

REUSABLE CODE

For Good or For Awesome!





...with [sing.] the image
...to south through the Pa
...the west side is one day earlier
...ast side. ➔ picture at earth

Internet /'ɪntənɪt/ (also *informa*
... [sing.] (computing) the international
...nputers that makes it possible for you to
...tion from all around the world on your
...to send information to other comput
...t it on the Internet. ➔ Compare Int
... **Wide Web**.

Interpol /'ɪntəpɒl/ noun [sing., with
...international organization th



BBC







Glow is a JavaScript library which gives you...

- Simplified DOM manipulation, event handling, animation, etc.
- A versatile set of user interface widgets
- Clear and comprehensive documentation
- W3C, Microsoft, Internet Explorer, Mozilla, Opera, Safari, Firefox, Chrome, and other browser compatibility

theTeam.

[Home](#)

[Services](#)

[Work](#)

[Opinion](#)

[About us](#)

[News](#)

[Events](#)

[Contact](#)





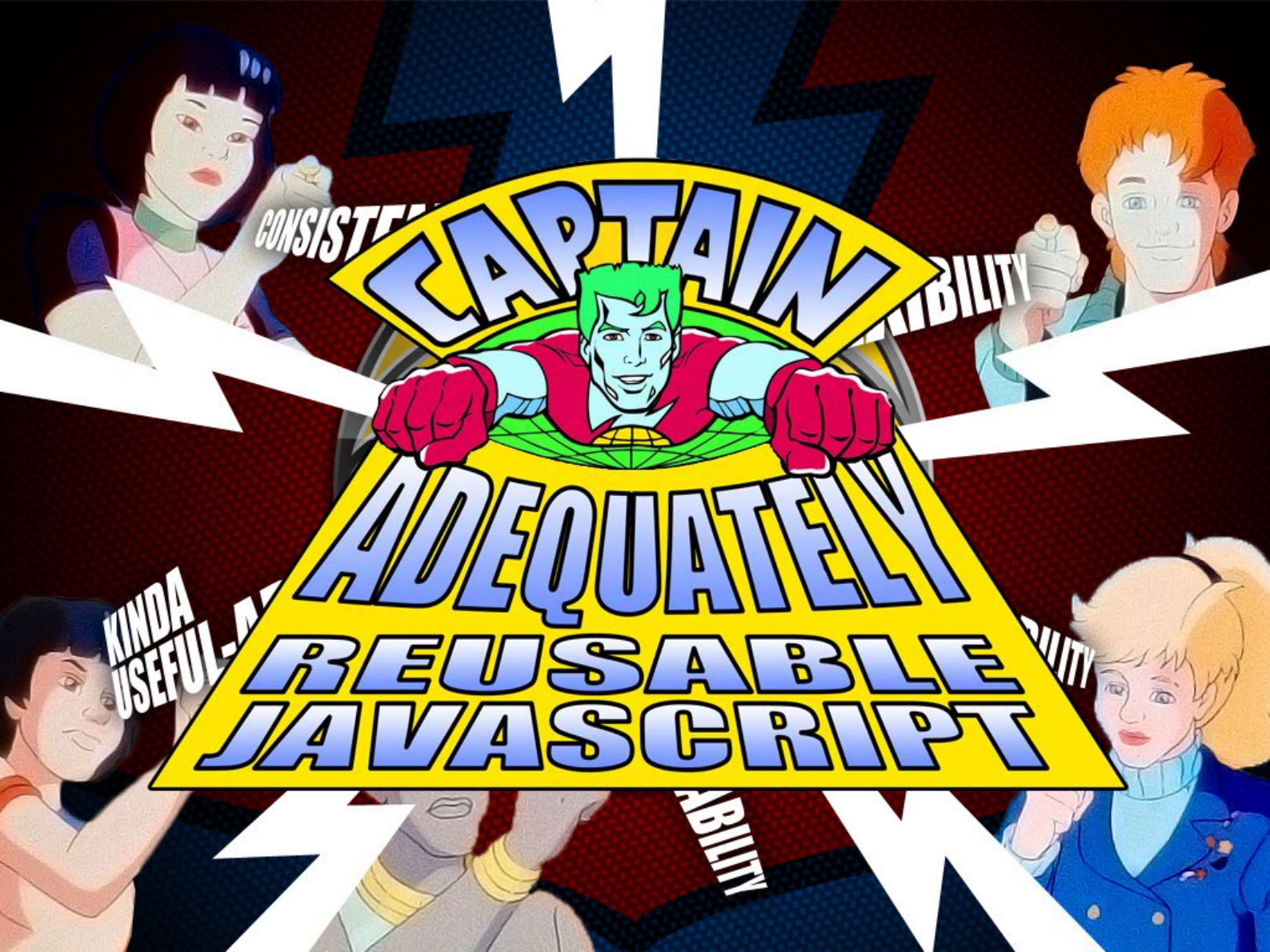
CONSISTENCY

FLEXIBILITY

**KINDA
USEFUL-ABILITY**

WRITABILITY

READABILITY



CAPTAIN

**ADEQUATELY
REUSABLE
JAVASCRIPT**

CONSISTENT

RELIABILITY

KINDA
USEFUL

QUALITY

ABILITY

STARTING

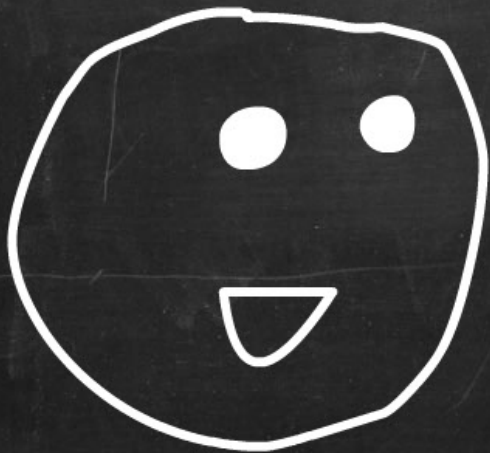
Your Journey

GETTING STARTED

- ➔ Find something worth building
- ➔ Could be an application...
- ➔ ...or a small component

RECONSTRUCTION

I made a laser!



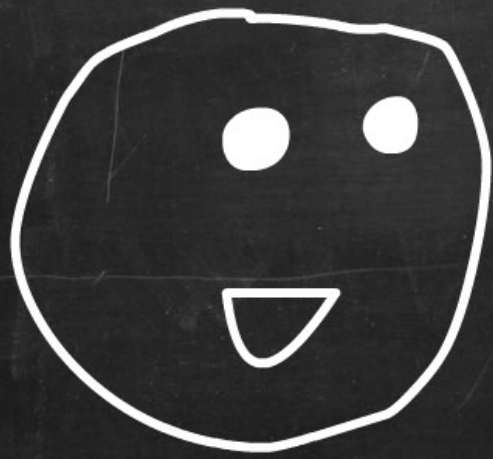
RECONSTRUCTION

You suck!



