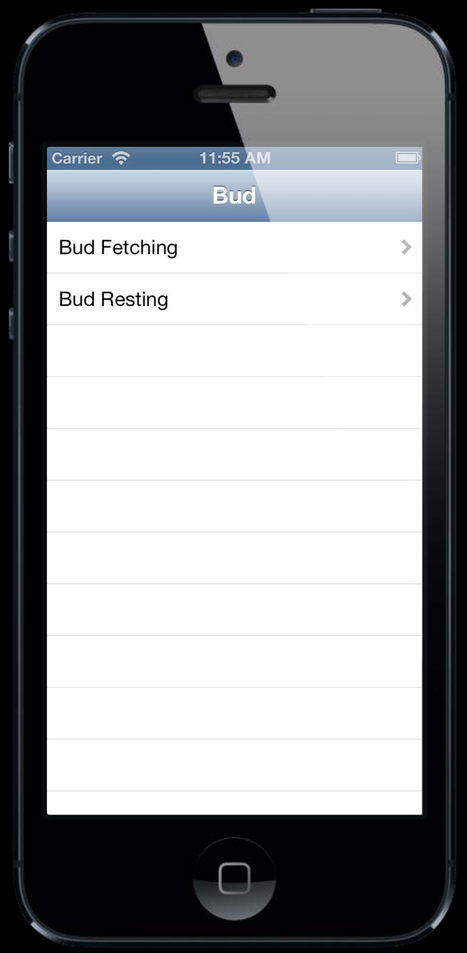
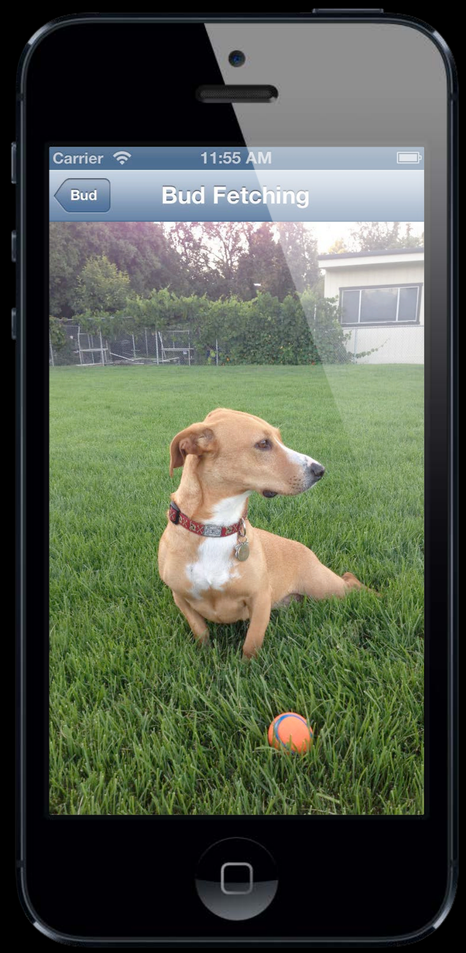
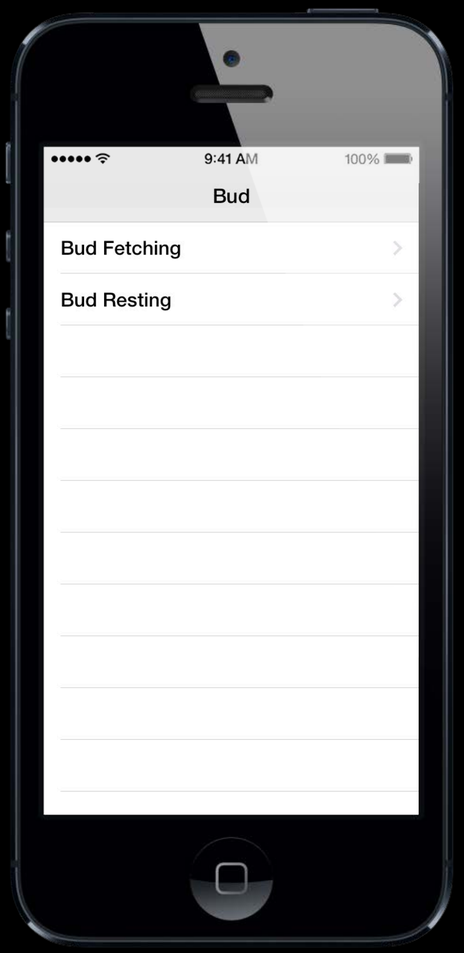
What's New in Cocoa Touch

Session 203

Chris Parker
UIKit Engineer


Carrier
11:55 AM
Bud
Bud Fetching
Bud Resting


Carrier
11:55 AM
Bud
Bud Fetching


9:41 AM
100%
Bud
Bud Fetching
Bud Resting


9:41 AM
100%
Bud
Bud Fetching

Multitasking

Multitasking

- Background fetching
- Remote notification
- Background transfers

Multitasking

Background fetching

- New background mode

  `fetch`

- Application launched opportunistically
- New delegate method on UIApplication is called

```
- (void)application:(UIApplication *)application
performFetchWithCompletionHandler:(void (^)(UIBackgroundFetchResult
result))completionHandler;
```

- Call the completion handler when fetch is complete

Multitasking

Background fetching

- New background mode

`fetch`

- Application launched opportunistically
- New delegate method on UIApplication is called

```
- (void)application:(UIApplication *)application
performFetchWithCompletionHandler:(void (^)(UIBackgroundFetchResult
result))completionHandler;
```

- Call the completion handler when fetch is complete

Multitasking

Background fetching

- Tuning the fetch interval

```
- (void)setMinimumBackgroundFetchInterval:(NSTimeInterval)minInterval;
```

- Constants

```
const NSTimeInterval UIApplicationBackgroundFetchIntervalMinimum
const NSTimeInterval UIApplicationBackgroundFetchIntervalNever
```

Multitasking

Background fetching

- Tuning the fetch interval

```
- (void)setMinimumBackgroundFetchInterval:(NSTimeInterval)minInterval;
```

- Constants

```
const NSTimeInterval UIApplicationBackgroundFetchIntervalMinimum
const NSTimeInterval UIApplicationBackgroundFetchIntervalNever
```

- You must call this at launch
- Your own values work too!

