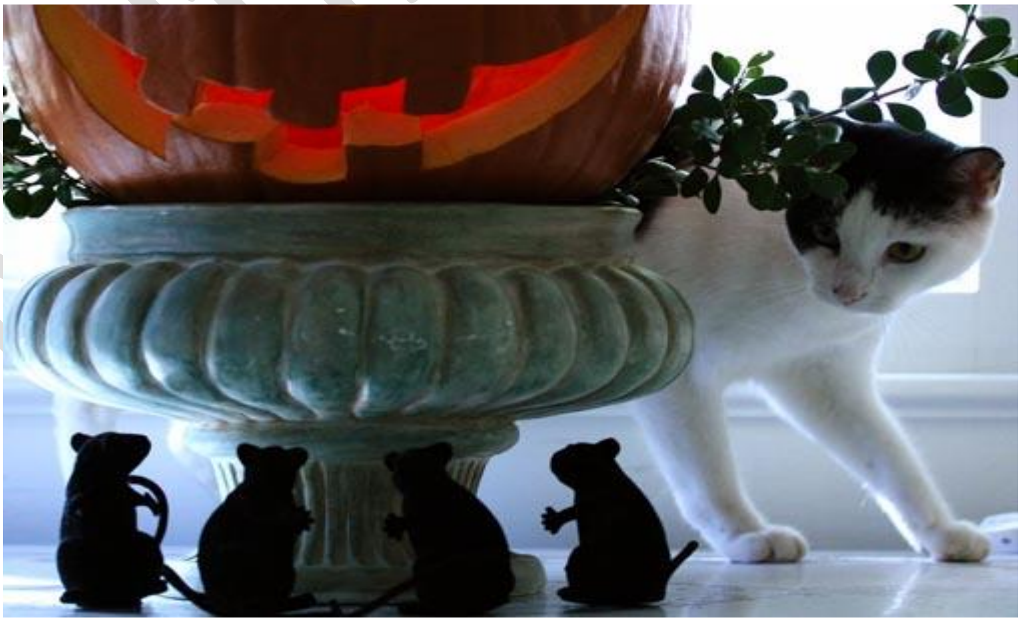
The easiest color correction tool is Auto Color. Auto Color often (though not always) gives very good results. Auto Color attempts to color correct images automatically by analyzing the highlights, shadows, and midtones of each channel.

Choose **Image>Auto Color** to use the Auto Color command and see if you like the results.

The following is an example of an image before Auto Color was applied. The image is good but slightly more blue than I would prefer:



The following is the image after applying Auto Color. The difference is subtle, but Auto Color added a little warmth to the image.



Removing Imperfections

Using the Clone tool

The clone tool ( ), also called the rubber stamp, allows you to remove imperfections such as dust and scratches by covering them with samples of nearby pixels. For best results, zoom in close and use a small brush.

1. Zoom in close to the area you want to edit.

2. Choose the clone tool, and make sure "aligned" is checked in the Options bar.

3. In the options bar, choose a brush size (this determines the size of the sample). The cursor shows the size of the brush, as shown below. The circle is the cursor (brush).

(**Note:** if the cursor doesn't show the brush size, go to Edit>Preferences>Display & Cursors, and choose Brush Size for Painting Cursors.)

4. Alt-click to define initial source point (sample).

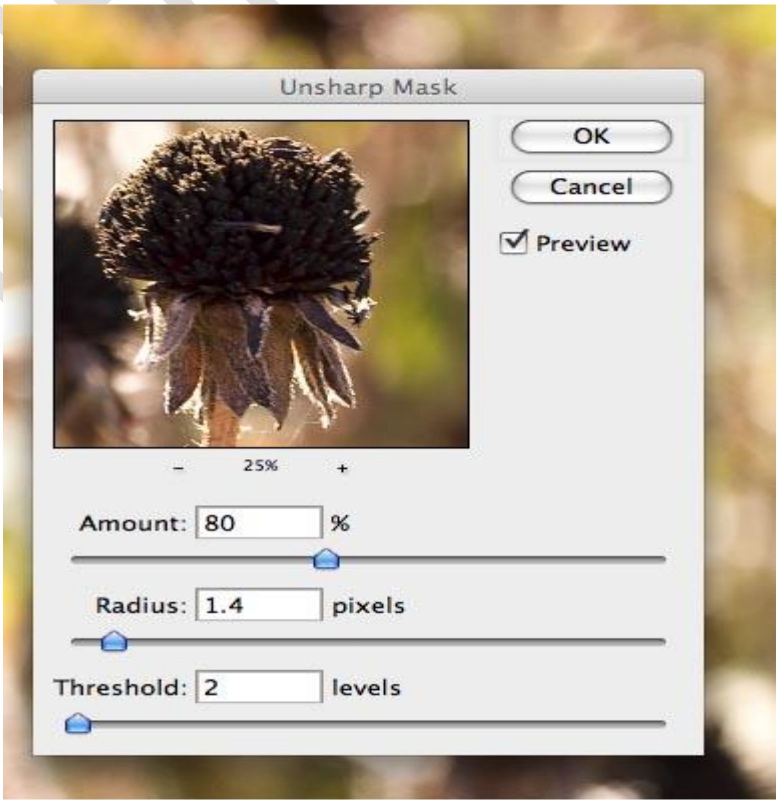
5. Move the cursor over the area you want to cover, and click to cover it with the sample.

Using the Healing Brush Tool

The steps for using the healing brush tool ( ) are the same as for the clone tool. The healing brush attempts to take the texture from the sample without affecting the brightness of the area to which it is applied. In some cases this is more effective than the clone tool.

Sharpening

Use Filter>Sharpen>Unsharp Mask to sharpen an image. In the dialog box that appears, keep the preview box checked. This allows you to see the effects of different settings.



☑ Preview

Amount: 80 %

Radius: 1.4 pixels

Threshold: 2 levels

Amount -- How much sharpening is applied. Consider using a value between 50% and 150% for typical images.

Radius -- The number of pixels affected around edges. Consider using a value between 0.2 and 2.0. (Deke McClelland recommends using 0.1 of radius for every 15 ppi in the image. For example, for an 150 ppi image, you could use a radius value of 1; for 300 ppi, you could use a radius of 2.)

Threshold -- Defines what brightness difference qualifies as an edge (0 is default, sharpens all pixels). Consider using a value somewhere between 1and 5 (for Threshold, the higher the value, the less pronounced the sharpening effect).

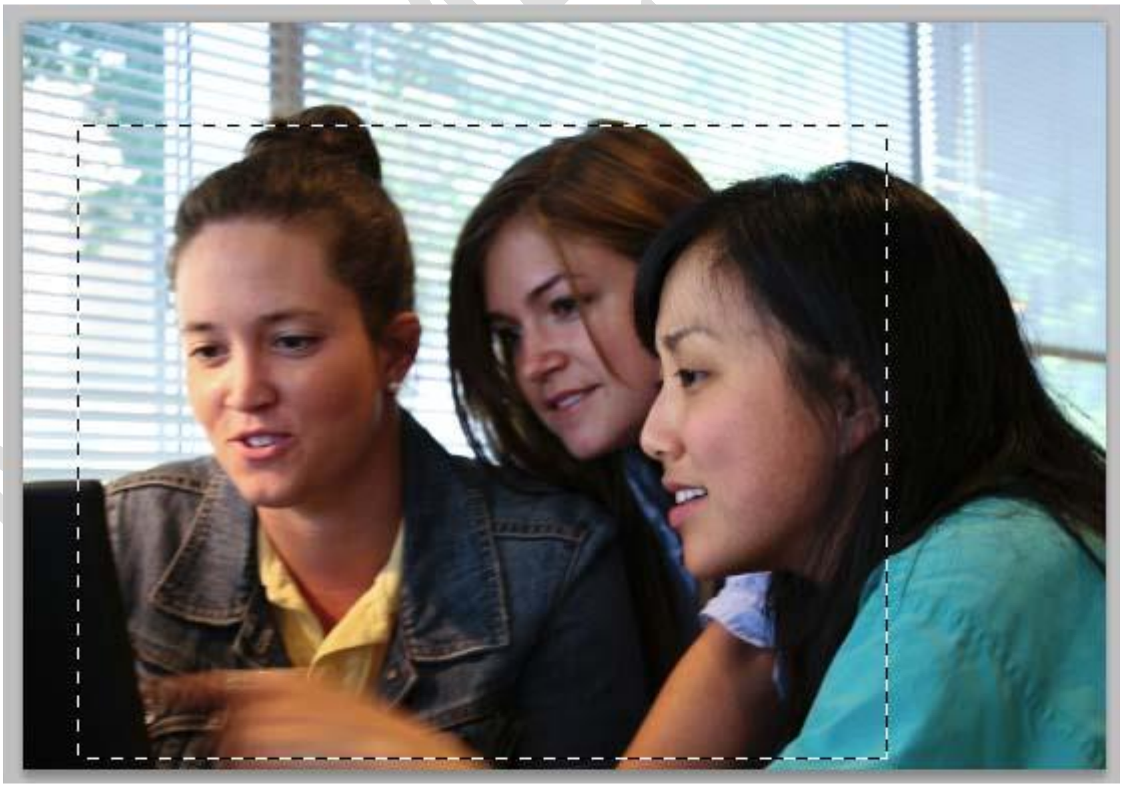
Note: Sharpening is more an art than a science, and is somewhat subjective. The above settings are only suggestions. For best results, you need to look at the particular image and figure out for yourself what looks best! Make sure that in Photoshop the image is at 100% magnification, and keep the Preview box checked in the Unsharp Mask dialog box .

Selection Techniques

In some cases, you may wish to isolate part of an image for editing, or to combine it with other images. In Photoshop, this is called a **selection**. When part of an image has been selected, the image editing tools and commands work the same way, but only the selected part of the image is affected.

When you make a selection, the part of the image that is protected (that is, not selected) is sometimes referred to as a **mask**. Just as you might use masking tape to protect part of a wall when you are painting a room, in Photoshop you are masking the part of the image that you want to protect when you have selected part of an image for editing.

When you make a selection, a selection outline appears to show you the selected area. In the following example, a rectangular area has been selected:



Note: In some cases, pixels may be partially selected (see feathering, below). In such a case, the selection outline is only approximate because there is no sharp boundary between selected and masked areas.

Using Selection Tools

To make a selection, click a selection tool icon in the toolbox. Then use the tool as described below.

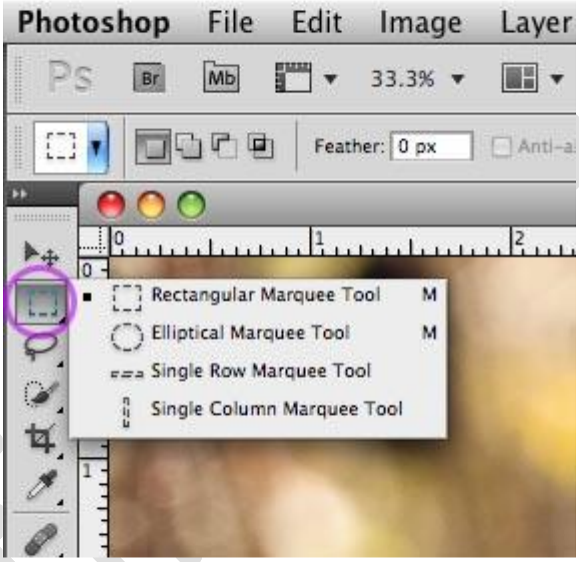
Note: Take a look at the Options bar for the tool you are using. (If the options are not visible, choose Window>Options.) You may want to adjust the options for the tool (for example, to make the selection feathered).

In some cases, tools may be hidden "behind" other tools. Click and hold the tool to see the hidden tools.

Selecting Rectangles, Ellipses, and Rows: the Marquee tools

The rectangular marquee tool ( ) is probably the most frequently used selection tool. It allows you to select rectangular areas. Click and drag diagonally with it to make a selection.

The elliptical and row marquee tools are less frequently used. They are hidden "behind" the Marquee tool. To select an elliptical area or a single row of pixels, type M to cycle through the marquee tools. Then drag (with Elliptical tool) or click (with Single Row tools) to make a selection.

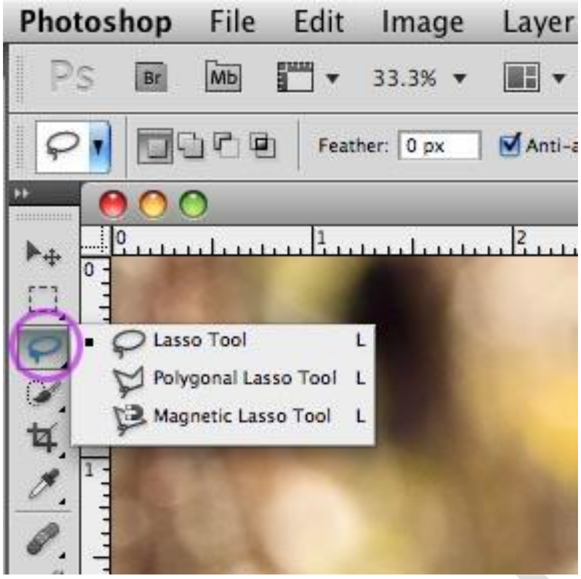


Selecting Irregularly Shaped Areas: the Lasso Tools

To create freeform selections, you can click and drag with the Lasso tool ( ). But the lasso tool may not give you enough control. For more control, you may find the polygon lasso and magnetic lasso tools easier to use.

The Polygon Lasso tool ( ) is often quite useful for selecting irregularly shaped areas. To use it, click on the image, then move to another place and click again to create a segment. Click multiple times to create a selection around an irregular area in your image. Double-click to finish the selection. **Note**: If you are in the middle of selecting and you make a mistake, press the backspace key to go back one segment, or click the Esc key to undo the selection entirely.

Without pressing the mouse button, move the magnetic lasso tool around an area to select it. The tool will look for edges. You can adjust its sensitivity to edges in the options bar.



Selecting by brightness and color: Magic Wand tool (W)

Click somewhere on the image with the magic wand tool to select adjacent portions of the image based on color brightness. The higher the tolerance entered in the Options palette, the more pixels are selected.

Note: In addition to the above tools, there are some additional more advanced techniques for making selections. For example, you can use the Pen tool to create paths. You can convert a path into a selection by choosing Make Selection from the Path palette menu.

General Selection techniques

- To hide the selection outline, press Ctrl-H. To display the outline again, press Ctrl-H again.
- To invert a selection, choose Select > Inverse. Everything that was selected becomes masked, and everything that was masked now becomes the selection.
- To deselect, click outside of the selection, or choose Select > None.
- You can also save a selection (Select > Save Selection) for later use (Select > Load Selection).

Adding to and subtracting from current selections

- To add to an existing selection, press Shift
- To subtract from an existing selection, press Alt
- To add to an existing selection based on color, choose Select > Similar.

Moving and Cloning Selections

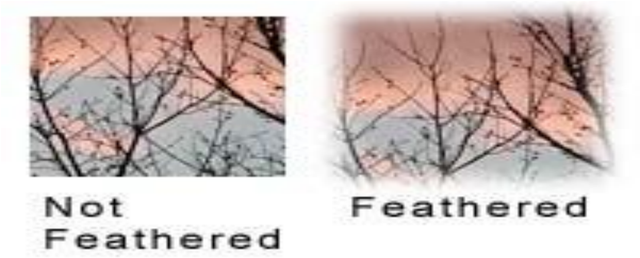
- To move a selection outline, put the selection cursor inside the selected area and drag.
- To move the contents of a selection, choose the move tool and drag the selection.
- To clone a selection, press Alt while using the move tool.

Saving Selections for Later Use

- To save a selection for later use, choose Select > Save Selection. The active selection will be saved, and you will be prompted to give it a name.
- To load a saved selection, choose Select >Load Selection.

Feathering Selections

Feathering gives a selection a softer edge. You can feather a selection by entering a value in the Feather checkbox in the options bar. Or you can feather an existing selection by choosing Select > Feather.



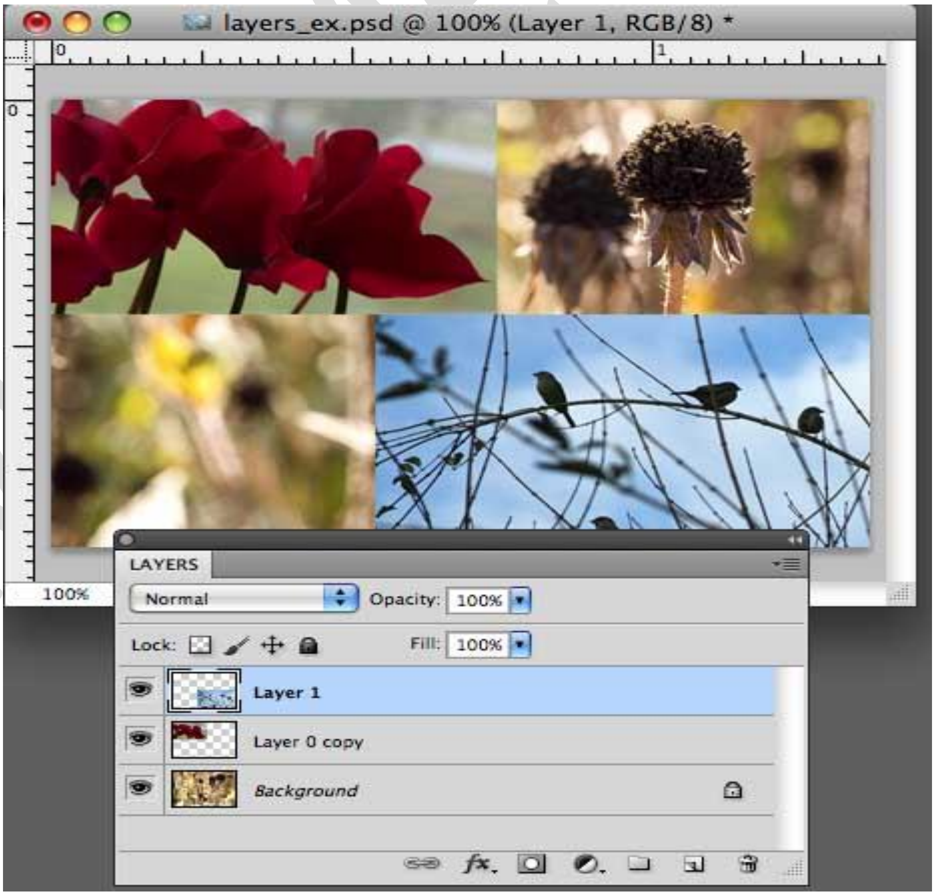
Working with Layers

Images in Adobe Photoshop are made up of layers. Initially, each image has only one layer (called the background layer), but you can add more layers to the image.

Layers are very useful for editing images and creating special effects. For example, you could make a copy of a layer and make some changes to it. If you don't like the results, you can delete the layer and still have your original layer as a backup. You can also combine images by putting different content on different layers.

Note: Keep in mind that layers (except for adjustment layers, described below) require a substantial amount of memory. Photoshop files with large numbers of layers typically are large files.

To work with layers, display the Layers palette by choosing Layers from the Window menu. The following is an example of the layers palette along with an image.



layers_ex.psd @ 100% (Layer 1, RGB/8) *

LAYERS

Normal Opacity: 100%

Lock: Fill: 100%

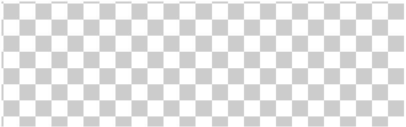
Layer 1

Layer 0 copy

Background

The first column of the layers palette shows which layers are currently visible, indicate by the eye icon (👁). In the above example all the layers are visible. You can change a layer's visibility by clicking in the first column for the layer.

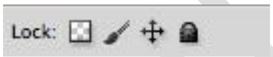
> **Note**: Even when a layer is visible, part of it may be transparent. For example, this happens if you select part of an area and then press the delete key. When part of a layer is transparent, the layers underneath it can show through.
>
> Photoshop represents transparent areas by a checkerboard pattern:



You can change the opacity of a layer by entering a value between 0 and 100 in the Opacity field at the top of the layers palette. For example, at an opacity of 10%, the layer would be nearly transparent; at an opacity of 90%, it would be almost completely opaque. The opacity default is 100%.

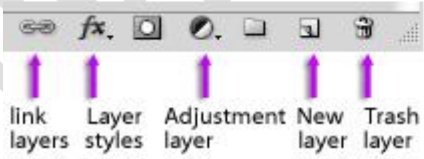
Editing you do within Photoshop affects the current layer. In the example above, "Layer 1" is the current layer.

- To make a layer the current layer, click on the layer in the Layers palette.
- To change the name of the layer, double-click on the layer's name, then type in a new name.



To lock (prevent changes to) a layer, click one of the Lock: boxes near the top of the layers palette. From left to right, they are: lock transparent areas, lock pixels (prevent changes to non-transparent areas), lock layer position, and lock all.

Other actions can be performed by clicking on icons at the bottom of the layers palette:



- To link two or more layers, first select them by shift-clicking on them in the layers palette. Then click the link layers icon. When they are linked they move together, so if you reposition a layer you will simpultaneously reposition all layers linked to it.

- To add effects to a layer (for example, shadows), click on the layer styles icon.

- To add an adjustment layer, click the Adjustment layer icon (adjustment layers are discussed below).

- To create a new layer, click the new layer icon at the bottom of the Layers palette, or duplicate a layer by dragging the layer to the new layer icon.

- To delete a layer, click the trash icon, or drag the layer to the trash icon (next to the new layer icon).

Adjustment Layers

Adjustment layers are special layers that you can use along with several image editing commands, including Levels, Photo Filters, Hue/Saturation and others. Adjustment layers allow you to make non-destructive (editable) corrections to your images. For example, if you create a Levels adjustment layer, you can go back to the Levels dialog box later and change settings at any time.

To create an adjustment layer, click on the adjustment layer button at the bottom of the Layers palette, then choose one of the menu items (Levels, Curves, Color Balance, etc.). Alternatively, you can choose Layer > New Adjustment Layer, and you will see the same menu of items.

When you choose e menu item, an adjustment layer will be created, and a dialog box appears that allows you to change the settings.

The advantage of this over an ordinary Levels command is that the adjustment layer is fully editable. At any time, you can click the adjustment layer thumbnail in the layers palette to edit the layer. When you do, the dialog box will reappear and you can change the settings.

Unlike ordinary layers, adjustment layers require minimal memory and do not significantly increase the size of the file.

Adding Type

To add type to images use the Type tool ("T" in the Toolbox).

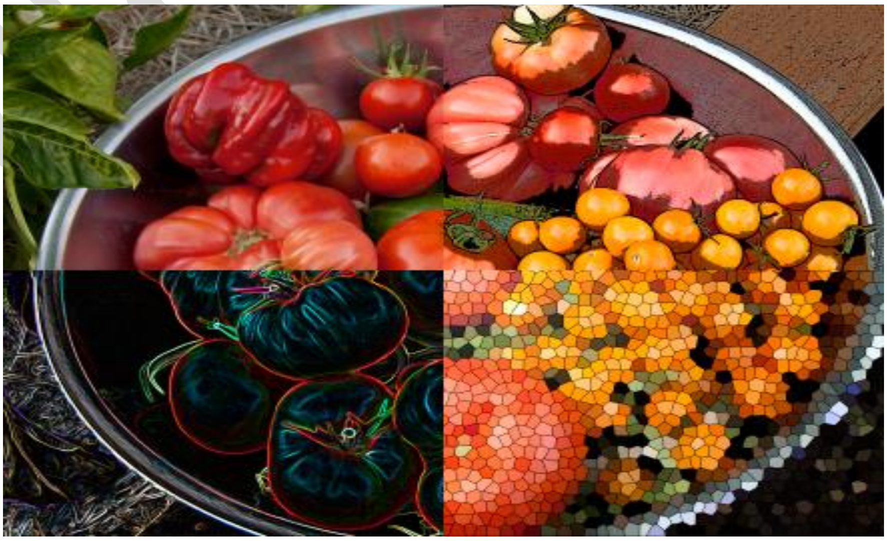
- To enter a single line of text, click on the image with the type tool. Then type. A type layer is automatically created.
- To create a paragraph (column) of text, click and drag with the type tool, then start typing. When you enter the text, it will wrap at the point where you stopped dragging.
- To reposition the text while you are using the Type tool, press the Ctrl key.The Type will change into the move tool so you can move the text box.
- When you are finished typing, click the check mark in the options bar. To cancel, click the X mark in the options bar.
- To adjust the color and format of your type, select the type and then change the settings in the options bar (at the top of the Photoshop desktop), or use the Character or Paragraph palettes (to display these palettes, click on the text palettes button ( ), which appears in the Options bar whenever the Type tool is selected).