RECONSTRUCTION

Yey for lasers!!



GETTING STARTED

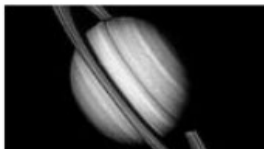
- ➔ Find something worth building
- ➔ Could be an application...

WATCH AND LISTEN TO CLIPS FROM PAST PROGRAMMES

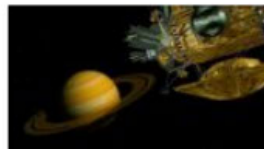
TV clips [9]



▶ **Touchdown on Titan**
The Huygens probe lands on Saturn's moon.



▶ **Voyager 1 uncovers Titan puzzle**
Saturn's moon intrigues scientists.



▶ **The Cassini-Huygens mission**
Researchers describe the Cassini-Huygens probe's design.



▶ **Titan's surface conditions**
The Cassini-Huygens probe finds ice volcanoes and methane rain on Titan.

Watch/Listen

'A Pakistan village slowly drowning'

China troops blast landslide dam

LIVE BBC News Channel

LIVE BBC Radio 5 live

is all about the Sunday surgery tonight on Radio 1

Home Shows Music Games Presenters

Visit IXtra: Games Website

43 Nick Grimshaw and Annie Mac :
08/08/2010
Rounding off your weekend...

Visit Nick

DOCTOR WHO

Home Episodes Characters Monsters News & Features Fun & Games Past Series

The Doctor Amy Pond Craig Owens Vincent Van Gogh Rory Williams

iPlayer

TV HIGHLIGHTS

Restoration Roadsho...

Women's Institute

English Heritage

Young, Dumb and Liv...

TRY THE NEW **BBC iPlayer** BETA

Radio Stations Categories A to Z

GO

TV RADIO

Have I Got News for You

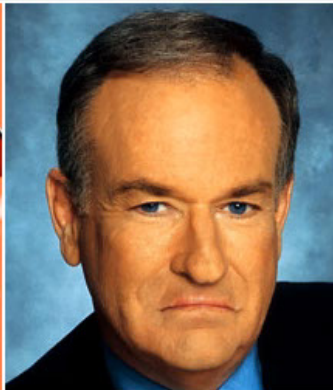
REQUIREMENTS

➔ What does the thing need to do?

Massive Pricks



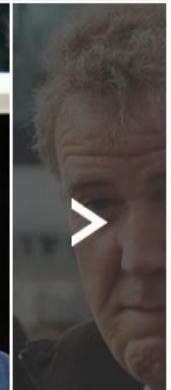
Fernando Alonso



Bill O'Reilly



Prince Charles



Jeremy

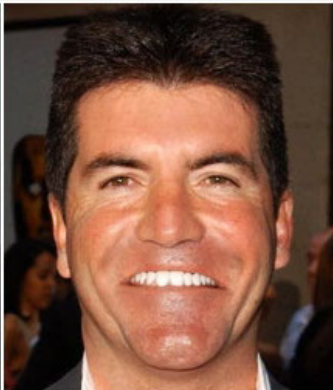
REQUIREMENTS

➔ What does the thing need to do?

Massive Pricks



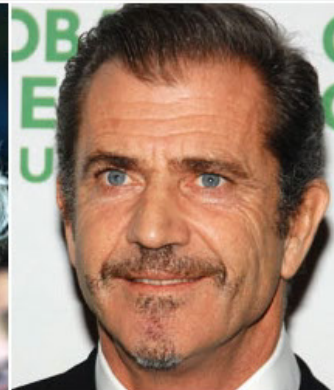
Mark Jackson



Simon Cowell



Flavio Briatore



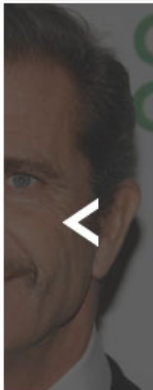
Mel Gibson



REQUIREMENTS

➔ What does the thing need to do?

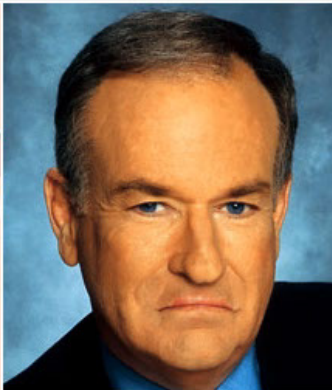
Massive Pricks



son



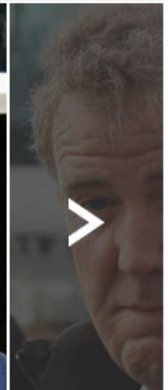
Fernando Alonso



Bill O'Reilly



Prince Charles



Jeremy

API & DOCS

Getting them right...



Historical Fallacies

*A look at common
misconceptions people hold
about historical events*

Historical ~~Fallacies~~

Phalluses

Cocks. Lots of them.

WHY SO SERIOUS?

- ➔ The better the API, the more the user will get out of your code
- ➔ A bad API may hide or distract from the best way of doing something
- ➔ APIs are difficult to change

WHY SO SERIOUS?

- ➔ The better the API, the less code
- ➔ A bad API may hide something
- ➔ APIs are difficult to



will get out of your

the best way of doing

WHY SO SERIOUS?

- ➔ The better the API, the more the user will get out of your code
- ➔ A bad API makes something
- ➔ APIs are different



way of doing

A FUNCTION

But where to put it?

~~Global scope~~

Custom 'namespace'

Existing object

Library plugin

NAMESPACES

➔ Namespaces group similar things together

```
jToys.carousel();
```

```
jToys.accordion();
```

NAMESPACES

- ➔ Namespaces group similar things together
- ➔ Namespaces can contain other namespaces

```
jToys.ui.carousel();
```

```
jToys.ui.accordion();
```

```
jToys.net.searchFlickr();
```


NAMESPACES

- ➔ Namespaces group similar things together
- ➔ Namespaces can contain other namespaces
- ➔ Be careful not to over-organise

```
jToys.ui.carousel();
```

NAMESPACES

- ➔ Namespaces
- ➔ Namespaces
- ➔ Be careful

jToys.ui.ca



NAMESPACES

- ➔ Namespaces group similar things together
- ➔ Namespaces can contain other namespaces
- ➔ Be careful not to over-organise
- ➔ Organise when necessary
- ➔ Alias for backwards compatibility

```
QUnit.same(obj1, obj2, 'Check objects are the same');
```