Multitasking

Remote notifications

- New background mode

```
remote-notification
```

- New delegate method on UIApplication is called

```
- (void)application:(UIApplication *)application
didReceiveRemoteNotification:(NSDictionary *)userInfo
      fetchCompletionHandler:(void (^)(UIBackgroundFetchResult
                                       result))completionHandler;
```

- Call the completion handler when fetch is complete

Multitasking

Remote notifications

- New background mode

```
remote-notification
```

- New delegate method on UIApplication is called

```
- (void)application:(UIApplication *)application
didReceiveRemoteNotification:(NSDictionary *)userInfo
      fetchCompletionHandler:(void (^)(UIBackgroundFetchResult
                                       result))completionHandler;
```

- Call the completion handler when fetch is complete

Multitasking

Fetch results

- Completion handler

```
void (^)(UIBackgroundFetchResult result))completionHandler;


typedef NS_ENUM(NSUInteger, UIBackgroundFetchResult) {
    UIBackgroundFetchResultNewData,
    UIBackgroundFetchResultNoData,
    UIBackgroundFetchResultFailed
}
```

Multitasking

Fetch results

- You call the completion handler

```
void (^)(UIBackgroundFetchResult result))completionHandler;


typedef NS_ENUM(NSUInteger, UIBackgroundFetchResult) {
    UIBackgroundFetchResultNewData,
    UIBackgroundFetchResultNoData,
    UIBackgroundFetchResultFailed
}
```

Multitasking

Background transfers

- NSURLSession
  - Replacement API for NSURLConnection
  - Data, upload, download tasks
  - Sessions have identifiers

- New delegate method on UIApplication is called

```
- (void)application:(UIApplication *)application
handleEventsForBackgroundURLSession:(NSString *)identifier
completionHandler:(void (^)())completionHandler;
```

- Call the completion handler when you're done handling callbacks

What's New in Foundation Networking	Mission Wednesday 9:00AM	

Views and Images

Image Rendering Modes

- Creating an image with a rendering mode

```
- (UIImage *)imageWithRenderingMode:(UIImageRenderingMode)renderingMode;
```

- Pass the mode

```
typedef NS_ENUM(NSInteger, UIImageRenderingMode) {
    UIImageRenderingModeAutomatic,
    UIImageRenderingModeAlwaysOriginal,
    UIImageRenderingModeAlwaysTemplate
}
```

Image Rendering Modes

- Creating an image with a rendering mode

```
- (UIImage *)imageWithRenderingMode:(UIImageRenderingMode)renderingMode;
```

- Pass the mode

```
typedef NS_ENUM(NSInteger, UIImageRenderingMode) {
    UIImageRenderingModeAutomatic,
    UIImageRenderingModeAlwaysOriginal,
    UIImageRenderingModeAlwaysTemplate
}
```

Image Rendering Modes

- Creating an image with a rendering mode

```
- (UIImage *)imageWithRenderingMode:(UIImageRenderingMode)renderingMode;
```

- Pass the mode

```
typedef NS_ENUM(NSInteger, UIImageRenderingMode) {
    UIImageRenderingModeAutomatic,
    UIImageRenderingModeAlwaysOriginal,
    UIImageRenderingModeAlwaysTemplate
}
```

Image Rendering Modes

- Creating an image with a rendering mode

```
- (UIImage *)imageWithRenderingMode:(UIImageRenderingMode)renderingMode;
```

- Pass the mode

```
typedef NS_ENUM(NSInteger, UIImageRenderingMode) {
    UIImageRenderingModeAutomatic,
    UIImageRenderingModeAlwaysOriginal,
    UIImageRenderingModeAlwaysTemplate
}
```

Tint

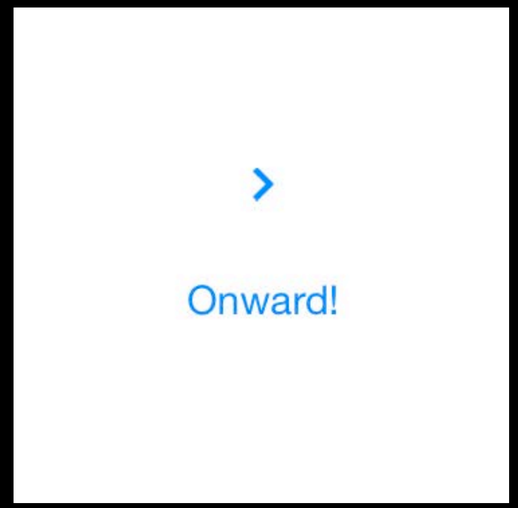
- New UIView property

```
@property (nonatomic, retain)
UIColor *tintColor;
```

Tint

- New UIView property

```
@property (nonatomic, retain)
UIColor *tintColor;
```



Tint

- New UIView property

```
@property (nonatomic, retain)
UIColor *tintColor;
```

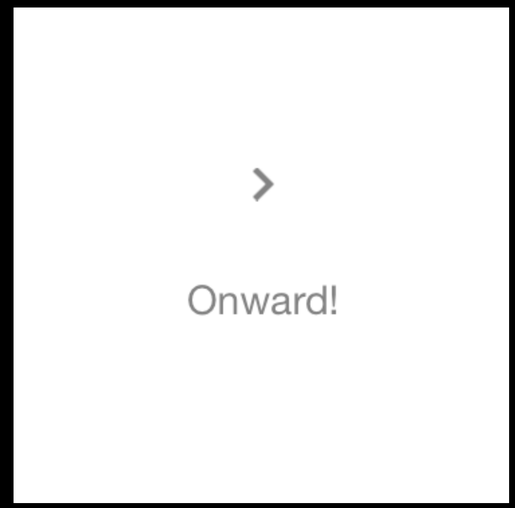
Tint

- New UIView property

```
@property (nonatomic, retain)
UIColor *tintColor;
```

- Dimming adjustment behavior

```
@property (nonatomic)
UIViewTintAdjustmentMode
tintAdjustmentMode;
```