When you use the Type tool, a type layer is automatically created. As long as your type is on a Type layer it is editable. If you convert it to an image layer by rasterizing it (Layer>Rasterize), you will no longer be able to edit it

<u>PhotoShop Filters</u>

Filter

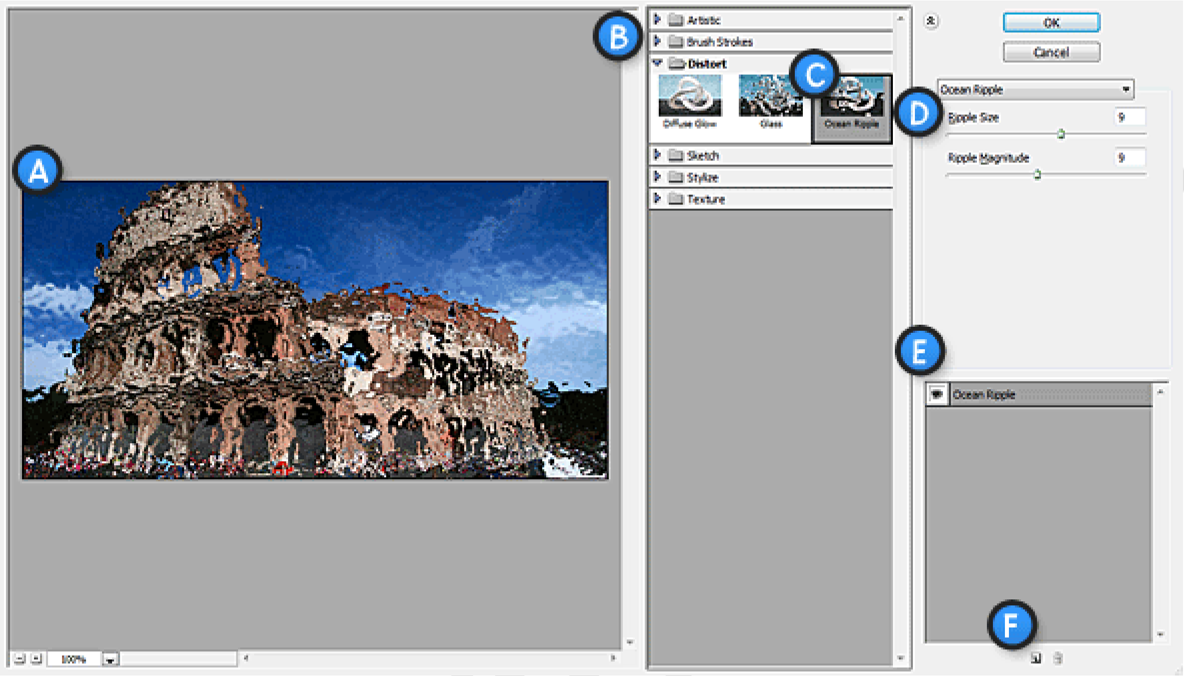
Automated ways to alter the look of an image, for instance, to make it look like a mosaic, add unique lighting, apply distortions, and so on.



Using Filters from the Filter Gallery

Photoshop's **Filter Gallery** is basically a one-stop place for working with filters in your documents. In the Filter Gallery, you can browse through many different types of filters, apply them individually to your image, or even stack them on top of one another like you would with layers. You can get to the Filter Gallery by going to ***Filter > Filter Gallery***.

Here's a quick overview of the Filter Gallery:



Ocean Ripple

Ripple Size 9

Ripple Magnitude 9

Ocean Ripple

Filter Gallery Interface

A. Filter Preview

B. Filter List (*Note: You can show or hide this list by clicking the arrow icon to the top right of it*)

C. Currently Selected Filter

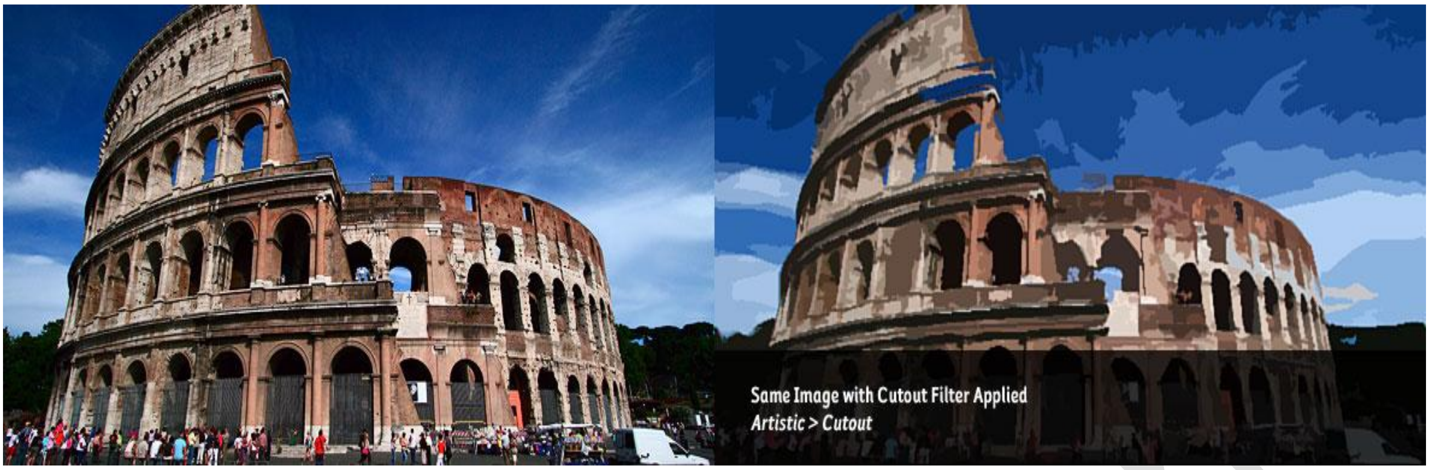
D. Filter Options

E. Effect Layers

F. New / Delete Effect Layer

The Filter Effect Layers work just like ordinary filters in Photoshop, and can be dragged and dropped on top of one another for desired results.

Applying a filter from the gallery is very easy to do. Simply **Click on a filter**, adjust it's options to your liking, and click **OK** to apply. To stack multiple filters on top of one another, just hit the **New Effect Layer ('F' in diagram above)**, and select another filter. You can expand filter types by clicking the folders in the filter list.



The Cutout Filter quickly transforms a photograph into an abstract piece of art.

Quicker Access to Filters

The Filter Gallery is really just a browser of sorts. If you know exactly what kind of filter you want to use, you can access it directly from the **Filter menu** in Photoshop to speed things up.

Using Other Filters

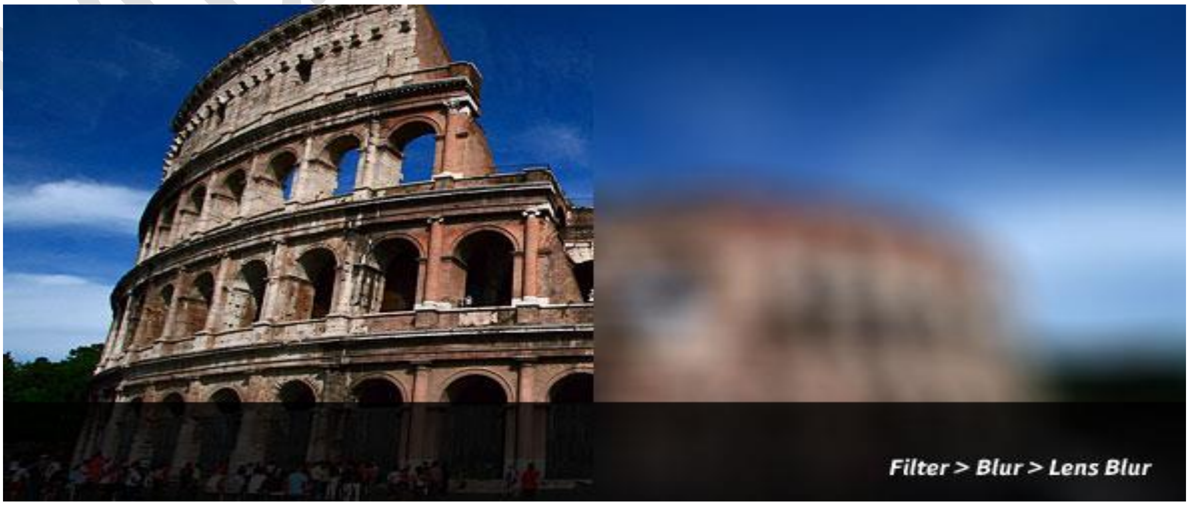
Not all filters in Photoshop are available through the Filter Gallery. Many of them must be accessed directly through the **Filter Menu**, and have unique interfaces and options of their own.

We're not going to go over every individual filter in this tutorial (*doing so would take a very long time*), but I highly suggest doing some experimentation on your part with different filters to get an idea of what some of them do.

Let's look at some of the more common filters used frequently in design.

Blur Filters (Filter > Blur)

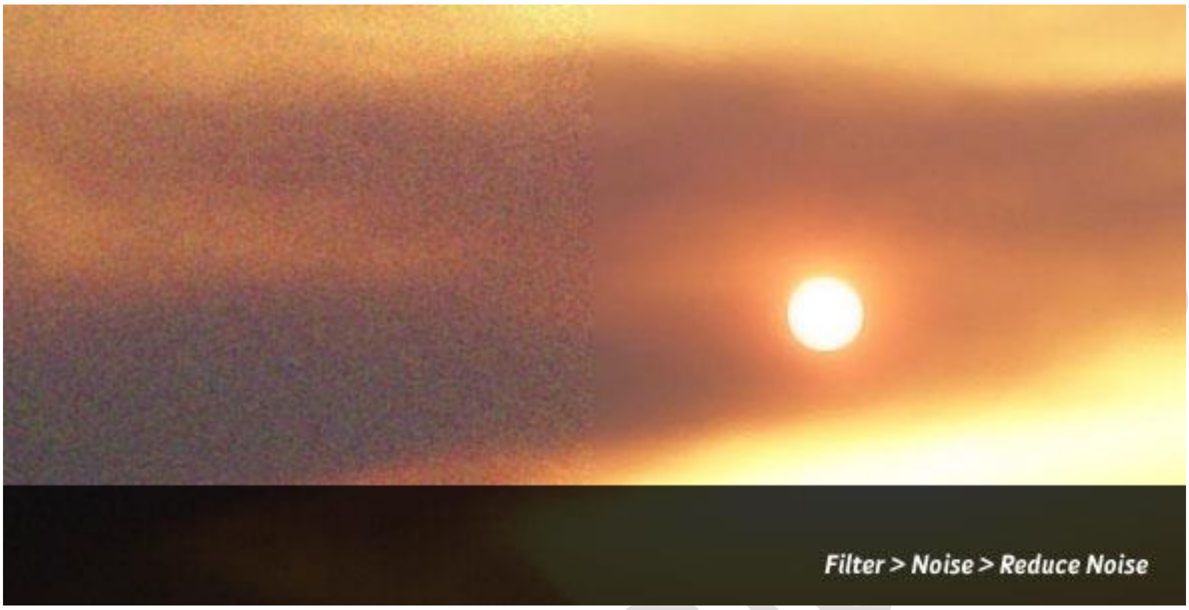
The **Blur Filters** are useful in lots of situations. We've actually written several popular tutorials on Tutorial9 that depend on the Blur Filters, such as our article on Adding Depth of Field in Photoshop, and our Windows Vista Aurora Effect Tutorial.



*The **Lens Blur Filter** makes it very easy to create a realistic lens blur effect in Photoshop.*

Noise Filters (Filter > Noise)

Noise Filters are great for adding, or reducing noise and grain in photographs. You may find filters such as the **Reduce Noise Filter** extremely useful if you work with old, damaged, or dusty photographs that need repair work done to them. The **Add Noise Filter** can also come in handy, and has some creative applications of its own.



*Noise can easily be reduced with the **Reduce Noise Filter** in Photoshop.*

Sharpen Filters (Filter > Sharpen)

The **Sharpen Filters** are also great for correcting imperfections in photographs, as well as putting emphasis on important elements in a design. When working with blurred images, a sharpen filter can be used to clarify and better define edges by increasing contrast between pixels.

*The **Smart Sharpen Filter** used to sharpen a photograph of a statue.*

Filters Disabled?

While all filters are available to be used on 8-bit RGB images, many will be unavailable if working with other color spaces, 16-bit, or 32-bit documents. Filters will also be unavailable for use in Bitmap and Indexed-color documents.

If you're not able to use filters, you **may need to go to Image > Mode**, and make sure you're set to **RGB**, and **8 Bits/Channel**, although some image data may be lost.

Exporting and Printing

Putting Images on the Web

When saving or exporting files for use on the web, you need to save images in a Web-compatible file format, such as GIF or JPEG. (Native Photoshop format is not Web-compatible.) To do this you can use File > Save For Web or File Save As. Save for Web gives you the most options, and allows you to preview your image at different compression levels.

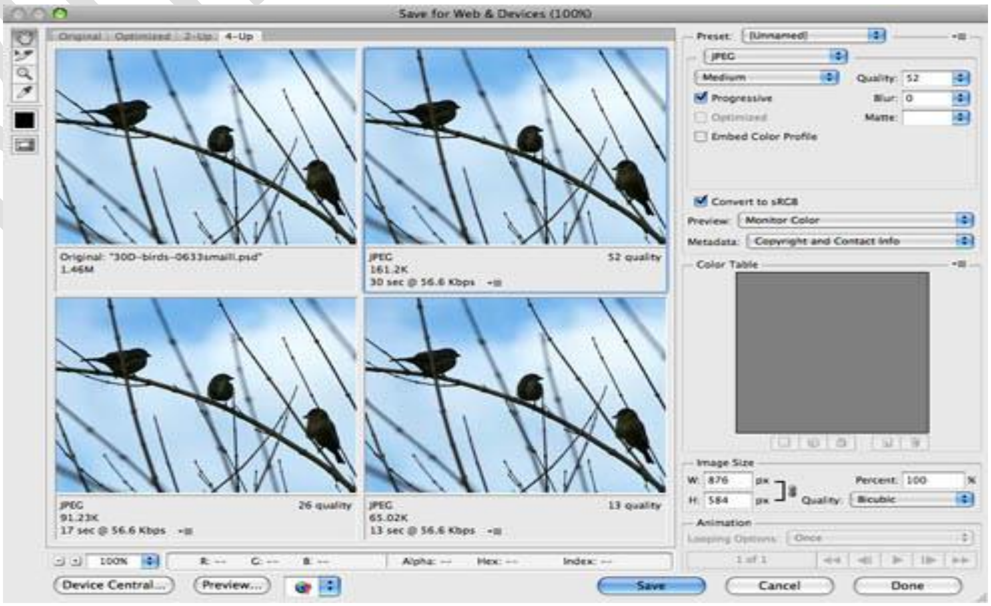
Here are some things to keep in mind when saving images for the the Web:

- Photoshop's native format allows layers, but Web file formats do not. When you export a file for the Web, any multiple-layer file will be flattened. If you use File > Save for Web, Photoshop will export your GIF or JPEG as a single-layer file (combining the currently visible layers of your Photoshop file). If you use File > Save As, you may need to merge layers before saving in a Web format.
- When you export images from Photoshop to a Web file format, keep a native Photoshop file as backup. This will help you if you need to modify any of the images later.
- Make file sizes as small as possible while maintaining acceptable image quality.
- When working on the Web, remember to think in terms of pixels (not inches). Images will appear on the Web at about the same size as they do when you are looking at them in Photoshop at 100% view.

The Save for Web & Devices Command

A good way to save images for the Web is to choose File > Save for Web & Devices. The advantage of Save For Web & Devices is that it presents previews of your image with different settings. You can apply different settings to different previews and compare them before you decide which settings you want to use when you save the image.

When you choose Save For Web & Devices, the following dialog box appears:



Original: "30D-birds-0633small.psd"
1.46M

JPEG 52 quality
161.2K
30 sec @ 56.6 Kbps

JPEG 26 quality
91.23K
17 sec @ 56.6 Kbps

JPEG 13 quality
65.02K
13 sec @ 56.6 Kbps

100% R: -- G: -- B: -- Alpha: -- Hex: -- Index: --

Preset: [Unnamed]
JPEG
Medium Quality: 52
☑ Progressive Blur: 0
□ Optimized Matte:
□ Embed Color Profile
☑ Convert to sRGB
Preview: Monitor Color
Metadata: Copyright and Contact Info

W: 876 px Percent: 100 %
H: 584 px Quality: Bicubic

Looping Options: Once
1 of 1

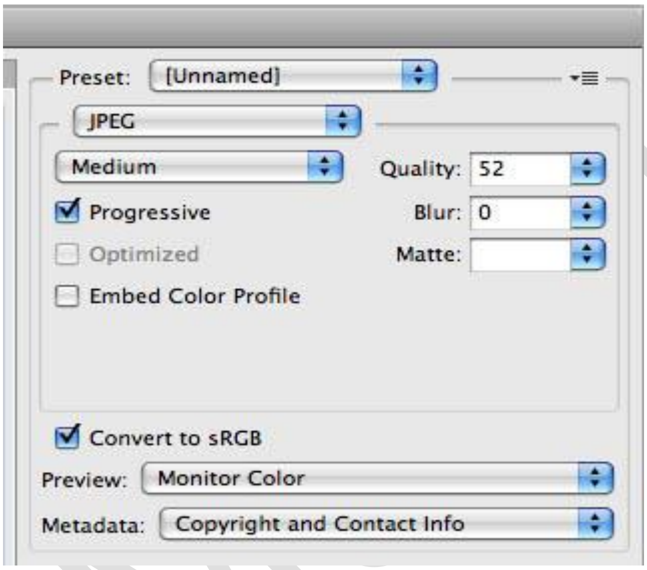
The original image appears in the upper left corner of the dialog box, and the others are previews. Click on one of the previews to select it. (A thin blue line appears around the currently selected preview.) You can change the selected preview's settings in the upper right area of the dialog box. The available settings change depending on whether you choose JPEG or GIF from the popup menu in the upper-right area.

The file size for each preview is shown below the preview. Compare the file sizes and display quality of the previews. In some cases either GIF or JPEG is acceptable.

In general, for color photographs and images with gradients, use JPEG format. For images or graphics with flat colors (such as illustrations and logos), use GIF format.

For JPEG format:

- Choose JPEG from the popup menu in the upper-right area.



Preset: [Unnamed]

JPEG

Medium — Quality: 52

☑ Progressive — Blur: 0

□ Optimized — Matte:

□ Embed Color Profile

☑ Convert to sRGB

Preview: Monitor Color

Metadata: Copyright and Contact Info

- If you wish, choose one of the Preset options for defining your settings; otherwise choose the settings you want to specify from the other popup menus.
- Choose Quality, either by choosing Low, Medium or High or by choosing a value between 0 and 100. A high number such as 90 will result in a very high quality image, but the file will be larger and will take much longer to download. Most images can be saved at fairly low quality and still look good on the Web.
- For most purposes, keep the Progressive checkbox checked. Progressive displays the image gradually in multiple passes (usually a good idea; this gives the user the sense that something is happening).
- In some cases, you may want to blur your image slightly. This won't have much effect on the quality of the displayed image but it may reduce the file size.
- If the edge of your image includes some transparent pixels (if you want the edge pixels to blend with the background color of your Web page), choose a Matte color that is the same as the background color of your Web page. Otherwise, set Matte to None.

For GIF Format:

- Choose GIF from the popup menu in the upper-right area.
- If you wish, choose one of the Preset options for defining your settings; otherwise choose the settings you want to specify from the other popup menus.
- Choose a color reduction palette. Usually Selective, Adaptive, and Perceptual work well. You can experiment and see which looks best to you.
- Specify the number of colors you want the GIF image to have (the more colors the larger the file size)
- Specify Dithering (choose none for smaller file size; choose diffusion if image has subtle gradations
- Choose Interlaced if you want the image to start appearing more quickly (recommended).

Importing Images into PowerPoint

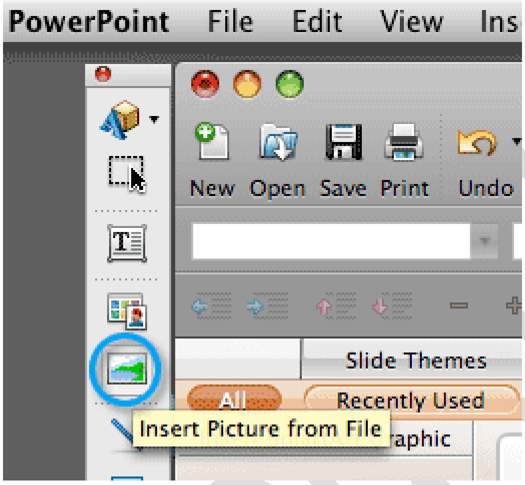
To insert an image into a Power point presentation, do the following:

1. Save your image in JPEG format.

2. In Powerpoint, create a blank slide where you will place your image.

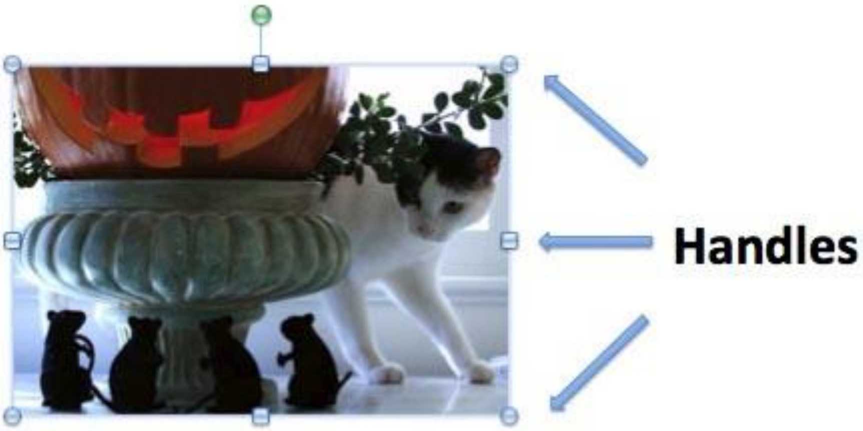
3. In PowerPoint, if the Drawing Toolbar is currently displayed, click on the "Insert Picture from File" icon.

Note: To display the Drawing toolbar, choose View menu > Toolbars > Drawing.