- 7.1. If the expected value is a Date object, the actual value is equivalent if it is also a Date object that refers to the same time.
- 7.2. If the expected value is a Date object, the actual value is equivalent if it is also a Date object that refers to the same time.
- 7.3. Other pairs that do not both pass `typeof value == "object"`, equivalence is determined by `==`.
- 7.4. For all other object pairs, including Array objects, equivalence is determined by having the same number of owned properties (as verified with `Object.prototype.hasOwnProperty.call`), the same set of keys (although not necessarily the same order), equivalent values for every corresponding key, and an identical "prototype" property. Note: this accounts for both named and indexed properties on Arrays.

```
assert.deepEqual(actual, expected, message_opt);
```

8. The non-equivalence assertion tests for any deep inequality.

NAMESPACES

- ➔ Namespaces group similar things together
- ➔ Namespaces can contain other namespaces
- ➔ Be careful not to over-organise
- ➔ Organise when necessary
- ➔ Alias for backwards compatibility

```
QUnit.same(obj1, obj2, 'Check objects are the same');  
QUnit.deepEqual(obj1, obj2, 'Check objects are the same');  
QUnit.same = QUnit.deepEqual;
```

~~Global scope~~

Custom 'namespace'



Existing object

Library plugin

ADDING TO EXISTING OBJECTS

➔ In JS all objects can be modified

```
jToys.sum( [4, 8, 15, 16, 23, 42] ); // 108
```

```
[4, 8, 15, 16, 23, 42].sum(); // 108
```

ADDING TO EXISTING OBJECTS

- ➔ In JS all objects can be modified
- ➔ Can create sexy-looking readable code

```
(6).weeks().fromNow();
```

```
Number.prototype.weeks = function() {  
    return // a Date 'this' number of weeks from zero  
};
```

```
Date.prototype.fromNow = function() {  
    return // 'this' Date plus now Date  
};
```


ADDING TO EXISTING OBJECTS

- ➔ In JS all objects can be modified
- ➔ Can create sexy-looking readable code
- ➔ What's the catch?

```
document.getElementsByClassName('comment').slice(0, 1);
```

object returned by the earlier call. In other cases, a new `NodeList` object is returned.

The `getElementsByClassName(classNames)` method on the `HTMLDocument` interface must return a live `NodeList` with the nodes that the `HTMLDocument.getElementsByTagName()` method would return when passed the same argument(s), excluding any elements that are not descendants of the `HTMLDocument` object on which the method was invoked. When the method is invoked on the same object again with the same argument, the user agent must return the same `NodeList` object as the object returned by the earlier call. In other cases, a new `NodeList` object must be returned.

`HTML`, `SVG`, and `MathML` elements define what classes they have. Elements with no namespace with the same `class` attribute are considered to be in the same namespace. Elements with different namespaces are considered to be in different namespaces.

ADDING TO EXISTING OBJECTS

- ➔ In JS all objects can be modified
- ➔ Can create sexy-looking readable code
- ➔ What's the catch?

```
var comments = document.getElementsByClassName('comment');  
comments.length; // 2  
$('<div class="comment"></div>').appendTo('body');  
comments.length; // 3
```

ADDING TO EXISTING OBJECTS

- ➔ In JS all objects can be modified
- ➔ Can create sexy-looking readable code
- ➔ What's the catch?
- ➔ Don't mess with objects you don't own!

~~Global scope~~

Custom 'namespace'



~~Existing object~~

Library plugin

WRITING A PLUGIN

➔ Similar to extending core types, but with less risk



WIKIPEDIA
The Free Encyclopedia

- View page
- Contents
- Related content
- Recent events
- Random article
- Help
- Support Wikipedia
- Community portal
- Wikipedia:About
- Wikipedia:FAQ

Article

Discussion

Read

Edit

View history

View source

Jake Weary

From Wikipedia, the free encyclopedia

Jake Weary (born February 14, 1990) is an *American actor*.

Contents [hide]
1 Biography
1.1 Movie career
1.2 Music career
1.3 Personal life
2 References
3 External links

Biography

References

WRITING A PLUGIN

➔ Similar to extending core types, but with less risk

```
$('#whatever').carousel();  
  
jQuery.fn.carousel = function() {  
    // ...  
};  
  
$ === jQuery;  
  
jQuery.fn === jQuery.prototype;
```


WRITING A PLUGIN

- ➔ Similar to extending core types, but with less risk
- ➔ You can still have collisions

~~Global scope~~

Custom 'namespace'



~~Existing object~~

Library plugin



SO, WHICH?

- ➔ Namespaces or plugin, you pick
- ➔ Consider the users!
- ➔ ...but be consistent

```
("I LIKE POTATOES").toLowerCase();  
String.fromCharCode(63);
```

```
// How it should be...  
(42.2).round();  
Number.random();
```

```
// How it is...  
Math.round(42.2);  
Math.random();
```

SO, WHICH?

- ➔ Namespaces or plugin, you pick
- ➔ Consider the users!



FUNCTIONS

& Parameters

FUNCTIONS & PARAMETERS

➔ Start with documentation

```
/**  
  @name jToys.carousel  
  @function  
  @description Create an interactive carousel  
  
  @param {selector} container Element to enhance  
    Direct children of this item will be treated as  
    carousel items.  
  
  @example  
    jToys.carousel('#whatever');  
*/
```



Index

woosh

woosh._LibraryTest

woosh._TestFrame

woosh._TestRunner

woosh._utils

woosh.AsyncTest

woosh.Conductor

woosh.DummyTest

woosh.LibraryResult

woosh.Result

woosh.ResultComparison

woosh.Test

woosh.TimeTest

woosh.views

woosh.VisualTest

Full API

Writing Plugins

Writing Tests

Methods

- ▶ [apply](#) Copies properties from one object to another
- ▶ [constructorName](#) Gets the constructor name for a function
- ▼ [extend](#) Copies the prototype of one object to another.

Synopsis

```
woosh._utils.extend(sub, base, additionalProperties)
```

Description

The 'subclass' can also access the 'base class' via subclass.base

Parameters

sub

Type Function

Required Yes

Class which inherits properties.

base

Type Function

FUNCTIONS & PARAMETERS

➔ Start with documentation

```
/**
  @name jQuery#carousel
  @function
  @description
    Turn each element in the set of matched elements into a carousel.

    Direct children of each matched element will be treated as
    carousel items.

  @example
    $('#whatever').carousel();
 */
```


`.appendTo()`

Insert every element in the set of matched elements to the end of the target.

`.attr()`

Get the value of an attribute for the first element in the set of matched elements.

[Attributes, General Attributes](#)

`.before()`

Insert content, specified by the parameter, before each element in the set of matched elements.

[DOM Insertion, Out](#)

`clone()`

create a deep copy of the set of matched elements.