Tint

- New UIView property

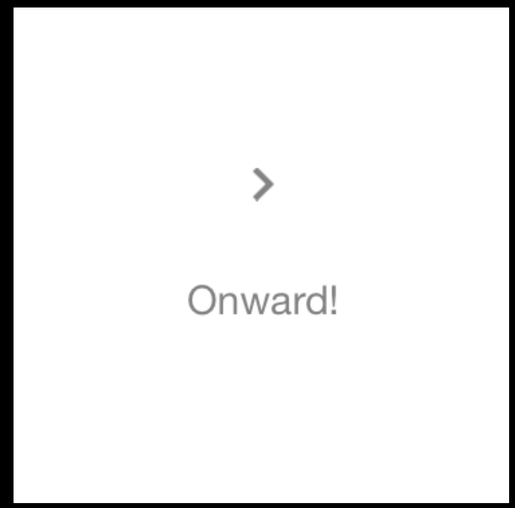
```
@property (nonatomic, retain)
UIColor *tintColor;
```

- Dimming adjustment behavior

```
@property (nonatomic)
UIViewTintAdjustmentMode
tintAdjustmentMode;
```

- Finding out about changes

```
– (void)tintColorDidChange;
```



View Animation

- No animations!

```
+ (void)performWithoutAnimation:(void (^)(void))actionsWithoutAnimation;
```

View Animation

Keyframes

```
+ (void)animateKeyframesWithDuration:(NSTimeInterval)duration
                               delay:(NSTimeInterval)delay
                             options:(UIViewKeyframeAnimationOptions)options
                          animations:(void (^)(void))animations
                          completion:(void (^)(BOOL finished))completion;
```

```
+ (void)addKeyframeWithRelativeStartTime:(double)frameStartTime
                        relativeDuration:(double)frameDuration
                              animations:(void (^)(void))animations;
```

View Animation

Keyframes

```
+ (void)animateKeyframesWithDuration:(NSTimeInterval)duration
                               delay:(NSTimeInterval)delay
                             options:(UIViewKeyframeAnimationOptions)options
                          animations:(void (^)(void))animations
                          completion:(void (^)(BOOL finished))completion;
```

```
+ (void)addKeyframeWithRelativeStartTime:(double)frameStartTime
                        relativeDuration:(double)frameDuration
                              animations:(void (^)(void))animations;
```

View Animation

Keyframe animation options

```
typedef NS_OPTIONS(NSUInteger, UIViewKeyframeAnimationOptions) {
    UIViewKeyframeAnimationOptionLayoutSubviews =
                                        UIViewAnimationOptionLayoutSubviews,
    UIViewKeyframeAnimationOptionAllowUserInteraction =
                                    UIViewAnimationOptionAllowUserInteraction,
    UIViewKeyframeAnimationOptionBeginFromCurrentState =
                                    UIViewAnimationOptionBeginFromCurrentState,
    UIViewKeyframeAnimationOptionRepeat = UIViewAnimationOptionRepeat,
    UIViewKeyframeAnimationOptionAutoreverse = UIViewAnimationOptionAutoreverse,
    UIViewKeyframeAnimationOptionOverrideInheritedDuration =
                                UIViewAnimationOptionOverrideInheritedDuration,
    UIViewKeyframeAnimationOptionCalculationModeLinear = 0 << 9,
    UIViewKeyframeAnimationOptionCalculationModeDiscrete = 1 << 9,
    UIViewKeyframeAnimationOptionCalculationModePaced = 2 << 9,
    UIViewKeyframeAnimationOptionCalculationModeCubic = 3 << 9,
    UIViewKeyframeAnimationOptionCalculationModeCubicPaced = 4 << 9
}
```

Motion Effects

- Applies relative values to keypaths of a target view
- Affected by device “pose” or position
- Affects animatable properties only

Motion Effects

UIInterpolatingMotionEffect

- Initialization

```
- (instancetype)initWithKeyPath:(NSString *)keyPath
                           type:(UIInterpolatingMotionEffectType)type;
```

- Types

```
typedef NS_ENUM(NSInteger, UIInterpolatingMotionEffectType) {
    UIInterpolatingMotionEffectTypeTiltAlongHorizontalAxis,
    UIInterpolatingMotionEffectTypeTiltAlongVerticalAxis
};
```

Motion Effects

UIInterpolatingMotionEffect

- Initialization

```
- (instancetype)initWithKeyPath:(NSString *)keyPath
                           type:(UIInterpolatingMotionEffectType)type;
```

- Types

```
typedef NS_ENUM(NSInteger, UIInterpolatingMotionEffectType) {
    UIInterpolatingMotionEffectTypeTiltAlongHorizontalAxis,
    UIInterpolatingMotionEffectTypeTiltAlongVerticalAxis
};
```

Motion Effects

UIInterpolatingMotionEffect

- Initialization

```
- (instancetype)initWithKeyPath:(NSString *)keyPath
                           type:(UIInterpolatingMotionEffectType)type;
```

- Types

```
typedef NS_ENUM(NSInteger, UIInterpolatingMotionEffectType) {
    UIInterpolatingMotionEffectTypeTiltAlongHorizontalAxis,
    UIInterpolatingMotionEffectTypeTiltAlongVerticalAxis
};
```

Motion Effects

UIInterpolatingMotionEffect

- Initialization

```
- (instancetype)initWithKeyPath:(NSString *)keyPath
                           type:(UIInterpolatingMotionEffectType)type;
```

- Types

```
typedef NS_ENUM(NSInteger, UIInterpolatingMotionEffectType) {
    UIInterpolatingMotionEffectTypeTiltAlongHorizontalAxis,
    UIInterpolatingMotionEffectTypeTiltAlongVerticalAxis
};
```

- Managing values

```
@property (retain, nonatomic) id minimumRelativeValue;
@property (retain, nonatomic) id maximumRelativeValue;
```

Motion Effects

UIMotionEffect

- Abstract superclass
- One method(!)

```
- (NSDictionary *)keyPathsAndRelativeValuesForViewerOffset:
                                                (UIOffset)viewerOffset;
```

```
typedef struct UIOffset {
    CGFloat horizontal, vertical;
} UIOffset;
```

Motion Effects

UIView

- Adding and removing

```
- (void)addMotionEffect:(UIMotionEffect *)effect;
- (void)removeMotionEffect:(UIMotionEffect *)effect;
```