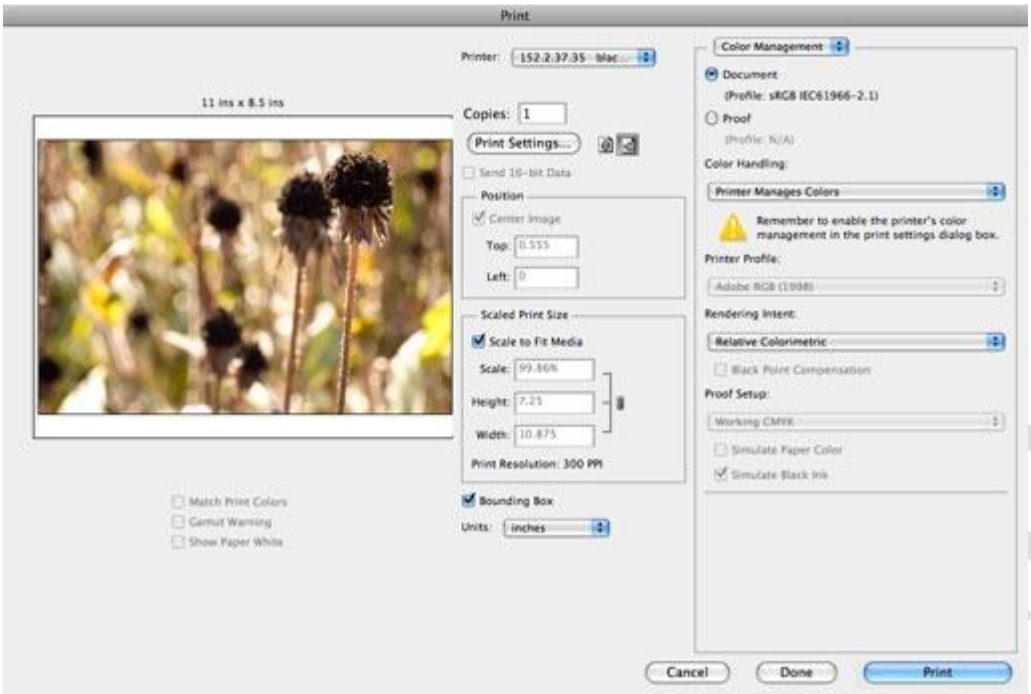
Or, choose Insert > Picture > From File, and choose the file you want to insert.

4. The image will appear on your PowerPoint slide. You can then adjust the size of the image within PowerPoint by clicking and dragging the handles on the edges of the image.



Printing Images

To print an image, choose File menu > Print. The print dialog box appears.



Print

Printer: 152.2.37.35 - blac...

11 ins x 8.5 ins

Copies: 1

Print Settings...

□ Send 16-bit Data

Position

☑ Center Image

Top: 0.555

Left: 0

Scaled Print Size

☑ Scale to Fit Media

Scale: 99.86%

Height: 7.25

Width: 10.875

Print Resolution: 300 PPI

☑ Bounding Box

Units: inches

□ Match Print Colors

□ Gamut Warning

□ Show Paper White

Color Management

● Document

(Profile: sRGB IEC61966-2.1)

○ Proof

(Profile: N/A)

Color Handling:

Printer Manages Colors

Remember to enable the printer's color management in the print settings dialog box.

Printer Profile:

Adobe RGB (1998)

Rendering Intent:

Relative Colorimetric

□ Black Point Compensation

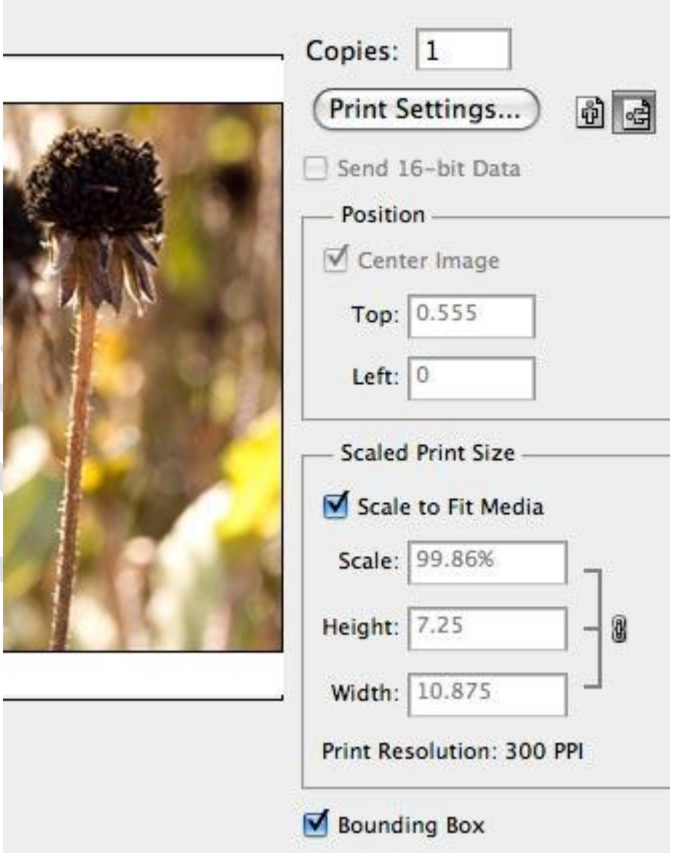
Proof Setup:

Working CMYK

□ Simulate Paper Color

☑ Simulate Black Ink

Adjust print settings if you wish. For example, you could click on one of the orientation icons (to the right of the "Print Settings" button) to change the orientation from portrait to landscape or vice-versa. Or you can change the position of the image on the paper or the size of the image (print size).



Copies: 1

Print Settings...

□ Send 16-bit Data

Position

☑ Center Image

Top: 0.555

Left: 0

Scaled Print Size

☑ Scale to Fit Media

Scale: 99.86%

Height: 7.25

Width: 10.875

Print Resolution: 300 PPI

☑ Bounding Box

Click Print to print your image.

Some of photoshop Design Photos(Texts Photo):

THAN
PSD
S
lg
OK
N
JN
rk
rt
n
GO
TA

Lab. 10: SwishMax

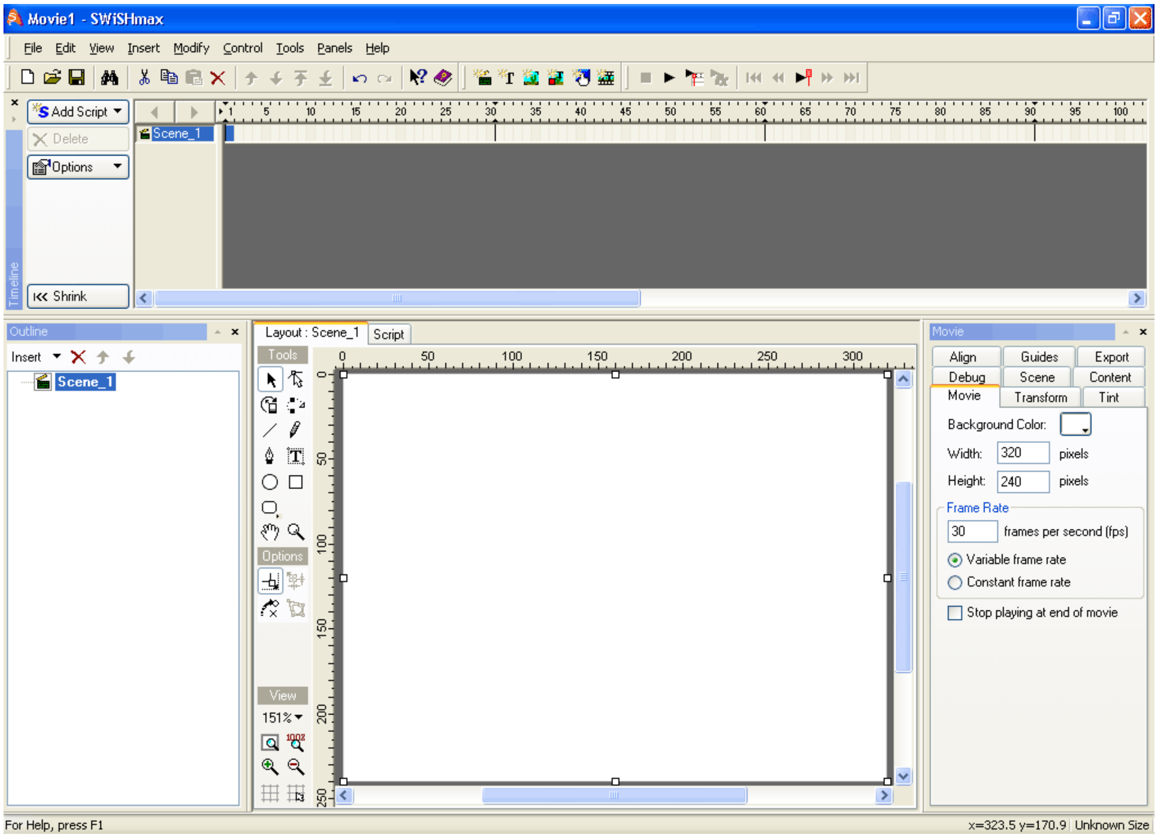
Reference: SwishMax unleashed/ BOKA EDUARD ZORAN/ 2004

1. SwishMax presentation

1.1. About SwishMax

SwishMax will allow you to create Flash™ content to add images, animation, sounds, and interactivity to your web site. Also you can create games and interactive applications. You can use this application to create introductions, commercials, banners, menus, and complete web sites. SwishMax revolutionized the Flash™ industry with its first release in April 2000. For the first time, complex-text effects could be created in minutes that had previously taken hours to create in Flash™. SwishMax is so intuitive and easy to use young children enjoy using it, and it is being used in schools throughout the world. SwishMax developers are dedicated to keeping the application intuitive and easy enough for the non-professional to use, while adding those features most requested by professional users.

1.2. SwishMax presentation



Background Color:

Width: 320 pixels

Height: 240 pixels

Frame Rate

30 frames per second (fps)

● Variable frame rate

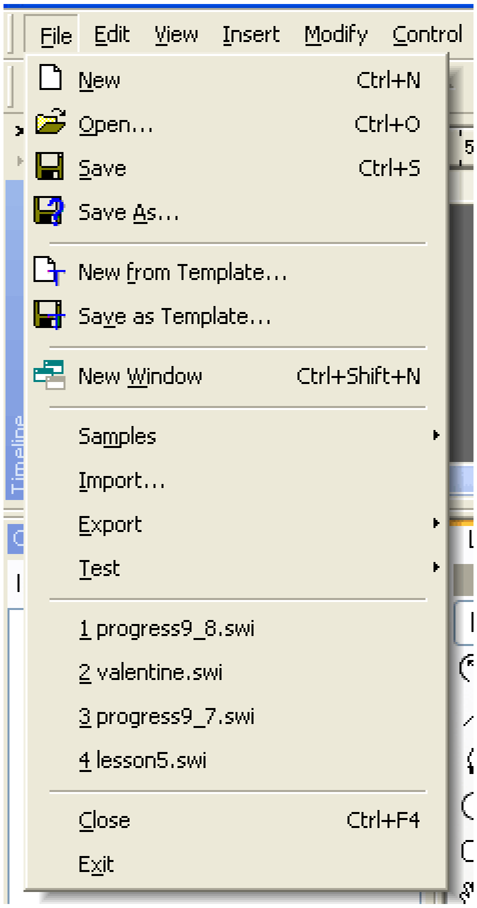
○ Constant frame rate

□ Stop playing at end of movie

In the above picture we have a sample of the SwishMax layout. We can see the menu, the timeline, the outline, the toolbar, layout and properties. We are going to take each of them and analyze them separately.

The menu

Provides links to all possible options in SwishMax, from basic to advanced, grouped into categories. It is not used very much, because all of the most needed options appear as special icons.



File

Provides a shortcut to options that allow us to:

Create new files: ***New***
Open earlier created files: ***Open***
Save our work: ***Save*** and ***Save As…***

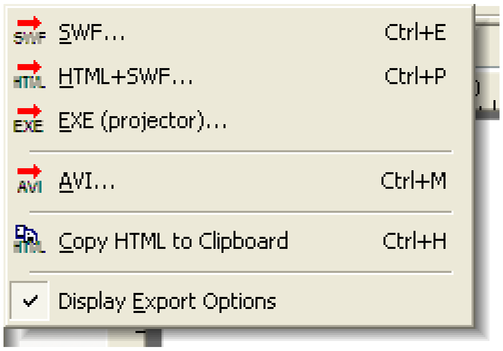
Create new file from a template: ***New from template…***
Save the project as a template: ***Save as template***

Open a new SwishMax window: ***New window***

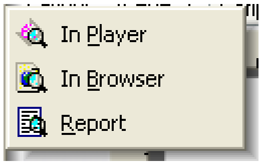
Open files that come with SwishMax: ***Samples***
Import files that can be used in the project: ***Import…***
Export files and make them ready to use: ***Export***
Test files in the flash player, browser: ***Test***

Open recently created files

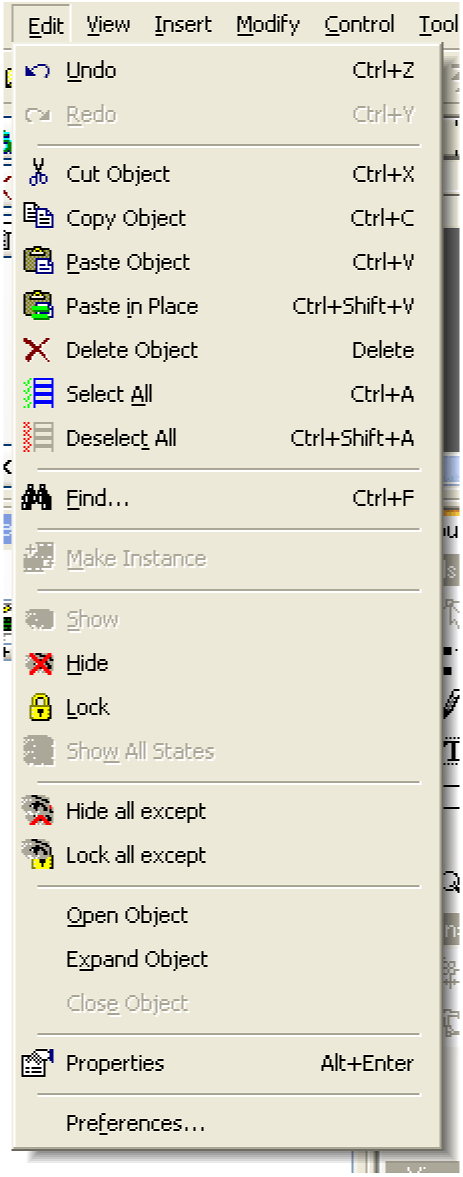
Close the file you are currently working on: ***Close***
Exit SwishMax: ***Exit***



This is the ***Export*** menu extension. This lets us choose for what format we want to export our movie. **Swf** is the default extension for Flash™ movies. You need the Flash player in order to play these movies. **Html + Swf** saves you work as a webpage that can be accessed by running the html file. **Exe** exporting is very useful for presentations and for people that do not have Flash™ player installed. **Avi** is the default extension used for movies. You can save your movie as such an animation too. **Copy HTML to Clipboard** lets you copy the movies html code to the clipboard.



The ***Test*** menu extension allows you to test the current movie you are working on in the Flash™ Player and in your default browser. **Report** is a very useful option that generates a report on your movie and allows you to see what objects are taking up the most of disk space in the movie.



Edit

A short description of the items in the Edit menu is listed below:

Undo & Redo actions are commonly used in all programs to undo the very last action you took and redo is if you later decide you didn't want to undo an action. As shortcut keys to undo is Ctrl +Z and I advise you to learn this combination as you will use this option a lot.

Editing tools for the objects in the scene. Very useful to duplicate objects in the movie (using copy and after that paste), delete objects in the movie, select all objects in the movie or deselect all objects in the movie.

Find allows users to find a specific text in the SwishMax scripting environment.

Make Instance creates an instance of a sprite you have in you movie. Sprites will be discussed later in the book

Visibility options for the objects in the scene are very useful if you just want to specifically work on a object and don't want to see other objects, don't want the objects to interact, etc.

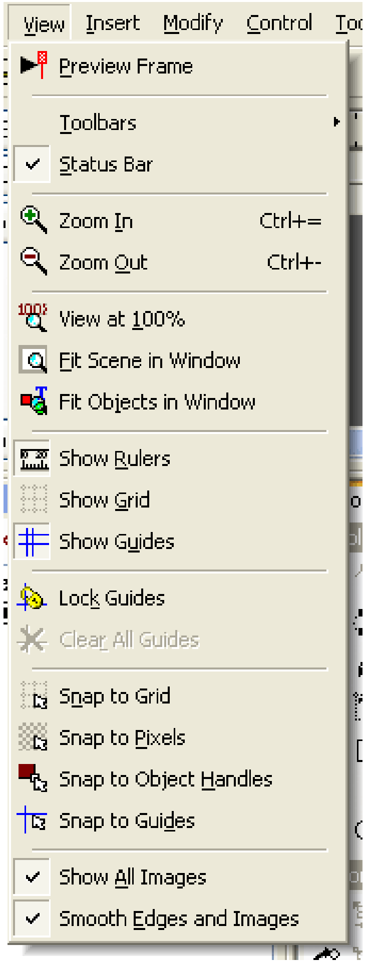
Hide/lock all except are useful for single object editing, allowing the user to **hide** all objects except those selected, or just **lock** those objects.

Open, Expand and **Close object** refer to sprites, groups and buttons that have the "⊞" character in front of them. This means that they contain other objects too. When expanding such an object the "⊟ "sign will appear as a visual explanation that the object is expanded.

The **Properties** option lets the user view the properties of the currently selected object.

By clicking **Preferences**, it brings up the preferences menu for SwishMax that allow users to change interactivity options between users and the program.

Note It is pretty important to learn shortcut keys by heart, to make your work a lot faster and easier, accessing the wanted option only by pressing a simple key combination.



View

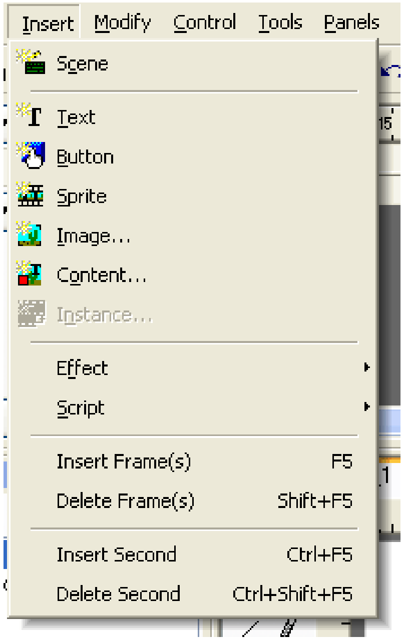
The view menu provides access to options regarding viewing of objects, SwishMax toolbars, grids and rulers. A short description of the menu is as follows.

Preview Frame lets you preview a frame as it would look like at runtime.

Toolbars option lets the user select the wanted toolbars that are displayed in the SwishMax workspace.
The **Status Bar** is the bar at the bottom of SwishMax that displays helpful information: mouse coordinates current movie size etc.

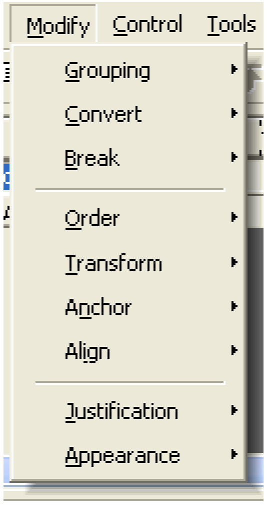
The zooming options are very useful to work precisely on a certain point in the movie.

Rulers, grids and guides are very useful when creating precise applications. Snapping to grid, pixels, guides makes the object that is being built to snap (go to) to the handles. Combining these options you will get very precise graphics, object placements, etc.



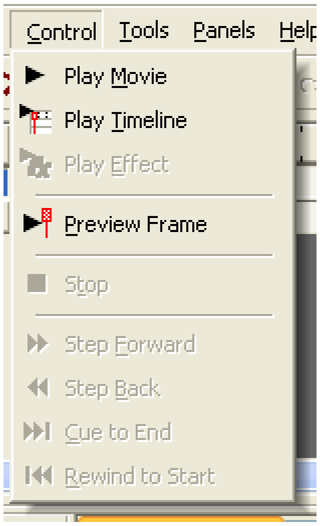
Insert

Here you can insert new scenes (**Scene**), new objects (**Text, Button, Sprite, Image, Content, and Instance**). Also you can add effects to the objects (**Effect**), or add a **Script** to objects you inserted. Also you can add frames, delete frames, insert seconds and delete seconds from the effects you just inserted.
Later on in the book there will be an explicit explanation of these objects, explanation of effects and a complete presentation of Swish Scripting.



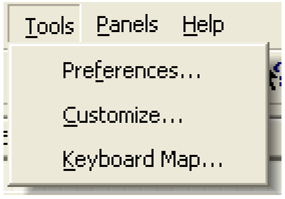
Modify

In the modify panel we can modify object properties like grouping, conversion, breaking and arrangement of objects in the scene, preset transformations than can be applied, vertex modeling (will be discussed later in the book), alignments , justification of text and appearance of text boxes.



Control

In this panel we can set options regarding the testing of our movie. We can play the movie, preview frames in the movie, and browse through the movie.



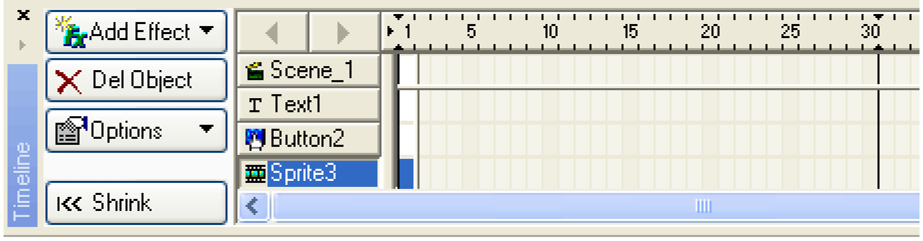
Tools

Here you can set general preferences for SwishMax, customize toolbars for you use and a list of shortcut keys that can be used (**Keyboard Map**).

Panels – here you can set visibility of panels. I warmly recommend to leave them all checked

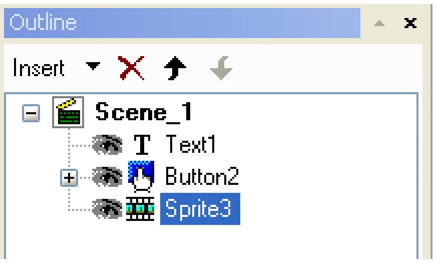
Help – contains help provided from the creators of SwishMax, tutorials, and useful links. Also here you can check out the version of your build of SwishMax. As the program is constantly improving, you can download the latest build from www.swishzone.com .

The timeline



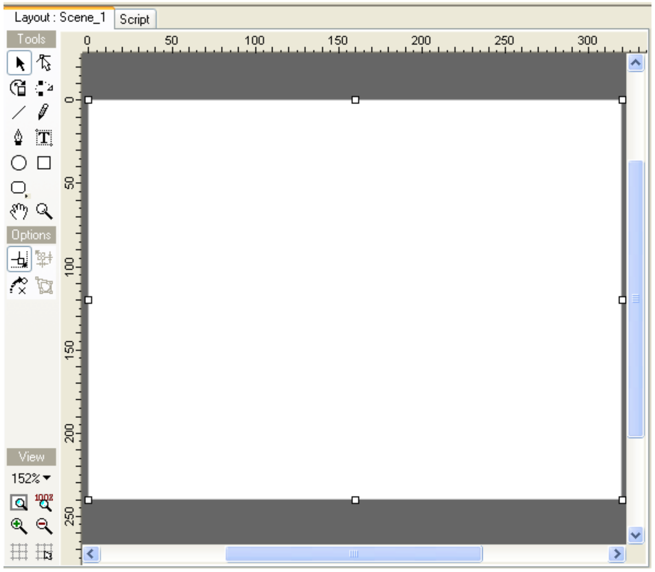
One of the most important toolbars in SwishMax allows users to view the objects in the current scene, **add effects**, and **delete objects**. Also if you notice there is a suite of numbers at the top of the timeline. This represents the frames. A certain amount of frames (can be set and will be discussed later on), form a second (30 in our case, as you see the line at frame 30).
The timeline also has options (display options) that can be accessed by pressing the options button. If you think that the text of the buttons is taking too much of you valuable work space, you can shrink them by pressing **Shrink**.
You will see later on the true importance of the timeline.

