

# Objective-J

Bringing Desktop Applications to  
the Web Browser

*<http://objective-j.org/>*

## Team Helpdesk Fish

Andrew Carlson

Greg Jackson

David Loughry

# First Some Background...

- Objective-J is named after Objective-C because it uses similar syntax, including Smalltalk-style message passing, but has a base language of JavaScript rather than C
- Cappuccino is an Objective-J library modeled after Apple's Cocoa framework
- Objective-J is developed and maintained by 280 North
- 280 Slides is a presentation application which demonstrates the power of Objective-J and Cappuccino

# What is it good for?

- Desktop quality applications in the web browser
- AJAX style web applications
- Rich features such as drag and drop and document saving
- Cross-browser development

# What is it NOT good for?

- Static web pages
- Computationally intensive web applications
- Direct HTML/DOM/CSS manipulation - use plain JavaScript

# What do I need to get started?

## 1) A text editor

- SubEthaEdit, Coda, TextMate, VIM, and Xcode are supported but are not required.

## 2) A web browser

- Objective-J code is translated within the browser through JavaScript.
- No web server is required for basic programs.
- It is useful to have a JavaScript debuggers such as Firebug, but is not required.

# Classes

```
@implementation ImageLayer : CALayer
{
    PageView      _pageView;
    CPArray       _images;
    int           _currentImageID;
    ...
}
@end
```

# Methods

```
- (void) addImage: (CPImage) anImage
{
    _images = [_images arrayByAddingObject: anImage];
}
```

# Object and Method Use

```
var _imageLayer =  
    [[ImageLayer alloc] initWithPageView:self];  
  
[_imageLayer  
 addImage:  
     [[CPIImage alloc]  
      initWithContentsOfFile:@"image1.jpg"  
      size:CGSizeMake(333, 500)]];
```

Demo

# Issues

- Documentation is often lacking
- Debugging can be difficult
- The line between JavaScript and Objective-J can be blurry
- The library isn't comprehensive
- Few extensions available