

Tips & Tricks of

Effective

iOS Developers



BEN SCHEIRMAN
@subdigital



ChaiOne

NS *Screencast*

Weekly Screencasts
on **iOS DEVELOPMENT**

\$9/month

Tips & Tricks of

Effective

iOS Developers



effective |i' fektiv|

adjective

*successful in producing a
desired or intended result*

Effective

DEVELOPERS

Effective

DEVELOPERS

FIX PROBLEMS

Effective

DEVELOPERS

WRITE CODE FASTER

Effective

DEVELOPERS

CONFIDENTLY REFACTOR

Effective

DEVELOPERS

WRITE BETTER CODE

Effective

DEVELOPERS

WRITE FEWER BUGS

Effective

DEVELOPERS

GET THINGS DONE

Effective

DEVELOPERS

SHIP BETTER SOFTWARE

SHIP BETTER SOFTWARE

ME?

I HAVE

NO IDEA WHAT I'M DOING



I aspire to be

Effective

Objective-C

Tips

Always use braces for conditionals

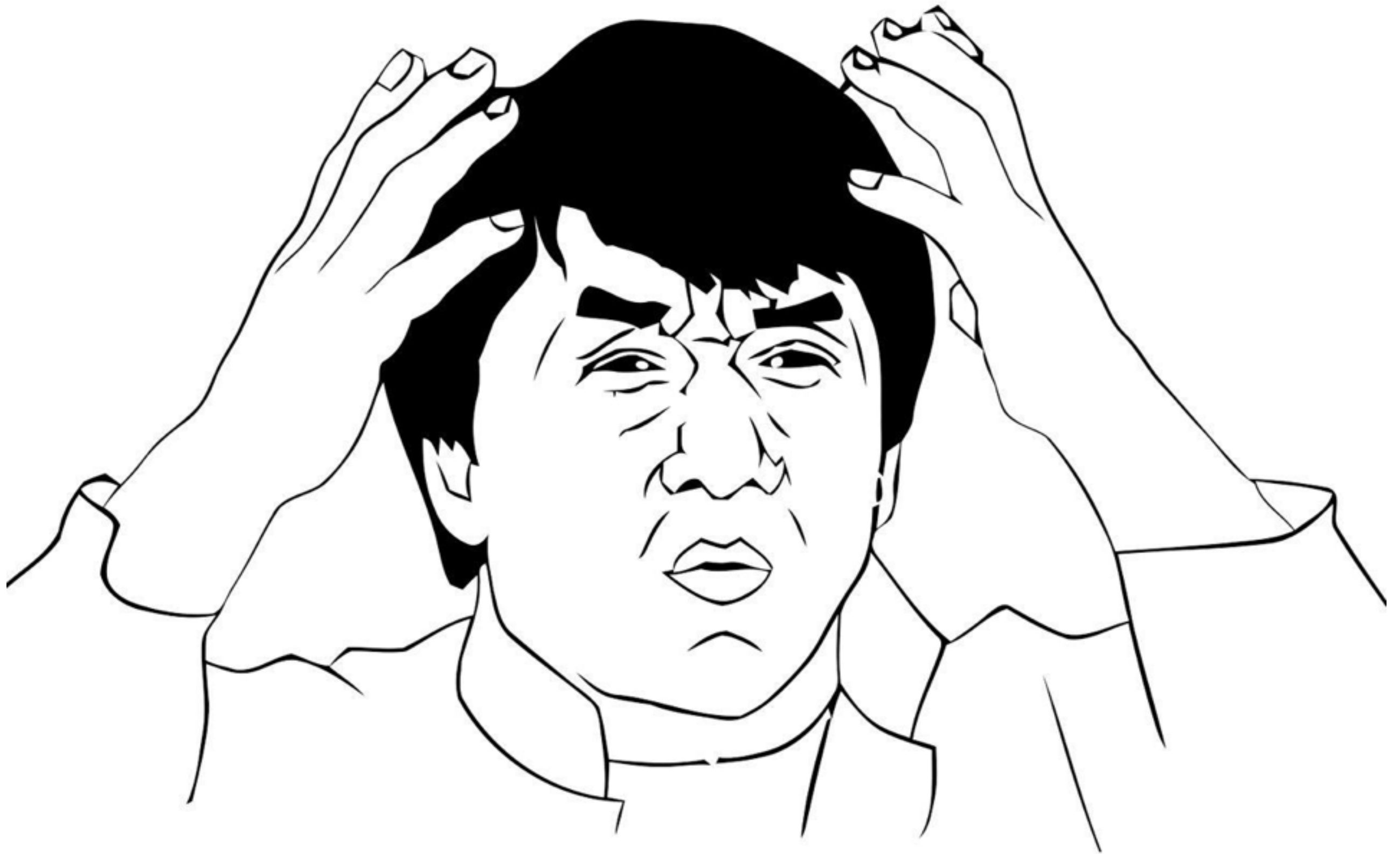
```
if ([self isHungry])  
    [self eatBurger];  
    [self takeNap];
```

```
if ([self isHungry]) {  
    [self eatBurger];  
    [self takeNap];  
}
```

Always use braces for conditionals

```
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
goto fail;
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;
...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
```



Use Class Continuations for Private Properties

Player.h

```
@interface Player

- (void)moveLeft;
- (void)moveRight;
- (void)jump;
- (void)hit;

@end
```

Player.m

```
@interface Player ()
@property CGFloat health;
@property Texture *texture;
@end

@implementation Player

...

@end
```

Use Class Continuations for Private Properties

```
@interface ViewController ()  
@property IBOutlet UILabel *label;  
@end  
  
@implementation ViewController  
...  
@end
```

Use intention revealing method names

```
- (NSDate *)nextDate:(NSDate *);
```

```
- (NSDate *)nextBillingDateAfterDate:(NSDate *);
```

Name single parameter BOOL arguments

```
[self showPanel:NO];
```

```
-(void)showPanel:(BOOL)animated;
```


Name single parameter BOOL arguments

```
[self showPanelAnimated:NO];
```

```
-(void)showPanelAnimated:(BOOL)animated;
```

Add Comment Documentation to Public Methods

```
/*! Returns the next billing date after the specified date. If the date falls on a banking holiday, then
 * the billing date will be pushed back by a day.
 * \param date The reference date to use. Can pass nil for today's date.
 * \returns an NSDate representing the next billing date.
 */
- (NSDate *)nextBillingDateAfterDate:(NSDate *)date;
```

- (NSDate *)nextBillingDateAfterDate:(NSDate *)date {

Declaration - (NSDate *)nextBillingDateAfterDate:(NSDate *)date;

Description Returns the next billing date after the specified date. If the date falls on a banking holiday, then the billing date will be pushed back by a day.

Parameters date The reference date to use. Can pass nil for today's date.

Returns an NSDate representing the next billing date.

Declared In ViewController.h

Lazily Initialize Properties

```
- (void)viewDidLoad {  
    [self.fetchedResultsController performFetch:nil];  
}
```

```
- (NSFetchedResultsController *)fetchedResultsController {  
    if (_frc == nil) {  
        _frc = [[NSFetchedResultsController alloc]  
                initWithFetchRequest: self.fetchRequest  
                ...  
                ]  
    }  
  
    return _frc;  
}
```

Lazily Initialize Properties

```
- (NSFetchRequest *)fetchRequest {  
    if (_fetchRequest == nil) {  
        _fetchRequest = ...  
    }  
    return _fetchRequest;  
}
```

Lazily Initialize Properties

```
- (void)didReceiveMemoryWarning {  
    self.fetchResultsController = nil;  
    self.fetchRequest = nil;  
  
    ...  
}
```

Use Refactor Method judiciously

```
- (UITableViewCell *)tableView:(UITableView *)tv cellForRowAtIndexPath:(NSIndexPath *)indexPath {  
    UITableViewCell *cell = [tv dequeueReusableCellWithIdentifier:@"cell"];  
    NSArray *peopleInSection = self.sections[indexPath.section];  
    Person *person = peopleInSection[indexPath.row];  
    cell.textLabel.text = [NSString stringWithFormat:@"%@ %@", person.firstName,  
                                                                    person.lastName];  
    cell.detailTextLabel.text = person.email;  
    return cell;  
}
```

Use Refactor Method judiciously

```
- (UITableViewCell *)tableView:(UITableView *)tv cellForRowAtIndexPath:(NSIndexPath *)indexPath {  
    UITableViewCell *cell = [tv dequeueReusableCellWithIdentifier:@"cell"];  
    Person *person = [self personAtIndex:indexPath:indexPath];  
    cell.textLabel.text = [NSString stringWithFormat:@"%@ %@", person.firstName,  
                                                                    person.lastName];  
  
    cell.detailTextLabel.text = person.email;  
    return cell;  
}
```

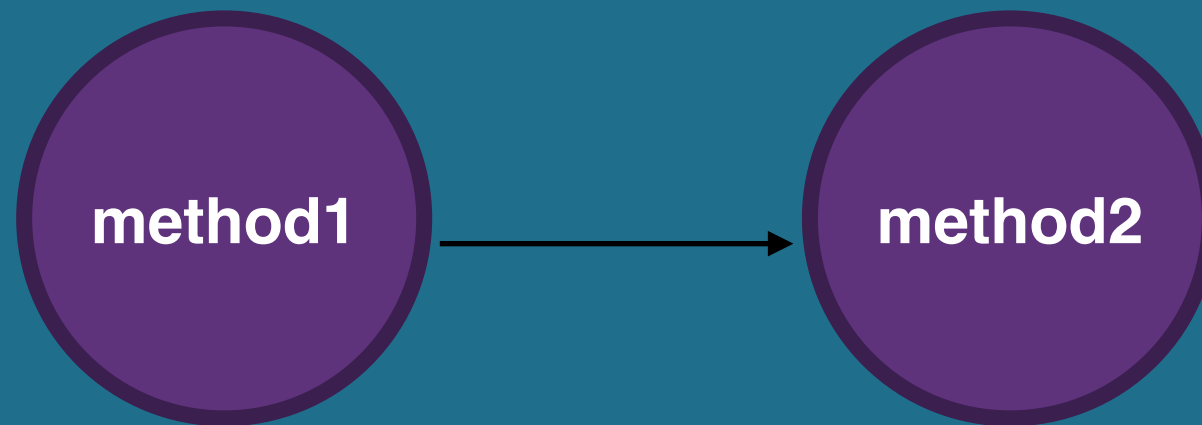

Use Refactor Method judiciously

```
- (UITableViewCell *)tableView:(UITableView *)tv cellForRowAtIndexPath:(NSIndexPath *)indexPath {  
    UITableViewCell *cell = [tv dequeueReusableCellWithIdentifier:@"cell"];  
    Person *person = [self personAtIndex:indexPath:indexPath];  
    cell.textLabel.text = person.fullName;  
    cell.detailTextLabel.text = person.email;  
    return cell;  
}
```


Use Refactor Method judiciously

```
- (UITableViewCell *)tableView:(UITableView *)tv cellForRowAtIndexPath:(NSIndexPath *)indexPath {  
    UITableViewCell *cell = [tv dequeueReusableCellWithIdentifier:@"cell"];  
    Person *person = [self personAtIndex:indexPath:indexPath];  
    [self updateCell:cell forPerson:person];  
    return cell;  
}
```