The outline



In the outline we can see how many scenes we have, objects and their current position in the movie.
Also by pressing Insert we can insert new objects, using the ✕ mark we can delete objects in our scene. By using the up and down arrows we can modify the position of a certain object in the movie, relatively to other objects.

Toolbars and layout

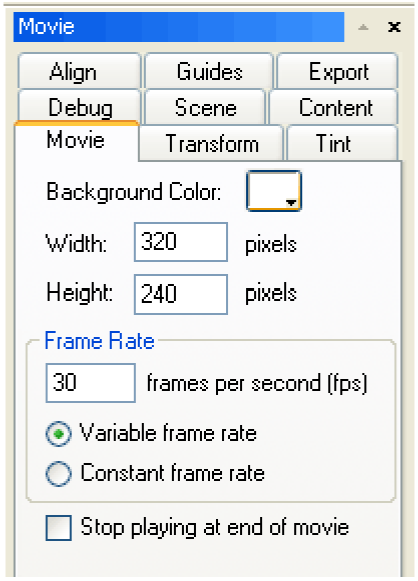


The toolbar represents easy access to creating new shapes in the movie and arrangement, modeling of shapes. The layout on the other hand is the workspace of SwishMax. Here you can view you objects, modify them after liking.

By clicking Script the layout will transform itself to a new workspace that lets you write SwishScript.

__Note__ SwishScript is a term used for the programming language in SwishMax. Almost as complex as the Action Script used in Flash™ and with almost the same characteristics.

Properties



Background Color:

Width: 320 pixels

Height: 240 pixels

30 frames per second (fps)

● Variable frame rate

○ Constant frame rate

□ Stop playing at end of movie

As you probably noticed, in the right side of SwishMax there is this panel. It contains properties for all items used in the movie and others. It contains Export properties for the movie, properties for the movie, selected object etc. Later on in the book there will be an explanation of each item separately for a better understanding.

2. Shapes and Objects

2.1. Shapes

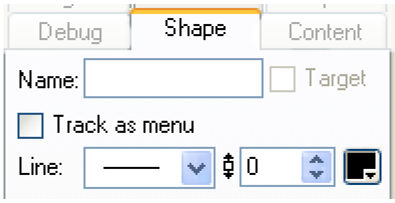
Shapes represent one of the most important things in SwishMax. Using shapes we can draw different objects that can be later modified and animated. By creating a drawing in SwishMax using shapes, we create vector graphics. Vector graphics have a lot of advantages, as they can take up very little hard disc space (or virtual space on the internet). Also they can be zoomed in very much, without losing any quality, or to become pixilated as normal bitmap graphics.

2.1.1. Creating shapes

Line

To create a line, press the icon from the toolbar. Now on the layout left click and drag the line till the desired length and positioning is reached.

Tip Holding the SHIFT key while dragging the line, you will draw only straight lines.



As line properties, we can modify the type of the line (solid, dashed, custom etc.) from the drop down combo. Also you can change line thickness from the input box. Color can be modified by clicking on the black rectangle (default). You can also specify a name for the line.

Pencil

To create a freehand line, you can use the pencil tool. To draw using pencil, click the icon from the toolbar. Now left click on the layout and move you mouse and draw the desired shape. The properties are the same as for line, because it is only a variation of a line (a freehand line technically). If the shape drawn with the pencil is closed, then it will gain the properties of a filled object. Then it has the properties of an ellipse.

Bezier

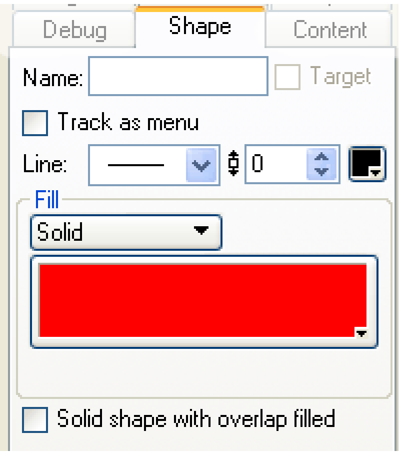
A Bezier line is a precise line that can vary from straight to curve. It all depends on how it is drawn. To draw a Bezier line or shape, click on the icon from the SwishMax toolbar. Now left click on the layout and move you mouse. You will see that a shape outline is being formed. You can add other points to the line by right clicking in the desired places. When you want to finish your drawing, double left click on the layout in case the line is not closed, and a single click to close the shape, thus it will become a filled shape.
The Bezier line has the same properties as a line, but the Bezier shape has the same properties as an ellipse.

Ellipse

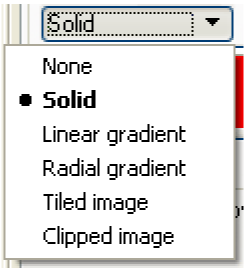
To draw an ellipse, click the ○ icon from the toolbar. Now left click on the layout and drag the shape till the desired shape is reached. Now release the left click mouse button.

Tip Holding the SHIFT key while dragging the ellipse will create only circles.

As line and filling properties for the ellipse shape we have the following:



As you can probably see we have the line properties and we have a new menu that contains Fill properties.
By clicking the ▼ button, you will see the following popup:



Here you can choose the type of filling for the shape. This can be Solid, Linear gradient, Radial gradient, Tiled Image and Clipped image. For more information on colors please view chapter **2.1.2.**

Rectangle

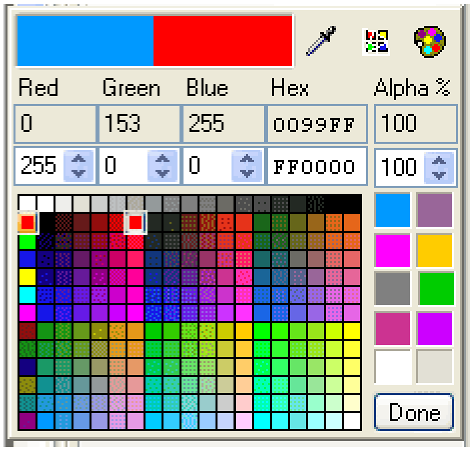
Clicking the □ icon on the SwishMax toolbar, you will be able to draw rectangles. It can be drawn exactly like an ellipse.

Tip Holding the SHIFT key while dragging the rectangle will create only squares.

The filling properties and line properties are the same as the rectangles.

2.1.2. Colors

Colors are widely used in SwishMax. Most commonly they can be set for shapes, but they are also used for setting background colors of the movie etc. We are going to take for example now the filling of a shape, so you can figure out colors better.
When we press the red rectangle in the Fill properties of a shape, we will see the following dialog box:



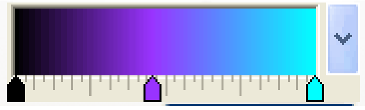
Red	Green	Blue	Hex	Alpha %
0	153	255	0099FF	100
255	0	0	FF0000	100

Here, you will see mostly used colors in SwishMax, marked with big blue in the top left corner of the dialog box is the old color of the object, and marked with red is the color we are currently on with our mouse, and can make a visual comparison. Also in this dialog box we have input boxes for Red, Green and Blue colors that form the RGB. These input boxes can have values from 0-255 for each color, making 16581375 unique combinations of colors. Also we have an input box for hexadecimal introduction of colors, useful to copy a color we want with just one line of code. We can also change the transparency of the shape, named alpha and it is represented in percents. This is from 0-100%, 0 being transparent and 100% being opaque.

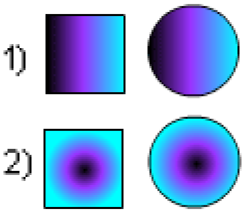
By pressing the 🎨 icon we will be shown the Windows color chooser. Here we can also modify hue, saturation and luminosity for a specified color.
Pressing the ▦ icon will force SwishMax to use only web safe colors, thus making your site viewable even on a 256 color depth display.

Linear and radial gradients

Linear and radial gradients are used to make a shape change a color gradient (fade to a color) or to fade to transparency. For a shape the options look like this:

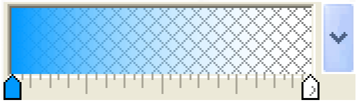


In the big rectangle we can see a preview of the gradient. Underneath this rectangle there are 3 arrows (in my example) that have different colors. As you can see the final result is a fading between the colors. To add more arrows left click on the ruler. To remove arrows, drag the desired arrow out of the rulers' domain.
The difference between linear gradient and radial gradient is that linear gradient fills a shape linearly with color, while the radial fills it circularly (radial).

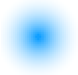


Here we have an example of linear and radial gradients applied to a rectangle and circle. In the first case (1) we have a linear gradient applied to a rectangle and then to a circle. You can clearly see it looks better on the rectangle. In the second case we have the radial gradient applied to a rectangle and circle. We can see that it looks better on the circle, because in the rectangle we have a color "leftover" that fills the corner of the rectangle. This doesn't happen in the case of the circle.

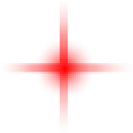
To fade out the endings of a rectangle or an ellipse we can set transparency to a certain arrow in the gradient.



I have in my example a simple fading between blue and a transparent (0% alpha) white. Appling this gradient to a circle as radial gradient we'll have the following shape:



Homework Try to make the following star using two rectangles and a circle:

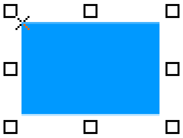


Tips to create it you will need a rectangle that has 3 arrows, center arrow is red and the outer arrows are transparent white. Copy + Paste it and rotate the pasted rectangle with 90º. Now create a radial gradient circle and place on top of the two rectangles. Have fun with your first vector graphic!

2.1.3. Shape modeling

Shape modeling is a very useful feature that allows users to modify how a shape looks like. You can create complex shapes starting from a line, rectangle, or an ellipse.

To do this, create a rectangle first, in your toolbar, the select button will be selected and your rectangle will have eight little hollow square handles, just like in the picture below. This is in case

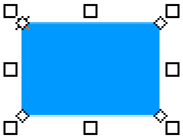


the Scale or Stretch option is activated in the toolbar:
By dragging these squares you will modify the rectangles height, width.

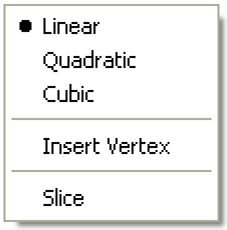
Now, if you activate the rotate or skew option in the toolbar, these handles will turn into eight little round handles. By dragging these, the rectangle will rotate or skew as you drag it.

Using the distort transformation tool located in the toolbar, you will see four star-like handles "⌧" in the corners of the rectangle. By dragging any of them, you will see that the whole rectangle will be distorted after the handle you are dragging.

A more complex shaping can be achieved by using key points, called vertexes. To enter in vertex mode and modify the rectangle, click on the reshape icon in the SwishMax toolbar. Now your rectangle will look like in the image below:

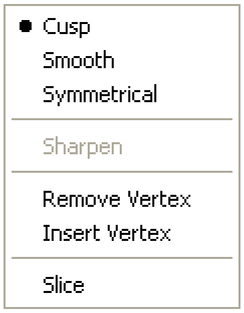


To reshape the rectangle drag the little hollow circles. You know that you are doing this right, because your mouse arrow will turn to . As you can see there are infinite forms for you rectangle, you just need imagination and you can draw it.

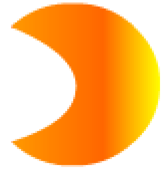


If you want to add more vertex points, or want to delete existing vector points, right click on the outline of the shape. Your mouse arrow will turn into: and the following menu will appear:
You can insert a new vertex or transform the outline between two vertexes to linear, quadratic or cubic. Experiment with these, and you will see the differences between them and realize the importance that they have.

Now if we right click an existing vertex, the following menu will be shown:



Here we can set the type of vertex, Cusp, Smooth or Symmetrical. Also we can sharpen the vertex in case it is smooth, or we could just remove the vertex. Usually the vertexes are cusp for rectangles and Smooth for circles. Now if you select smooth for a vertex of a rectangle, you will see the whole area around the vertex will gain a round shape and two cyan small circle handles appear “◆”. The best way to learn their meaning is to experiment. By selecting symmetrical, the “◆” handles are equally oriented from the vertex, that is situated at the half way between them.



Homework Try to create a shape similar to a half moon, like the one shown in the picture. Start from a circle and use all techniques that you’ve learned so far.

2.1.4. Gradient modeling

This can be achieved by creating a filled shape and selecting a type of gradient (linear or gradient). Now click on icon in the toolbar. Eight blue handles will appear. These can be squares or circles, depending on which icon is selected: the scale or stretch one: (square handles) or the rotate or skew icon: (circle handles). You can drag these handles, depending on what you want to do with the objects gradient.

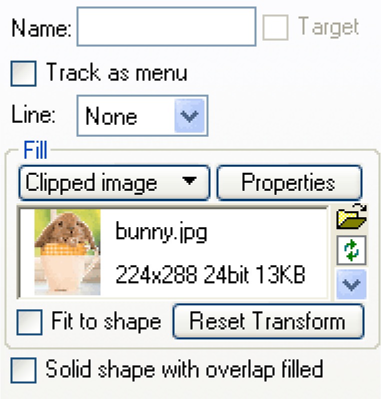
2.2. Objects

The Scene

It is a “home” for all other objects present in the movie. A movie can have more than one scene, and these can be linked with one another.

Images

You can insert images by clicking on the insert image button. The images properties appear as below:



Name: □ Target

□ Track as menu

Line: None

Fill

Clipped image ▼ | Properties

bunny.jpg
224x288 24bit 13KB

□ Fit to shape | Reset Transform

□ Solid shape with overlap filled

By clicking on the properties button a new panel will be displayed:

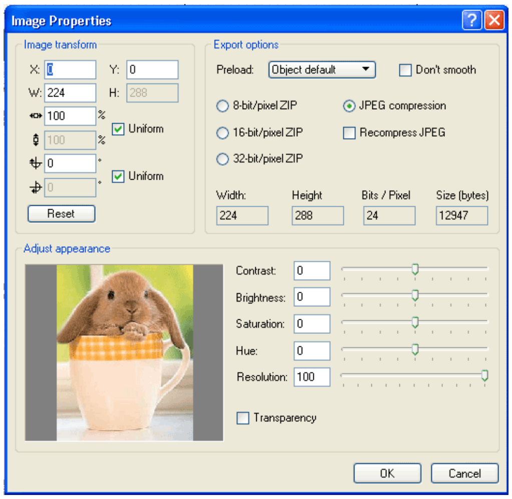


Image Properties

Image transform

X: 0 Y: 0
W: 224 H: 288
100 % ☑ Uniform
100 %
0 ° ☑ Uniform
0 °
Reset

Export options

Preload: Object default ▼ □ Don't smooth
○ 8-bit/pixel ZIP ● JPEG compression
○ 16-bit/pixel ZIP □ Recompress JPEG
○ 32-bit/pixel ZIP

Width:	Height	Bits / Pixel	Size (bytes)
224	288	24	12947

Adjust appearance

Contrast: 0
Brightness: 0
Saturation: 0
Hue: 0
Resolution: 100
□ Transparency

OK | Cancel

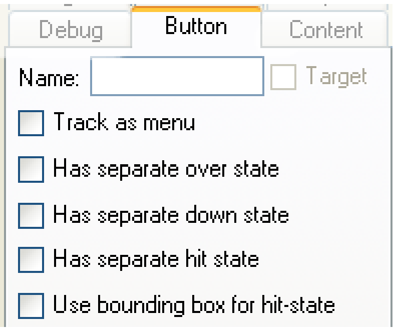
Here you can modify compression of the image, position of the image (X & Y axes), Width, Height, Rotation. Also you can modify the Contrast, Brightness, Saturation, Hue and Resolution of the image. These are very powerful build in features of SwishMax, letting you practically obtain a different image only by modifying a few values.

By clicking on the transparency checkbox, a new menu will appear, letting you choose the color you want to make transparent and the tolerance (how close to that color) should the pixels be made transparent.

Tip If you insert images in your movie, and you want to keep a maximum of clarity and minimum file size, try to compress large image files and save them as .jpg . This helps you movie to stay small with very little image clarity loss. For the small image files, a .png or .gif compression is recommended, the file size is very small and the clarity is the same. So my tip is to avoid .bmp extension images, as they are very large. Also in the properties panel of the image, test different compression for jpg and see how much loss you have in image quality and how it affects the file size of the new image have.

Buttons

Buttons are special objects in SwishMax. You can create them by pressing the insert button icon. In the outline the buttons icon will be this one:
A button has the following properties (shown in the image):



Debug | Button | Content

Name: □ Target

□ Track as menu

□ Has separate over state

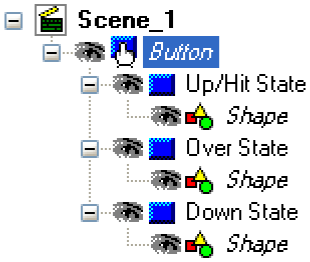
□ Has separate down state

□ Has separate hit state

□ Use bounding box for hit-state

A button has four states: Up, Over, Down and Hit state. They are referring to the properties of the button on different mouse actions: When the mouse is not over the button (Up), when the mouse is over the button (Over), when the left mouse button is pressed on the button (Down). The hit state refers only to the virtual rectangle that surrounds the button and triggers the events. To add more states to a button check the state you want in the buttons properties.
The most accurate and easiest way to create a button is to create an object and group it as a button. After that, editing the objects will not represent a problem.

We would like to create a simple button that consists of a rectangle. The normal color of the button is red. When our mouse moves over the button it turns to blue and when the button is pressed it turns to black.
To do this, first create a rectangle. Set its color to red. Now group it as button. In the buttons properties panel check *Has separate over state*, *Has separate down state*. Now in the outline panel click the "⊞" icon. Your button will look as following in the outline panel:



Scene_1
Button
Up/Hit State
Shape
Over State
Shape
Down State
Shape

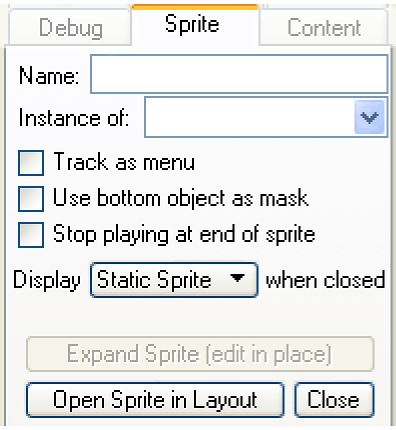
On the up/hit state click on the shape . Now check its properties. The color should be red. Now click the shape in the Over state. Change the color to blue. On the down state, change the color of the shape to black. Now test your movie. Your button should work exactly like described above. Congratulations! You have just created your first button.
If it doesn't look right to you, or you example didn't come out right, check out ***button.swi*** provided with the book. **77**

Sprites

Sprites are the most complex objects in SwishMax. The equal to a movie clip in Flash™, the sprite can contain different objects in it. It is like a little movie in you movie. It has the ability to play a different animation and different actions in each one. That is how you can play multiple animations, play multiple sounds at once. Another good thing about sprites is that they are interactive... meaning each one can communicate with the other.
Sprites can be created by pressing the insert sprite button: . In the outline, the sprites icon will be the following: , followed by the name of the sprite.

The most efficient way to create a sprite (in my opinion… saves a lot of time), is to select an object or multiple objects and group them as a sprite. This way the objects keep their position in the layout and it can be used to lighten up the outline if there are many objects that need separate animating.

The sprite has a specific timeline and the following properties (shown in the image):


Debug
Sprite
Content
Name:
Instance of:
□ Track as menu
□ Use bottom object as mask
□ Stop playing at end of sprite
Display Static Sprite when closed
Expand Sprite (edit in place)
Open Sprite in Layout
Close

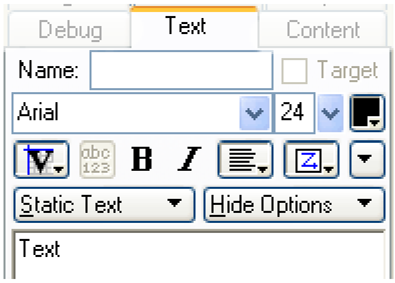
One of the power features of a sprite is masking. By checking the *Use bottom object as mask* makes the sprite show only the information as much as the bottom object lets. To be more specific, create a rectangle as the bottom object and an ellipse that that surpasses the rectangle and has a different color. When the movie is tested, the sprite will display only the ellipse, trimmed by the bottom rectangle.

Another advantage of using Sprites in your movies is that you can easily reuse them throughout your movie, saving time and with only a slight increase to the file size of the movie.

77

Text

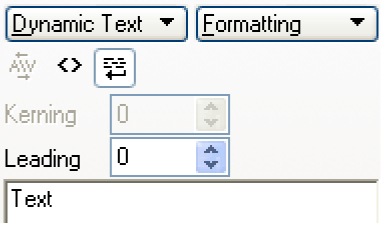
Texts are often used in SwishMax. You can use them to show from static data to dynamic data. Referring to static data, I mean text that does not change at runtime. Dynamic texts have the ability to change at runtime. The text properties look like in the image below:



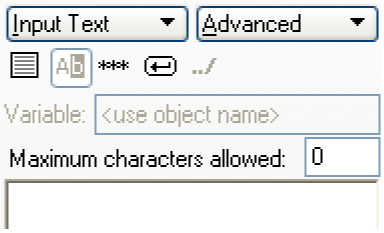
You can specify a name for the text, select your font type, font size, font color, type of alignment of text, type of text (normal, bold **B**, italic ***I***.
Also you can select the type of text (Vector or Pixel) from the dropdown menu drawn with a **V** in the image. Vector fonts are normal fonts and usually these kinds of fonts are used. Pixel fonts on the other hand are fonts that are very clear to read even at 8 pixels. These are used for professional presentations and web pages.

From the Static Text drop down menu you can see the following options:

Static text is generally used only to display static text.
Dynamic text is used to add interactivity to the text. It can change at runtime.
Input text is used to input data that can be later used in the program.



Other options:
If we select dynamic text, and formatting we will see a new set of options. A good option to use is *Render text as HTML*. This is very useful if we import external texts that have html formatting, thus the text displayed can contain Bold, Italic, Underlined texts, etc. This option can be also used internally too. To write with bold write you text between <b>sample text</b>. To write with italic text, use: <i>sample text</i>. For underlined text, use <u>sample text</u>. To begin a new paragraph with a bullet, use <li>



Input Text ▾ | Advanced ▾

Variable: <use object name>

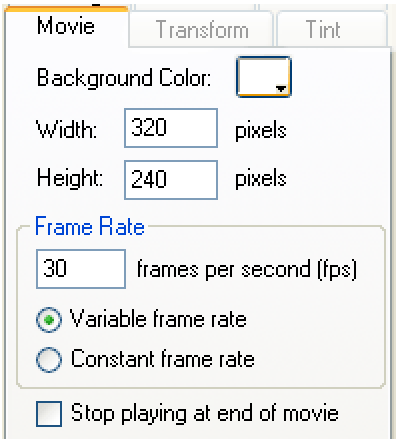
Maximum characters allowed: 0

If we select Input text for text type and Advanced from the drop down menu we will see some options specific for input boxes. These are masking the text as a password, using the " * " character. Also by clicking on the ▤ icon, the input box text will be framed in a black border with a white background. By selecting the ↵ icon, the input box accepts the enter key and supports multiple line text.

