

*“If you’re writing jQuery and not structuring it as a plugin, **you’re doing it wrong**”*

- some random dude on Twitter

Doing it Right: Building jQuery Plugins

Ian Robinson

Ian Robinson

Top 10%  for [jquery](#) [asp.net](#) [cms](#) [dotnetnuke](#)

Top 20%  for [javascript](#) [css](#)

- **Session Feedback:** <http://bit.ly/jqstldodn>
- Twitter [@i-robinson](#)
- GitHub <http://github.com/i-robinson>
- StackOverflow Careers <http://bit.ly/i-robinson>
- jQuery Crash Course
 - <http://jquery.ian-robinson.com>

Platinum



Microsoft®

SAINT LOUIS
DOT NET
USER GROUP



DOTNETNUKE™

TEK
systems.

Gold



POWER DNN
DOTNETNUKE® DONE RIGHT!



talentporte

TALX



Silver



The Value of JavaScript Libraries

- **Web Statelessness**
 - responsiveness doesn't come natural
- **JavaScript**
 - misunderstood and abused raw power
- **Browsers**
 - each do things a bit differently

[jQuery]

jQuery

- Is a **JavaScript abstraction**
- Is a **collection of solved problems**(shortcuts, utilities)
- Has an sweet **plugin model**
- Is totally **ubiquitous**
- Has a **vibrant community**
- Originally made JavaScript **fun again**
- Now just makes JavaScript **completely awesome**

So it's basically a **clean foundation**
from which to build rich user
interaction

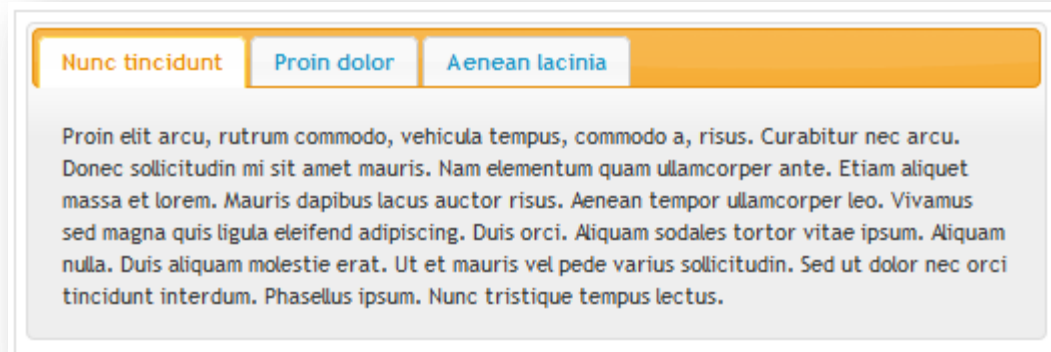
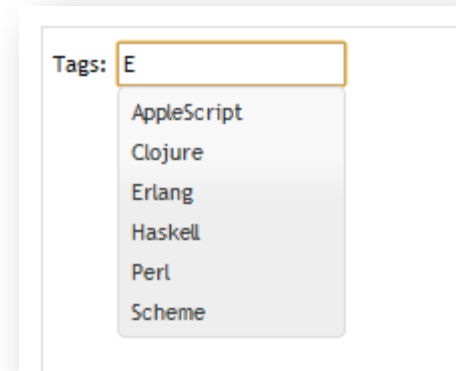
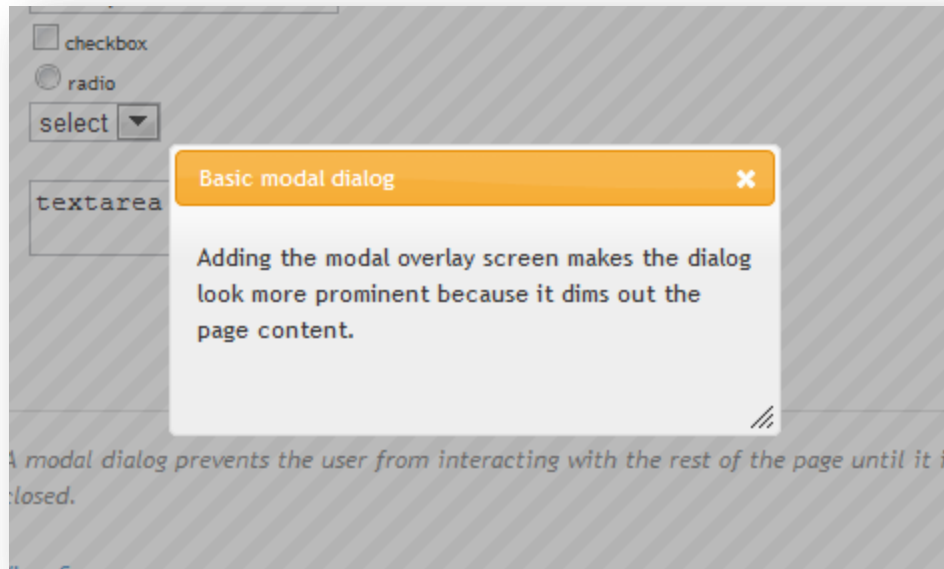
[Plugins]

And a plugin is...?