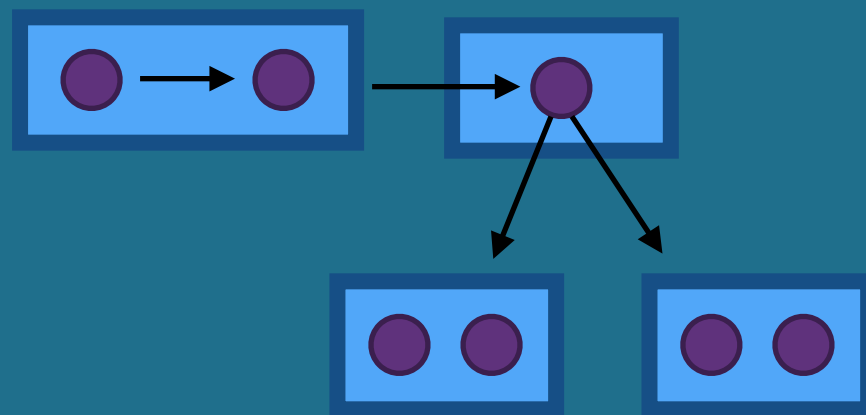
Understand Dependencies



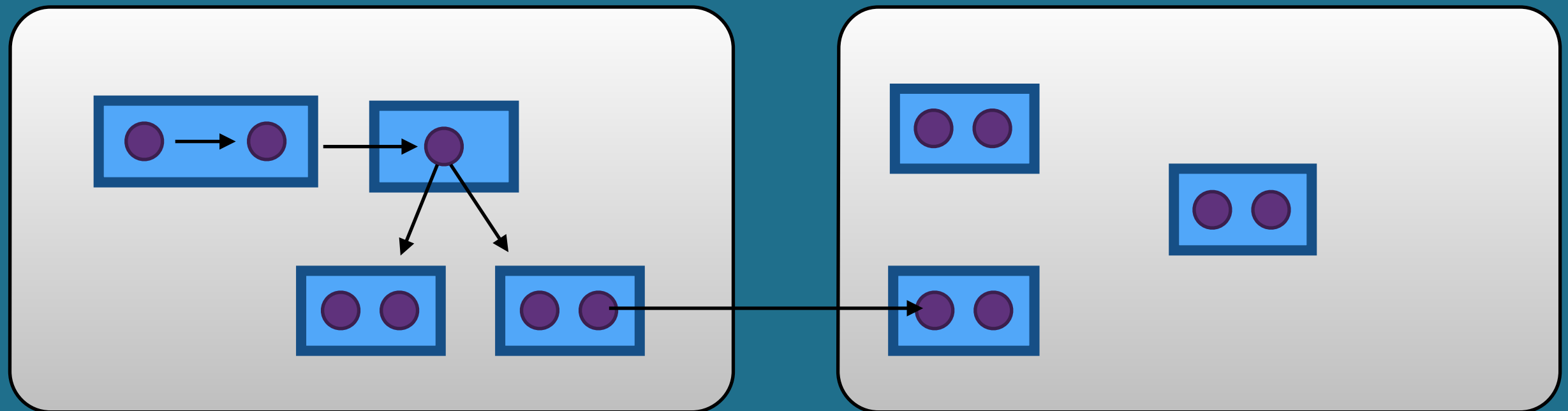
Understand Dependencies



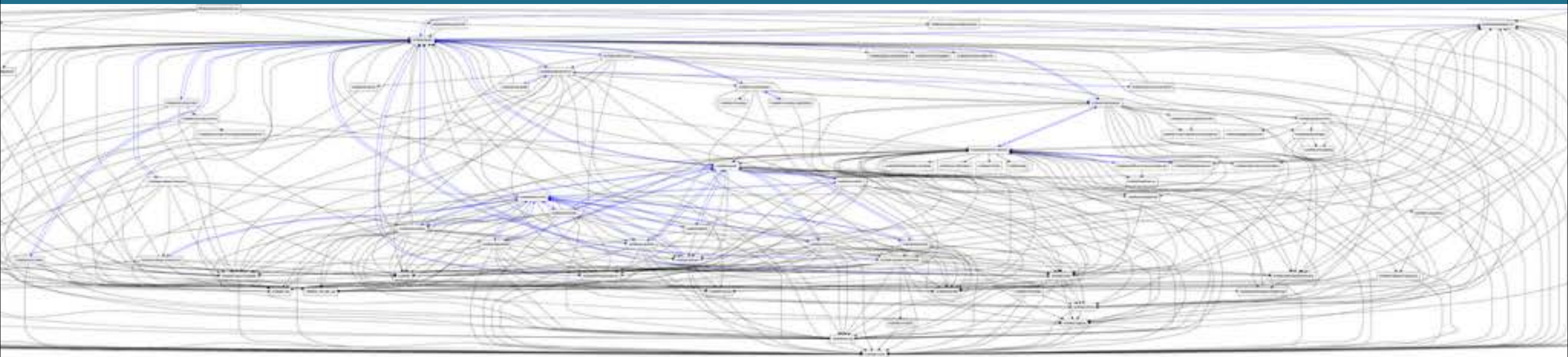
Understand Dependencies



Understand Dependencies

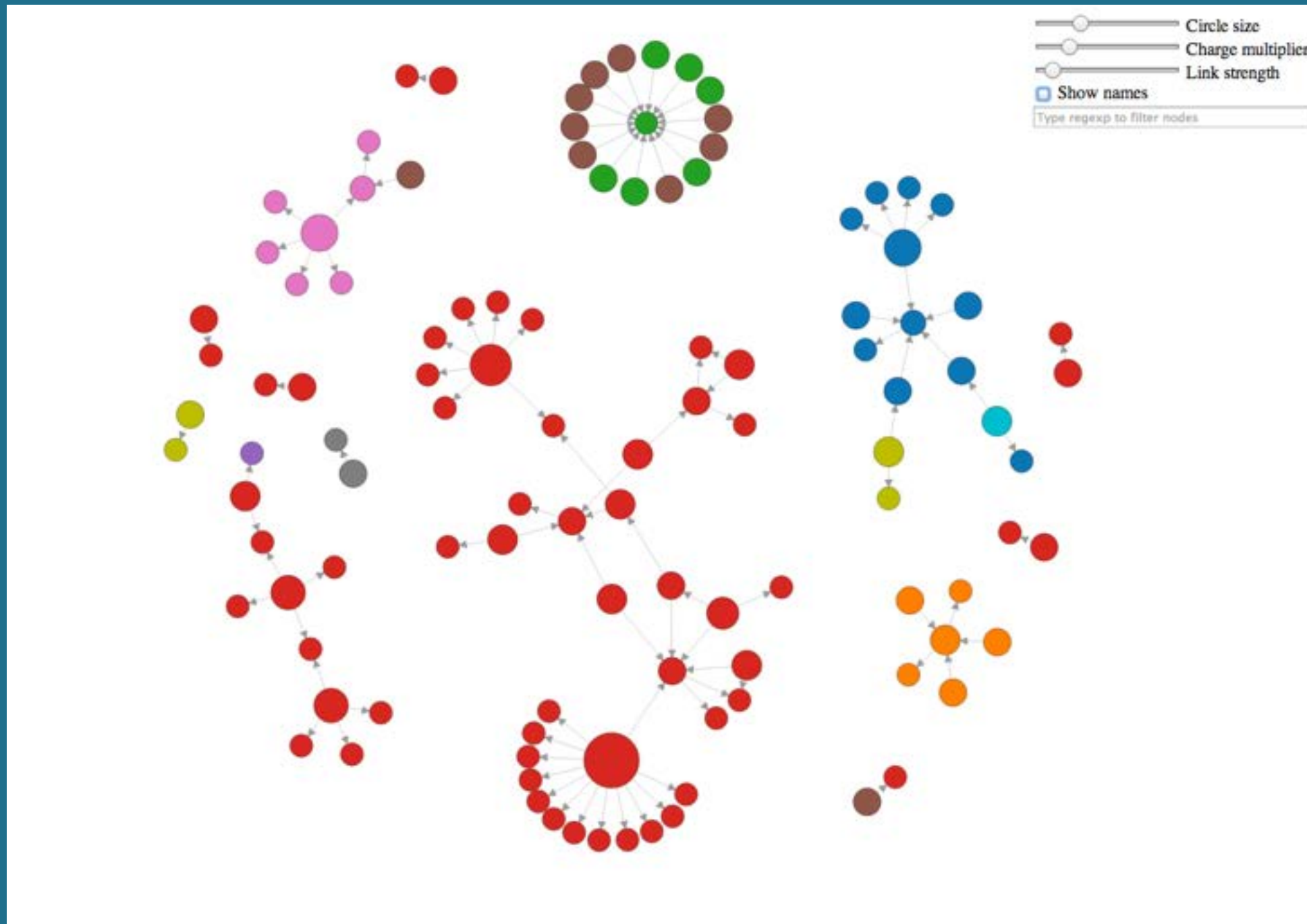


Keep your dependencies in check

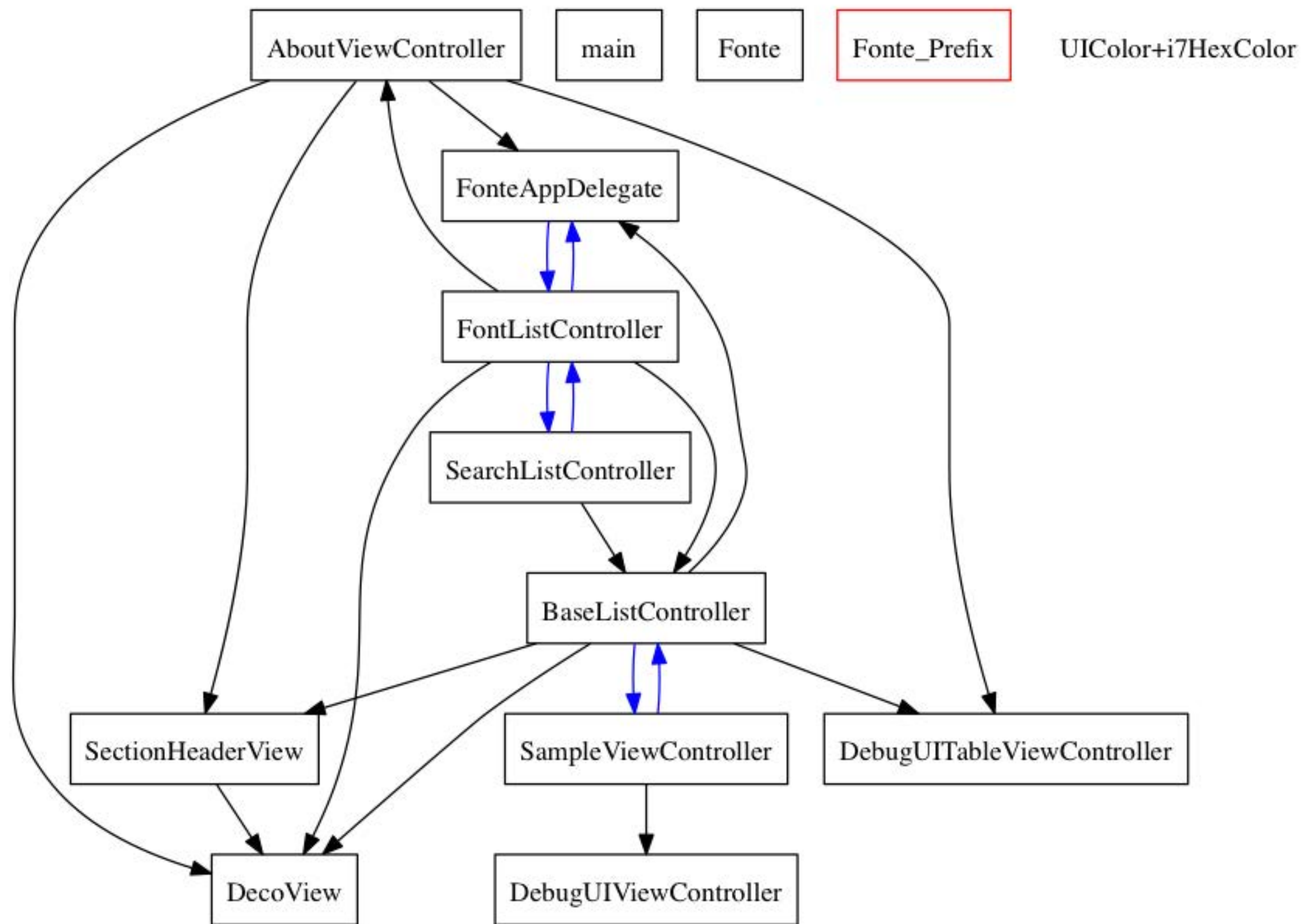


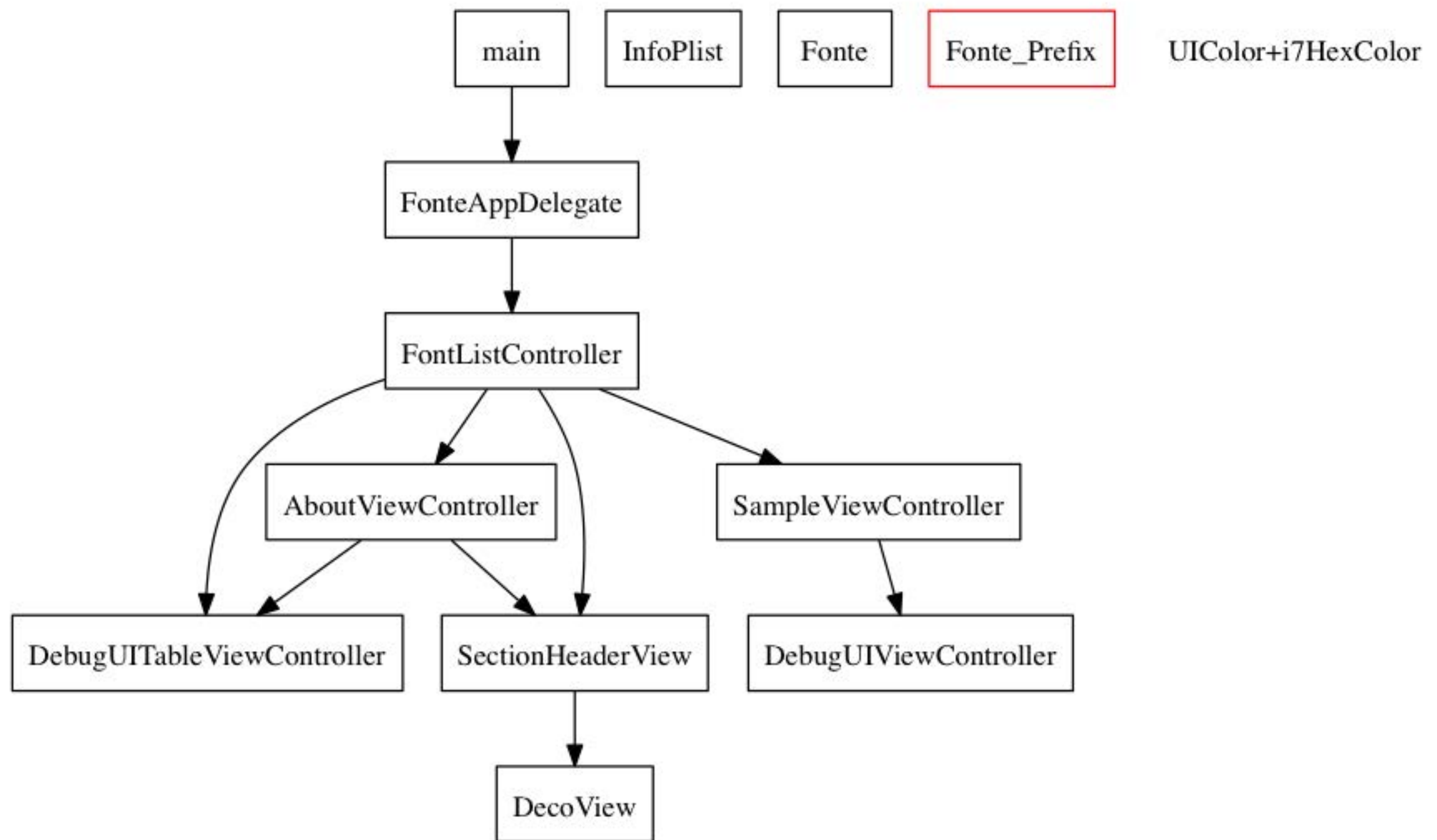
00 is your friend

objc-dependency-visualizer



<http://paultaykalo.github.io/objc-dependency-visualizer>





Reduce Dependencies by...