WIKIPEDIA
The Free Encyclopedia

- [Main page](#)
- [Contents](#)
- [Featured content](#)
- [Current events](#)
- [Random article](#)

- ▼ [Interaction](#)
 - [About Wikipedia](#)
 - [Community portal](#)
 - [Recent changes](#)
 - [Contact Wikipedia](#)
 - [Donate to Wikipedia](#)
 - [Help](#)

- ▶ [Toolbox](#)
- ▶ [Print/export](#)

- ▼ [Languages](#)

[Article](#)

[Discussion](#)

[Read](#)

[Edit](#)

[View history](#)



Consistency

From Wikipedia, the free encyclopedia

Not PHP^[1].

See also

[\[edit\]](#)

- [Disaster](#) - the collective noun for PHP developers, e.g. "They hired a *disaster* of PHP developers"

Footnotes

[\[edit\]](#)

1. [The screams of millions of developers](#)

This page was last modified on 22 August 2010 at 13:48.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Contact us](#)

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#)



FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently

```
// prefixed with 'str', no underscores / camelCasing  
strtolower('Hello World');
```

```
// prefixed with 'str', but now also underscores!  
str_pad($i, 4, '0', STR_PAD_LEFT);
```

```
// Now suffixed with 'string'  
convert_cyr_string('cBMΠΛΠ', 'w', 'm');
```

```
// A mix of no underscores & underscores (i.e. taking the piss)  
htmlspecialchars_decode('&lt;p&gt;Hello&lt;/p&gt;');
```



FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently

```
/**  
  @name jToys.carousel  
  @function  
  @description Create an interactive carousel  
  
  @param {selector} container Element to enhance  
    Direct children of this item will be treated as  
    carousel items.  
  @param {boolean} [loop=false]  
    Loop the carousel from the last item to the first.  
  @param {number} [duration=200]  
    Duration of scrolling animations in milliseconds.  
  
  @example  
    jToys.carousel('#whatever', true, 500);  
*/
```

FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently

```
jToys.carousel('#whatever', true, 500);
```

FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently
- ➔ Try to make arguments read well

```
$('#<div>Hello</div>').appendTo('#main');
```


FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently
- ➔ Try to make arguments read well

```
something.size(640, 360);
```

FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently
- ➔ Try to make arguments read well

```
element.cloneNode(true);
```

FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently
- ➔ Try to make arguments read well

```
// match dates like 2010/10/12  
var re = /^s*(\d{2}|\d{4})[-\/](\d\d?)[-\/](\d\d?)s*$/
```



3

1

1



too nerdy?

FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently
- ➔ Try to make arguments read well

```
jToys.carousel('#whatever', undefined, 500);
```

FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently
- ➔ Try to make arguments read well

```
swfobject.embedSWF('movie.swf', 'vid', 500, 300, 9, false, {foo:'bar'}, ...);
```

FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently
- ➔ Try to make arguments read well
- ➔ Use option objects for optional arguments

```
jToys.carousel('#whatever', {  
  loop: true,  
  duration: 500  
});
```


FUNCTIONS & PARAMETERS

- ➔ Start with documentation
- ➔ Name functions and parameters consistently
- ➔ Try to make arguments read well
- ➔ Use option objects for optional arguments

```
/**  
  ...  
  @param {selector} container Element to enhance  
    Direct children of this item will be treated as  
    carousel items.  
  @param {Object} [opts] Options  
    @param {boolean} [opts.loop=false]  
      Loop the carousel from the last item to the first.  
    @param {number} [opts.duration=200]  
      Duration of scrolling animations in milliseconds.  
*/
```

ADDING OPTIONS

```
jToys.carousel('#whatever', {  
  loop: false,  
  duration: 200,  
  spotlightSize: 3,  
});
```

ADDING OPTIONS

```
jToys.carousel('#whatever', {  
  loop: false,  
  duration:  
  spotlight:  
});
```



ADDING OPTIONS

```
jToys.carousel('#whatever', {  
  loop: false,  
  duration: 500,  
  spotlights: 1,  
});
```



ADDING OPTIONS

- ➔ Use metaphors if necessary, but careful they don't mislead

```
jToys.carousel('#whatever', {
  loop: false,
  duration: 200,
  spotlightSize: 3,
  step: 3
});

/**
  ...
  @param {number} [opts.step=1]
    Number of places to move at a time
  ...
*/
```

ADDING OPTIONS

- ➔ Use metaphors if necessary, but careful they don't mislead
- ➔ Consider common usage, make those things easy

```
jToys.carousel('#whatever', {  
  paged: true  
});
```

```
jToys.carousel('#whatever', {  
  step: 2  
});
```

```
/**
```

```
...
```

```
@param {boolean} [opts.paged=false]
```

```
  Move by the number of items in the spotlight
```

```
@param {number} [opts.step=1]
```

```
  Number of places to move at a time
```

```
...
```

```
*/
```

ADDING OPTIONS

- ➔ Use metaphors if necessary, but careful they don't mislead
- ➔ Consider common usage, make those things easy

```
jToys.carousel('#whatever', {
  paged: true,
  step: 2
});

/**
  ...
  @param {boolean} [opts.paged=false]
    Move by the number of items in the spotlight
  @param {number} [opts.step=1]
    Number of places to move at a time
  ...
*/
```

ADDING OPTIONS

- ➔ Use metaphors if necessary, but careful they don't mislead
- ➔ Consider common usage, make those things easy
- ➔ Avoid options that depend on or clash with other options

```
#whatever {  
  overflow-x: hidden;  
  overflow-y: auto;  
}
```


however, we should have started by assuming that no scrollbars are needed. In that case the content height of d1 is exactly the maximum height of 5em, proving that the assumption was correct and d1 indeed should not have scrollbars.

The computed values of `overflow-x` and `overflow-y` are the same as their specified values, except that some combinations with `visible` are not possible: if one is specified as `visible` and the other is `scroll` or `auto`, then `visible` is set to `auto`. The computed value of `overflow` is equal to the computed value of `overflow-x` if `overflow-y` is the same; otherwise it is the pair of computed values of `overflow-x` and `overflow-y`.

The scrolling mechanism depends on the UA. The most common mechanism is a scrollbar, but panners, hand cursors, page flickers, etc. are also possible. A value of `scroll` would tell UAs that support a visible scrolling mechanism to display one so that users can access the clipped content. The `overflow: auto` property sets an author intent to have a visible scrolling mechanism.

