

<!doctype html⁵.in>

HTML5 Media++

Audio, Video, SVG and Canvas


Who Am I?

- Vishal Parpia (a.k.a.: viz/vizkid)
 - Part designer part developer
 - Building web sites since 1995 (far too long)
 - ActivElement: Founder + Director
 - MetroGuild: Partner

The next 45 minutes

- Why you should care about this stuff
- Quick overview of HTML5's media, canvas and SVG tags
- Practical ways to use the tags today
- Fallbacks and browser compatibility
- The bleeding edge
- Q+A

Why should you care?

- Flash is dead, long live HTML5
- The  Factor
- Growing browser support, you can use many things right away
- Undoubtedly the way forward, but not quite fully cooked yet

Audio

- `<audio>`
- Does exactly what you would expect
- Great JavaScript support
- Plays in-line on iPhones, iPods and iPads
- MP3 vs. Ogg Vorbis
- Fallback to Flash

Video

- `<video>`
- Déjà vu
- H.264 vs. Ogg Theora vs. VP8/WebM
- H.264 is patent-encumbered but not for your users
 - As long as it is free, i.e.: if you charge users to see the video, restrictions apply
- Ogg Theora and VP8 are probably not free either

`<!doctype html5.in>`

Using video

- `<video src=video.mp4>`
- Video with the appropriate fallback support:
 - `<video width="640" height="360" controls>`
 - `<source src="__VIDEO__.MP4" type="video/mp4" />`
 - `<source src="__VIDEO__.OGV" type="video/ogg" />`
 - `<object ...>`
 - `//Flash fallback code comes here`
 - `</object>`
 - `</video>`
- Kroc Camen uses this code for maximum compatibility
 - http://camendesign.com/code/video_for_everybody

Javascript Control

- Video+Audio elements can be controlled via Javascript

```
- function toggle() {  
    if (video.paused) {  
        video.play();  
        this.innerHTML = 'Pause';  
    } else {  
        video.pause();  
        this.innerHTML = 'Play';  
    }  
}
```

```
var play = document.getElementById('play');  
play.onclick = toggle;
```

<!doctype html⁵.in>

<!doctype html⁵.in>

Demo

<http://nh7.in>

SVG

- `<svg>`
- Vector-based
- Plays nice with the DOM
 - Very important for interactivity
 - z-index
 - (Relatively) accessible
- Can manipulate individual items easily
- Raphaël - [http://raphaeljs.com/](http://dmitrybaranovskiy.github.io/raphael/)

`<!doctype html5.in>`

Canvas

- `<canvas>`
- Bitmap based
- Allows per-pixel manipulation
- Unbelievably powerful
- Not accessible
- Google's ex-canvas:
 - <http://code.google.com/p/explorercanvas/>