Taking the helpfulness a bit further

- Often a **reusable** component
- Structured for **maintainability** and **compatibility**

Canonical Example: jQuery UI




Popular in the Community



- Interaction
 - Cycle – slideshows...
 - Hover Intent – sophisticated hovering
 - Table Sorter – spiff up tables
- Utility
 - Back Button & Query Library (BBQ)
 - Cookie

Admin Host Modules Pages Tools Mode **Edit**

Getting Started Home About Us Our Services News & Promotions

 **DOTNetNUKE**

LOGOUT | SUPERUSER ACCOUNT

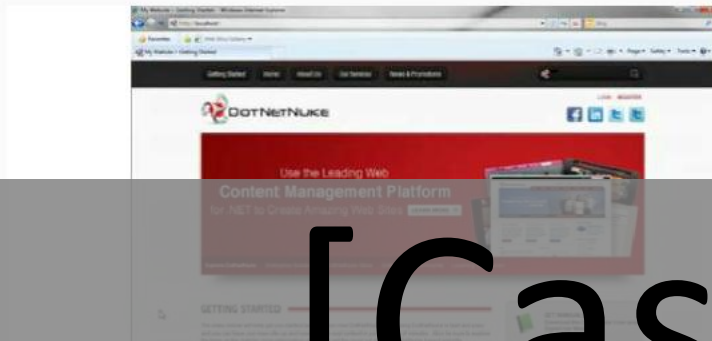
Manage    

Use the Leading Web
Content Management Platform
for .NET to Create Amazing Web Sites [LEARN MORE](#)

Explore DotNetNuke Enterprise Solutions DotNetNuke Store DotNetNuke Community Learning Resources

GETTING STARTED

The video below will help get you started building your new DotNetNuke site. Using DotNetNuke is fast and easy and you can have your new site up and running with real content in just a matter of minutes. Also be sure to explore the links on the right for more information on how to get the most out of your DotNetNuke based website.



GET MANUALS

Download the DotNetNuke User and SuperUser Manuals



GET TRAINING

View training videos or register for upcoming sessions



FIND APPS

Select from nearly 10,000 commercial and open source apps



ENGAGE WITH COMMUNITY

Engage with other members of the DotNetNuke community



ATTEND SERVICES

Look for DotNetNuke conferences and events in your area

[Case Study]

Goals

- Updated, modern, user interface
- Consistent user experience
- Ease of implementation
- Reuse by third party developers

[Demo]

Demo Review

- Collection of plugins (UI widgets)
- Use smart defaults
- Based on established HTML Conventions
- Public API used for configuration as needed
- Generally very simple to consume

[Creation]

Creating a Plugin

- Wrap business logic with plugin logic
- Logic can stay largely the same
- Modify to **inject settings and context**
- Modify for safety and to play well with others

First things first

- Always wrap in a closure

```
(function($){  
    // ahhh...safety  
})(jQuery);
```

Define Your Plugin

- No selecting needed

```
$.myPlugin = function(options){...};
```

- Select and return (chain)

```
$.fn.myPlugin = function(options){  
    return this.each(function(){  
        ...  
    });  
};
```

Establish Default Options

```
$.fn.myPlugin.defaultOptions = {  
    speed: 'slow'  
};
```

```
var opts = $.extend({},  
$.fn.myPlugin.defaultOptions, options),
```

Inject HTML Context

- Options aren't just for “settings”
- Inject all context through options

```
$( '.nav' ).myPlugin({  
    speed: 'fast'  
    subNavSelector: 'li.subnav',  
});
```

Events

1. Sure you need to enforce doc ready?
2. If you're ever unbinding events, bind them with a namespace first.

Favor this:

```
$('#elem').unbind('click.Namespace',  
function(){...});
```


Instead of this:

```
$('#elem').unbind('click', function(){...});
```

[Code Demo]

Ian Robinson

Top 10%  for [jquery](#) [asp.net](#) [cms](#) [dotnetnuke](#)

Top 20%  for [javascript](#) [css](#)

- **Session Feedback:** <http://bit.ly/jqstldodn>
- Twitter [@i-robinson](#)
- GitHub <http://github.com/i-robinson>
- StackOverflow Careers <http://bit.ly/i-robinson>
- jQuery Crash Course
 - <http://jquery.ian-robinson.com>

References

- <http://nathansjslessons.appspot.com>
- <http://api.jquery.com>
- <http://benalman.com>
- <http://msdn.microsoft.com/en-us/scriptjunkie>
- <http://marcgrabanski.com/articles/jquery-plugins-toolbox>
- <http://www.dotnetnuke.com/Resources/Wiki/Page/Reusable-DotNetNuke-jQuery-Plugins.aspx>