- What's already there?

```
@property (copy, nonatomic) NSArray *motionEffects;
```

Motion Effects

UIView

- Adding and removing

```
- (void)addMotionEffect:(UIMotionEffect *)effect;
- (void)removeMotionEffect:(UIMotionEffect *)effect;
```

- What's already there?

```
@property (copy, nonatomic) NSArray *motionEffects;
```

Collection Views

Collection View

Transitions between layouts

```
- (void)setCollectionViewLayout:(UICollectionViewLayout *)layout
                       animated:(BOOL)animated
                     completion:(void (^)(BOOL finished))completion;
```

Collection View

UICollectionViewTransitionLayout

- Initializing

```
- (id)initWithCurrentLayout:(UICollectionViewLayout *)currentLayout
                 nextLayout:(UICollectionViewLayout *)newLayout;
```

- Managing values

```
- (void)updateValue:(CGFloat)value forAnimatedKey:(NSString *)key;
- (CGFloat)valueForAnimatedKey:(NSString *)key;
```

- Progress

```
@property (assign, nonatomic) CGFloat transitionProgress;
```

Collection View

UICollectionViewTransitionLayout

- Initializing

```
- (id)initWithCurrentLayout:(UICollectionViewLayout *)currentLayout
                 nextLayout:(UICollectionViewLayout *)newLayout;
```

- Managing values

```
- (void)updateValue:(CGFloat)value forAnimatedKey:(NSString *)key;
- (CGFloat)valueForAnimatedKey:(NSString *)key;
```

- Progress

```
@property (assign, nonatomic) CGFloat transitionProgress;
```

Collection View

UICollectionViewTransitionLayout

- Initializing

```
- (id)initWithCurrentLayout:(UICollectionViewLayout *)currentLayout
                 nextLayout:(UICollectionViewLayout *)newLayout;
```

- Managing values

```
- (void)updateValue:(CGFloat)value forAnimatedKey:(NSString *)key;
- (CGFloat)valueForAnimatedKey:(NSString *)key;
```

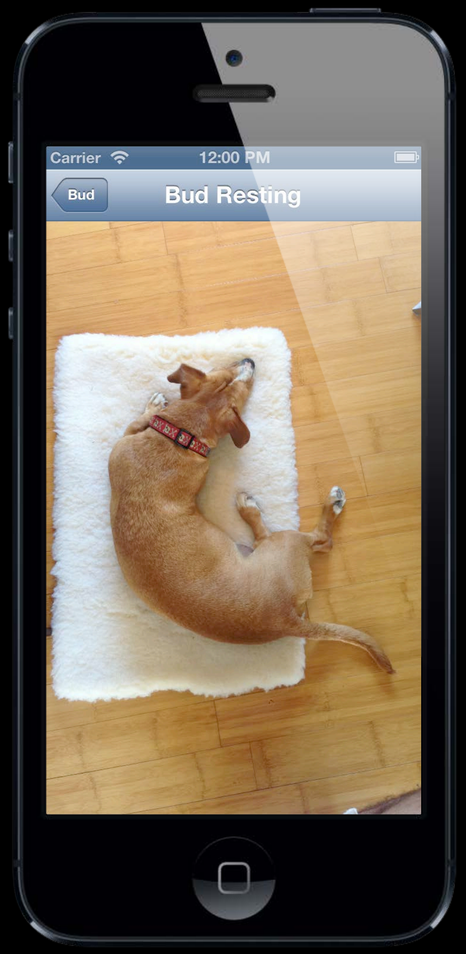
- Progress

```
@property (assign, nonatomic) CGFloat transitionProgress;
```