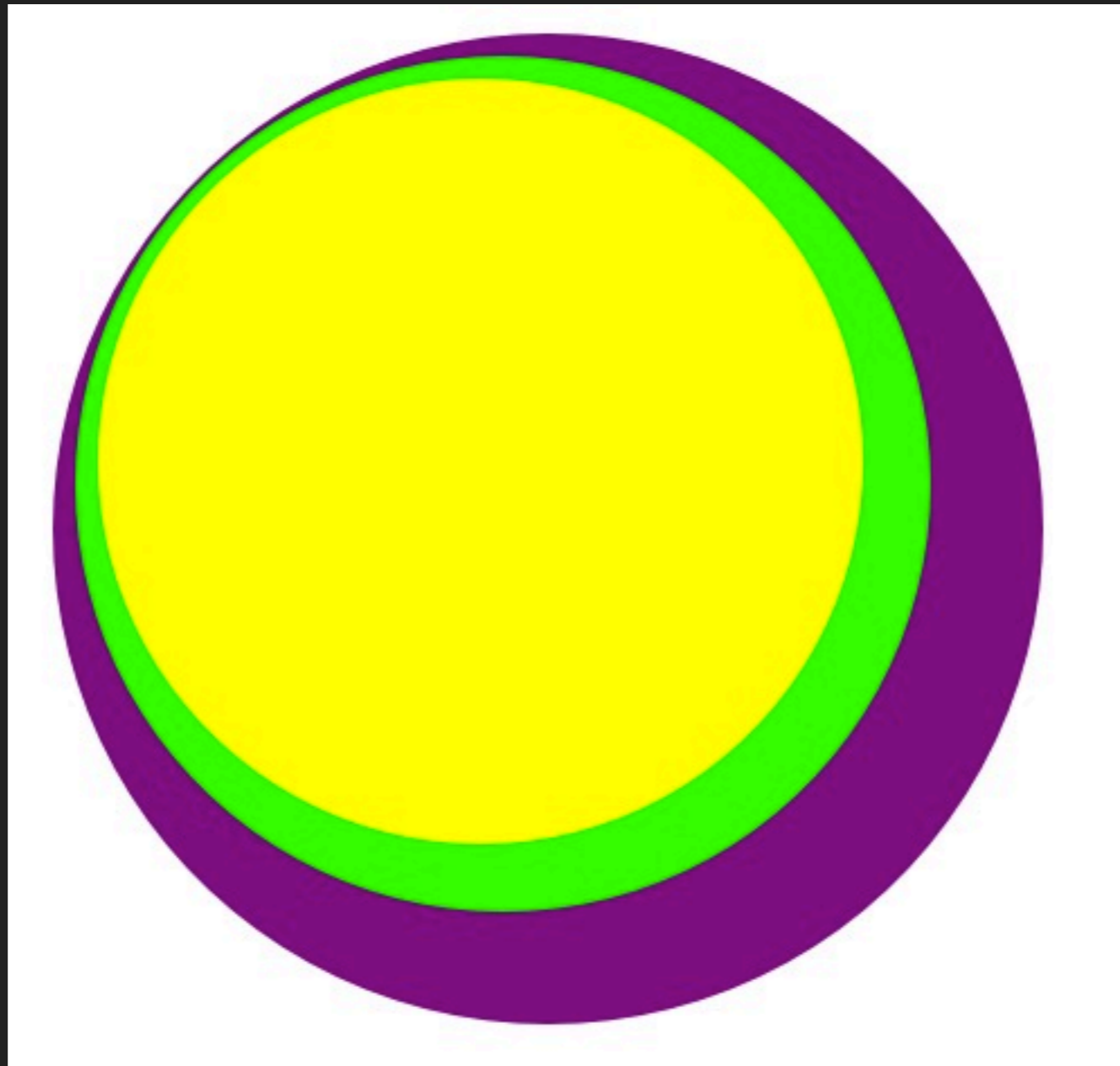
Using SVG

- Draw 3 circles one on top of the other

```
- <svg width="100%" height="100%"  
  version="1.1" xmlns="http://www.w3.org/  
  2000/svg">  
  <circle cx="240" cy="100" r="220"  
  style="fill:purple" />  
  <circle cx="220" cy="70" r="190"  
  style="fill:lime" />  
  <circle cx="210" cy="45" r="170"  
  style="fill:yellow" />  
</svg>
```

<!doctype html⁵.in>

SVG Output



<!doctype html⁵.in>

<!doctype html⁵.in>

Demo

<!doctypeHTML5.in> Logo

Using Canvas

- All the magic happens in Javascript
- Canvas uses contexts (ala DirectX and OpenGL)
 - `var ctx = canvas.getContext('2d');`
- It gives you a method to read/write pixels directly to the screen
 - `ctx.getImageData(x,y,w,h);`
 - `ctx.fillRect(x,y,w,h);`

<!doctype html⁵.in>

Demo

MetroLeads Dashboard

Advanced Canvas

- Very powerful per pixel manipulation
 - Image editors
 - Video playback
 - Sprite-based game systems
- Proves once and for all Javascript is actually really powerful

<!doctype html⁵.in>

Demo

Mugtug, Akihabara + more

Choosing between Canvas and SVG

- Relatively simple and interactive - SVG
 - Graphs should probably be in SVG
- Complicated and bitmap heavy - canvas
 - Rendering engines need to be in canvas
- There are exceptions to every rule
- SVG has you drawing shapes
 - Resolution Independence
- Canvas draws straight to screen, you're manipulating pixels on a display

<!doctype html⁵.in>

Thanks!

For links and a copy of this presentation go to:

<http://vizkid.com/html5>