ADDING OPTIONS

- ➔ Use metaphors if necessary, but careful they don't mislead
- ➔ Consider common usage, make those things easy
- ➔ Avoid options that depend on or clash with other options

```
jToys.carousel('#whatever', {  
  paged: true  
});
```

ADDING OPTIONS

- ➔ Use metaphors if necessary, but careful they don't mislead
- ➔ Consider common usage, make those things easy
- ➔ Avoid options that depend on or clash with other options

```
jToys.carousel('#whatever', {  
  paged: 2  
});
```

ADDING OPTIONS

- ➔ Use metaphors if necessary, but careful they don't mislead
- ➔ Consider common usage, make those things easy
- ➔ Avoid options that depend on or clash with other options
- ➔ Blah blah consistency again etc etc

```
#whatever {  
    color: rgba(50%, 50%, 50%, 0.5);  
}
```



ADDING OPTIONS

- ➔ Use metaphors if necessary, but careful they don't mislead
- ➔ Consider common usage, make those things easy
- ➔ Avoid options that depend on or clash with other options
- ➔ Blah blah consistency again etc etc

```
#whatever {  
    color: rgba(255, 127, 0, 50%);  
}
```

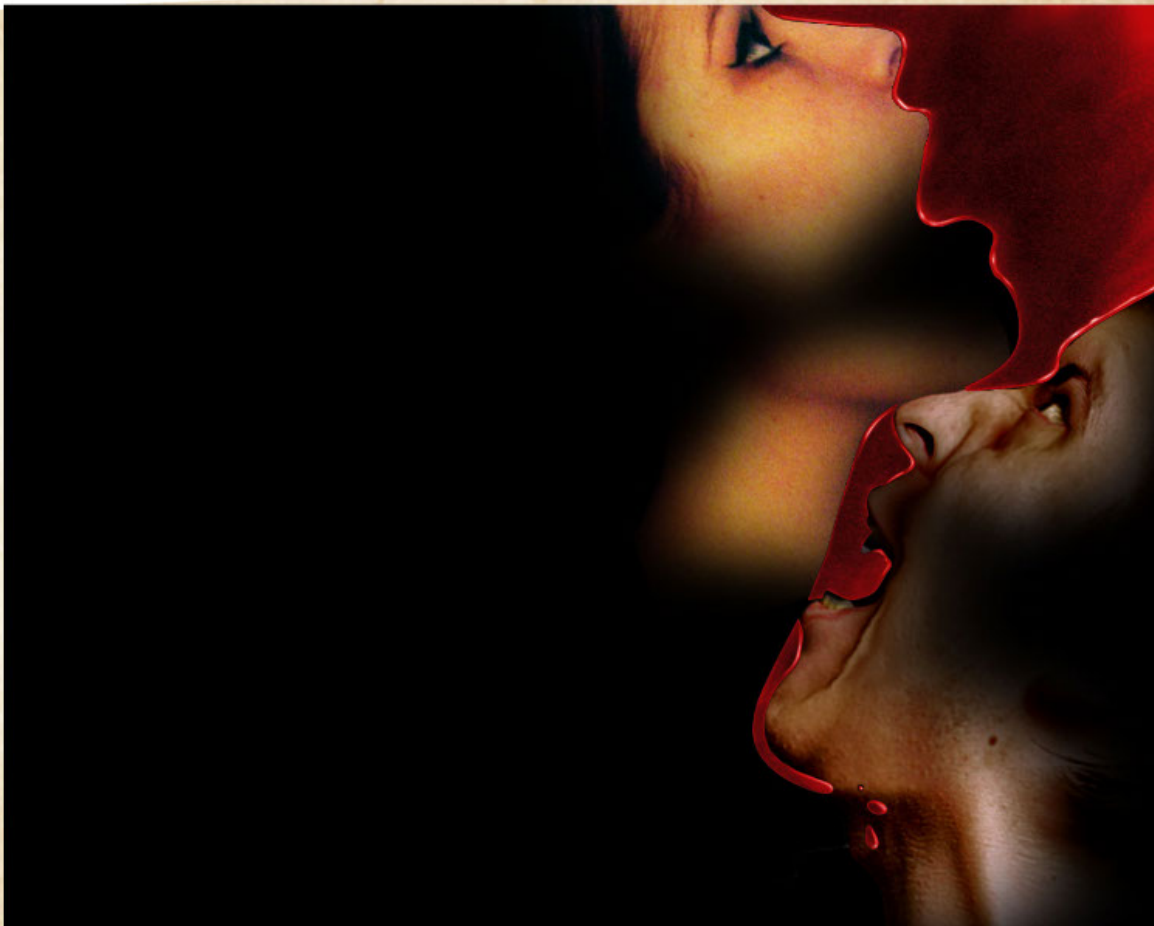
ARE WE...

...on the right tracks?

TESTING YOUR API



TESTING YOUR API



TESTING YOUR API

- ➔ Show your code examples to someone else
- ➔ Do they have a rough idea what it does?
- ➔ Let them read your documentation
- ➔ Capture any questions they have - are they missing from your docs?

TESTING YOUR API

- ➔ Show
- ➔ Do
- ➔ Let
- ➔ Cap
- you



om

FLEXIBLE APIS

Allowing the user to do something a bit different

FLEXIBLE APIS

- ➔ Resign yourself to the idea that what you make will never do exactly what the user wants

FLEXIBLE APIS

➔ Resign yourself to the idea that what you make will never

Massive Pricks



Fernando
Alonso



FLEXIBLE APIS

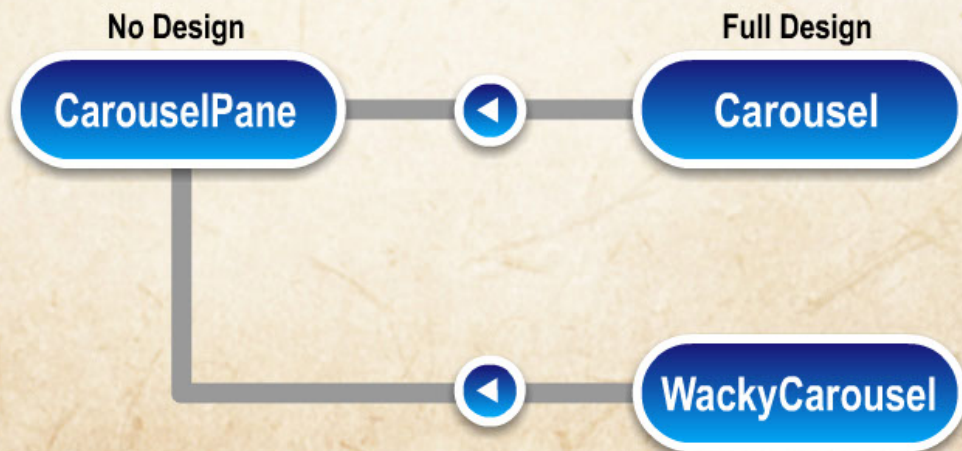
- ➔ Resign yourself to the idea that what you make will never do exactly what the user wants
- ➔ How can we add more flexibility to our carousel?
- ➔ Build something more abstract to use or inherit from