- Loose Coupling
- Programming to interfaces
- Dependency Inversion
- Reduce God Objects
- Reduce Singletons

With Less Dependencies you can...

Refactor

Change your mind

Change your implementation

Add features

“Ruthlessly modularize functionality. Design, implement, and package everything as if it were to be distributed as 3rd-party code”

“Ruthlessly modularize functionality. Design, implement, and package everything as if it were to be distributed as 3rd-party code”



Mattt Thompson

Slim down your classes

```
1439 }
1440 - (void)showFindFriendsControll
1441     [self showViewControllerAtM
1442 }
1443
1444 - (void)showFindFriendsControll
1445     self.findFriendsVC.shouldLo
1446     [self showFindFriendsContro
1447 }
1448
1449 @end
1450
```


1446

[se

1447

}

1448

1449

@end

1450

Try to impose limits



Photograph by Mikey Schaefer

Challenge Yourself

Max LOC for .m: ~ 150

Max LOC for method: 4-5



Cesare Rocchi

Automate it

STAND BACK



I'M GOING TO TRY

~~SCIENCE~~

MADE BY THE SCIENCE COM

awk

```
find "${SRCROOT}" \( -name "*.h" -or -name "*.m" \)
-and \( -path "${SRCROOT}/Pods/*"
-prune -o -print0 \) |
xargs -0 wc -l |
awk '$1 > 400 && $2 != "total" {
    for(i=2;i<NF;i++) {
        printf "%s%s", $i, " "
    }
    print $NF ":1: warning: File more
than 400 lines (" $1 "), consider
refactoring." }'
```

Xcode

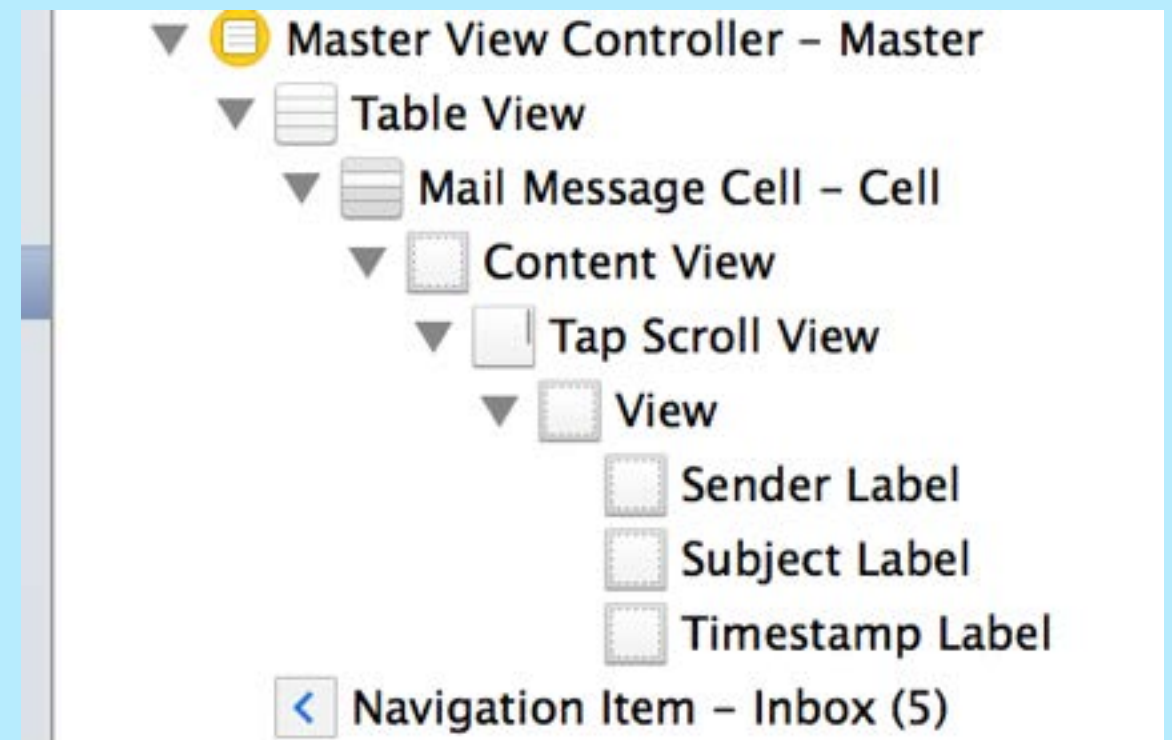
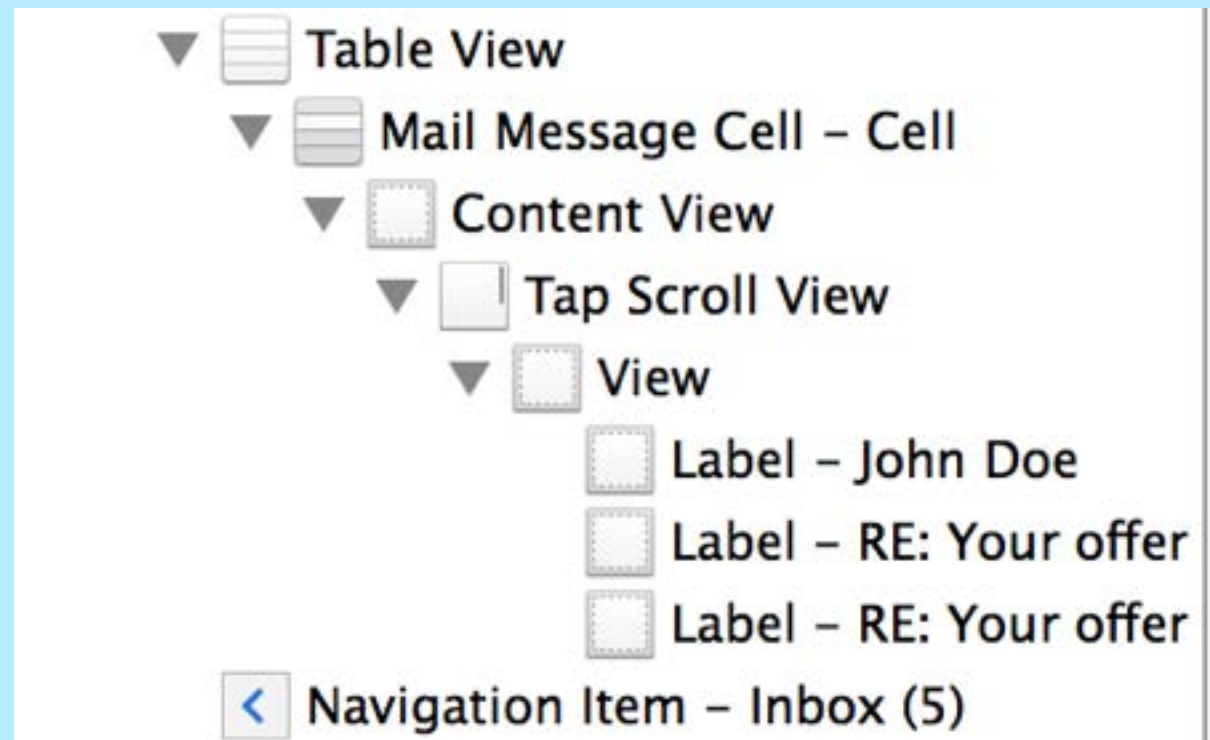
Tips

Completing Methods

“- ta”

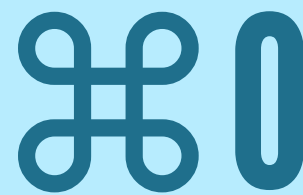
```
36 Tells the data source to return the number of rows in a given section of a table view. (required) More...
37
38 [M] (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath
39 [M] (NSInteger)tableView:(UITableView *)tableView indentationLevelForRowAtIndexPath:(NSIndexPath *)indexPath
40 [M] (void)tableView:(UITableView *)tableView moveRowAtIndexPath:(NSIndexPath *)sourceIndexPath toIndexPath:(N...
41 [M] (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
42 [M] (void)tableView:(UITableView *)tableView performAction:(SEL)action forRowAtIndexPath:(NSIndexPath *)index...
43 [M] (NSInteger)tableView:(UITableView *)tableView sectionForSectionIndexTitle:(NSString *)title atIndex:(NSIn...
44 [M] (BOOL)tableView:(UITableView *)tableView shouldHighlightRowAtIndexPath:(NSIndexPath *)indexPath
45 [M] (BOOL)tableView:(UITableView *)tableView shouldIndentWhileEditingRowAtIndexPath:(NSIndexPath *)indexPath
46 [M] (BOOL)tableView:(UITableView *)tableView shouldHighlightRowAtIndexPath:(NSIndexPath *)indexPath
47 [M] (BOOL)tableView:(UITableView *)tableView shouldIndentWhileEditingRowAtIndexPath:(NSIndexPath *)indexPath
48 [M] (BOOL)tableView:(UITableView *)tableView shouldIndentWhileEditingRowAtIndexPath:(NSIndexPath *)indexPath
49 - tableView:numberOfRowsInSection:
50
51
```

Label your views in Interface Builder



Learn Keyboard Shortcuts





SwipeToRevealCell.xcodeproj — MailMessage.m

Build SwipeToRevealCell: Succeeded | 2/27/14 at 9:35 AM | No Issues

SwipeToRevealCell > iPhone Retina (3.5-inch)

SwipeToRevealCell

- 2 targets, iOS SDK 7.0
 - SwipeToRevealCell
 - AppDelegate.h
 - AppDelegate.m
 - Main.storyboard
 - MasterViewController.h
 - MasterViewController.m
 - Images.xcassets
 - Supporting Files
 - MailMessage.h
 - MailMessage.m
 - MailMessageCell.h
 - MailMessageCell.m
 - BSTapScrollView.h
 - BSTapScrollView.m
 - SwipeToRevealCellTests
 - Frameworks
 - Products

```
//  
// MailMessage.m  
// SwipeToRevealCell  
//  
// Created by ben on 2/10/14.  
// Copyright (c) 2014 NSScreencast. All rights reserved.  
//  
#import "MailMessage.h"  
  
@implementation MailMessage  
  
@end
```

Identity and Type

Name: MailMessage.m

Type: Default - Objective-C Source File

Location: Relative to Group

MailMessage.m

Full Path: /Users/ben/nsscreencast/sourcecode/107-swipe-to-reveal-cells/SwipeToRevealCell/SwipeToRevealCell/MailMessage.m

Target Membership

- ☒ SwipeToRevealCell
- ☐ SwipeToRevealCellTests

Text Settings

Text Encoding: Unicode (UTF-8)

Line Endings: Default - OS X / Unix (LF)

Indent Using: Spaces

Widths: 4 Tab 4 Indent

☐ Wrap lines

Source Control

Repository: sourcecode

Type: Git

Current Branch: master

Version: --

Status: No changes

View Controller - A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller - A controller that manages a table view.

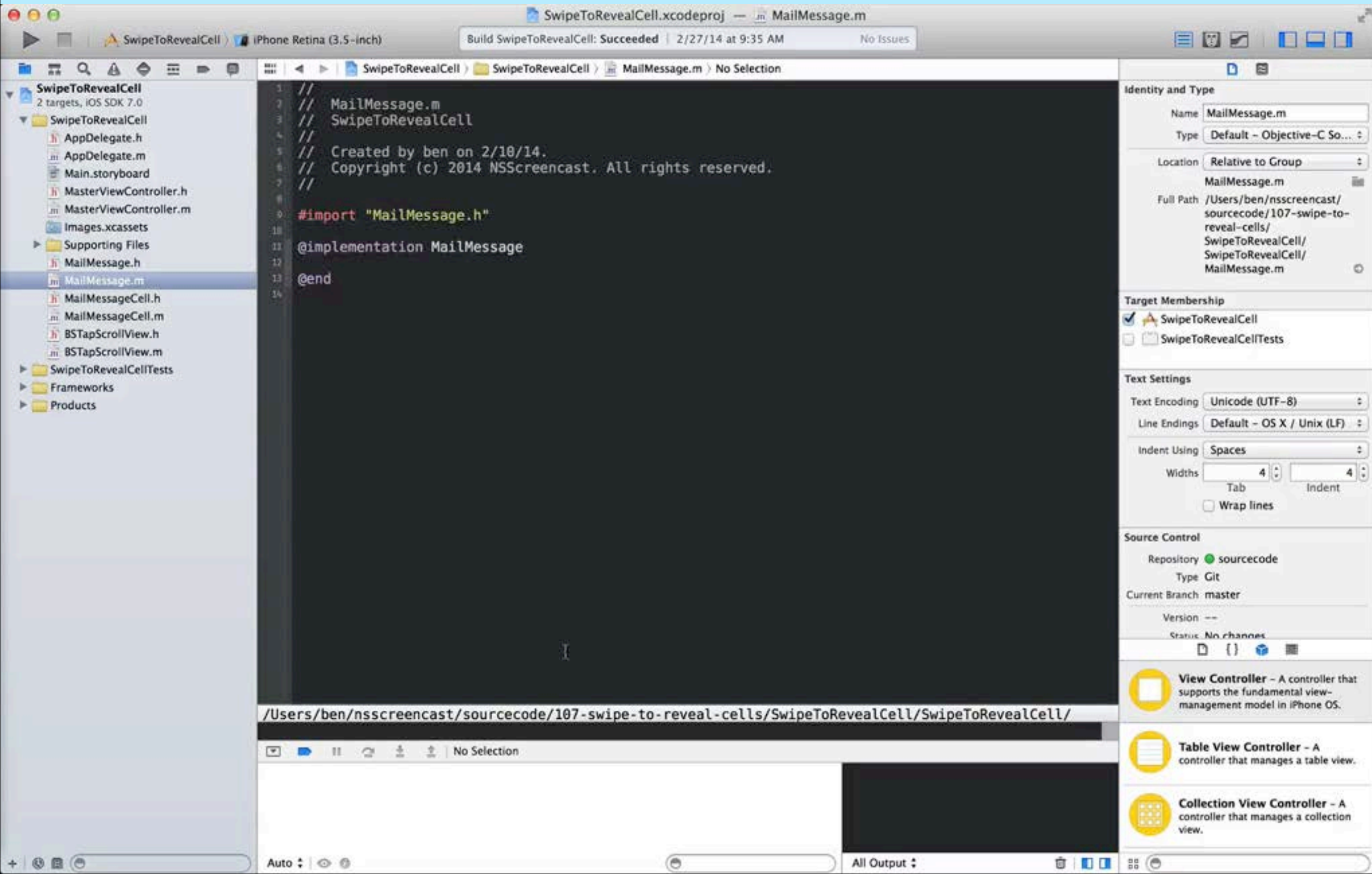
CollectionView Controller - A controller that manages a collection view.