View Controllers

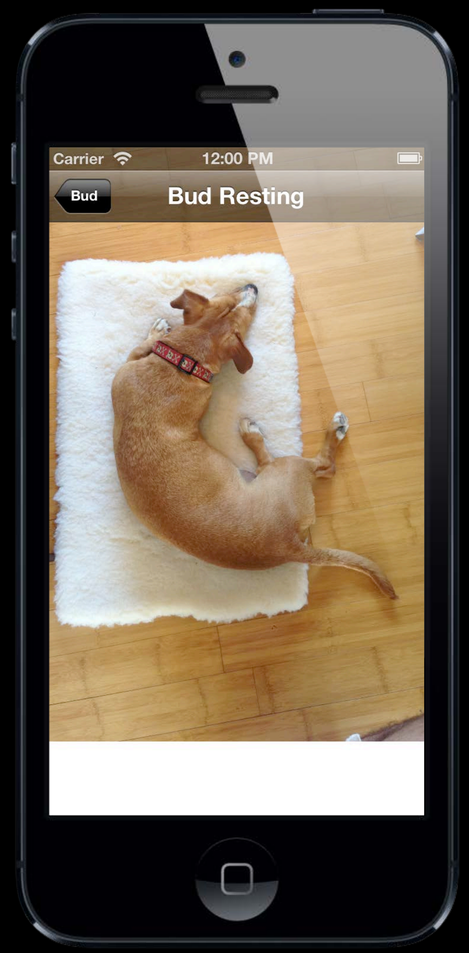
Layout

wantsFullScreenLayout



Layout

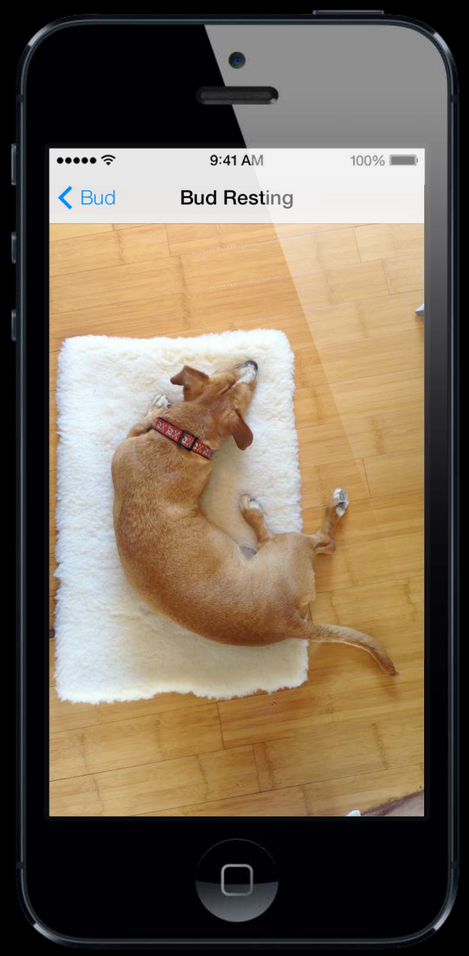
wantsFullScreenLayout



Layout

wantsFullScreenLayout

- Deprecated in iOS 7.0



Layout

Extended edges

```
typedef NS_OPTIONS(NSUInteger, UIExtendedEdge) {
    UIExtendedEdgeNone = 0,
    UIExtendedEdgeTop  = 1 << 0,
    UIExtendedEdgeLeft = 1 << 1,
    UIExtendedEdgeBottom = 1 << 2,
    UIExtendedEdgeRight  = 1 << 3,
    UIExtendedEdgeAll  = UIExtendedEdgeTop |
                         UIExtendedEdgeLeft |
                         UIExtendedEdgeBottom |
                         UIExtendedEdgeRight
};

@property(nonatomic,assign) UIExtendedEdge edgesForExtendedLayout;
@property(nonatomic,assign) BOOL extendedLayoutIncludesOpaqueBars;
@property(nonatomic,assign) BOOL automaticallyAdjustsScrollViewInsets;
```

Layout

Extended edges

```
typedef NS_OPTIONS(NSUInteger, UIExtendedEdge) {
    UIExtendedEdgeNone = 0,
    UIExtendedEdgeTop  = 1 << 0,
    UIExtendedEdgeLeft = 1 << 1,
    UIExtendedEdgeBottom = 1 << 2,
    UIExtendedEdgeRight  = 1 << 3,
    UIExtendedEdgeAll  = UIExtendedEdgeTop |
                         UIExtendedEdgeLeft |
                         UIExtendedEdgeBottom |
                         UIExtendedEdgeRight
};

@property(nonatomic,assign) UIExtendedEdge edgesForExtendedLayout;
@property(nonatomic,assign) BOOL extendedLayoutIncludesOpaqueBars;
@property(nonatomic,assign) BOOL automaticallyAdjustsScrollViewInsets
```

Layout

Extended edges

```
typedef NS_OPTIONS(NSUInteger, UIExtendedEdge) {
    UIExtendedEdgeNone = 0,
    UIExtendedEdgeTop  = 1 << 0,
    UIExtendedEdgeLeft = 1 << 1,
    UIExtendedEdgeBottom = 1 << 2,
    UIExtendedEdgeRight  = 1 << 3,
    UIExtendedEdgeAll  = UIExtendedEdgeTop |
                         UIExtendedEdgeLeft |
                         UIExtendedEdgeBottom |
                         UIExtendedEdgeRight
};

@property(nonatomic,assign) UIExtendedEdge edgesForExtendedLayout;
@property(nonatomic,assign) BOOL extendedLayoutIncludesOpaqueBars;
@property(nonatomic,assign) BOOL automaticallyAdjustsScrollViewInsets
```

Layout

Extended edges

```
typedef NS_OPTIONS(NSUInteger, UIExtendedEdge) {
    UIExtendedEdgeNone = 0,
    UIExtendedEdgeTop  = 1 << 0,
    UIExtendedEdgeLeft = 1 << 1,
    UIExtendedEdgeBottom = 1 << 2,
    UIExtendedEdgeRight  = 1 << 3,
    UIExtendedEdgeAll  = UIExtendedEdgeTop |
                         UIExtendedEdgeLeft |
                         UIExtendedEdgeBottom |
                         UIExtendedEdgeRight
};

@property(nonatomic,assign) UIExtendedEdge edgesForExtendedLayout;
@property(nonatomic,assign) BOOL extendedLayoutIncludesOpaqueBars;
@property(nonatomic,assign) BOOL automaticallyAdjustsScrollViewInsets
```

Layout

Content size

```
@property (nonatomic) CGSize preferredContentSize;
```

Layout

Status bar appearance

- New behavior for the status bar
- New status bar style

```
typedef NS_ENUM(NSInteger, UIStatusBarStyle) {
    UIStatusBarStyleDefault
    UIStatusBarStyleLightContent     NS_ENUM_AVAILABLE_IOS(7_0),
    UIStatusBarStyleBlackTranslucent,
    UIStatusBarStyleBlackOpaque
}
```

Layout

Status bar appearance

- New behavior for the status bar
- New status bar style

```
typedef NS_ENUM(NSInteger, UIStatusBarStyle) {
    UIStatusBarStyleDefault
    UIStatusBarStyleLightContent     NS_ENUM_AVAILABLE_IOS(7_0),
    UIStatusBarStyleBlackTranslucent,
    UIStatusBarStyleBlackOpaque
}
```

Layout

Status bar appearance

- New behavior for the status bar
- New status bar style

```
typedef NS_ENUM(NSInteger, UIStatusBarStyle) {
    UIStatusBarStyleDefault
    UIStatusBarStyleLightContent     NS_ENUM_AVAILABLE_IOS(7_0),
    UIStatusBarStyleBlackTranslucent,
    UIStatusBarStyleBlackOpaque
}
```

Layout

Status bar appearance

- New behavior for the status bar
- New status bar style

```
typedef NS_ENUM(NSInteger, UIStatusBarStyle) {
    UIStatusBarStyleDefault
    UIStatusBarStyleLightContent     NS_ENUM_AVAILABLE_IOS(7_0),
    UIStatusBarStyleBlackTranslucent,
    UIStatusBarStyleBlackOpaque
}
```

Custom Transitions

- Bounded, "canned" transitions
  - Navigation controller push and pop
  - Presentation and dismissal
- Interactive, user-driven transitions
  - Driven by gestures or other events
- New delegate method on UIViewController

```
@property (nonatomic,retain) id <UIViewControllerTransitioningDelegate>
transitioningDelegate;
```