Many options are only available for different types of text boxes. It is up to you to discover which option is enabled for which type of text. **76**

The movie

The movie contains all possible objects, scenes, scripts etc. It is created when pressing the New 🗋 icon. As for properties, these are shown in the image below:



Background Color:

Width: 320 pixels

Height: 240 pixels

Frame Rate

30 frames per second (fps)

● Variable frame rate

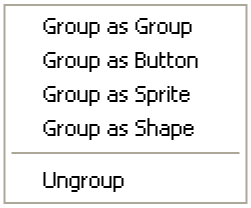
○ Constant frame rate

□ Stop playing at end of movie

You can set the movies background color, set the width and height of the movie.
Frame rate is available for all types of movies. It represents the number of frames (images) displayed in a certain amount of time. In our case, frames per second (fps). Movies on you DVD, or on you TV are running from a frame rate of 24 to max 30. The human eye does not distinguish all the frames after the maximum of 30 fps. By making a slow frame rate, the movie will run choppy and it doesn't look nice. For a smooth transition use a frame rate higher than 24.
Variable frame rate (VBR) is an option used especially in music MP3 files. This is used for little quality loss and smaller files size. If the check box *Stop playing at end of movie* is checked, the movie will stop after running the animation in the timeline, it doesn't loop infinitely as usual.

2.3. Grouping

Grouping is one of the most useful options in SwishMax, letting the user group multiple objects together, making the outline look cleaner, and also useful for sorting different objects. For example you want to create a sun with shapes, you create a circles and some triangles around it, it would be useful to group them as one and name the group sun… right?
Grouping can be achieved in many ways, but the fastest way is to right click on the selected objects, and move your mouse over grouping. Now the following menu will appear:



Grouping as group, creates a new object called group with this ▣ icon. Now you can rename it to whatever you like. It contains all your objects that are inside this group.
Grouping as button, creates a button that contains the selected objects.
Grouping as sprite, groups your objects into a sprite, creating a new movie inside you movie.
Grouping as shape, groups all your objects into a shape. This is a very helpful option especially for texts because texts gain the ability of filling transformations. Now you can add gradient to your text! There is only one bug to the whole problem. After grouping your text as shape, you cannot go back to text mode and modify the text.

3. Basic actions and effects

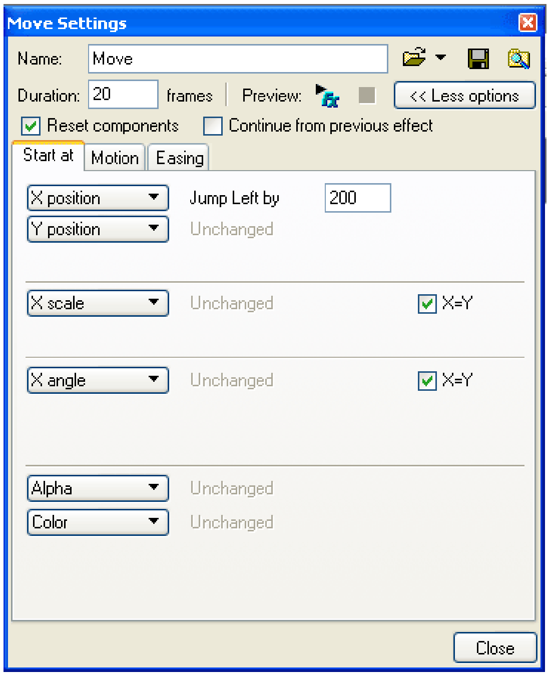
The stop action

One of the most used actions in SwishMax is the stop action. The easiest way in my opinion to introduce new code in SwishMax is to right click on the timeline and select the desired option. For the stop action, right click and select: movie control → stop() . Now the movie will continue playing until the stop action is reached. In the script are you will see the following code:

```
onFrame (frame_number) {
   stop();
}
```

The move effect

Movement applied to any object makes it move to specified coordinates. Also it can be used to change the color of an object, change the transparency of an object, rotate an object and many other options.
To make your first movement, first create a rectangle in a new movie. Now right click on the timeline (rectangle) where you want the move effect to start and select move. A 10 frame move action will appear in the timeline for the rectangle. Double click the move effect. You will see the following dialog box:



Name: Move
Duration: 20 frames
☑ Reset components □ Continue from previous effect
X position — Jump Left by 200
Y position — Unchanged
X scale — Unchanged ☑ X=Y
X angle — Unchanged ☑ X=Y
Alpha — Unchanged
Color — Unchanged

Now click on the X position button and select jump left by. An input box will appear and you can set the amount of movement for the rectangle. In my example I entered 200. This number represents the number of pixels that the rectangle will move. For duration I set the length of 20 frames. You can choose how much you like. The more frames, the longer the animation will last. Now test you movie. You will see it repeats the moving effect. Congratulations! You have just made your first animation!
Have fun and test the movement on vertical, horizontal, test scaling, angles, alpha (transparency) and color.

Place and remove

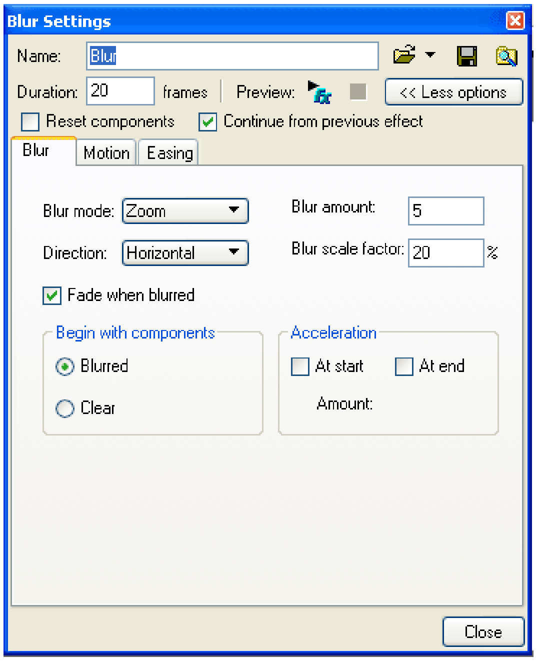
Probably the most used actions used in SwishMax allow users to show a certain object at runtime at a specific frame. Also useful to create animations by using sequential images. Try to import a .gif file (animated graphic). You will see it creates a sprite that contains a lot of little images that have the place and remove action attributed to them. Now let me explain a little better by giving you an example.
Create a rectangle in the layout and an ellipse. On frame 1 for both objects put a remove action. On frame 30 put a place action for the rectangle, on frame 40 a place action for the ellipse and a

remove action for the rectangle and on frame 60 a remove action for the ellipse. Now test you movie. You will see that after a little time the rectangle appears. When the ellipse appears, the rectangle disappears and after a while the ellipse disappears too. Easy enough right?

Fade in & Fade out

Fade in and Fade out is used for a specific object to “fade in” (become visible) or “fade out” (become invisible). Just place an object in the layout and on the timeline right click and select fade in and after that a fade out effect. You will see that the object fades in and out.

Blur



Name: Blur
Duration: 20 frames
☐ Reset components ☑ Continue from previous effect
Blur mode: Zoom
Blur amount: 5
Direction: Horizontal
Blur scale factor: 20 %
☑ Fade when blurred
Begin with components
● Blurred
○ Clear
Acceleration
☐ At start ☐ At end
Amount:

Blur is a processor consuming effect, but the results of a blurring an object is hardly replaced by another. You can set the blur mode, blurs direction, Blur amount and blur scale factor. Test these options out and you will see the results of the blur effect for different values.

Core effects

Using core effects, you can almost create all of the 200 build in effects of SwishMax. Also you can design you own effects. The largest usability of these effects is found for text objects. You can edit any core effect by double clicking the effect in the timeline. As following I will explain some of the most used core effects.

- Transform

This core effect is very similar to the move effect but with a lot more options

- Snake

A very powerful effect lets you move objects after a virtual trajectory, given by a mathematical formula. Doesn’t hurt if you know some math, but that isn’t really a problem because you can experiment with the values.

- Wave

Useful on texts, makes the letters wave in different ways.

- Typewriter

Used for text objects, it creates the idea of writing on a typewriter, by displaying one character at a time (or as you set in the effects preferences).

4. An introduction to Swish Script

Swish script is the new feature added to swish. It makes SwishMax very powerful and prepares it for professional use.

4.1. Variables

Variables exist in all programming languages, and thus present in Swish Script too. These are of three types: **STRING**, **EXPRESSION** and **BOOLEAN**. These exist to tell the programming language to remember something and use later.
Now let's take a moment to explain all of these variables.

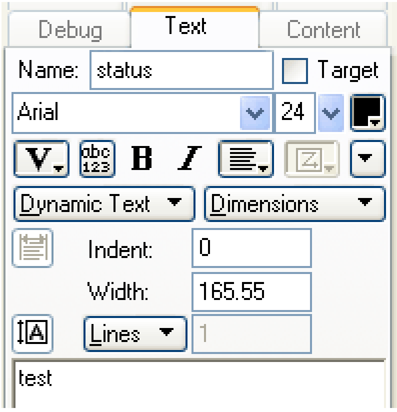
STRING

The string variable is used to identify information that is in brackets

Ex: *name = "John"*

Our programming language (well I am referring to SwishMax now) will remember that the variable ***name*** has assigned to it ***John*** information. Also we can have numbers in brackets: Ex: *weight = "80"*. This tells the variable ***weight*** to remember ***80***.

As a concise example, crete a new movie. In it create a dynamic textbox, and assign it the name ***status***, containing the text *test*.



Name: status □ Target
Arial 24
Dynamic Text Dimensions
Indent: 0
Width: 165.55
Lines 1
test

Remember that only dynamic text support variable handling and calculations.

Now click on ***Scene_1*** and add the following code:

```
onEnterFrame() {
name = "john"}
```

Now draw a rectangle on the layout and add the following code to its script:

```
on (release) {
status = name}
```

Now test your movie in the flash player (or html as you prefer) and click the rectangle you drew. Magic! The text ***test*** will transform into ***john***.
Explanation: we declared a variable called name and we told it to remember john (the text in brackets). On the button action release we told the text ***status*** to replace its current containing information with the variables content. And that is how we made magic! If you can't figure it out exactly, or want to check my example, open the ***variables1.swi*** file provided in the package.

Now let's move to our next variable, **EXPRESSION**

Just like his brother, the **STRING**, the **EXPRESSION** is used to remember numbers and operations with numbers. Unlike the String, the expression does not support letters (only if they are variables that declare numbers), only numbers. You can specify the text like this(in scene_1 script):

```
onLoad() {
age = 22}
```

Now we told SwishMax to remember 22 as a number. To display the number we use the same tutorial from above (at STRINGS) and just replace name = "john" with age = 22. On the rectangle script, write:

```
on (release) {
status = age}
```

When you press the rectangle, Magic again! The textbox will display 22. Just as easy. Now let's make a little modification and tell status something else:

```
on (release) {
status = age + 10}
```

Now, when pushing the rectangle, the text ***status*** will display 32. Why? SwishScript automatically calculates what you told him.

Another good example (having fun are we) follows:

In Scene_1 replace the code with the following:

```
onLoad () {
age = 22}
```

Write as script on the rectangle:

```
on (release) {
age += 1;
status = age}
```

Now test your movie and push the rectangle several times. What happens? It adds up the age with 1. Why? Because we told him with += to add the variable ***age***, 1.

This code can also be written like:

```
age = age + 1;
```

It’s the same thing, but using += save us precious coding time and is easier to follow.
To see a running example open ***variables2.swi*** . The blue rectangle exemplifies the first example, the red the second and the green the third example.

Ok, now forward to our friend the **BOOLEAN**. He is used to return TRUE or FALSE, true being of value 1 and false being of value 0. Using this we can find out information about the state of a different object (sprite) or we can tell it something.
Let’s follow the next example closely:
Create a new file. Draw a circle. Convert it to sprite. Name the sprite ***circle***.
Draw a rectangle, and add the following code to it’s script

```
on (release) {
circle._visible = false}
```

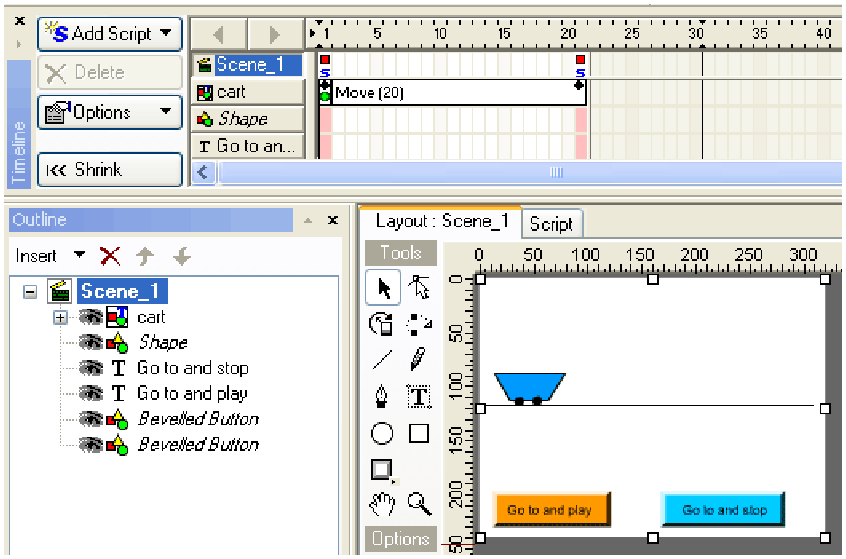
Now test your movie, and click the rectangle. Magic! The circle disappears. If you want to take it further create one more rectangle and add the following code:

```
on (release) {
circle._visible = true}
```

Now when testing the movie, when you click the first rectangle, the circle disappears and when pressing the second rectangle the circle will appear. How nice.
For a working example, run ***variables3.swi*** .

4.2. GotoandPlay(); GotoandStop()

GotoandPlay(*frame number*) and GotoandStop(*frame number*) actions are very often used in SwishMax. They are used to tell the movie to go to a certain frame number and play the action from there or just stop the action on a certain frame.
If you check out the example: ***gotoand.swi*** provided with the book you will see the following:



I made a cart (using a reshaped rectangle with 2 circles for wheels), a line (representing the ground and two beveled buttons one orange that has a text over it with the text “go to and play” and a cyan beveled button with the “Go to and stop” text over it.
In the timeline, I made the movie stop on frame1 and on frame 21(so the movie doesn’t loop, this can be removed, and the movie goes to frame 1 after the move action is finished). For the cart to be viewable at the beginning of the movie, I added a place action. After that follows a move action of 20 frames, making the cart move to _X=230.
For the orange beveled button I have added the following code:

```
on (press) {
gotoandplay(2)}
```

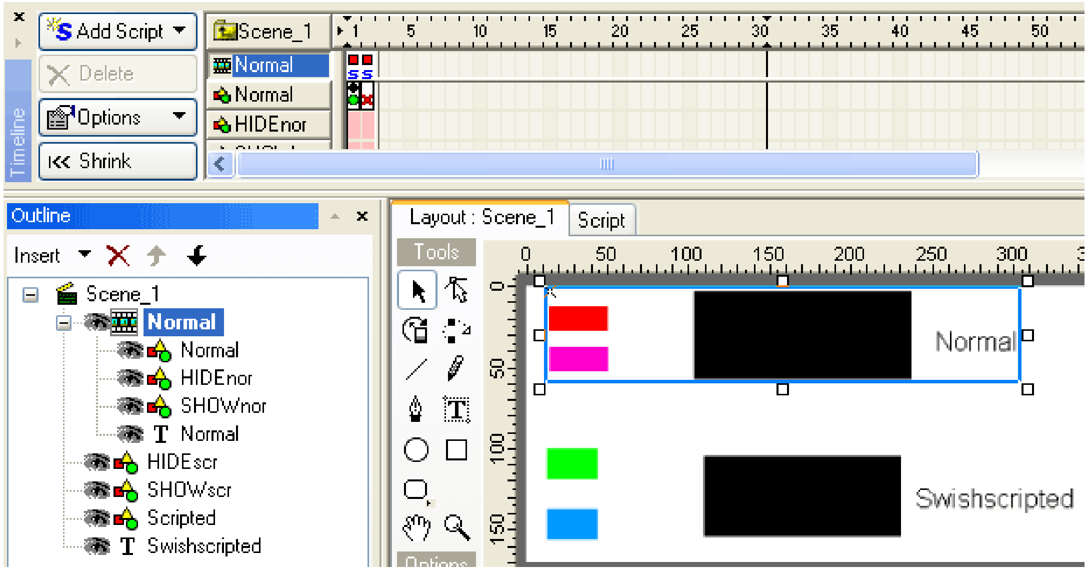
and for the cyan beveled button:

```
on (press) {
gotoandstop(10)}
```

When we test the movie, and press the orange beveled button, the movie plays the animation from frame 2. When the cyan beveled button is pressed, it goes to frame 10 and stops, thus making the cart stop in the middle of its course.
Going to frames and playing or stopping can be used to control sprites, scenes, etc. For other options regarding the goto command, use the guided mode in scripting.

4.3. Placing and removing objects

Open the ***visibility.swi*** movie. If you don’t want to learn from the created example, create your own.



Create a new file. In it, create a black rectangle and name it ***Normal***. Now create two other little rectangles and name them ***HIDEnor*** and ***SHOWnor*** . ***HIDEnor*** is red and ***SHOWnor*** is magenta. Also you can create a text and write in it normal. Group them as a sprite and name it ***Normal***.
Inside the sprite, in the timeline put on frame 1 and 2 stop actions. For the large rectangle add a place action on frame 1 and a remove action on frame 2.
Add the following code to the red rectangle:

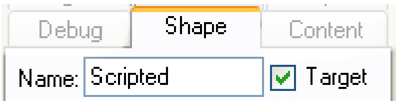
```
on (release) {
gotoandstop(2)}
```

,and this code to the magenta rectangle:

```
on (release) {
gotoandstop(1)}
```

If you test the movie, and press the red rectangle, the big black rectangle will disappear and if you press the magenta rectangle it will reappear. This is because on the red rectangle code we told it to “remove” and in the magenta code we told it to “place”.

Now for the swish scripted version of visibility (place and remove), create in the same movie, in _root (base layer in the scene) a big black rectangle and call it ***Scripted***. Check the target check box.



Name: Scripted ☑ Target

This is very important, otherwise the actions we send won’t be recognized. Now create 2 rectangles, a green and a blue one, called ***HIDEscr*** and ***SHOWscr***. Add the following code to the green one:

```
on (release) {
scripted._visible = false}
```

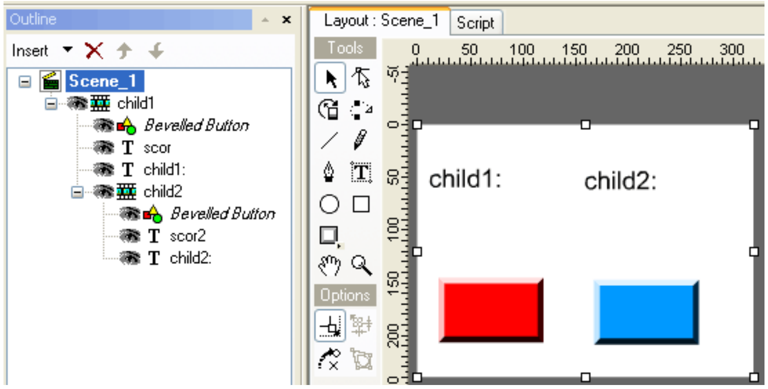
and for the blue one:

```
on (release) {
scripted._visible = true}
```

As you can see, when testing the movie, the scripted version does the same as the normal version, but more efficiently because there isn’t any timeline usage.

4.4. _root and _parent explanation

To exemplify _root and _parent better, we will create a little game.



Create a new movie. Now create a static textbox and enter the text ***child1:*** in it. Create another textbox (dynamic this time) and name it ***scor***. Place them one near another. Then create a ***beveled button*** (using auto shapes). Now select them all (in outline using the CTRL Key and left click on them and select *Grouping --> Group as Sprite*). Now rename your newly created Sprite into ***child1***. Now click on the ***child1*** sprite and copy it (CTRL +C) and paste it in ***child1*** (CTRL+V). There should be another sprite created. Move the second sprite, called ***child2*** as in the image above. Change the static text's content to ***child2:*** and name the dynamic text in ***child2***, ***scor2***. Also you can change the beveled buttons color to blue.

Now after all is done just like in the picture above, I will explain how _root works and _parent works:

_root and _parent describe paths, paths that we can use to control different objects. Using our newly created example, add the following code to the red beveled button:

```
on (release) {
scor = "OK"}
```

If you test your movie, the first dynamic text box (***scor***) will display OK. Now we want ***scor2*** to display OK. For this we have to give its path. This is done in the code below:

```
on (release) {
_root.child1.child2.scor2 = "OK"}
```

At runtime you can see that the second sprites dynamic text box changes to “OK”.

Note that _root represents the movie, the first level in the outline.

Now we want ***scor*** to display OK when pressing the blue beveled button. To do this add the following code to the blue beveled button:

```
on (release) {
_parent.scor="OK"}
```

_parent represents one path level below the current sprite.

As you test your movie, you will see that ***scor*** will display “OK”.

This can also be done by writing the exact path to ***scor*** like:

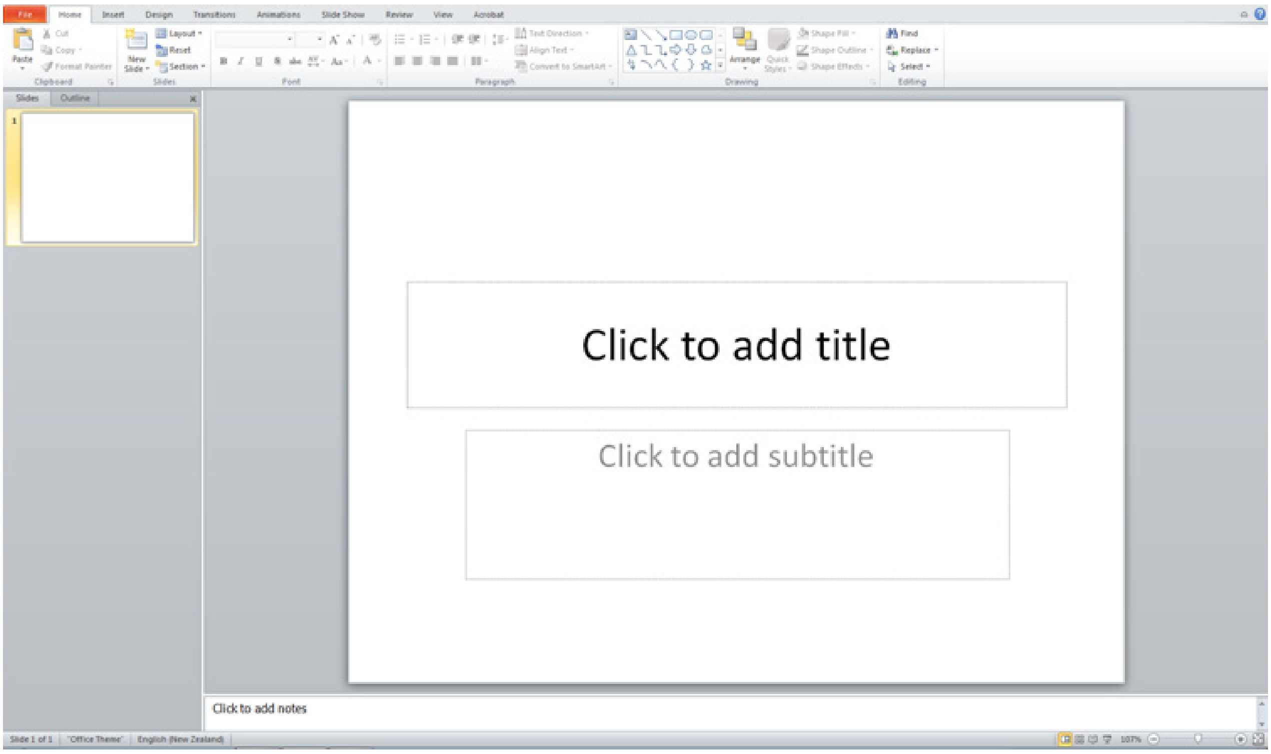
```
on (release) {
_root.child1.scor = "OK"}
```

If you are still experiencing problems please view the example provided with the book, ***_root_parent.swi*** .