/Users/ben/nsscreencast/sourcecode/107-swipe-to-reveal-cells/SwipeToRevealCell/SwipeToRevealCell/

No Selection

Auto

All Output





SwipeToRevealCell.xcodeproj — MailMessage.h

Build SwipeToRevealCell: Succeeded | 2/27/14 at 9:35 AM | No Issues

SwipeToRevealCell > iPhone Retina (3.5-inch)

SwipeToRevealCell > SwipeToRevealCell > MailMessage.h > No Selection

```
1 //
2 // MailMessage.h
3 // SwipeToRevealCell
4 //
5 // Created by ben on 2/10/14.
6 // Copyright (c) 2014 NSScreencast. All rights reserved.
7 //
8
9 #import <Foundation/Foundation.h>
10
11 @interface MailMessage : NSObject
12
13 @property (nonatomic, strong) NSString *subject;
14 @property (nonatomic, strong) NSString *dateString;
15 @property (nonatomic, strong) NSString *sender;
16 @property (nonatomic, strong) NSString *preview;
17
18 @end
```

Identity and Type

Name: MailMessage.h

Type: Default - C Header

Location: Relative to Group

MailMessage.h

Full Path: /Users/ben/nsscreencast/sourcecode/107-swipe-to-reveal-cells/SwipeToRevealCell/SwipeToRevealCell/MailMessage.h

Target Membership

- ☐ SwipeToRevealCell
- ☐ SwipeToRevealCellTests

Text Settings

Text Encoding: Unicode (UTF-8)

Line Endings: Default - OS X / Unix (LF)

Indent Using: Spaces

Widths: Tab: 4, Indent: 4

☐ Wrap lines

Source Control

Repository: sourcecode

Type: Git

Current Branch: master

Version: --

Status: No changes

View Controller - A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller - A controller that manages a table view.

CollectionView Controller - A controller that manages a collection view.

Auto | No Selection

All Output



SwipeToRevealCell.xcodeproj — MailMessage.h

Build SwipeToRevealCell: Succeeded | 2/27/14 at 9:35 AM | No Issues

SwipeToRevealCell > iPhone Retina (3.5-inch)

SwipeToRevealCell > SwipeToRevealCell > MailMessage.h > No Selection

SwipeToRevealCell
2 targets, iOS SDK 7.0

- SwipeToRevealCell
 - AppDelegate.h
 - AppDelegate.m
 - Main.storyboard
 - MasterViewController.h
 - MasterViewController.m
 - Images.xcassets
 - Supporting Files
 - MailMessage.h
 - MailMessage.m
 - MailMessageCell.h
 - MailMessageCell.m
 - BSTapScrollView.h
 - BSTapScrollView.m
- SwipeToRevealCellTests
- Frameworks
- Products

```
//  
// MailMessage.h  
// SwipeToRevealCell  
//  
// Created by ben on 2/10/14.  
// Copyright (c) 2014 NSScreencast. All rights reserved.  
//  
#import <Foundation/Foundation.h>  
  
@interface MailMessage : NSObject  
  
@property (nonatomic, strong) NSString *subject;  
@property (nonatomic, strong) NSString *dateString;  
@property (nonatomic, strong) NSString *sender;  
@property (nonatomic, strong) NSString *preview;  
  
@end
```

Identity and Type

Name: MailMessage.h
Type: Default - C Header
Location: Relative to Group
MailMessage.h
Full Path: /Users/ben/nsscreencast/sourcecode/107-swipe-to-reveal-cells/SwipeToRevealCell/SwipeToRevealCell/MailMessage.h

Target Membership

- ☐ SwipeToRevealCell
- ☐ SwipeToRevealCellTests

Text Settings

Text Encoding: Unicode (UTF-8)
Line Endings: Default - OS X / Unix (LF)
Indent Using: Spaces
Widths: Tab: 4, Indent: 4
☐ Wrap lines

Source Control

Repository: sourcecode
Type: Git
Current Branch: master
Version: --
Status: No changes

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

CollectionView Controller – A controller that manages a collection view.

/Users/ben/nsscreencast/sourcecode/107-swipe-to-reveal-cells/SwipeToRevealCell/SwipeToRevealCell/



SwipeToRevealCell.xcodeproj — MasterViewController.m

Running SwipeToRevealCell on iPhone Retina (3.5-inch) No Issues

SwipeToRevealCell > iPhone Retina (3.5-inch)

SwipeToRevealCell > SwipeToRevealCell > MasterViewController.m > -viewDidLoad

SwipeToRevealCell
2 targets, iOS SDK 7.0

- SwipeToRevealCell
 - AppDelegate.h
 - AppDelegate.m
 - Main.storyboard
 - MasterViewController.h
 - MasterViewController.m
- Images.xcassets
- Supporting Files
 - MailMessage.h
 - MailMessage.m
 - MailMessageCell.h
 - MailMessageCell.m
 - BSTapScrollView.h
 - BSTapScrollView.m
- SwipeToRevealCellTests
- Frameworks
- Products

```
@implementation MasterViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    if (!self.messages) {
        [self loadTestMessages];
    }
}

- (void)loadTestMessages {
    NSArray *senders = @[@"Barack Obama", @"Edward Snowden", @"Nobel Peace Prize Committee", @"Mr.
    NSArray *dateStrs = @[@"2/1/14", @"2/2/14", @"2/4/14", @"2/4/14", @"2/6/14"];
    NSArray *subjects = @[@"Dinner Invite", @"Get those files I sent?", @"Nomination", @"Need US A
    NSArray *previews = @[@"Are you free for dinner this weekend?", @"Hey did you get those files
        @"Greetings of the day....", @"We should jam soon. Call me."];

    self.messages = [NSMutableArray array];
    [senders enumerateObjectsUsingBlock:^(NSString *sender, NSUInteger idx, BOOL *stop) {
        MailMessage *msg = [[MailMessage alloc] init];
        msg.sender = sender;
        msg.dateString = dateStrs[idx];
        msg.subject = subjects[idx];
        msg.preview = previews[idx];
        [self.messages addObject:msg];
    }];
}

#pragma mark - Table View

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:(NSInteger)section {
    return self.messages.count;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    MailMessageCell *cell = [tableView dequeueReusableCellWithIdentifier:@"Cell" forIndexPath:indexPath];

    MailMessage *msg = self.messages[indexPath.row];

    cell.senderLabel.text = msg.sender;
    cell.subjectLabel.text = msg.subject;
    cell.previewLabel.text = msg.preview;

    return cell;
}
```

Identity and Type

Name: MasterViewController.m

Type: Default - Objective-C So...

Location: Relative to Group

MasterViewController.m

Full Path: /Users/ben/nsscreencast/sourcecode/107-swipe-to-reveal-cells/SwipeToRevealCell/SwipeToRevealCell/MasterViewController.m

Target Membership

- ☒ SwipeToRevealCell
- ☐ SwipeToRevealCellTests

Text Settings

Text Encoding: Default - Unicode (UTF-8)

Line Endings: Default - OS X / Unix (LF)

Indent Using: Spaces

Widths: 4 4

Tab Indent

☐ Wrap lines

Source Control

Repository: sourcecode

Type: Git

Current Branch: master

Version: --

Status: No changes

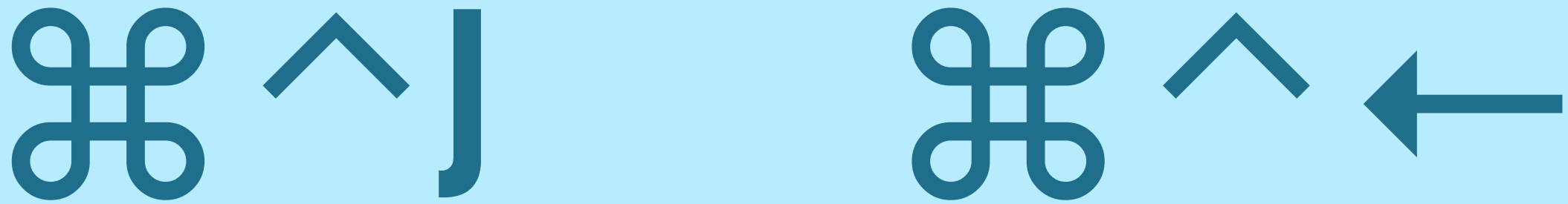
View Controller - A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller - A controller that manages a table view.

Collection View Controller - A controller that manages a collection view.

/Users/ben/nsscreencast/sourcecode/107-swipe-to-reveal-cells/SwipeToRevealCell/SwipeToRevealCell/

SwipeToRevealCell

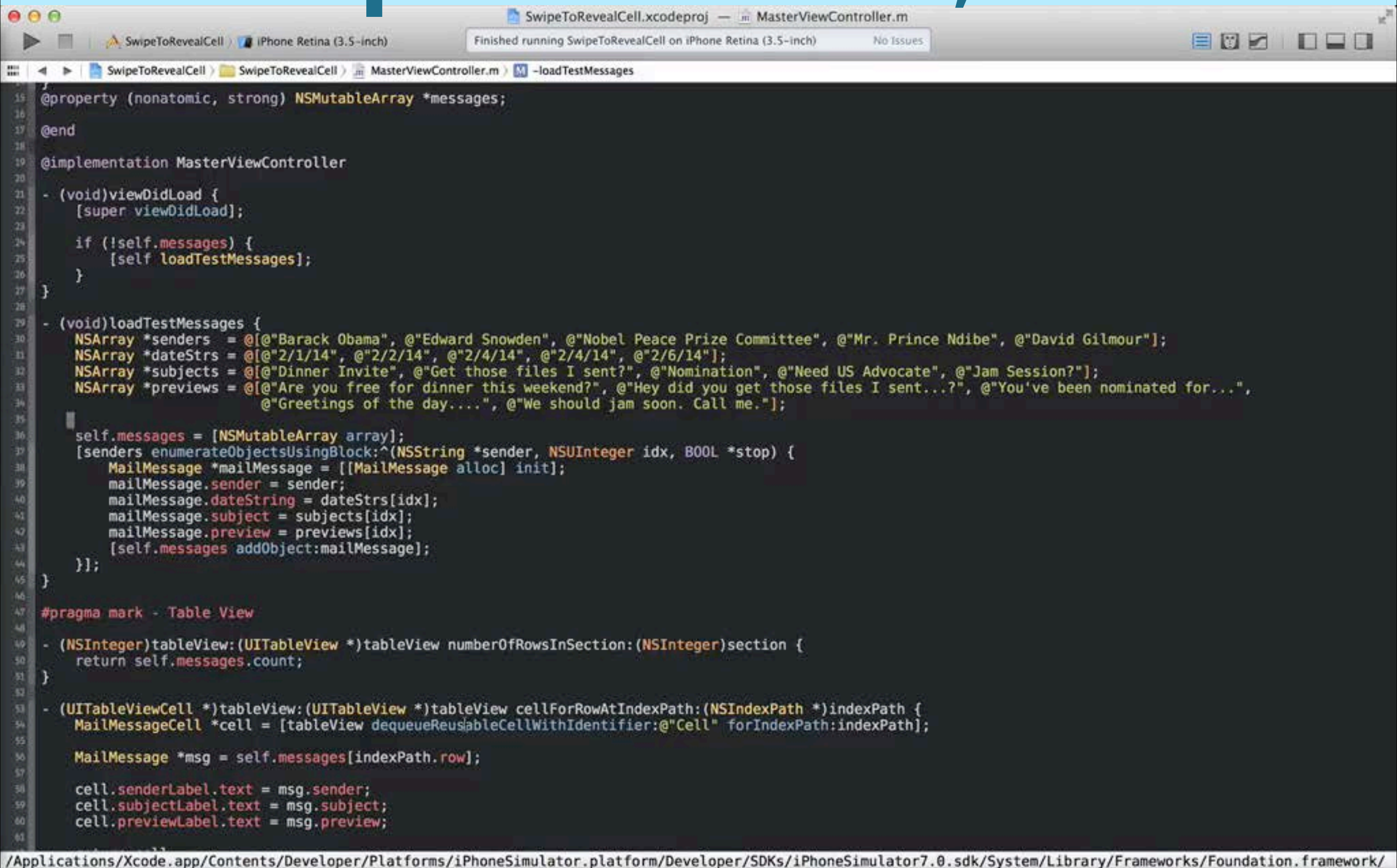


```
SwipeToRevealCell.xcodeproj — MasterViewController.m
Finished running SwipeToRevealCell on iPhone Retina (3.5-inch) No Issues

SwipeToRevealCell > iPhone Retina (3.5-inch) > MasterViewController.m > -loadTestMessages

15 @property (nonatomic, strong) NSMutableArray *messages;
16
17 @end
18
19 @implementation MasterViewController
20
21 - (void)viewDidLoad {
22     [super viewDidLoad];
23
24     if (!self.messages) {
25         [self loadTestMessages];
26     }
27 }
28
29 - (void)loadTestMessages {
30     NSArray *senders = @[@"Barack Obama", @"Edward Snowden", @"Nobel Peace Prize Committee", @"Mr. Prince Ndibe", @"David Gilmou"];
31     NSArray *dateStrs = @[@"2/1/14", @"2/2/14", @"2/4/14", @"2/4/14", @"2/6/14"];
32     NSArray *subjects = @[@"Dinner Invite", @"Get those files I sent?", @"Nomination", @"Need US Advocate", @"Jam Session?"];
33     NSArray *previews = @[@"Are you free for dinner this weekend?", @"Hey did you get those files I sent...?", @"You've been nomi"];
34     @"Greetings of the day....", @"We should jam soon. Call me."];
35
36     self.messages = [NSMutableArray array];
37     [senders enumerateObjectsUsingBlock:^(NSString *sender, NSUInteger idx, BOOL *stop) {
38         MailMessage *msg = [[MailMessage alloc] init];
39         msg.sender = sender;
40         msg.dateString = dateStrs[idx];
41         msg.subject = subjects[idx];
42         msg.preview = previews[idx];
43         [self.messages addObject:msg];
44     }];
45 }
46
47 #pragma mark - Table View
48
49 - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:(NSInteger)section {
50     return self.messages.count;
51 }
52
53 - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
54     MailMessageCell *cell = [tableView dequeueReusableCellWithIdentifier:@"Cell" forIndexPath:indexPath];
55
56     MailMessage *msg = self.messages[indexPath.row];
57
58     cell.senderLabel.text = msg.sender;
59     cell.subjectLabel.text = msg.subject;
60     cell.previewLabel.text = msg.preview;
```