Custom Transitions

UIViewControllerTransitioningDelegate

```
@protocol UIViewControllerTransitioningDelegate <NSObject>
@optional
- (id <UIViewControllerAnimatedTransitioning>)
    animationControllerForPresentedController:(UIViewController *)presented
                         presentingController:(UIViewController *)presenting
                             sourceController:(UIViewController *)source;

- (id <UIViewControllerAnimatedTransitioning>)
                 animationControllerForDismissedController:(UIViewController *)dismissed;

- (id <UIViewControllerInteractiveTransitioning>)interactionControllerForPresentation:
                                    (id <UIViewControllerAnimatedTransitioning>)animator;

- (id <UIViewControllerInteractiveTransitioning>)interactionControllerForDismissal:
                                    (id <UIViewControllerAnimatedTransitioning>)animator;
@end
```

Custom Transitions

UIViewControllerTransitioningDelegate

```
@protocol UIViewControllerTransitioningDelegate <NSObject>
@optional
- (id <UIViewControllerAnimatedTransitioning>)
    animationControllerForPresentedController:(UIViewController *)presented
                         presentingController:(UIViewController *)presenting
                             sourceController:(UIViewController *)source;

- (id <UIViewControllerAnimatedTransitioning>)
                 animationControllerForDismissedController:(UIViewController *)dismissed;

- (id <UIViewControllerInteractiveTransitioning>)interactionControllerForPresentation:
                                    (id <UIViewControllerAnimatedTransitioning>)animator;

- (id <UIViewControllerInteractiveTransitioning>)interactionControllerForDismissal:
                                    (id <UIViewControllerAnimatedTransitioning>)animator;
@end
```

Custom Transitions

UIViewControllerTransitioningDelegate

```
@protocol UIViewControllerTransitioningDelegate <NSObject>
@optional
- (id <UIViewControllerAnimatedTransitioning>)
    animationControllerForPresentedController:(UIViewController *)presented
                         presentingController:(UIViewController *)presenting
                             sourceController:(UIViewController *)source;

- (id <UIViewControllerAnimatedTransitioning>)
                 animationControllerForDismissedController:(UIViewController *)dismissed;

- (id <UIViewControllerInteractiveTransitioning>)interactionControllerForPresentation:
                                    (id <UIViewControllerAnimatedTransitioning>)animator;

- (id <UIViewControllerInteractiveTransitioning>)interactionControllerForDismissal:
                                    (id <UIViewControllerAnimatedTransitioning>)animator;
@end
```

Custom Transitions

UIViewControllerTransitioningDelegate

```
@protocol UIViewControllerTransitioningDelegate <NSObject>
@optional
- (id <UIViewControllerAnimatedTransitioning>)
    animationControllerForPresentedController:(UIViewController *)presented
                         presentingController:(UIViewController *)presenting
                             sourceController:(UIViewController *)source;

- (id <UIViewControllerAnimatedTransitioning>)
                 animationControllerForDismissedController:(UIViewController *)dismissed;

- (id <UIViewControllerInteractiveTransitioning>)interactionControllerForPresentation:
                                    (id <UIViewControllerAnimatedTransitioning>)animator;

- (id <UIViewControllerInteractiveTransitioning>)interactionControllerForDismissal:
                                    (id <UIViewControllerAnimatedTransitioning>)animator;
@end
```

Custom Transitions

UIViewControllerAnimatedTransitioning

```
@protocol UIViewControllerAnimatedTransitioning <NSObject>

- (NSTimeInterval)transitionDuration:
                (id <UIViewControllerContextTransitioning>)transitionContext;

- (void)animateTransition:
                (id <UIViewControllerContextTransitioning>)transitionContext;

@optional
- (void)animationEnded:(BOOL) transitionCompleted;

@end
```

Custom Transitions

UIViewControllerInteractiveTransitioning

```
@protocol UIViewControllerInteractiveTransitioning <NSObject>

- (void)startInteractiveTransition:
                (id <UIViewControllerContextTransitioning>)transitionContext;

@optional
- (CGFloat)completionSpeed;
- (UIViewAnimationCurve)completionCurve;

@end
```

Custom Transitions

UIViewControllerContextTransitioning

```
@protocol UIViewControllerContextTransitioning <NSObject>
- (UIView *)containerView;
- (BOOL)isAnimated;
- (BOOL)isInteractive;
- (BOOL)transitionWasCancelled;
- (UIModalPresentationStyle)presentationStyle;
- (void)updateInteractiveTransition:(CGFloat)percentComplete;
- (void)finishInteractiveTransition;
- (void)cancelInteractiveTransition;
- (void)completeTransition:(BOOL)didComplete;
- (UIViewController *)viewControllerForKey:(NSString *)key;
- (CGRect)initialFrameForViewController:(UIViewController *)vc;
- (CGRect)finalFrameForViewController:(UIViewController *)vc;
@end
```

Custom Transitions

UIViewControllerContextTransitioning

```
@protocol UIViewControllerContextTransitioning <NSObject>
- (UIView *)containerView;
- (BOOL)isAnimated;
- (BOOL)isInteractive;
- (BOOL)transitionWasCancelled;
- (UIModalPresentationStyle)presentationStyle;
- (void)updateInteractiveTransition:(CGFloat)percentComplete;
- (void)finishInteractiveTransition;
- (void)cancelInteractiveTransition;
- (void)completeTransition:(BOOL)didComplete;
- (UIViewController *)viewControllerForKey:(NSString *)key;
- (CGRect)initialFrameForViewController:(UIViewController *)vc;
- (CGRect)finalFrameForViewController:(UIViewController *)vc;
@end
```

Custom Transitions

UIViewControllerContextTransitioning

```
@protocol UIViewControllerContextTransitioning <NSObject>
- (UIView *)containerView;
- (BOOL)isAnimated;
- (BOOL)isInteractive;
- (BOOL)transitionWasCancelled;
- (UIModalPresentationStyle)presentationStyle;
- (void)updateInteractiveTransition:(CGFloat)percentComplete;
- (void)finishInteractiveTransition;
- (void)cancelInteractiveTransition;
- (void)completeTransition:(BOOL)didComplete;
- (UIViewController *)viewControllerForKey:(NSString *)key;
- (CGRect)initialFrameForViewController:(UIViewController *)vc;
- (CGRect)finalFrameForViewController:(UIViewController *)vc;
@end
```