Lab. 11: Microsoft PowerPoint

Reference: Working With Microsoft PowerPoint/ Eastern Institute of Technology

Microsoft PowerPoint is an electronic presentation program that helps people present a speech using a collection of slides. A PowerPoint presentation is a collection of slides that can be used to create oral presentations.



Click to add title

Click to add subtitle

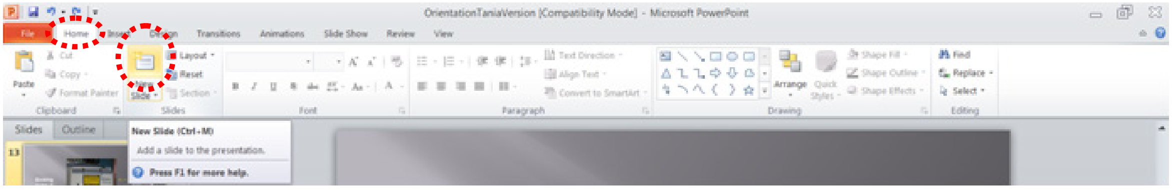
Click to add notes

Slide 1 of 1 "Office Theme" English (New Zealand) 107%

This is the standard first slide of a PowerPoint presentation.

Inserting a New Slide

Home << **New Slide**



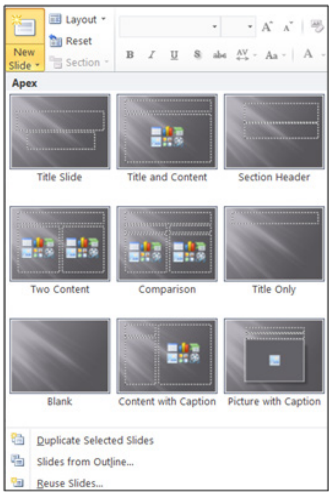
OrientationTaniaVersion [Compatibility Mode] - Microsoft PowerPoint

New Slide (Ctrl+M)

Add a slide to the presentation.

Press F1 for more help.

Choose the Layout required, the layouts are pre-set but can be customised as needed



Views

There are four different views in Powerpoint:

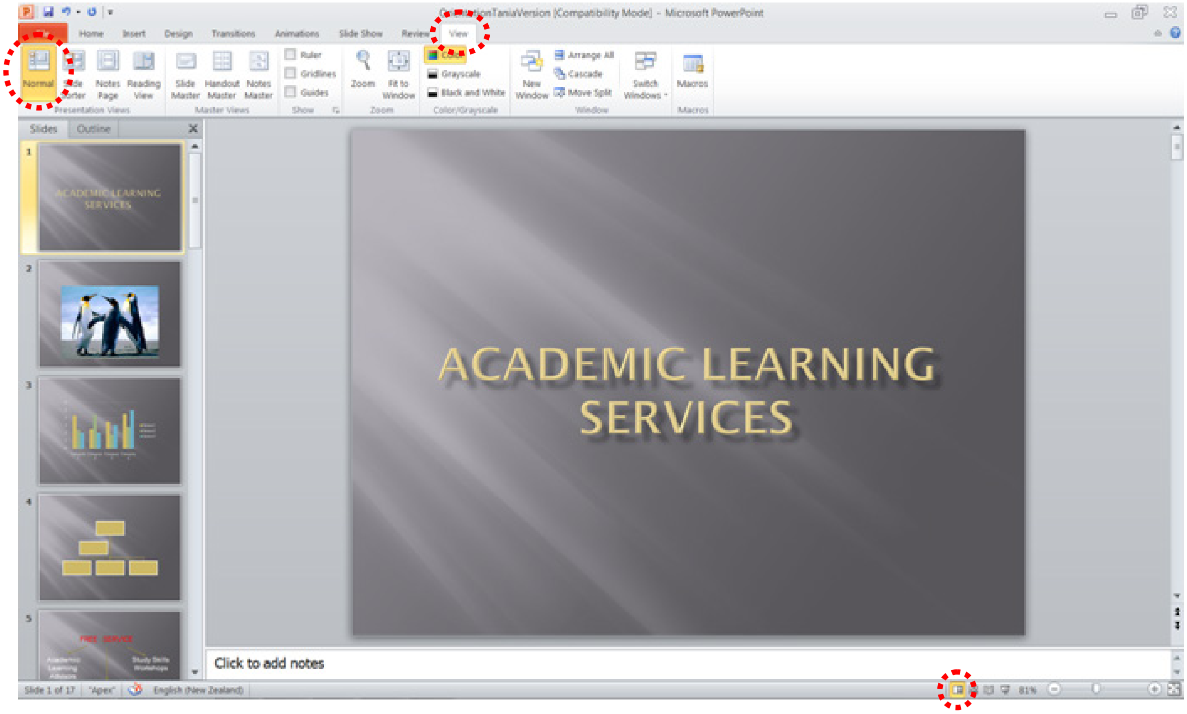
- Normal
- Slide sorter
- Notes page
- Slide show

Each view is used for a different step in creating your Powerpoint presentation.

Normal View

View >> **Normal**

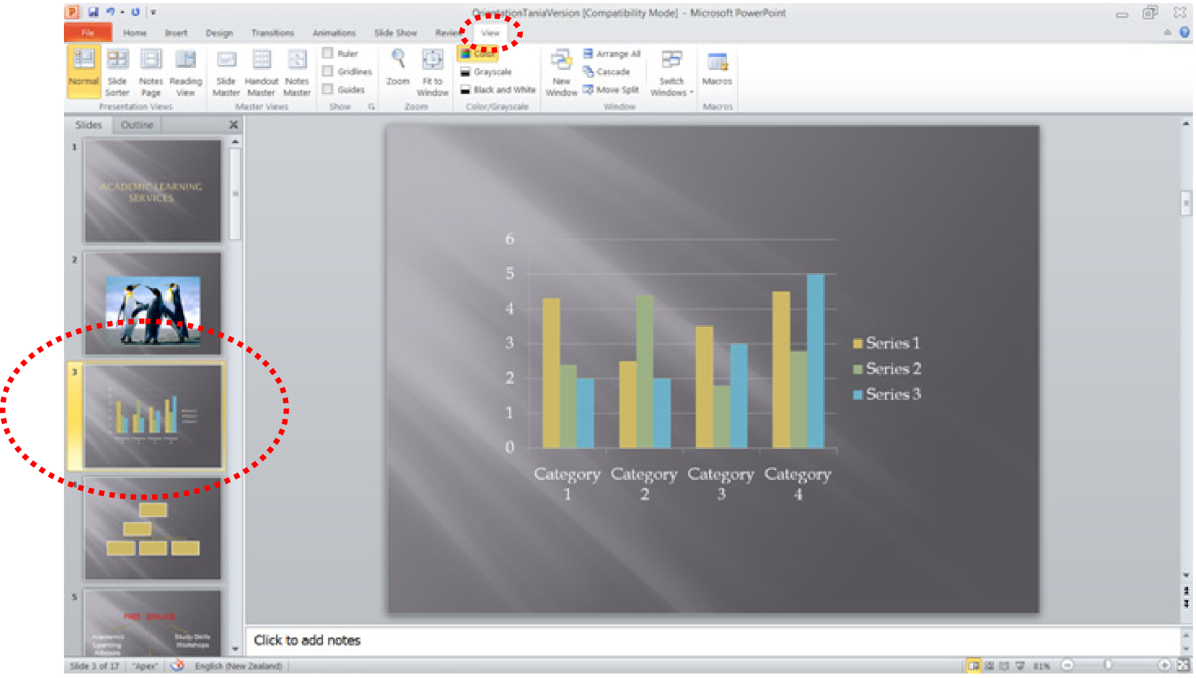
This view is used when creating and designing your slides.



NB: This view can also be altered using the icons at the bottom of the page on the task bar.

Selecting a Slide

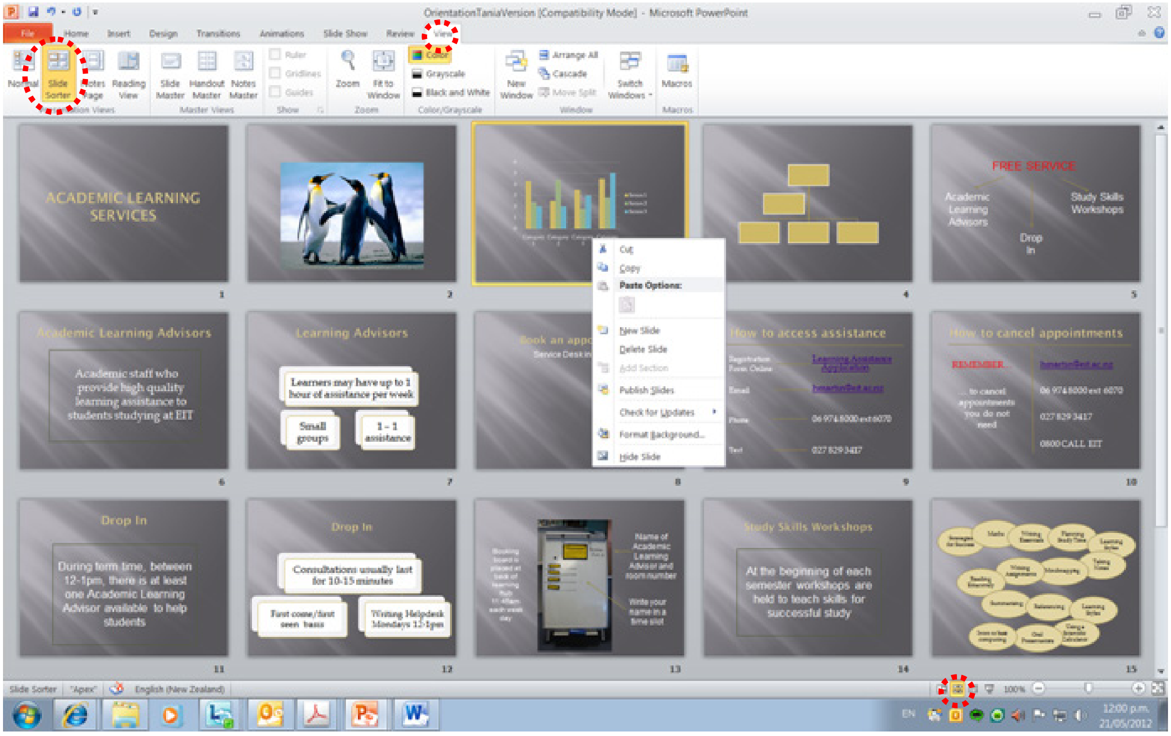
To select a slide, click the slide in the slides panel (a thick borderline appears around the slide).



Slide Sorter View

View >>Slide Sorter

This view is good to use when organising your slides. Use when you want to delete, copy, paste or move your slides.



Deleting a Slide

Select the slide that you would like to delete and press the Delete key on the keyboard OR right click on the selected slide and select Delete Slide from the menu.

Copying a Slide

Select the slide you would like to copy, right click on the selected slide and select Copy from the menu.

Pasting a Slide

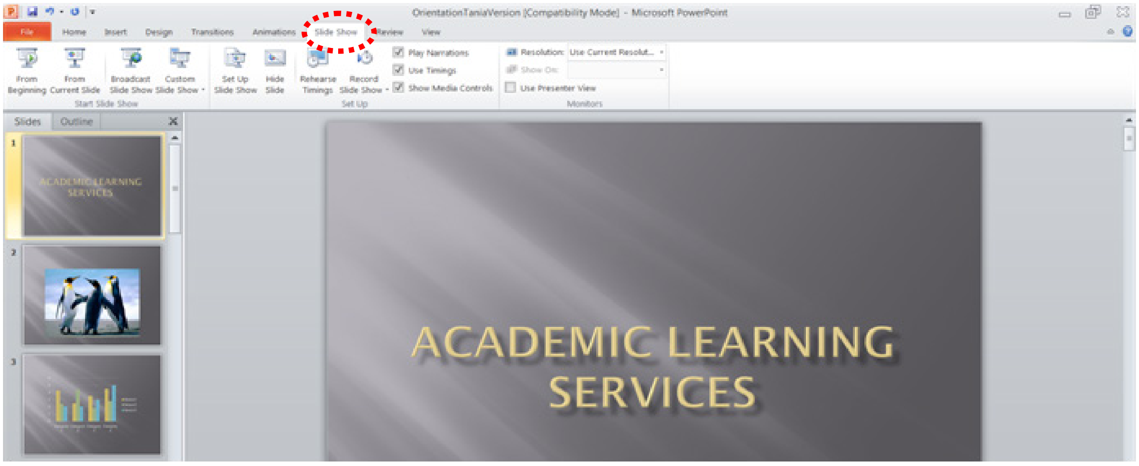
Position your cursor where you want to paste the slide (eg. between 6 & 7) << Ctrl + V

Moving a slide

After creating a Powerpoint presentation you may decide to change the order of your slides. Select the slide you wish to move (a border will appear around the selected slide), and drag the slide to where you want to reposition it and drop it into the position. The slide number sequence will automatically update.

Slide Show

View >>Slide Show



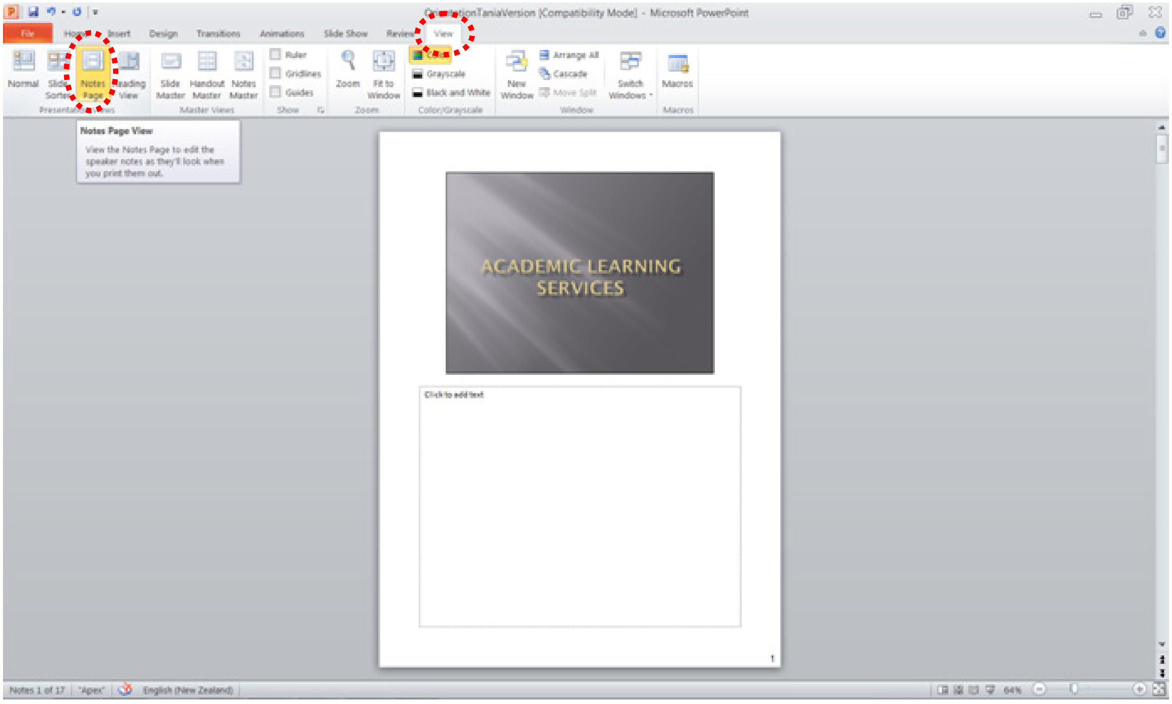
It shows your presentation in full screen. It can also be used to check any animations, transitions or timing at the final stage of preparation for your presentation

ACADEMIC LEARNING SERVICES

NB: Press ESC key to return to normal view.

Notes Page

View >> **Notes Page**



Notes Page View

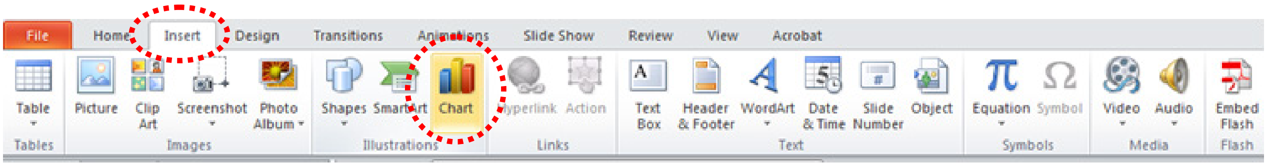
View the Notes Page to edit the speaker notes as they'll look when you print them out.

ACADEMIC LEARNING SERVICES

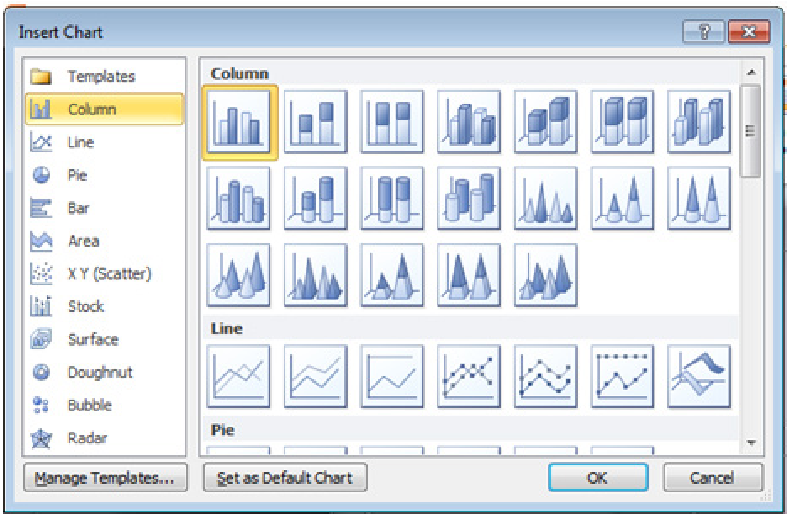
Click to add text

NB: You can add and view your notes for each slide.

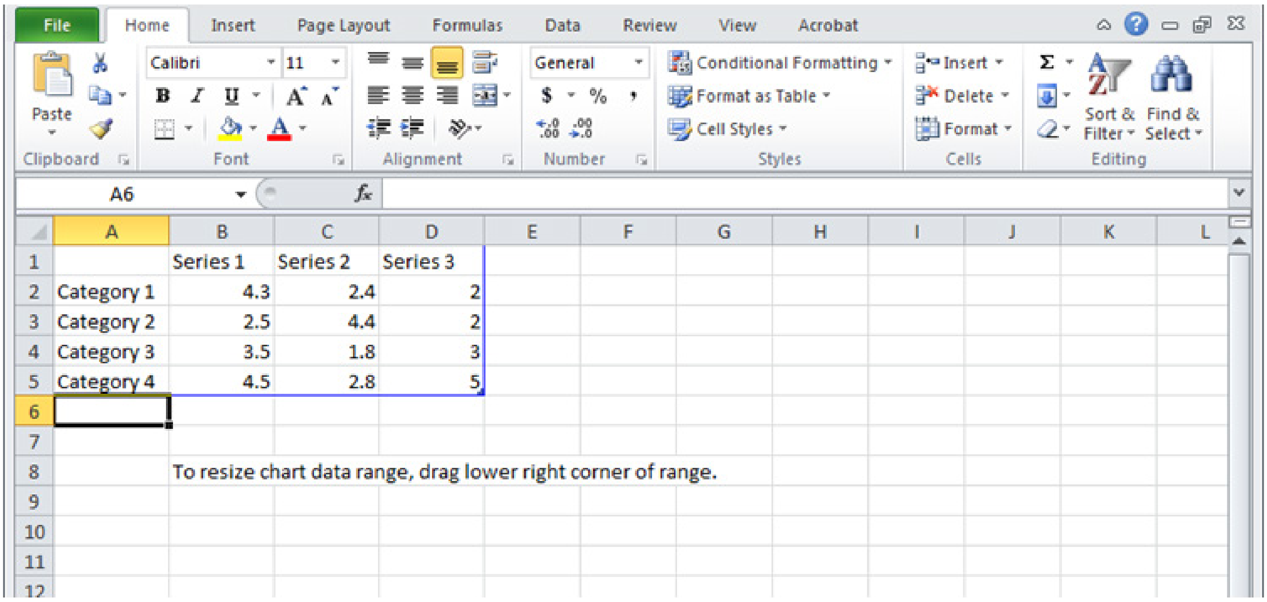
Select the slide you are going to put the chart on
Insert << Chart



Choose the type of chart you want to use << **OK**



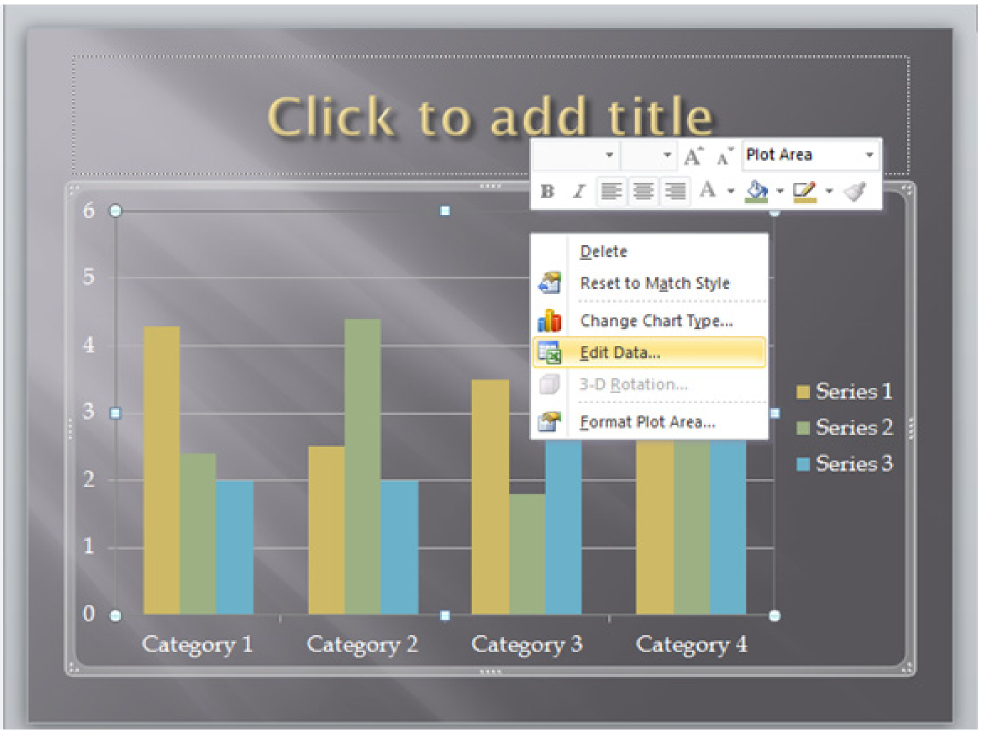
An excel window will open, so that you can edit / enter the data in your chart



	A	B	C	D
1		Series 1	Series 2	Series 3
2	Category 1	4.3	2.4	2
3	Category 2	2.5	4.4	2
4	Category 3	3.5	1.8	3
5	Category 4	4.5	2.8	5

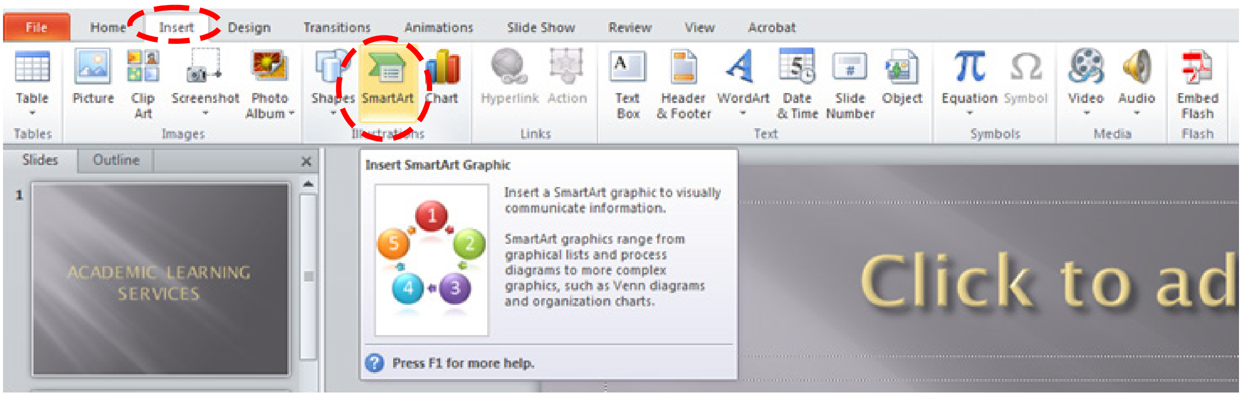
To resize chart data range, drag lower right corner of range.