Jump in Assistant, back



```
SwipeToRevealCell.xcodeproj — MasterViewController.m
Finished running SwipeToRevealCell on iPhone Retina (3.5-inch) No Issues

SwipeToRevealCell > SwipeToRevealCell > MasterViewController.m > -loadTestMessages

15 @property (nonatomic, strong) NSMutableArray *messages;
16
17 @end
18
19 @implementation MasterViewController
20
21 - (void)viewDidLoad {
22     [super viewDidLoad];
23
24     if (!self.messages) {
25         [self loadTestMessages];
26     }
27 }
28
29 - (void)loadTestMessages {
30     NSArray *senders = @[@"Barack Obama", @"Edward Snowden", @"Nobel Peace Prize Committee", @"Mr. Prince Ndibe", @"David Gilmour"];
31     NSArray *dateStrs = @[@"2/1/14", @"2/2/14", @"2/4/14", @"2/4/14", @"2/6/14"];
32     NSArray *subjects = @[@"Dinner Invite", @"Get those files I sent?", @"Nomination", @"Need US Advocate", @"Jam Session?"];
33     NSArray *previews = @[@"Are you free for dinner this weekend?", @"Hey did you get those files I sent...?", @"You've been nominated for...",
34                          @"Greetings of the day....", @"We should jam soon. Call me."];
35
36     self.messages = [NSMutableArray array];
37     [senders enumerateObjectsUsingBlock:^(NSString *sender, NSUInteger idx, BOOL *stop) {
38         MailMessage *mailMessage = [[MailMessage alloc] init];
39         mailMessage.sender = sender;
40         mailMessage.dateString = dateStrs[idx];
41         mailMessage.subject = subjects[idx];
42         mailMessage.preview = previews[idx];
43         [self.messages addObject:mailMessage];
44     }];
45 }
46
47 #pragma mark - Table View
48
49 - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:(NSInteger)section {
50     return self.messages.count;
51 }
52
53 - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
54     MailMessageCell *cell = [tableView dequeueReusableCellWithIdentifier:@"Cell" forIndexPath:indexPath];
55
56     MailMessage *msg = self.messages[indexPath.row];
57
58     cell.senderLabel.text = msg.sender;
59     cell.subjectLabel.text = msg.subject;
60     cell.previewLabel.text = msg.preview;
61 }
```

/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator7.0.sdk/System/Library/Frameworks/Foundation.framework/

⌘ ^E

(Edit all in Scope)

```
NSArray *previews = @[@"Are you free for dinner?",  
                      @"Greetings of the day."
```

```
self.messages = [NSMutableArray array];  
[senders enumerateObjectsUsingBlock:^(NSString *sender, NSUInteger idx, BOOL *stop, BOOL ^result) {  
    MailMessage *msg = [[MailMessage alloc] initWithSender:sender];  
    msg.sender = sender;  
    msg.dateString = dateStrs[idx];  
    msg.subject = subjects[idx];  
    msg.preview = previews[idx];  
    [self.messages addObject:msg];  
}];  
}
```

```
#pragma mark - Table View
```




(Re-indent selection)

```
if (!self.messages) {
    [self loadTestMessages];
}

- (void)loadTestMessages {
    NSArray *senders = @[@"Barack Obama", @"Edward Snowden", @"Nobel Peace Prize Co
    NSArray *dateStrs = @[@"2/1/14", @"2/2/14", @"2/4/14", @"2/4/14", @"2/6/14"];
    NSArray *subjects = @[@"Dinner Invite", @"Get those files I sent?", @"Nomination
    NSArray *previews = @[@"Are you free for dinner this weekend?", @"Hey did you ge
        @"Greetings of the day....", @"We should jam soon. Call me

    self.messages = [NSMutableArray array];
    [senders enumerateObjectsUsingBlock:^(NSString *sender, NSUInteger idx, BOOL *stop) {
        MailMessage *mailMessage = [[MailMessage alloc] init];
        mailMessage.sender = sender;
        mailMessage.dateString = dateStrs[idx];
        mailMessage.subject = subjects[idx];
        mailMessage.preview = previews[idx];
        [self.messages addObject:mailMessage];
    }];
}

#pragma mark - Table View

(NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)sec
```

So many to learn

Pick 2 you don't know

Deliberately Practice them

Repeat.





Snippets

Xcode / TextExpander

“pns”

@property (nonatomic, strong)

“propint”

@property (nonatomic, assign) NSInteger

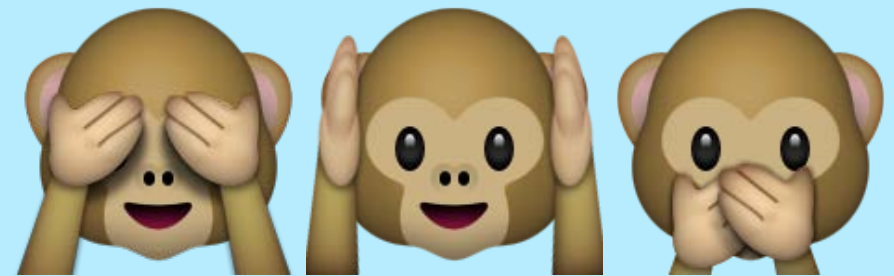
“propstr”

@property (nonatomic, copy) NSString *

“mark”

#pragma mark -

#pragma mark



#pragma mark 🍌

“tvds”

“SWf”

“tvdel”

“networkact”

```
[[UIApplication sharedApplication]  
setNetworkActivityIndicatorVisible:YES];
```

Repetitive code?
Boilerplate code?

Make a snippet.

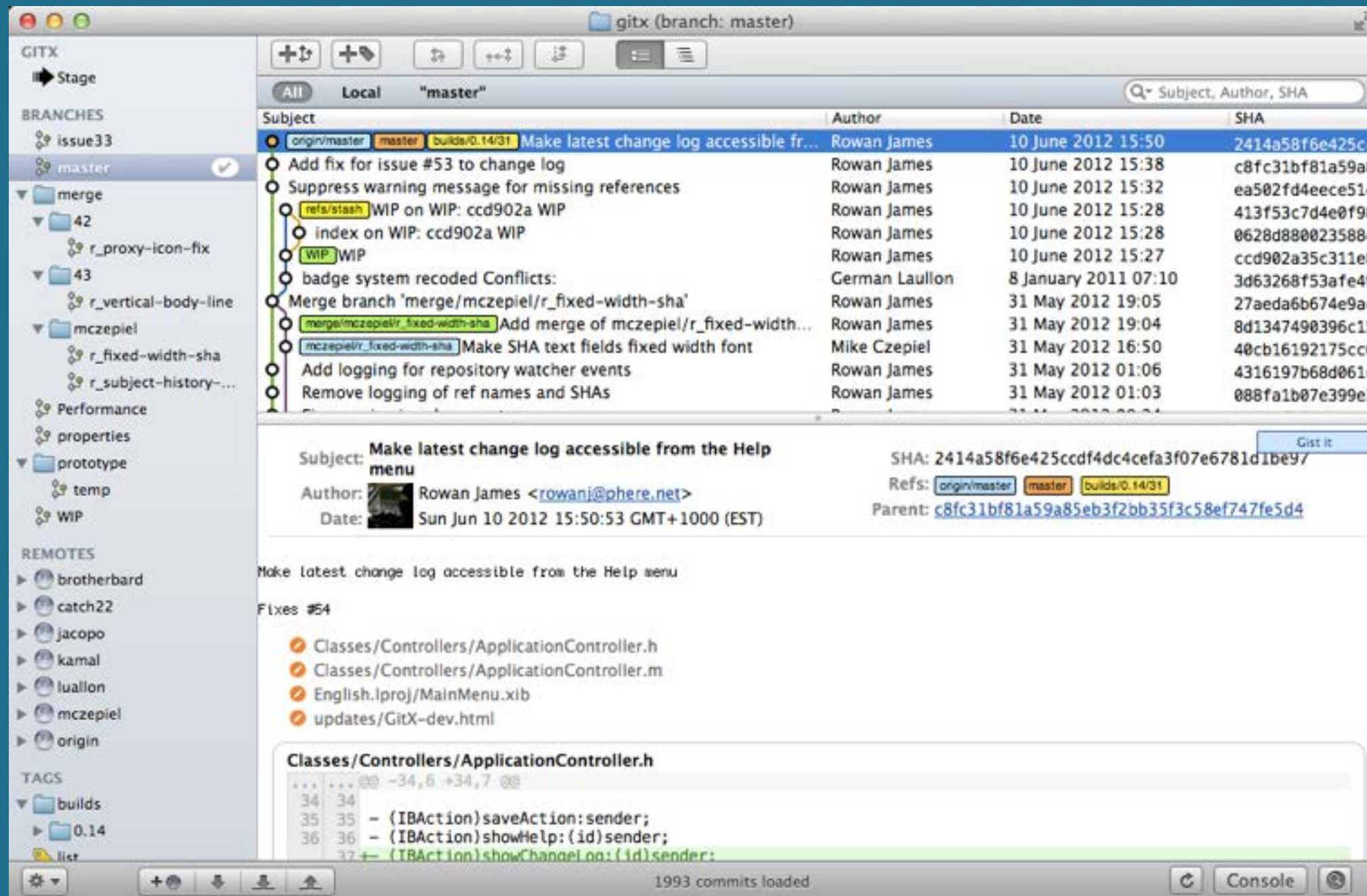
Source Control

Tips

(git)

Use the
command
line

**Leverage GUIs for
specific tasks**



GitX

<http://rowanj.github.io/gitx/>

Commits aren't just

⌘S

**Commit in logical
chunks.**

**Write a good commit
message**

Capitalized, short (50 chars or less) summary

More detailed explanatory text, if necessary. Wrap it to about 72 characters or so. In some contexts, the first line is treated as the subject of an email and the rest of the text as the body. The blank line separating the summary from the body is critical (unless you omit the body entirely); tools like rebase can get confused if you run the two together.

Write your commit message in the imperative: "Fix bug" and not "Fixed bug" or "Fixes bug." This convention matches up with commit messages generated by commands like git merge and git revert.

Further paragraphs come after blank lines.

- Bullet points are okay, too
- Typically a hyphen or asterisk is used for the bullet, followed by a single space, with blank lines in between, but conventions vary here
- Use a hanging indent

```
$ git log --oneline
```

```
9511f6c Add mixpanel gem
47a3e6d Add a sleep to not get throttled by facebook
ce30dcb Kick Jenkins build
2e6802f Merge pull request #981 from DeliRadio/venue-favorites-rabls
99eb628 stub master account to avoid fail when run in test suite.
b4e0435 before(:all) -> before(:each) to avoid certain test to fail when execute
e4b10b6 Update schema
7b1e053 Fix banner check for venues in autocomplete
83ca7cb Update ci reporter
6eae32f Fix admin manages events spec.
7c8902b Add new line at the end of file so Github likes it.
a153a48 Call Rails.application.eager_load! right before it is needed so that its
4a4d050 Refactoring test so it finishes faster.
4155ac6 fix crash when correcting country code from geolocated params
08c5bdb Proper auth token generation for facebook login:
4efaf1e Venue needs to have shareable_id and shareable_type implementation. Defa
a9cdd7a Implemented a test to check shareable_id and shareable_type. Since Favor
9e1e6c2 fix after_party task syntax
dde3ae9 add after_party task to fill in fb birthdays where we can
6e99a74 Implemented tests to check renderability of all ActiveRecord instance th
b174af9 Added four templates that are required for favoritable items.
07218a0 year_of_birth, not birth_year
be562bf update attribs from fb even for existing users
29041f0 Add logging to facebook oauth responses
```

Include declared_path in 'No podspec found..' error message for PathS...