Full Design

Carousel

FLEXIBLE APIS

- ➔ Resign yourself to the idea that what you make will never do exactly what the user wants
- ➔ How can we add more flexibility to our carousel?
- ➔ Build something more abstract to use or inherit from
- ➔ Eases the learning curve & adds perception of progress



FLEXIBLE APIS

- ➔ Resign yourself to the idea that what you make will never do exactly what the user wants
- ➔ How can we add more flexibility to our carousel?
- ➔ Build something more abstract to use or inherit from
- ➔ Eases the learning curve & adds perception of progress



FLEXIBLE APIS

- ➔ Resign yourself to the idea that what you make will never do exactly what the user wants
- ➔ How can we add more flexibility to our carousel?
- ➔ Build something more abstract to use or inherit from
- ➔ Eases the learning curve & adds perception of progress



FLEXIBLE APIS

- ➔ Resign yourself to the idea that what you make will never do exactly what the user wants
- ➔ How can we add more flexibility to our carousel?
- ➔ Build something more abstract to use or inherit from
- ➔ Eases the learning curve & adds perception of progress

```
$.get('someFile.txt', function(data) {  
    console.log(data);  
});
```

FLEXIBLE APIS

- ➔ Resign yourself to the idea that what you make will never do exactly what the user wants
- ➔ How can we add more flexibility to our carousel?
- ➔ Build something more abstract to use or inherit from
- ➔ Eases the learning curve & adds perception of progress

```
$.ajax({  
  url: 'someFile.txt',  
  success: function(data) {  
    console.log(data);  
  },  
  error: function() {  
    console.log('Oh noes!');  
  },  
  timeout: 4000  
});
```

INSTANCES...

...and Instance Properties & Methods

INSTANCES

```
var carousel = new jToys.CarouselPane('#whatever', {  
    loop: false,  
    page: true  
});  
  
carousel.moveTo(3).itemsInSpotlight().css('color', 'red');
```

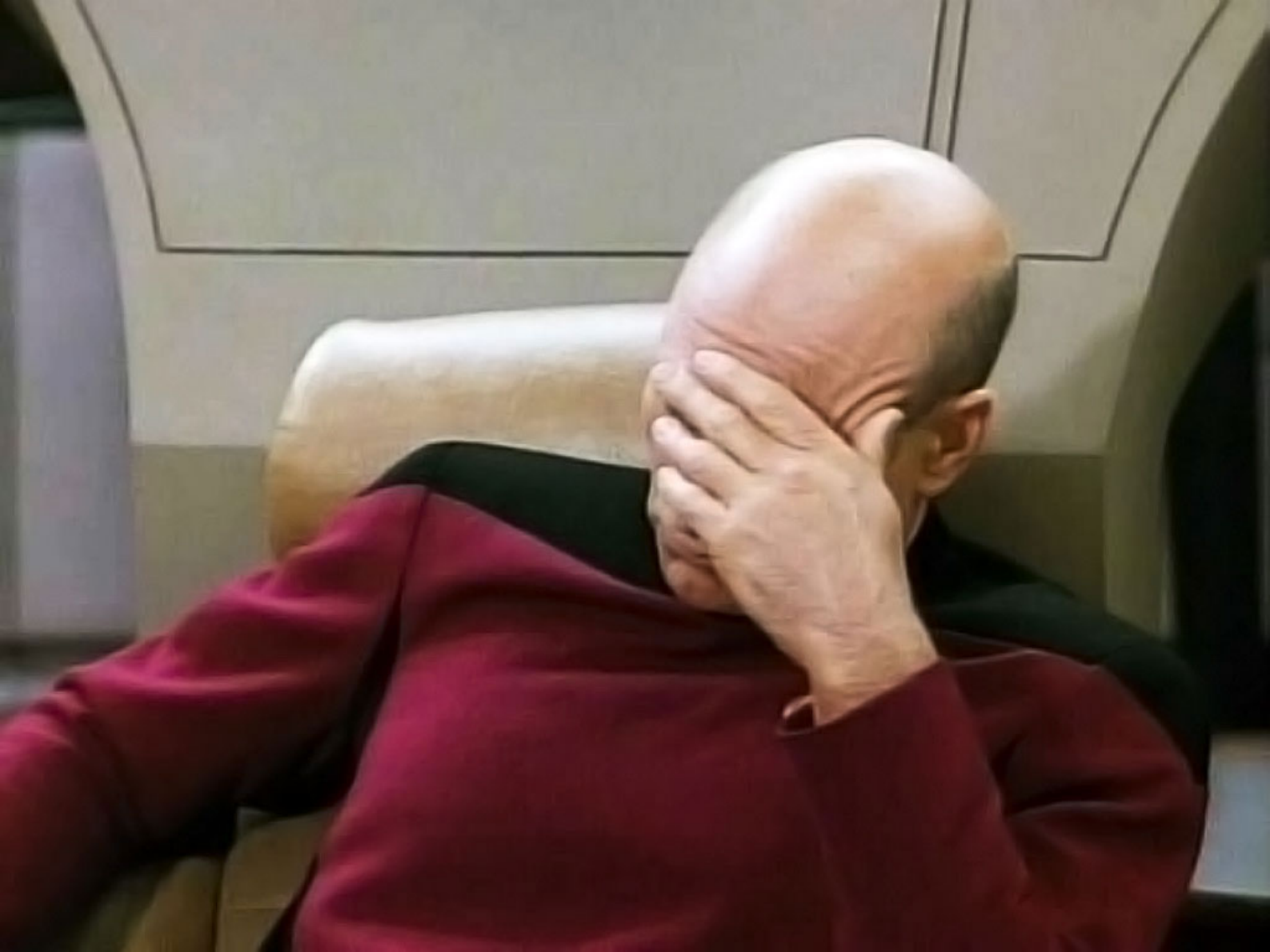
INSTANCES

```
var $carousel = $('#whatever').carouselPane({
  loop: false,
  page: true
});

$carousel.carouselPane('moveTo', 3)
  .carouselPane('itemsInSpotlight')
  .css('color', 'red');
```

INSTANCES

```
var $dialog = $('#whatever').dialog();  
  
$dialog.bind('dialogfocus', callback);  
  
$dialog.show();  
$dialog.hide();  
  
$dialog.dialog('open');  
$dialog.dialog('close');  
  
$dialog.dialog('cloes'); // silent fail  
  
dialog.cloes(); // undefined is not a function
```