NB: If the table disappears, right click on your chart >> **Edit Data** and the table will appear again.



NB: To change your chart type, right click on your chart >> Click on **Change Chart Type**

Choose **Insert** >> **Smart Art**



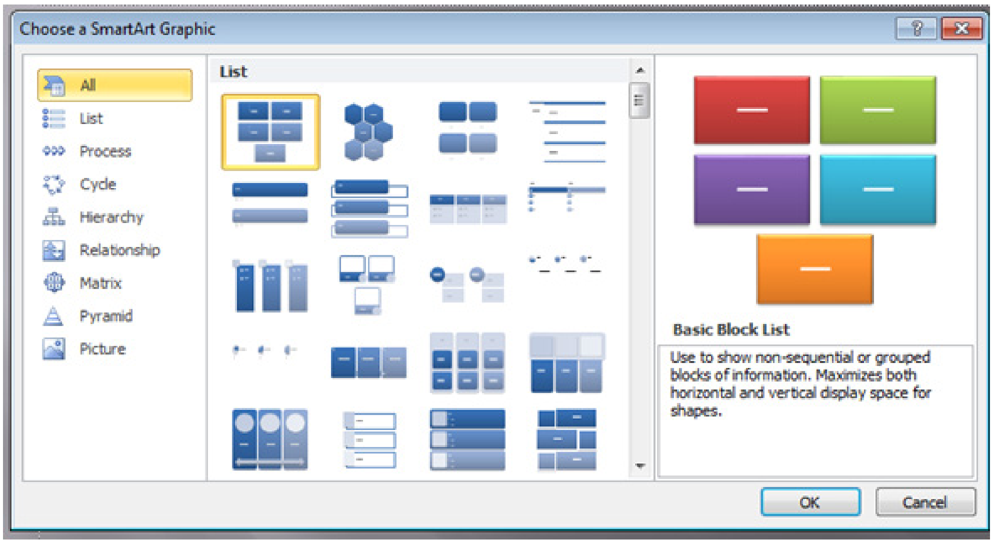
Insert SmartArt Graphic

Insert a SmartArt graphic to visually communicate information.

SmartArt graphics range from graphical lists and process diagrams to more complex graphics, such as Venn diagrams and organization charts.

Press F1 for more help.

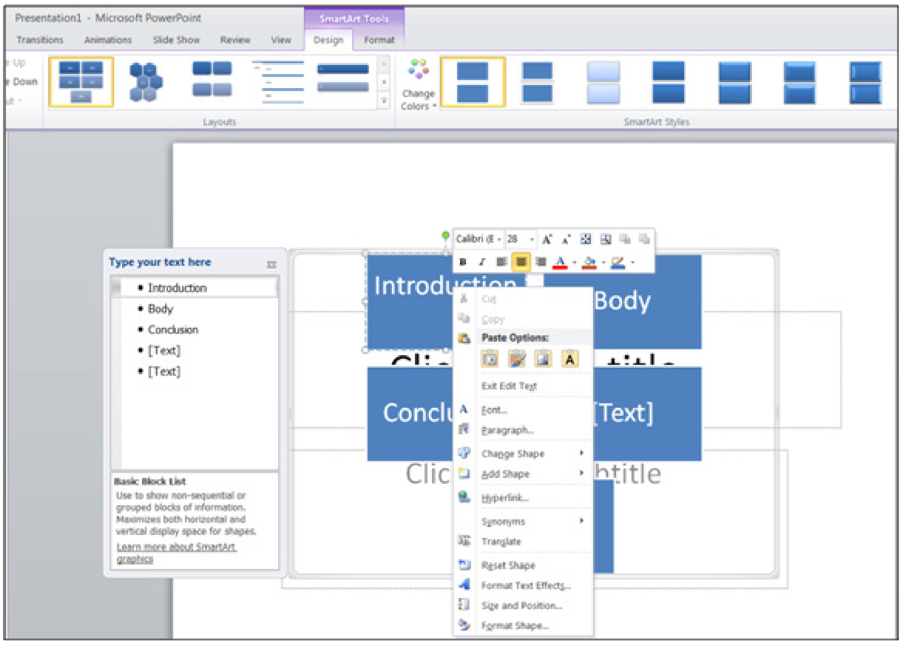
Choose a Diagram type << OK



Basic Block List

Use to show non-sequential or grouped blocks of information. Maximizes both horizontal and vertical display space for shapes.

Click on the objects within the diagram to edit them.



Type your text here

- Introduction
- Body
- Conclusion
- [Text]
- [Text]

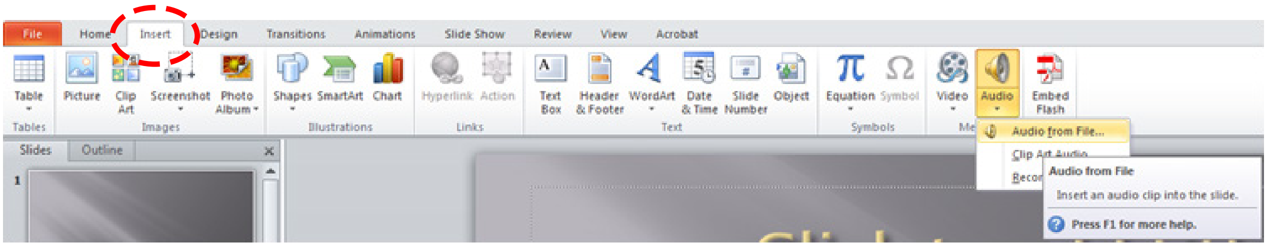
Basic Block List

Use to show non-sequential or grouped blocks of information. Maximizes both horizontal and vertical display space for shapes.

Learn more about SmartArt graphics

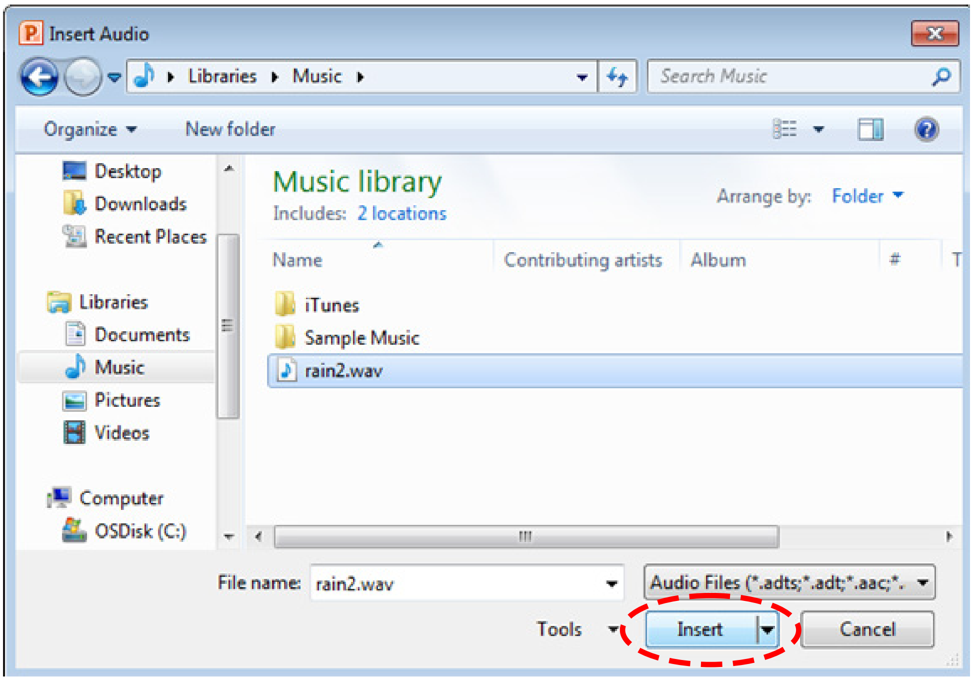
NB: Right click on the objects to change their properties.

Select the slide you are going to put the sound on
Insert << **Audio** >> **Audio from File**



Insert an audio clip into the slide.

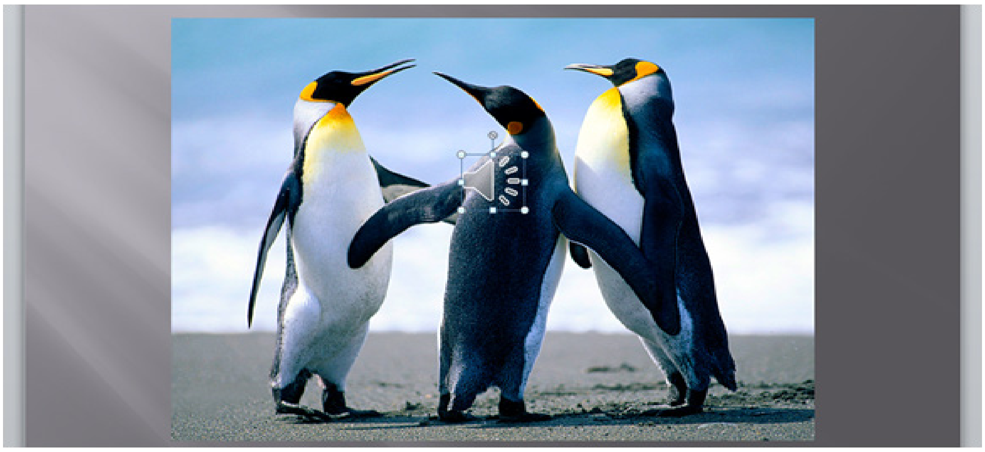
Browse for your sound file >> **Insert**



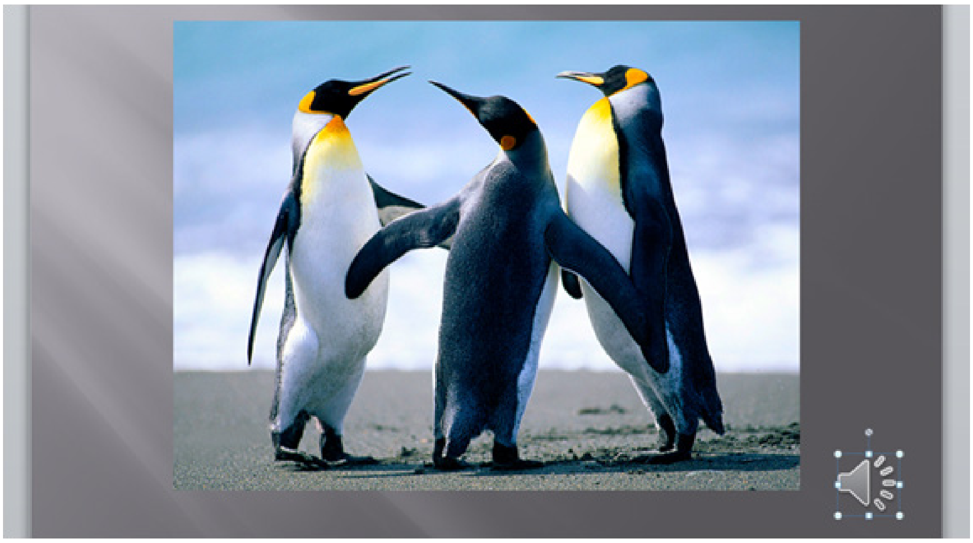
File name: rain2.wav

Audio Files (*.adts;*.adt;*.aac;*.

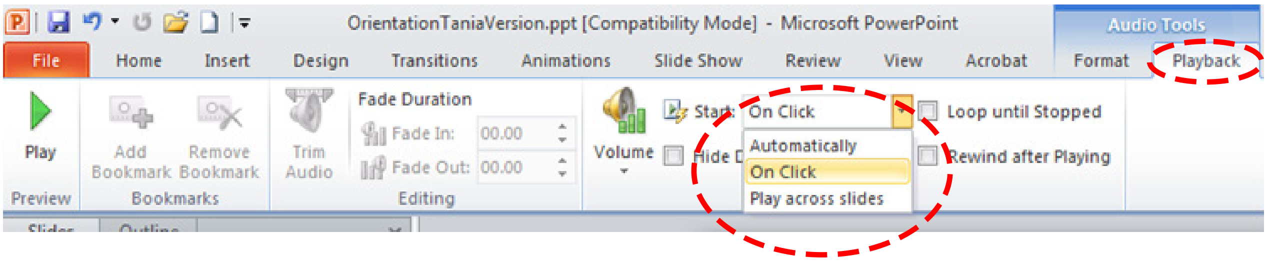
When you click on the sound icon your sound will start playing automatically.



To move your sound icon out of the middle of your powerpoint>> Click on the icon >> move it to where you want it placed.



To change the playback of your sound:
Click on the **Playback** tab << **Start** dropdown box << select **Automatically, On Click or Play across slides**



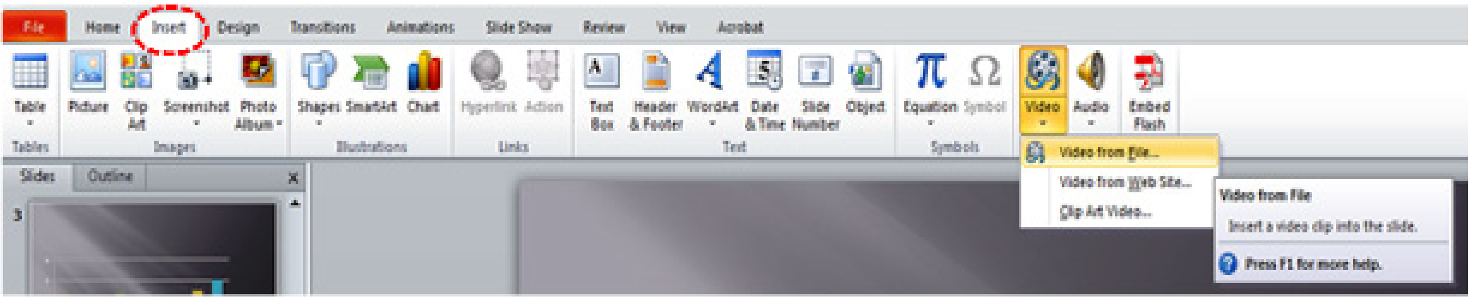
OrientationTaniaVersion.ppt [Compatibility Mode] - Microsoft PowerPoint

Audio Tools

If you chose **Automatically**, your sound will play straight away.
If you chose **On Click**, you will need to click on the sound icon.
If you chose **Play across slides**, when you click on your sound icon your sound will continue to play across all slides.

Select the slide you are going to insert a video into.

Insert >> **Video** >> **Video from File**

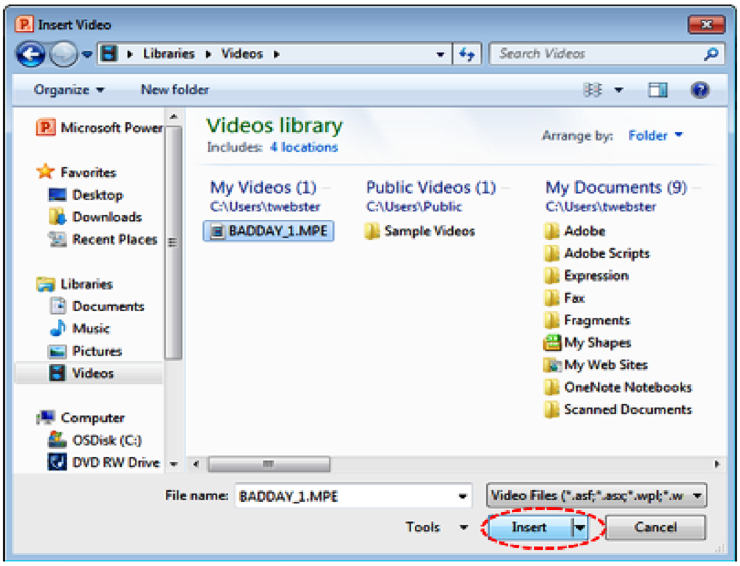


Video from File

Insert a video clip into the slide.

Press F1 for more help.

Browse for your video >> **Insert**


Insert Video

Libraries ▸ Videos ▸

Search Videos

Organize ▾ New folder

Microsoft Power

Favorites
Desktop
Downloads
Recent Places

Libraries
Documents
Music
Pictures
Videos

Computer
OSDisk (C:)
DVD RW Drive

Videos library
Includes: 4 locations

Arrange by: Folder ▾

My Videos (1)
C:\Users\twebster
BADDAY_1.MPE

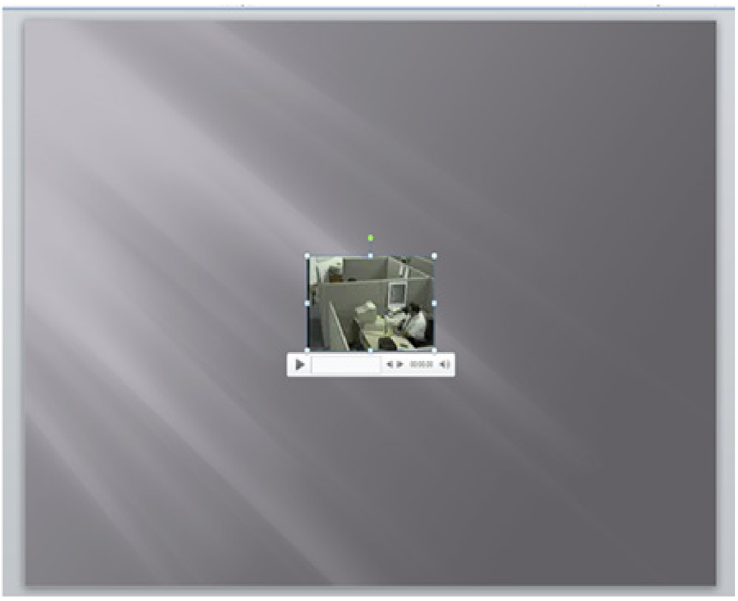
Public Videos (1)
C:\Users\Public
Sample Videos

My Documents (9)
C:\Users\twebster
Adobe
Adobe Scripts
Expression
Fax
Fragments
My Shapes
My Web Sites
OneNote Notebooks
Scanned Documents

File name: BADDAY_1.MPE

Video Files (*.asf;*.asx;*.wpl;*.w

Tools ▾ Insert Cancel



To change the playback of your video:

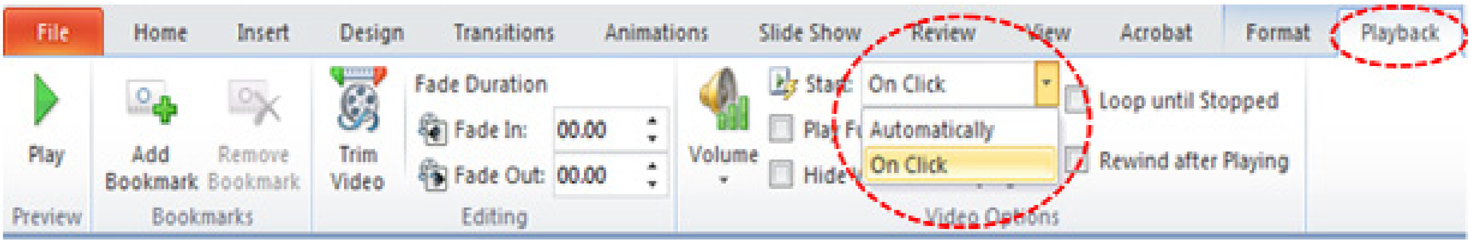
Click on the **Playback** tab

Click on the **Start** dropdown box

Choose between **Automatically** or **On Click**

  Automatically will play your movie clip straight away

  On Click means you have to click on your movie clip to get it to play

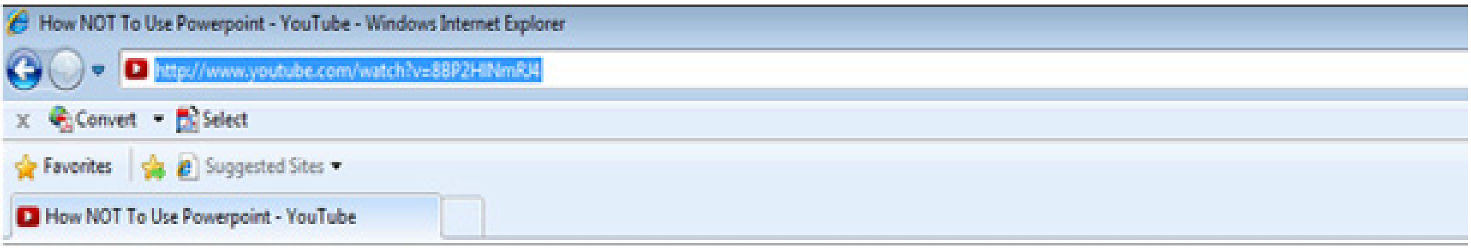


Inserting a Video from YouTube:

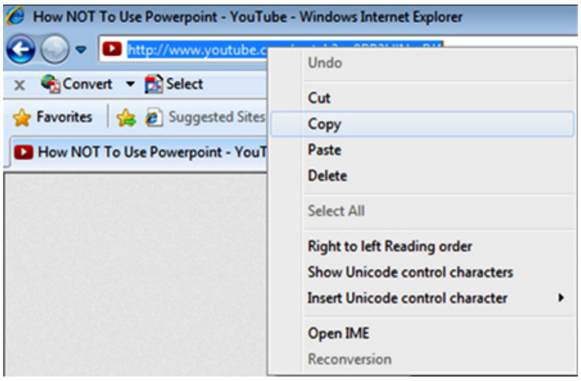
The safest way to use a video from YouTube is to hyperlink it to your slide; **do not** embed the file as this can breach the Copyright Act 1994.