Custom Transitions

UIViewControllerContextTransitioning

```
@protocol UIViewControllerContextTransitioning <NSObject>
- (UIView *)containerView;
- (BOOL)isAnimated;
- (BOOL)isInteractive;
- (BOOL)transitionWasCancelled;
- (UIModalPresentationStyle)presentationStyle;
- (void)updateInteractiveTransition:(CGFloat)percentComplete;
- (void)finishInteractiveTransition;
- (void)cancelInteractiveTransition;
- (void)completeTransition:(BOOL)didComplete;
- (UIViewController *)viewControllerForKey:(NSString *)key;
- (CGRect)initialFrameForViewController:(UIViewController *)vc;
- (CGRect)finalFrameForViewController:(UIViewController *)vc;
@end
```

Custom Transitions

UIViewControllerContextTransitioning

```
@protocol UIViewControllerContextTransitioning <NSObject>
- (UIView *)containerView;
- (BOOL)isAnimated;
- (BOOL)isInteractive;
- (BOOL)transitionWasCancelled;
- (UIModalPresentationStyle)presentationStyle;
- (void)updateInteractiveTransition:(CGFloat)percentComplete;
- (void)finishInteractiveTransition;
- (void)cancelInteractiveTransition;
- (void)completeTransition:(BOOL)didComplete;
- (UIViewController *)viewControllerForKey:(NSString *)key;
- (CGRect)initialFrameForViewController:(UIViewController *)vc;
- (CGRect)finalFrameForViewController:(UIViewController *)vc;
@end
```

Custom Transitions Using View Controllers	Pacific Heights Thursday 11:30AM	

State Restoration

Ignoring Snapshots

- Called from methods invoked by state restoration:

```
- (void)ignoreSnapshotOnNextApplicationLaunch;
```

Other Objects

- Non-view and non-view controller objects can now participate

```
+ (void)registerObjectForStateRestoration:(id<UIStateRestoring>)object
        restorationIdentifier:(NSString *)restorationIdentifier;
```

Bluetooth State Restoration

- Launch options keys

```
UIKIT_EXTERN NSString *const
UIApplicationLaunchOptionsBluetoothCentralsKey;

UIKIT_EXTERN NSString *const
UIApplicationLaunchOptionsBluetoothPeripheralsKey;
```

Bluetooth State Restoration

- Launch options keys

```
UIKIT_EXTERN NSString *const
UIApplicationLaunchOptionsBluetoothCentralsKey;

UIKIT_EXTERN NSString *const
UIApplicationLaunchOptionsBluetoothPeripheralsKey;
```

AirDrop

AirDrop

- Adopt UIActivityItemSourceProtocol
- Update application's Info.plist to create, register, and export UTI for custom document formats
- New Documents/Inbox directory
  - Check this on app activations
  - …even when you didn't get an -application:openURL:sourceApplication:annotation: call
- May be launched multiple times in quick succession
  - Might want to queue work up

Dynamics

Dynamics

- Fluid, responsive animations
- Enhances the interactions in your application
- Concentration on behaviors

Dynamics

UIDynamicAnimator

```
@interface UIDynamicAnimator: NSObject
- (instancetype)initWithReferenceView:(UIView*)view;

- (void)addBehavior:(UIDynamicBehavior *)behavior;
- (void)removeBehavior:(UIDynamicBehavior *)behavior;
- (void)removeAllBehaviors;

@property (nonatomic, readonly) UIView* referenceView;
@property (nonatomic, readonly, copy) NSArray* behaviors;

- (NSArray*)itemsInRect:(CGRect)rect;
@property (nonatomic, readonly, getter = isRunning) BOOL running;
- (NSTimeInterval)elapsedTime;

@property (nonatomic, assign) id <UIDynamicAnimatorDelegate> delegate;
@end
```

Dynamics

UIDynamicAnimator

```
@interface UIDynamicAnimator: NSObject
- (instancetype)initWithReferenceView:(UIView*)view;

- (void)addBehavior:(UIDynamicBehavior *)behavior;
- (void)removeBehavior:(UIDynamicBehavior *)behavior;
- (void)removeAllBehaviors;

@property (nonatomic, readonly) UIView* referenceView;
@property (nonatomic, readonly, copy) NSArray* behaviors;

- (NSArray*)itemsInRect:(CGRect)rect;
@property (nonatomic, readonly, getter = isRunning) BOOL running;
- (NSTimeInterval)elapsedTime;

@property (nonatomic, assign) id <UIDynamicAnimatorDelegate> delegate;
@end
```

Dynamics

UIDynamicAnimator

```
@interface UIDynamicAnimator: NSObject
- (instancetype)initWithReferenceView:(UIView*)view;

- (void)addBehavior:(UIDynamicBehavior *)behavior;
- (void)removeBehavior:(UIDynamicBehavior *)behavior;
- (void)removeAllBehaviors;

@property (nonatomic, readonly) UIView* referenceView;
@property (nonatomic, readonly, copy) NSArray* behaviors;

- (NSArray*)itemsInRect:(CGRect)rect;
@property (nonatomic, readonly, getter = isRunning) BOOL running;
- (NSTimeInterval)elapsedTime;

@property (nonatomic, assign) id <UIDynamicAnimatorDelegate> delegate;
@end
```

Dynamics

UIDynamicAnimator

```
@interface UIDynamicAnimator: NSObject
- (instancetype)initWithReferenceView:(UIView*)view;

- (void)addBehavior:(UIDynamicBehavior *)behavior;
- (void)removeBehavior:(UIDynamicBehavior *)behavior;
- (void)removeAllBehaviors;

@property (nonatomic, readonly) UIView* referenceView;
@property (nonatomic, readonly, copy) NSArray* behaviors;

- (NSArray*)itemsInRect:(CGRect)rect;
@property (nonatomic, readonly, getter = isRunning) BOOL running;
- (NSTimeInterval)elapsedTime;

@property (nonatomic, assign) id <UIDynamicAnimatorDelegate> delegate;
@end
```

Dynamics

UIDynamicBehavior

```
@interface UIDynamicBehavior : NSObject

- (void)addChildBehavior:(UIDynamicBehavior *)behavior;
- (void)removeChildBehavior:(UIDynamicBehavior *)behavior;

@property (nonatomic, readonly, copy) NSArray* childBehaviors;

@property (nonatomic,copy) void (^action)(void);

@end
```