ADDING HOOKS

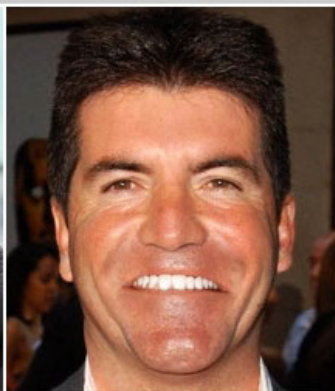
For More Flexibility

FEATURE REQUESTS

Massive Pricks



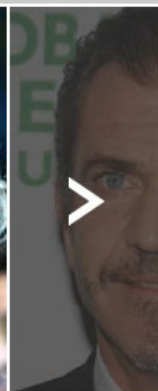
Jeremy Clarkson



Simon Cowell



Flavio Briatore



FEATURE REQUESTS

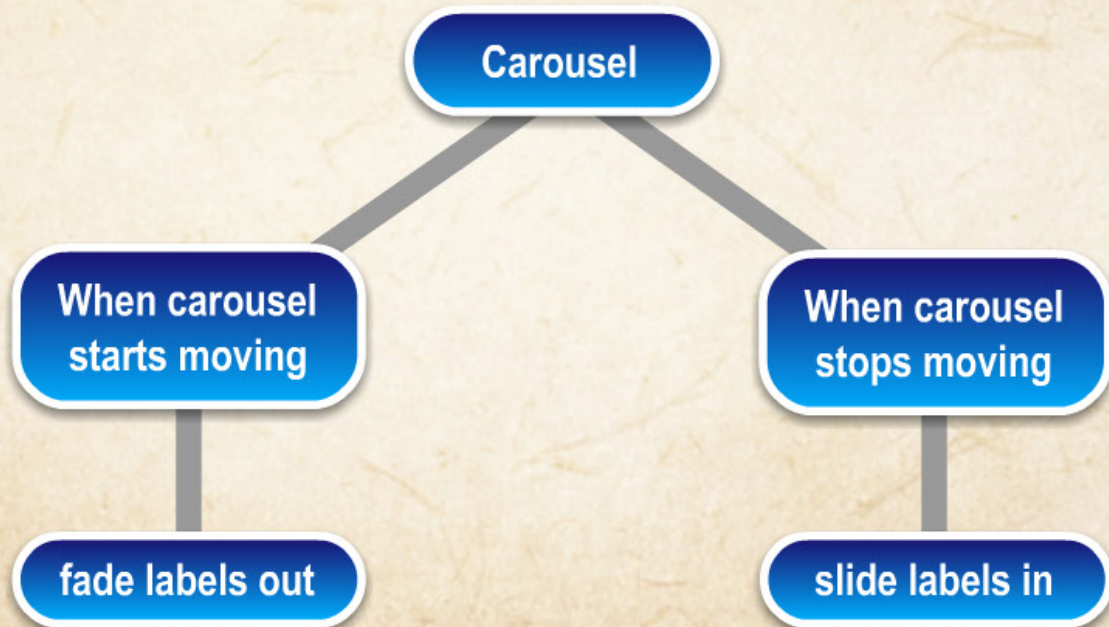
```
$('#whatever').carousel({  
  loop: true,  
  duration: 200,  
  spotlightSize: 3,  
  labels: '.label',  
  labelAnimIn: ?,  
  labelAnimOut: ?  
});
```



I BEAT
ANOREXIA

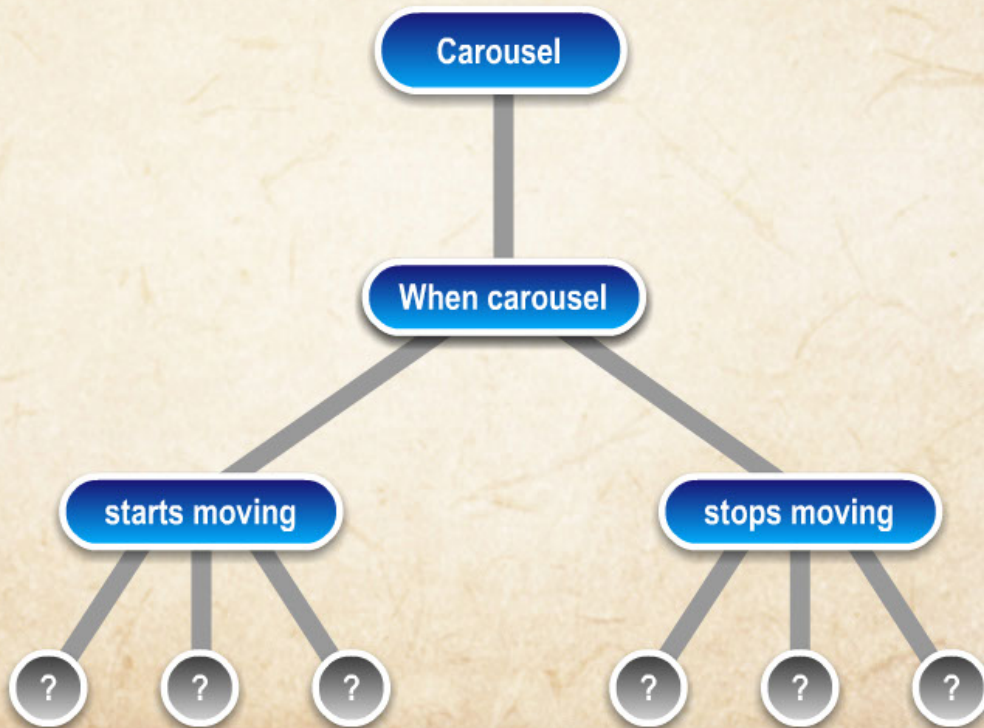
FEATURE REQUESTS

- ➔ Don't get into option-hell
- ➔ Break the request down into components



FEATURE REQUESTS

- ➔ Don't get into option-hell
- ➔ Break the request down into components



EVENTS

Firing & Listening For

EVENTS

```
// User's code
var carousel = new jToys.Carousel('#whatever');

carousel.on('move', function() {
    console.log('hello!');
});
```

Class **EventTarget**

Known Subclasses:

[AsyncQueue](#)

EventTarget provides the implementation for any object to produce custom events, and also allows other *EventTargets* to target the events from the other object. *EventTarget* is designed to be used with *EventCustom* in an interface that allows events to be listened to and it possible for implementing code to subscribe to an event that will or will not be created at all.

```
// abc
// !bc
```

EventEmitter

Many objects in Node emit events: a TCP server emits a `connection` event when a new connection is accepted, a child process emits an event when it exits. All objects that inherit from `events.EventEmitter`.

Events are represented by a camel-cased string, `eventName`, such as `'data'`, `'connect'`, `'error'`.

Listeners can be added to an event by calling `addListener(eventName, listener)`.

Listeners can be removed by calling `removeListener(eventName, listener)`.

Constructor Detail

EventDispatcher() Constructor

public function EventDispatcher(target: IEventDispatcher = null)

Language Version: ActionScript 3.0

Runtime Versions: Flash Player 9, AIR 1.0

Aggregates an instance of the EventDispatcher class.