[Browse code](#)

...ource (ref [#1825](#))

🔑 master

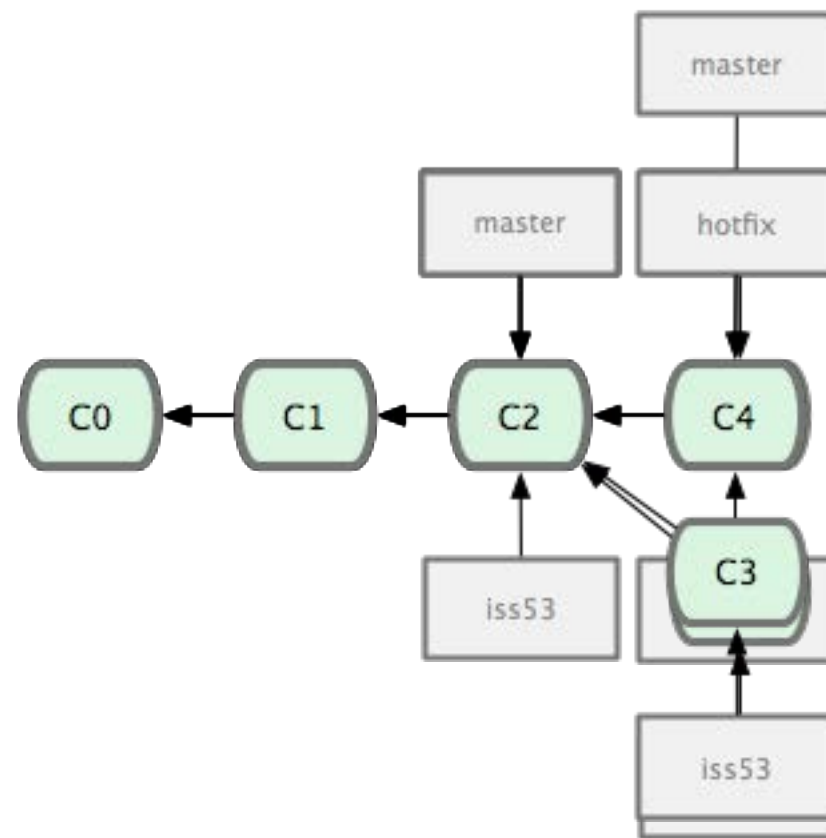


ericallam authored 15 days ago

1 parent [a3ec49e](#) commit [124ab63d3b4e93ba65826c0e84c94bcd779a6659](#)