Find the video you want to link to

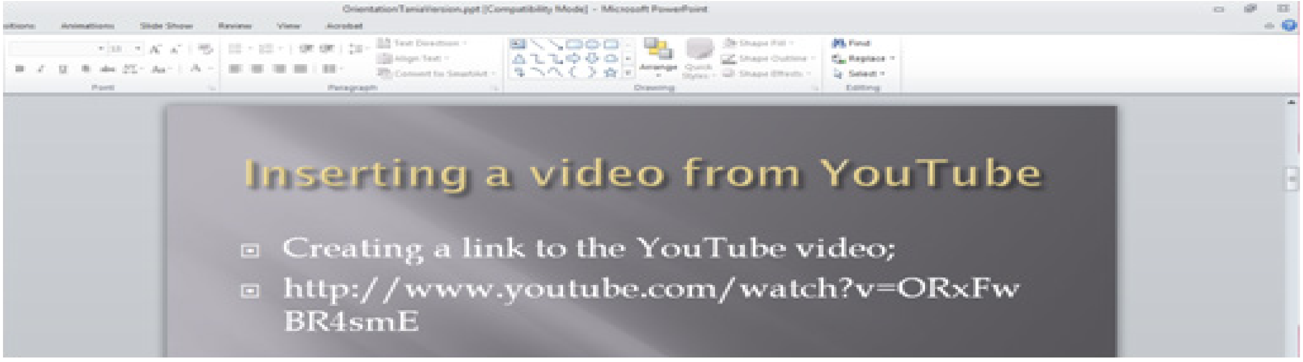
Click in the address bar, to copy your video link


How NOT To Use Powerpoint - YouTube - Windows Internet Explorer
http://www.youtube.com/watch?v=8BP2HlNmRJ4
Convert
Select
Favorites
Suggested Sites
How NOT To Use Powerpoint - YouTube

Press Ctrl C (Copy) **OR** Right click and choose Copy


How NOT To Use Powerpoint - YouTube - Windows Internet Explorer
http://www.youtube.c
Convert
Select
Favorites
Suggested Sites
How NOT To Use Powerpoint - YouT
Undo
Cut
Copy
Paste
Delete
Select All
Right to left Reading order
Show Unicode control characters
Insert Unicode control character
Open IME
Reconversion

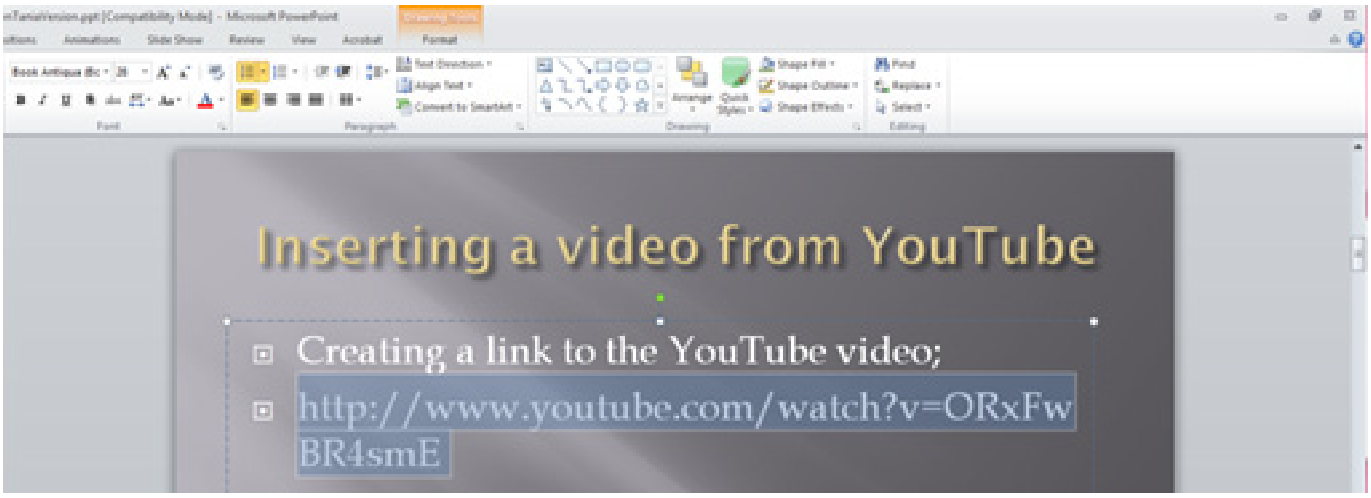
Go back to your PowerPoint slide
Click where you want to place your video link
Press Ctrl V (Paste) **OR** Right click and choose Paste



Inserting a video from YouTube

- Creating a link to the YouTube video;
- http://www.youtube.com/watch?v=ORxFwBR4smE

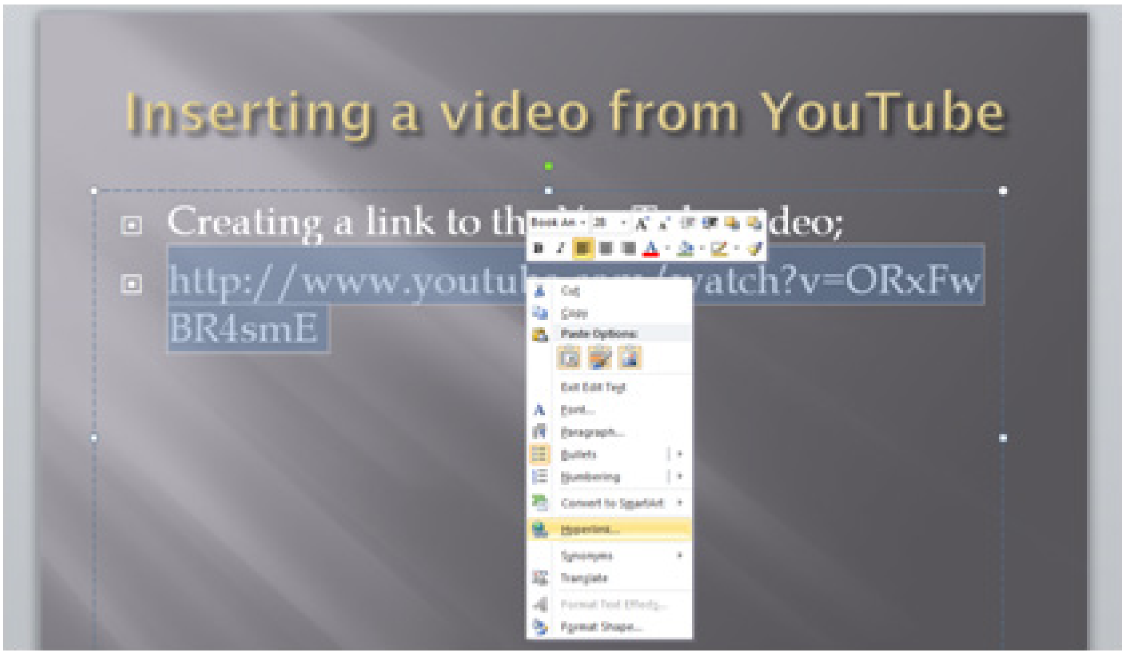
Highlight the link - http://www.youtube.com/watch?v=ORxFwBR4smE



Inserting a video from YouTube

- Creating a link to the YouTube video;
- http://www.youtube.com/watch?v=ORxFwBR4smE

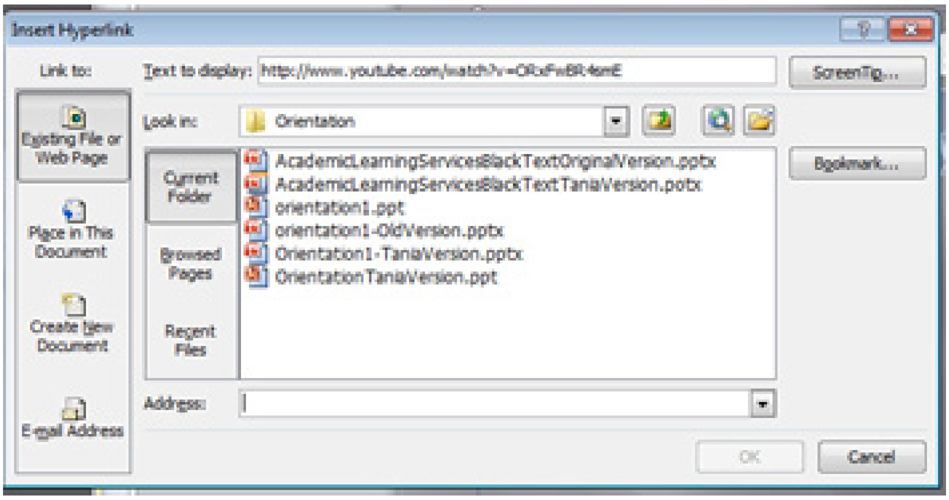
Right click on the link and choose hyperlink



Inserting a video from YouTube

- Creating a link to the YouTube video;
- http://www.youtube.com/watch?v=ORxFwBR4smE

The following dialogue box will appear;



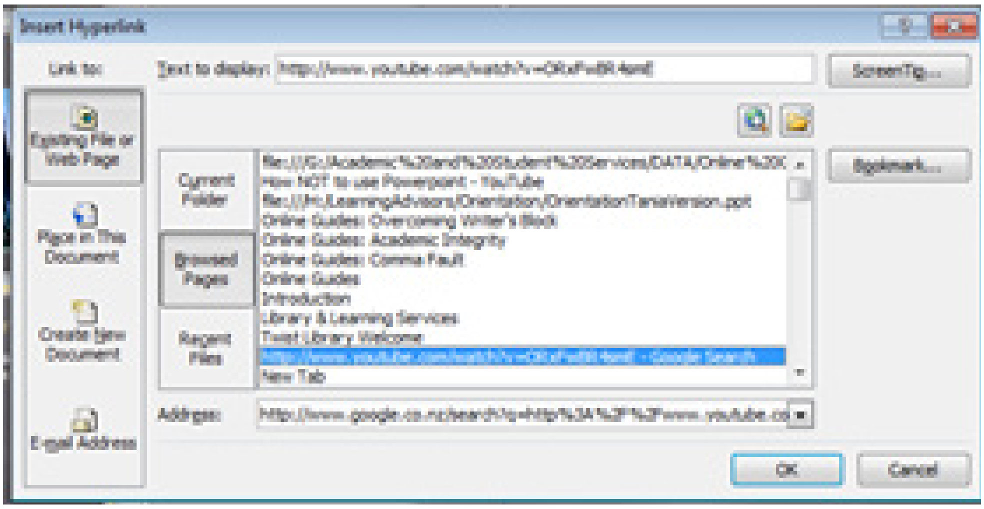
Text to display: http://www.youtube.com/watch?v=ORxFwBR4smE

Look in: Orientation

AcademicLearningServicesBlackTextOriginalVersion.pptx
AcademicLearningServicesBlackTextTaniaVersion.pptx
orientation1.ppt
orientation1-OldVersion.pptx
Orientation1-TaniaVersion.pptx
OrientationTaniaVersion.ppt

Address:

Choose **Browsed Pages** << select your **YouTube video link**

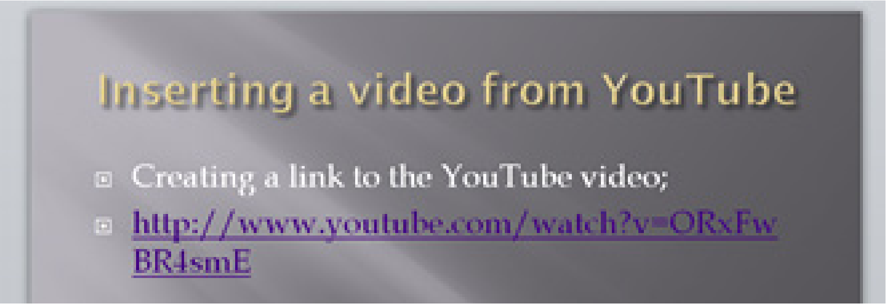


Text to display: http://www.youtube.com/watch?v=ORxFwBR4smE

file:///G:/Academic%20and%20Student%20Services/DATA/Online%20(
How NOT to use Powerpoint - YouTube
file:///H:/LearningAdvisors/Orientation/OrientationTaniaVersion.ppt
Online Guides: Overcoming Writer's Block
Online Guides: Academic Integrity
Online Guides: Comma Fault
Online Guides
Introduction
Library & Learning Services
Twist Library Welcome
http://www.youtube.com/watch?v=ORxFwBR4smE - Google Search
New Tab

Address: http://www.google.co.nz/search?q=http%3A%2F%2Fwww.youtube.co

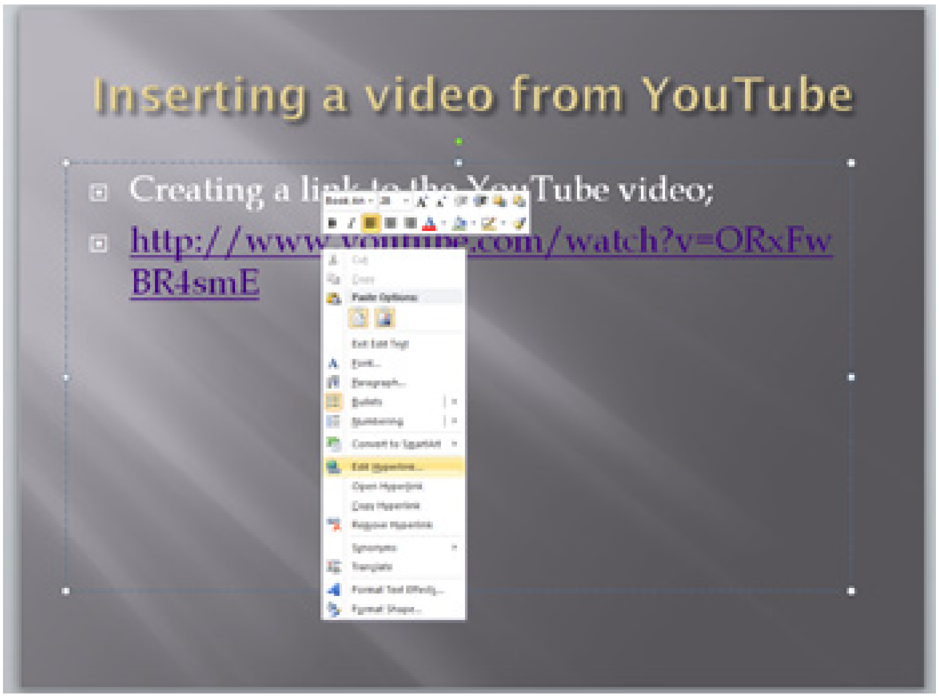
The hyperlink will change colour



Inserting a video from YouTube

- Creating a link to the YouTube video;
- http://www.youtube.com/watch?v=ORxFwBR4smE

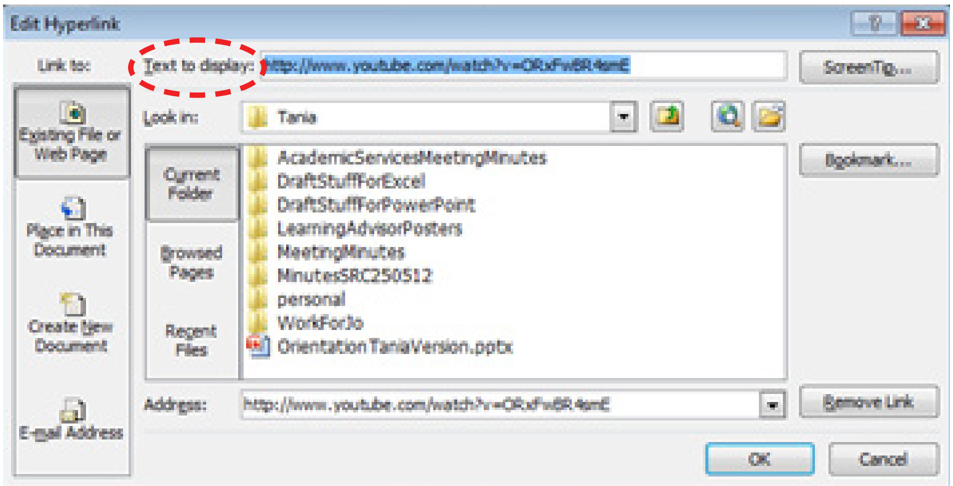
You can also edit the text of the hyperlink to make it more relevant
Right click on your hyperlink



Inserting a video from YouTube

- Creating a link to the YouTube video;
- http://www.youtube.com/watch?v=ORxFwBR4smE

Choose to **Edit Hyperlink**
The following dialogue box appears
Select **Text to display**



Edit Hyperlink

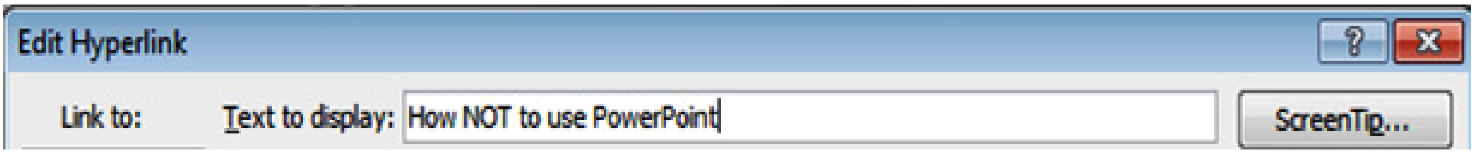
Text to display: http://www.youtube.com/watch?v=ORxFwBR4smE

Look in: Tania

- AcademicServicesMeetingMinutes
- DraftStuffForExcel
- DraftStuffForPowerPoint
- LearningAdvisorPosters
- MeetingMinutes
- MinutesSRC250512
- personal
- WorkForJo
- OrientationTaniaVersion.pptx

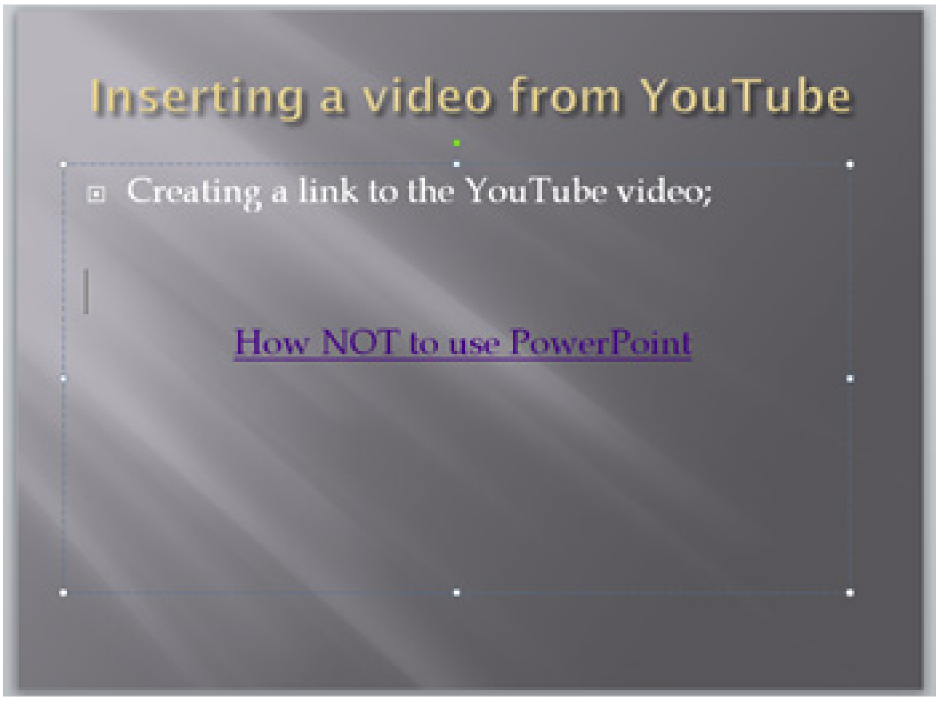
Address: http://www.youtube.com/watch?v=ORxFwBR4smE

Type in your text e.g. How NOT to use PowerPoint



Edit Hyperlink

Text to display: How NOT to use PowerPoint



Inserting a video from YouTube

- Creating a link to the YouTube video;

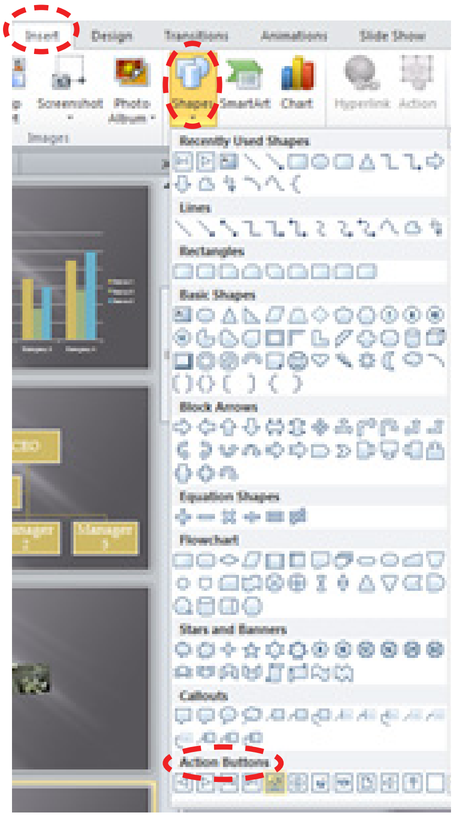
How NOT to use PowerPoint

Adding Action Buttons to your Presentation

By adding action buttons to your presentation you can tell a movie or sound to play or you could use an action button to navigate to another slide in your presentation.

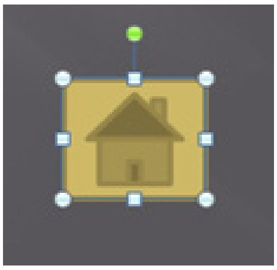
Action Buttons

Insert >> Shapes >> Action Buttons

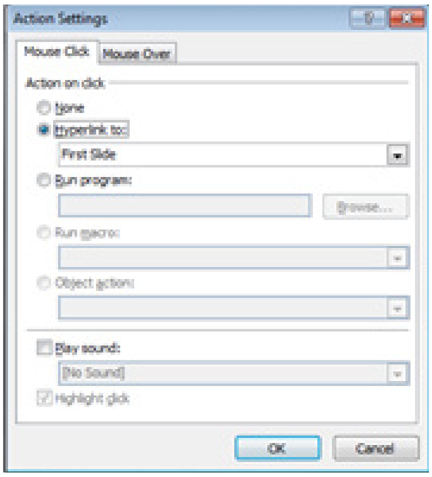


Choosing a button

Once you have chosen your button you will notice the mouse pointer has changed to +
Move your mouse onto your slide and click and drag to make your button



The action button dialogue box will now show, this is to enable you to choose the settings you want for your button



Action Settings

Mouse Click | Mouse Over

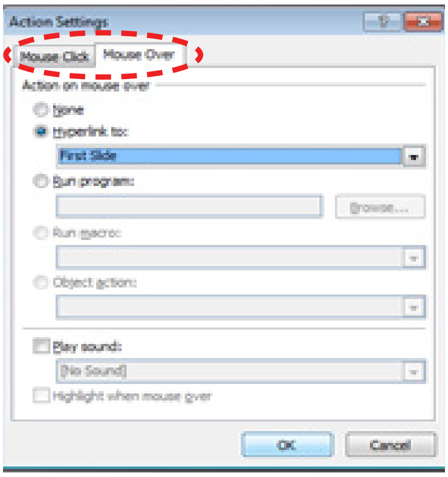
Action on click

○ None
● Hyperlink to:
First Slide
○ Run program:
Browse...
○ Run macro:
○ Object action:

□ Play sound:
[No Sound]
☑ Highlight click

OK | Cancel

Choose which tab you want to use; **Mouse Click** or **Mouse Over**



Action Settings

Mouse Click | Mouse Over

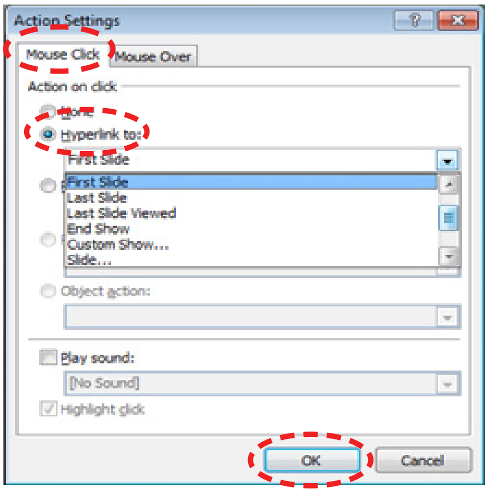
Action on mouse over

○ None
● Hyperlink to:
First Slide
○ Run program:
Browse...
○ Run macro:
○ Object action:

□ Play sound:
[No Sound]
□ Highlight when mouse over

OK | Cancel

Choose which action you want and click **OK**



Action Settings

Mouse Click | Mouse Over

Action on click

○ None
● Hyperlink to:
First Slide
First Slide
Last Slide
Last Slide Viewed
End Show
Custom Show...
Slide...
○ Object action:

□ Play sound:
[No Sound]
☑ Highlight click

OK | Cancel