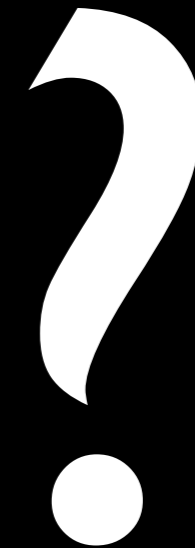
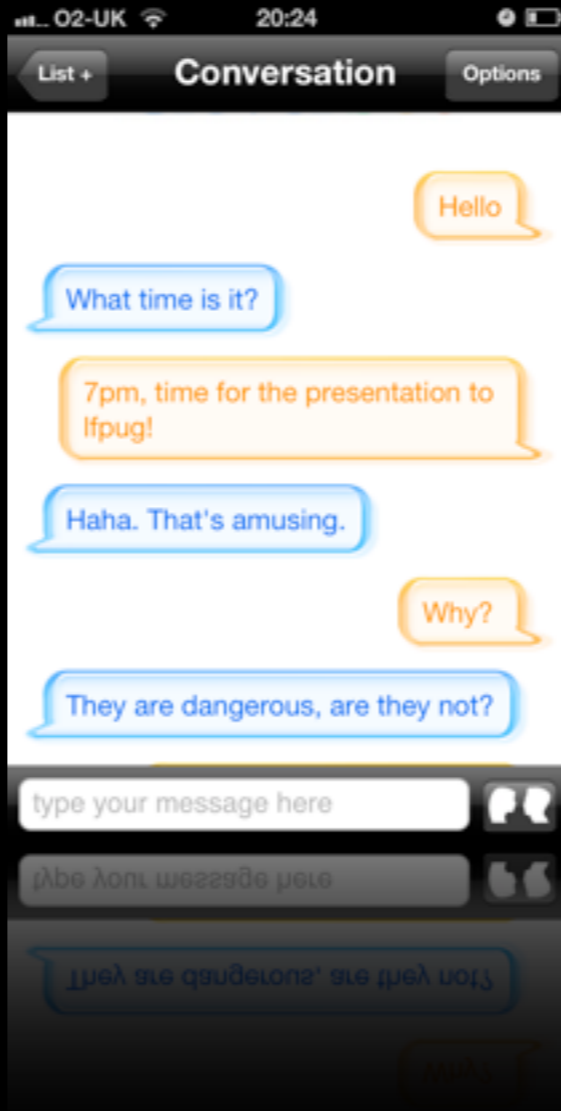
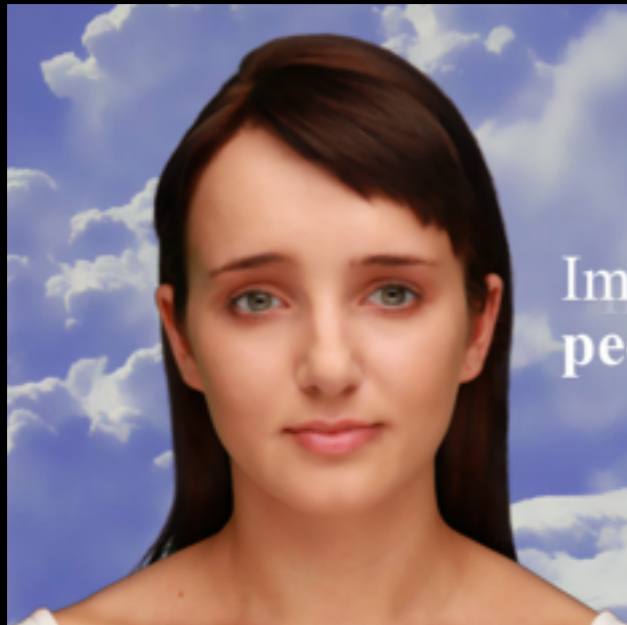


Liam Flynn

Animating HTML5



Loading



Standing on the shoulders of a giant.

Standards.



Options.

<Canvas>

Low level pixel setting and image pasting.

<SVG>

High level vector graphics.

CSS3

WebGL (Canvas3D)



Canvas2D

- Step 1 - Declare the instance.

```
<canvas id="theCanvas" width="320"  
height="480">
```

```
This message will show if your browser  
can not support canvas based fun.
```

```
</canvas>
```

- Step 2 - Get a reference to it's context.

```
var elem = document.getElementById("theCanvas");
```

```
var context = elem.getContext('2d');
```

- Step 3 - Call methods on context.

```
context.fillStyle = 'rgb(200,0,0)';  
context.fillRect(0, 0, 50, 50);
```

```
context.translate(50,50);
```

```
context.fillStyle = 'rgba(0,0,200, 0.5)';  
context.fillRect(20, 20, 60, 60);
```

- Step 4 - Save and reload context.



```
context.save();  
//do some stuff  
context.restore();
```



- Step 5 - Setup a frame loop.

```
//every 50 milliseconds  
setInterval(draw,50);
```

```
//on mouse move  
window.addEventListener('mousemove',draw, false);
```

A few quick examples of Canvas2D

- Step 42 - Construct a display list.

```
//pseudo code
class SceneObject{

var listOfChildObjects = new Array<SceneObject>();

function addChild(child:SceneObject){ listOfChildObjects.push(child);}

//called every frame
function draw(){

    context.save();

    //...draw the stuff for this object here.
    context.blahBlahBlah()

    for(var i:int = 0; i < listOfChildObjects.length(); i++){
        listOfChildObjects[i].draw();
    }

    context.reload();
}
```

EaselJS by Grant Skinner

Up and running.

SVG

- Step 1 - Declare the instance.

```
<svg id="theSvg" xmlns="http://www.w3.org/2000/svg"  
width="320" height="480">
```

```
</svg>
```

- Step 2 - Add some objects.

```
<svg id="theSvg" xmlns="http://www.w3.org/2000/svg"  
width="320" height="480">
```

```
<rect x="1" y="1" width="30" height="40" fill="blue"  
id="box" />
```

```
<circle cx="0" cy="0" r="15" fill="red" id="ball" />
```

```
</svg>
```

- Step 3 - Setup a frame loop.

```
//same as canvas, really.  
setInterval(draw,50);
```

- Step 4 - Get and set attributes.

```
var rect = document.getElementById('ball');  
  
rect.setAttribute("x", "200");  
rect.setAttribute("y", "100");  
  
...  
//get attribute value.  
var w = rect.getAttribute("width");
```

- Step 5 - Runtime objects.

```
var rect = document.createElementNS(  
    'http://www.w3.org/2000/svg', 'rect');  
  
rect.setAttribute("id", "runtimeRect");  
rect.setAttribute("x", "100"); etc...  
  
var theSVG = document.getElementById('theSVG');  
theSVG.appendChild(rect);
```

A few simple *SVG* examples.

Up and running in SVG

CSS3

Animations

- Step 1 - Add some scene objects.

```
<div class="box1">  
<div class="box2">
```

- Step 2 - Describe the objects.

```
.box1{  
  width:200px;  
  height:200px;  
  background:rgba(255,0,0,0.5);  
  position:absolute;  
  top:100px;  
  left:50%;  
  margin-left:-100px;  
}
```

- Step 2 - Describe the animation.

```
@-webkit-keyframes movingbox{  
  0%{left:90%;}  
  50%{left:10%;}  
  100%{left:90%;}  
}
```

- Step 3 - Apply an animation to an object.

```
.box2 {  
    -webkit-animation: movingbox 5s infinite;  
}
```

CSS3 in action.

Eg

www.liamflynn.co.uk/blog

l.flynn@live.co.uk