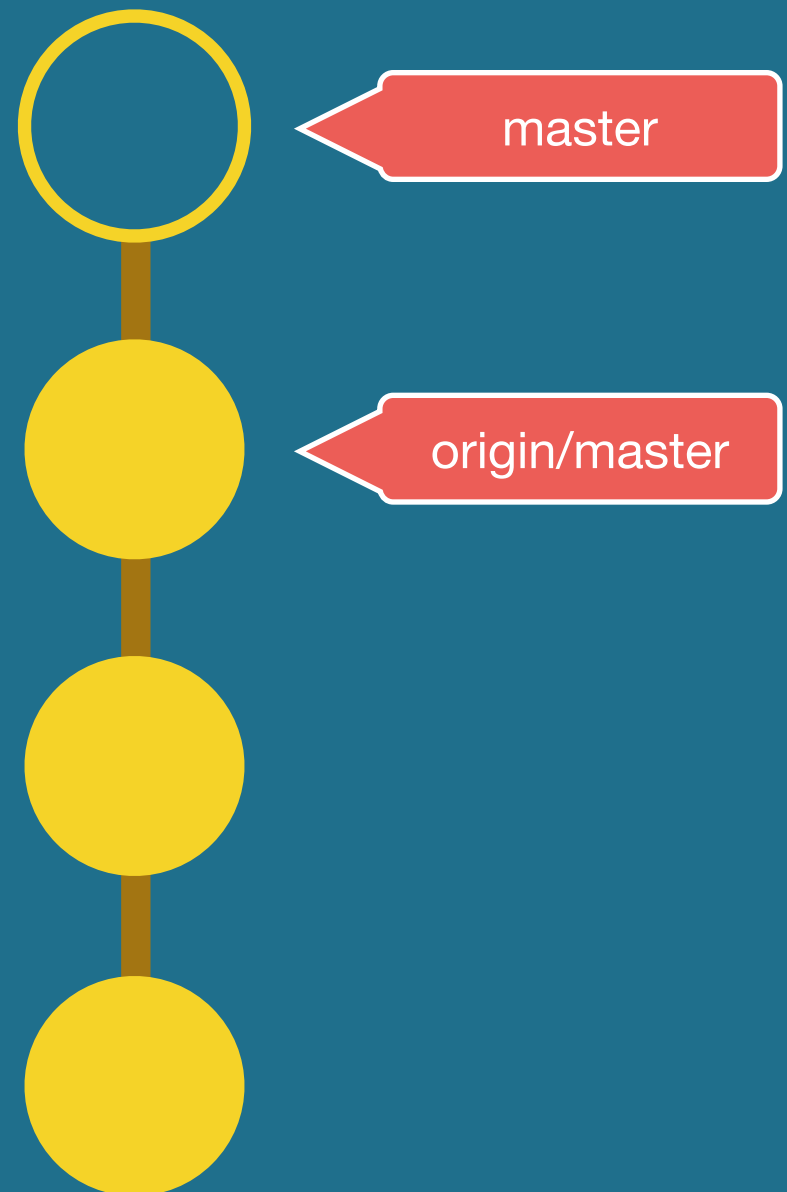


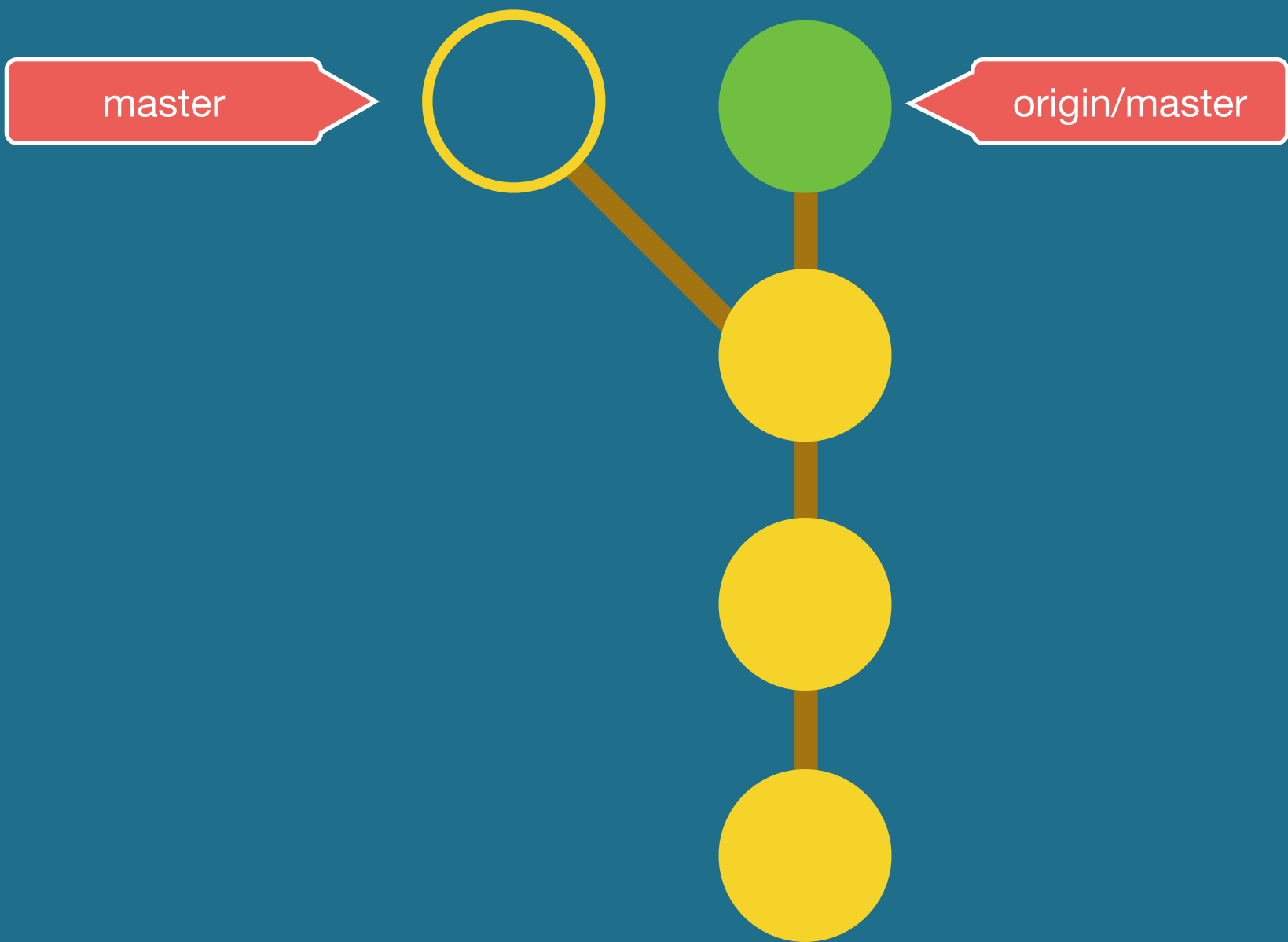
Branches are basically free

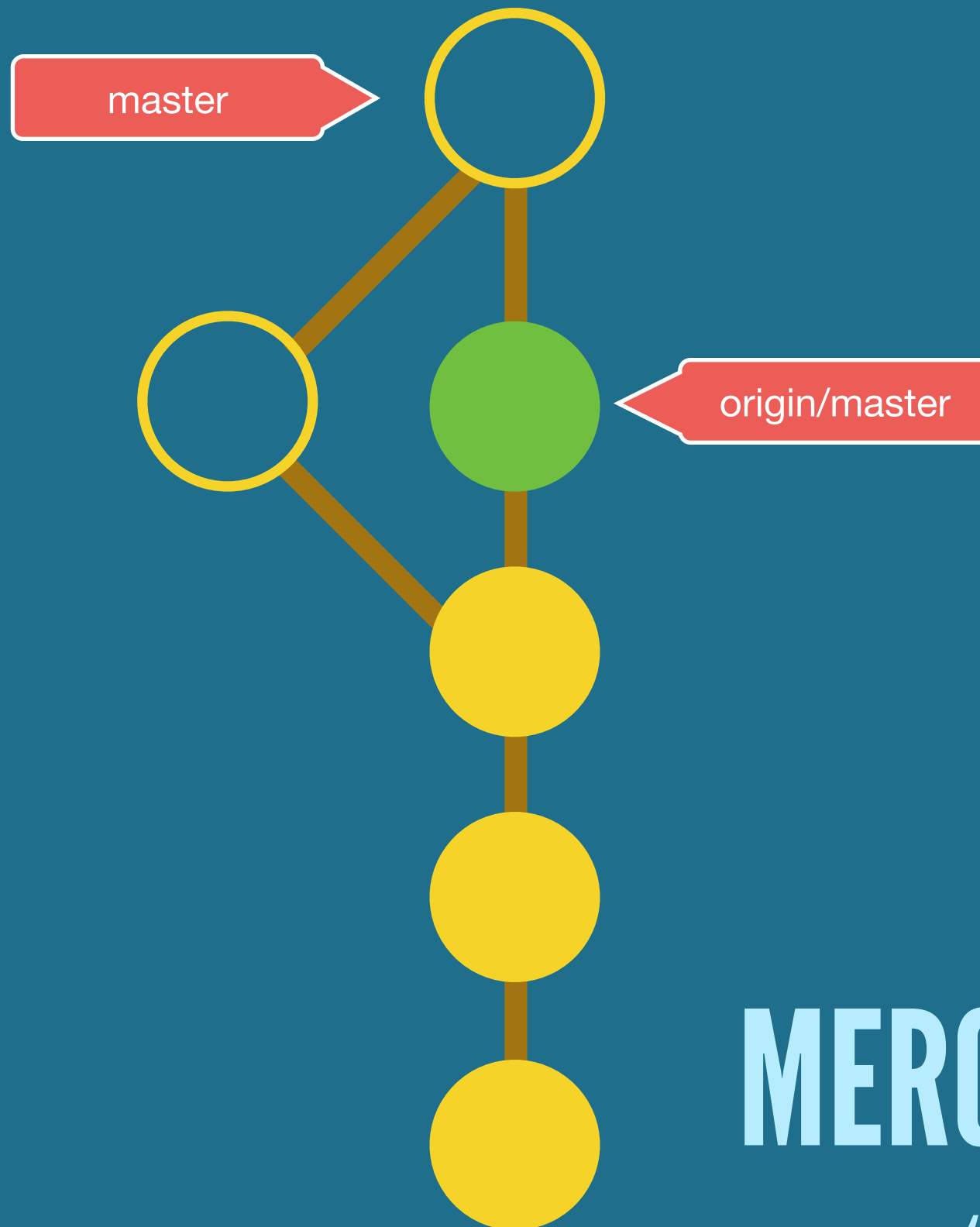


Learn to

REBASE

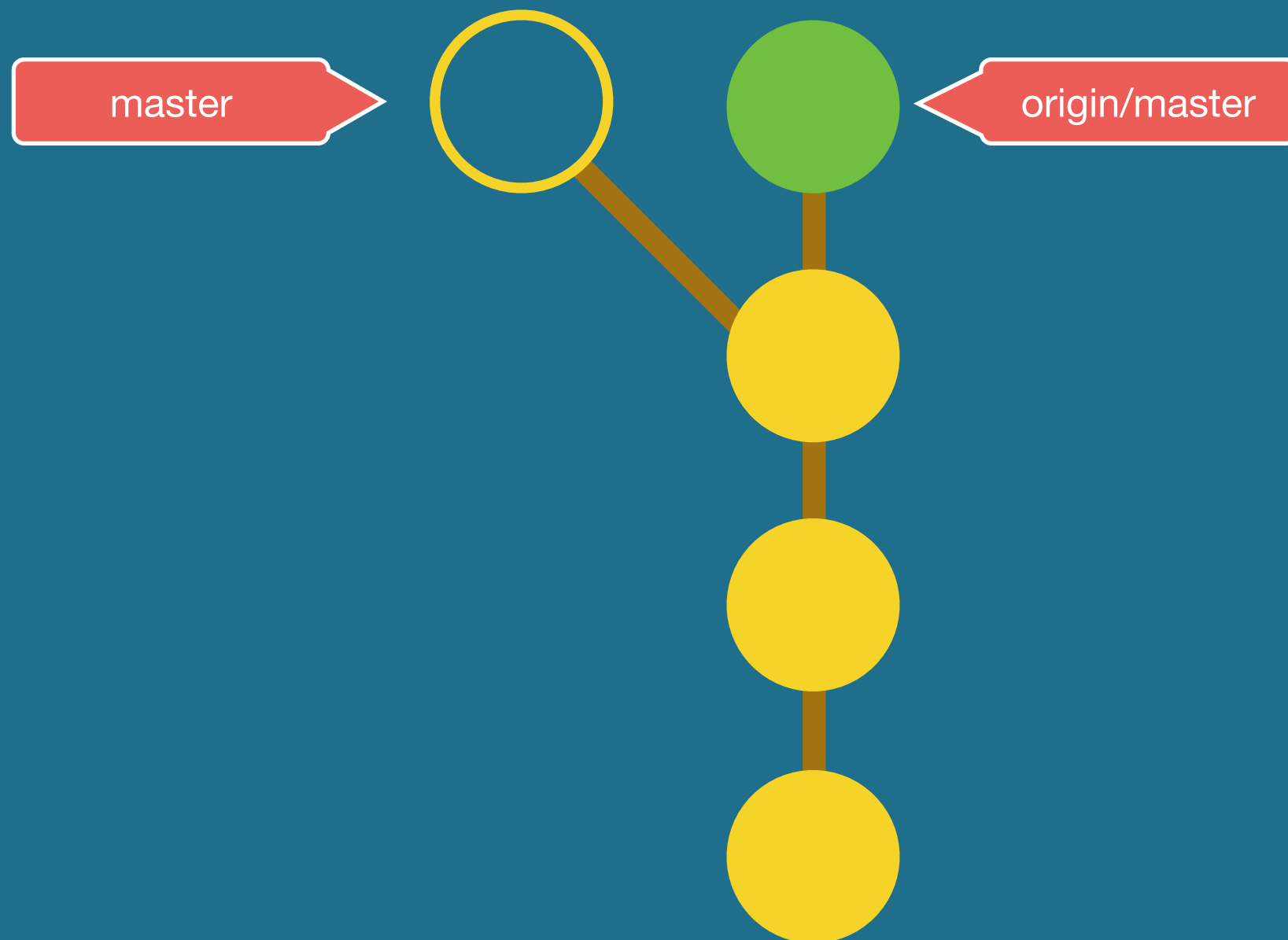


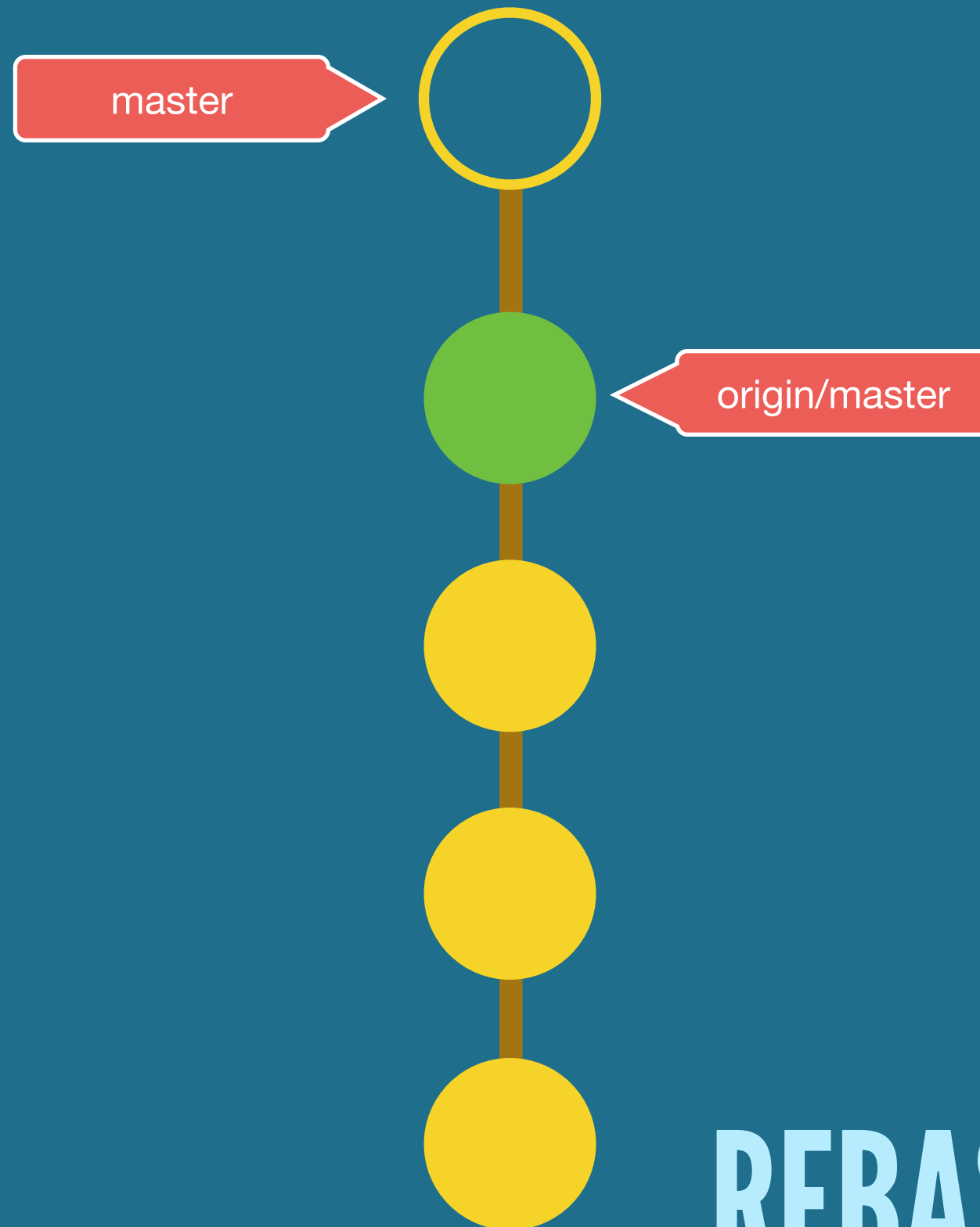




MERGE METHOD

“git pull” by default





REBASE METHOD

99% of the time you want

git pull --rebase

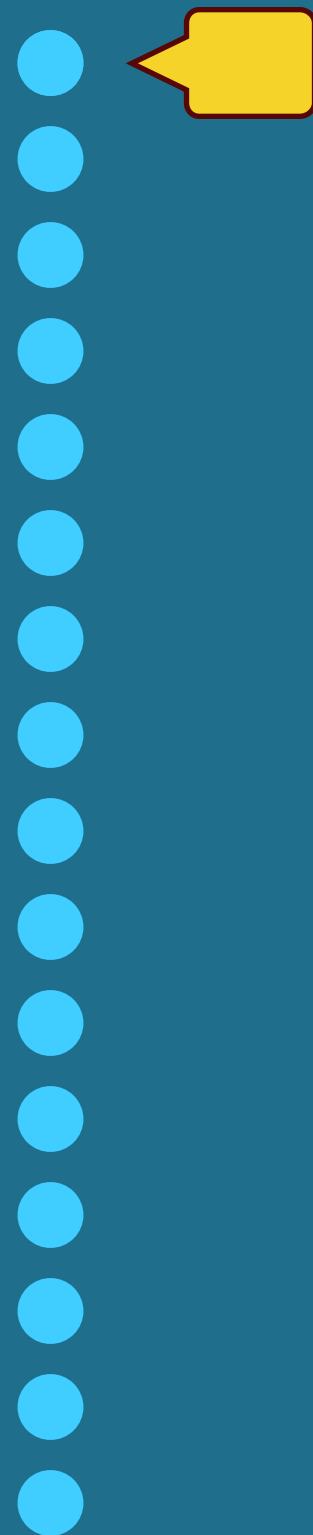
Make this the default

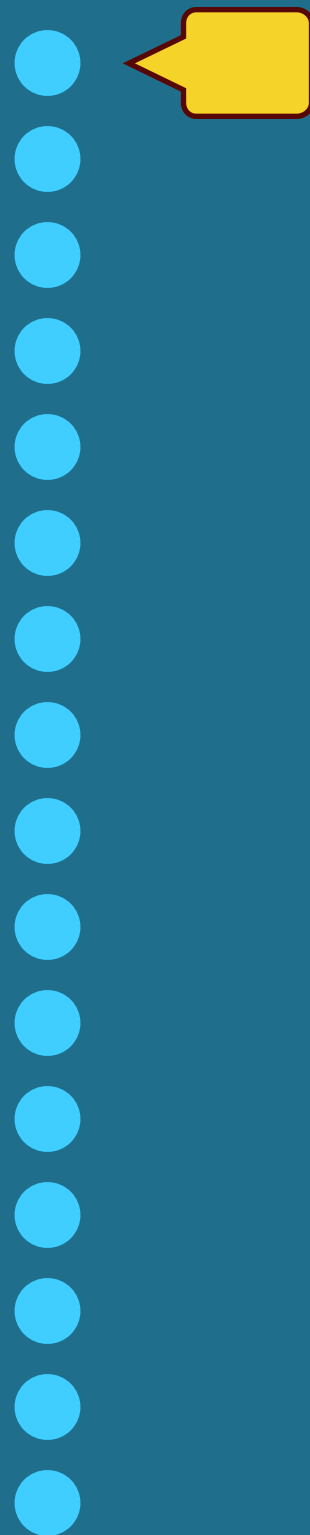
```
git config branch.autosetuprebase always
```



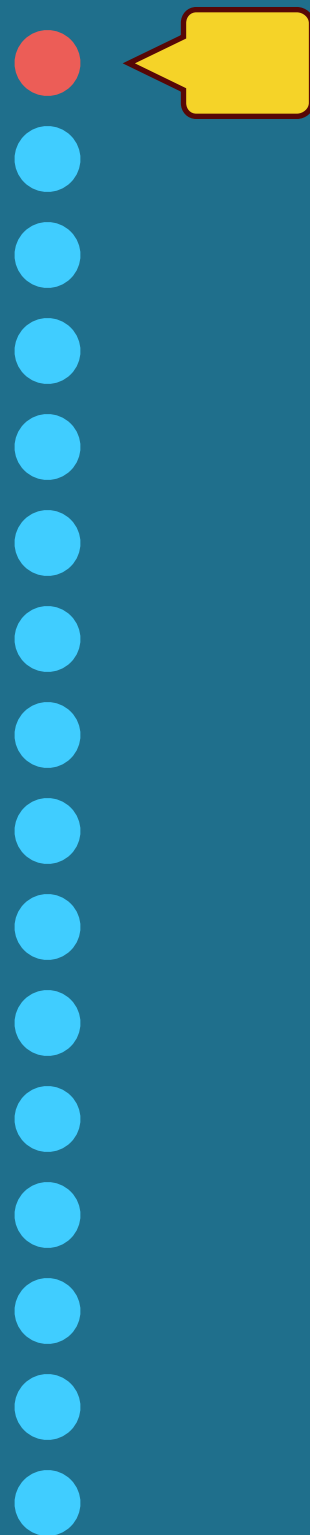
Never rebase public branches

git bisect

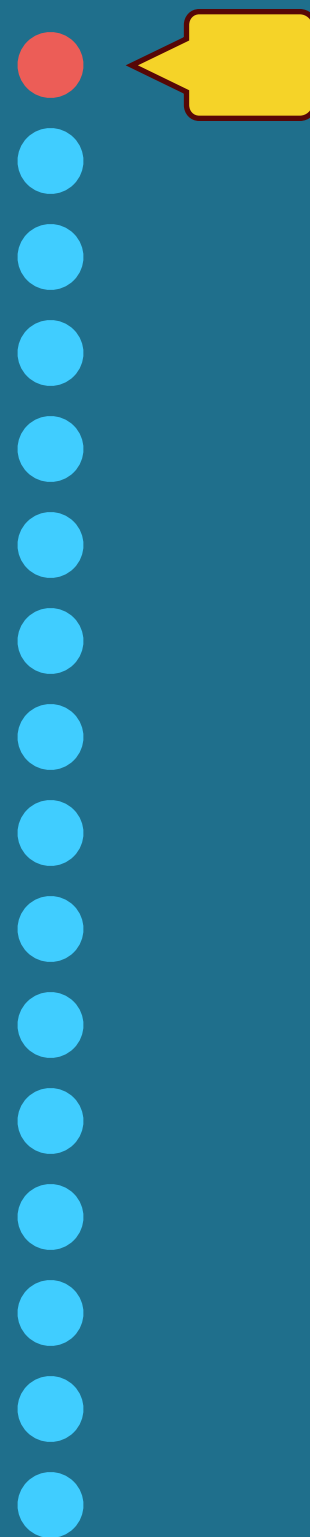




git bisect start



git bisect bad



git checkout HEAD~15



git bisect good



?