The EventDispatcher class is generally used as a base class, which means that most developers use the constructor function. However, advanced developers who are implementing the IEventDispatcher interface use this constructor. If you are unable to extend the EventDispatcher class and must instead implement the IEventDispatcher interface, use this constructor to aggregate an instance of the EventDispatcher class.

Parameters

target: IEventDispatcher (default = null) — The target object for events. Developers who implement the IEventDispatcher interface use this parameter to aggregate an instance of the EventDispatcher class. If a necessary so that the containing object can use the target for events. If not, a null value is used when a class extends EventDispatcher.

See Also

EventDispatcher class
IEventDispatcher interface

EVENTS

```
// User's code
var carousel = new jToys.Carousel('#whatever')
    .on('move', function() {
        console.log('hello!');
    });

carousel.moveBy(1); // logs 'hello!'

// Our code
jToys.Carousel.prototype._startMove = function() {
    this.fire('move');
    // then do moving stuff...
};
```

EVENTS

```
// User's code
var $carousel = $('#whatever').carousel()
    .bind('carouselmove', function() {
        console.log('hello!');
    });

$carousel.carousel('moveTo', 5); // logs 'hello!'
```

EVENTS

- ➔ Fire lots of events
- ➔ Fire events before an action happens
- ➔ Fire events after asynchronous actions
- ➔ Document them!

```
/**
 * @name jToys.Carousel#event:move
 * @description Fires when the carousel is about to move.
 *
 * @param {jToys.Event} event Event Object
 *   @param {number} event.moveBy
 *     The number of items the Carousel will move by.
 *     Negative when moving backwards.
 *
 * @example
 *   myCarousel.on('move', function(event) {
 *     console.log('Moving by ' + event.moveBy);
 *   });
 */
```

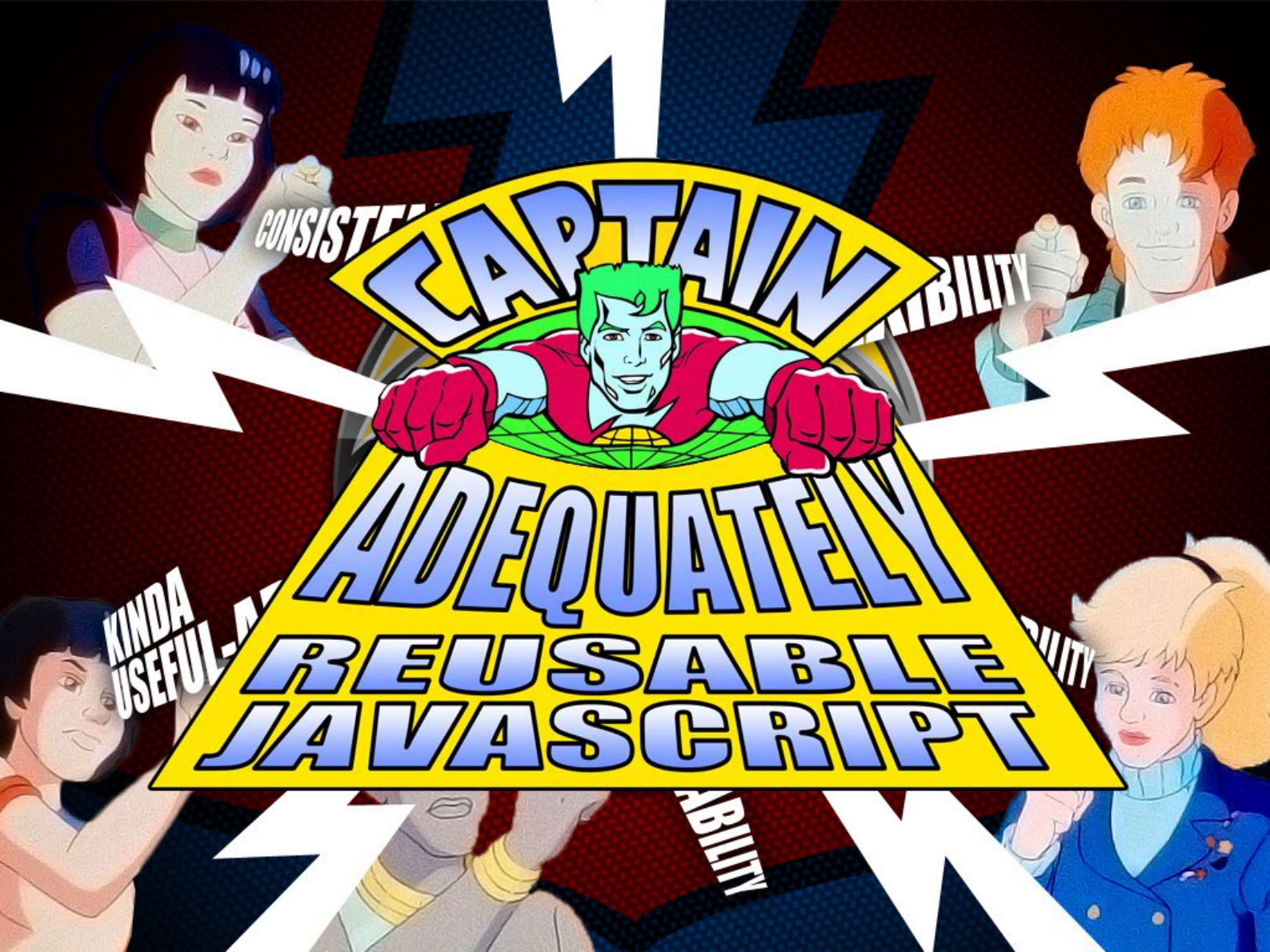

EVENTS

- ➔ Fire lots of events
- ➔ Fire events before an action happens
- ➔ Fire events after asynchronous actions
- ➔ Document them!

```
var carousel = new jToys.Carousel('#whatever')
  .on('move', function(event) {
    // fade labels out
  })
  .on('afterMove', function() {
    // slide labels in
  });
```

WRAPPING UP

- ➔ Consider the users...
- ➔ Write great docs BEFORE implementation
- ➔ Adopt existing conventions where relevant
- ➔ Ensure the naming of methods, properties, params & events reads well
- ➔ Use option objects for optional params
- ➔ Aim for a 1 line, 1 param usage
- ➔ But allow the user to get more by writing more
- ➔ Let the user get creative



CAPTAIN

**ADEQUATELY
REUSABLE
JAVASCRIPT**

CONSISTENT

ABILITY

KINDA
USEFUL

ABILITY

THAT'S IT!

- ➔ Email: jakearchibald@theteam.co.uk
- ➔ Twitter: [@jaffathecake](https://twitter.com/jaffathecake)
- ➔ Also twitter: [@theteam](https://twitter.com/theteam)
- ➔ JsDoc Toolkit: <http://code.google.com/p/jsdoc-toolkit/>