git bisect bad



git bisect bad





git bisect good



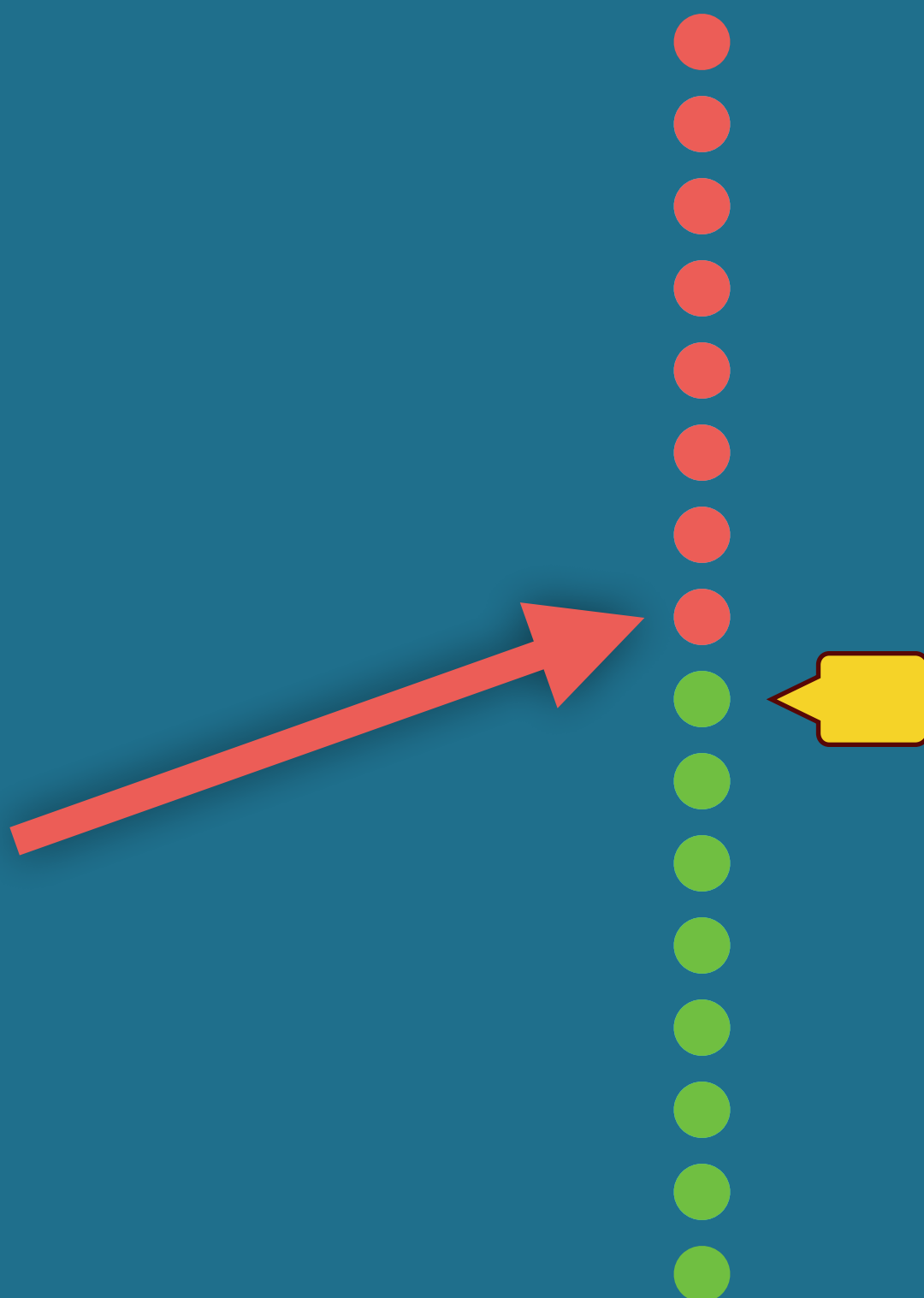


git bisect good





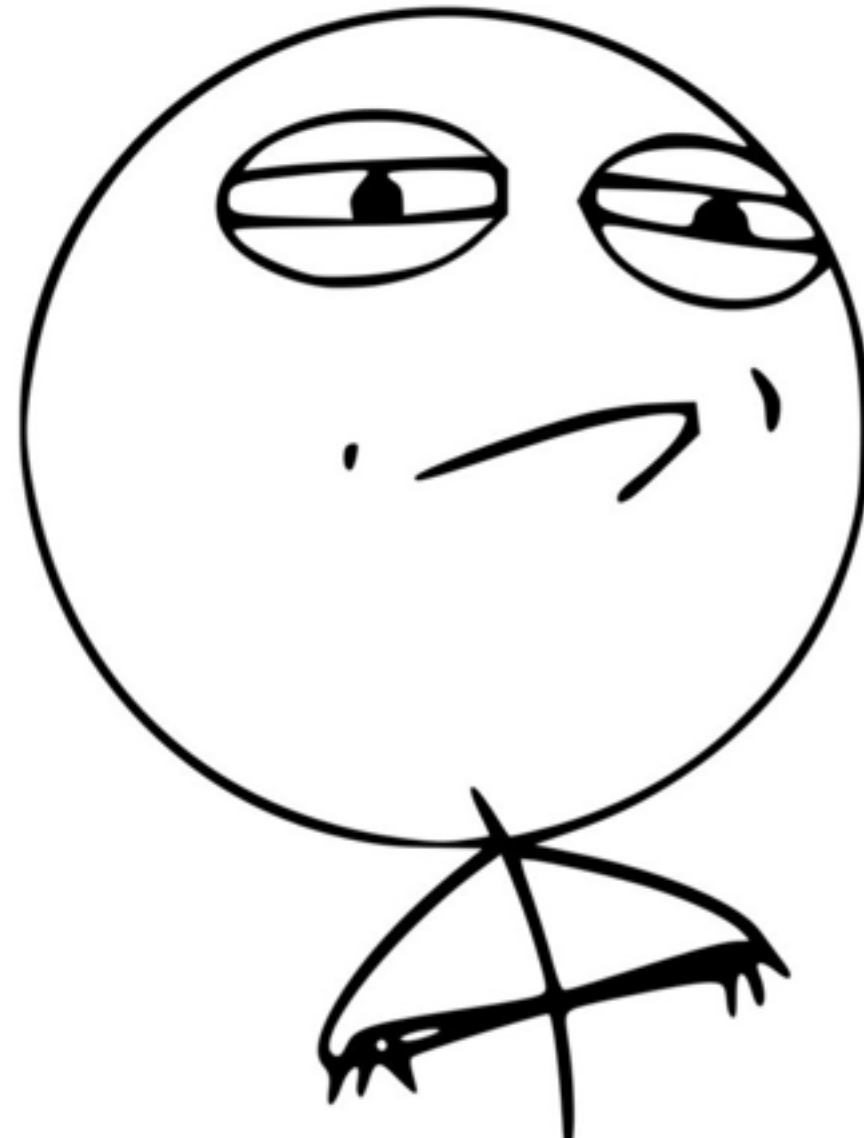
git bisect good





git bisect run <script>

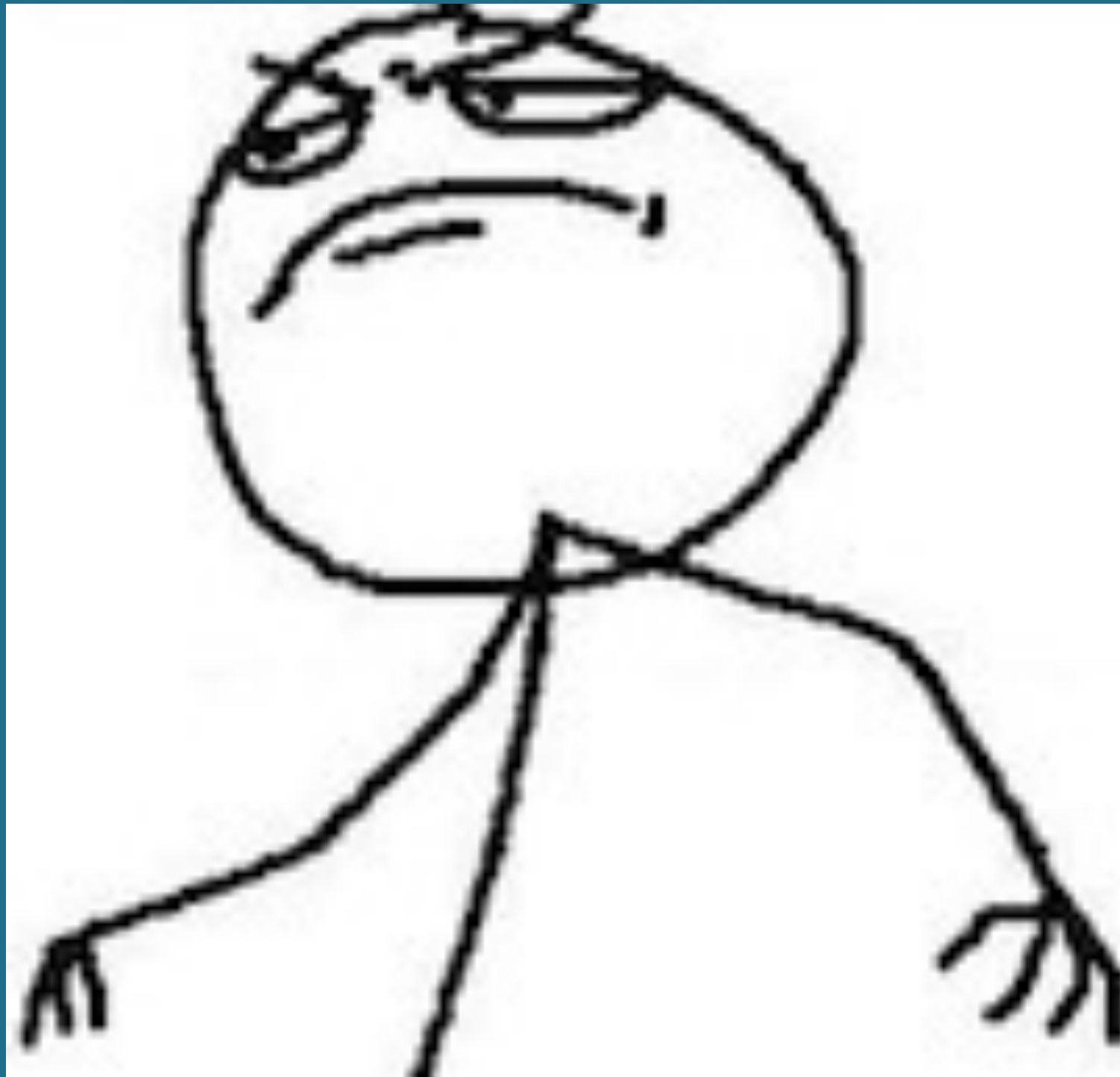
CHALLENGE ACCEPTED



Oops, I made a git mistake!

git reflog

```
jekyll master $ git reflog
6703083... HEAD@{0}: pull origin master: Fast forward
fe71d2b... HEAD@{1}: commit: Updating README with added
eac6b03... HEAD@{2}: commit: Making sure that posts fla
f682c8f... HEAD@{3}: commit: Added publish flag to post
9478f1b... HEAD@{4}: rebase: Modifying the README a bit
7e178ff... HEAD@{5}: checkout: moving from master to 7e
cda3b9e... HEAD@{6}: HEAD~1: updating HEAD
3b75036... HEAD@{7}: merge newfeature: Merge made by re
cda3b9e... HEAD@{8}: commit: Modifying the README a bit
6981921... HEAD@{9}: checkout: moving from newfeature t
7e178ff... HEAD@{10}: commit: Rewriting history is fun
4a97573... HEAD@{11}: commit: all done with TODOs
eb60603... HEAD@{12}: commit: README
```



Branch Per Feature

Keep Master Releasable

**Don't veer too far from
working software**

*“I attempt to keep the app in a working state as much as possible.
That means test driving small changes, committing often, and only
hooking things up to the UI when I think they're ready...”*

“... If I'm not in a position to push my code at least every hour or so, I feel like I haven't broken my work down properly”



Stew Gleadow





Effective

Habits

“I'm not a great programmer; I'm just a good programmer with great habits.”

- Kent Beck

Keep an improvement list

Have a side project

Read Other People's Code

“I form opinions about the code I study. Not all of it is good. I want to reflect on why I feel that way.”

Jonathan Penn



Learn how to ask for help

My client server code is not working correctly



0



I get some code of a client server program and i modify it by my own . But i could not find out why my code is not working correctly. Compilation is fine but getting an abort error in the client ? I ahve tried myself and i am new to programming. So please if you can have a look on the give client server code below and check whats the proble.

```
//Client side code

#include <iostream>
#include <winsock2.h>
#include <string>
#include <stdint.h>
#include <assert.h>
#include <time.h>
#include <stdlib.h>
#include <vector>
#include <algorithm>
#include <ctime>
#include "client.h"
#pragma comment(lib, "Ws2_32.lib")

using namespace std;

tcp_client::tcp_client()
{}

tcp_client::~~tcp_client()
{
}
```


Know someone who can answer? Share a link to this [question](#) via [email](#), [Google+](#), [Twitter](#), or [Facebook](#).

Your Answer

B *I* |     |     |  



☐ community wiki

.....

Don't ask

5

Questions.

Ask

1

Question.

Distill The Problem

**Reduce
Variables**

Also works for:



“Before asking for advice, take a careful look at what you know and what you are looking to have answered. Often that will put you in a spot where you answer your own question.”

Daniel Steinberg





Program
in
other
Languages

“Don't just program in Objective C, make sure you regularly use another language. There is a lot we can learn from other technologies and communities”

- Stew Gleadow

Write Tests.

Go Forth,
and be

Effective

Thank you

NS *Screencast*

BEN SCHEIRMAN

@subdigital

chaione.com