Dynamics

UIDynamicBehavior

```
@interface UIDynamicBehavior : NSObject

- (void)addChildBehavior:(UIDynamicBehavior *)behavior;
- (void)removeChildBehavior:(UIDynamicBehavior *)behavior;

@property (nonatomic, readonly, copy) NSArray* childBehaviors;

@property (nonatomic,copy) void (^action)(void);

@end
```

Dynamics

Supported behaviors

- UIAttachmentBehavior
- UICollisionBehavior
- UIGravityBehavior
- UIPushBehavior
- UISnapBehavior
- UIDynamicItemBehavior

Dynamics

UIDynamicItem

```
@protocol UIDynamicItem <NSObject>

@property (nonatomic, readwrite) CGPoint center;
@property (nonatomic, readonly) CGRect bounds;
@property (nonatomic, readwrite) CGAffineTransform transform;

@end
```

Dynamics

UIDynamicItem

```
@protocol UIDynamicItem <NSObject>

@property (nonatomic, readwrite) CGPoint center;
@property (nonatomic, readonly) CGRect bounds;
@property (nonatomic, readwrite) CGAffineTransform transform;

@end
```

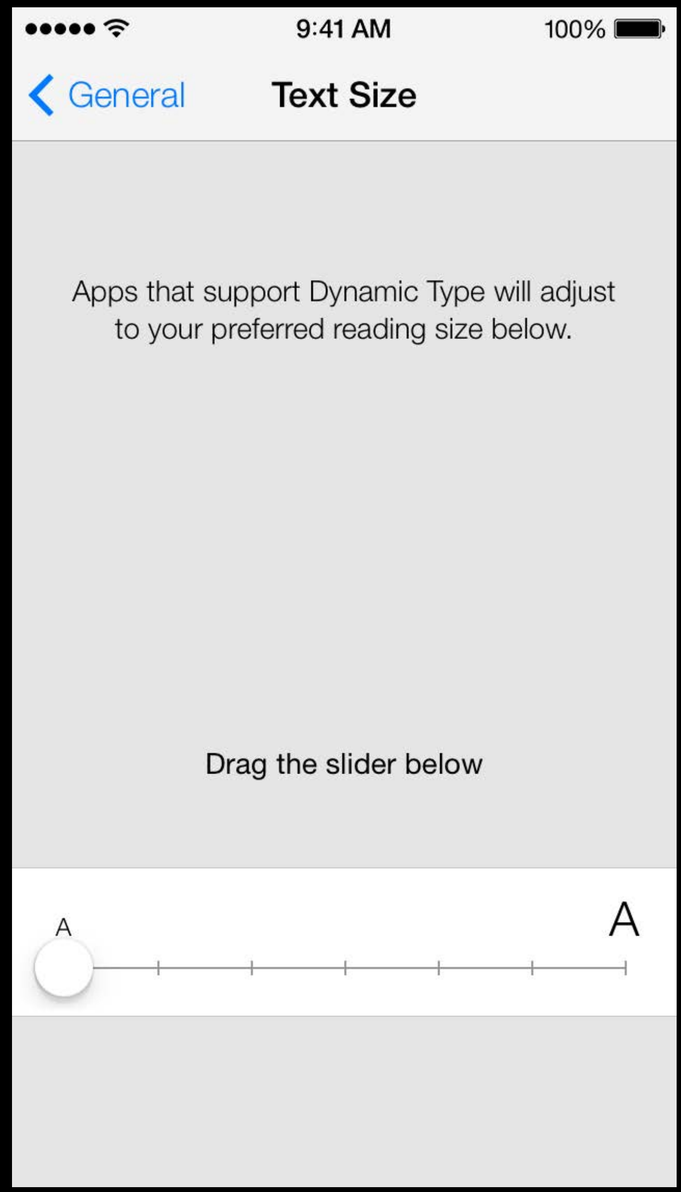
- UIView
- UICollectionViewLayoutItem

Getting Started with UIKit Dynamics	Presidio Tuesday 4:30PM	
Advanced Techniques with UIKit Dynamics	Presidio Thursday 3:15PM	

Text

Text

Dynamic type sizing



Text

Dynamic type sizing

```
@property(nonatomic,readonly) NSString *preferredContentSizeCategory;
```

Text

Dynamic type sizing

```
@property(nonatomic,readonly) NSString *preferredContentSizeCategory;

NSString *const UIContentSizeCategoryExtraSmall;
NSString *const UIContentSizeCategorySmall;
NSString *const UIContentSizeCategoryMedium;
NSString *const UIContentSizeCategoryLarge;
NSString *const UIContentSizeCategoryExtraLarge;
NSString *const UIContentSizeCategoryExtraExtraLarge;
NSString *const UIContentSizeCategoryExtraExtraExtraLarge;
```

Text

Dynamic type sizing

```
@property(nonatomic,readonly) NSString *preferredContentSizeCategory;

NSString *const UIContentSizeCategoryDidChangeNotification;
NSString *const UIContentSizeCategoryNewValueKey;
```

Text

UIFont

- Font scaling based on content size category

```
+ (UIFont *)preferredFontForTextStyle:(NSString *)style;

NSString *const UIFontTextStyleHeadline1;
NSString *const UIFontTextStyleHeadline2;
NSString *const UIFontTextStyleBody;
NSString *const UIFontTextStyleSubheadline1;
NSString *const UIFontTextStyleSubheadline2;
NSString *const UIFontTextStyleFootnote;
NSString *const UIFontTextStyleCaption1;
NSString *const UIFontTextStyleCaption2;
```

Text

UIFont

- Font scaling based on content size category

```
+ (UIFont *)preferredFontForTextStyle:(NSString *)style;

NSString *const UIFontTextStyleHeadline1;
NSString *const UIFontTextStyleHeadline2;
NSString *const UIFontTextStyleBody;
NSString *const UIFontTextStyleSubheadline1;
NSString *const UIFontTextStyleSubheadline2;
NSString *const UIFontTextStyleFootnote;
NSString *const UIFontTextStyleCaption1;
NSString *const UIFontTextStyleCaption2;
```

Text Kit

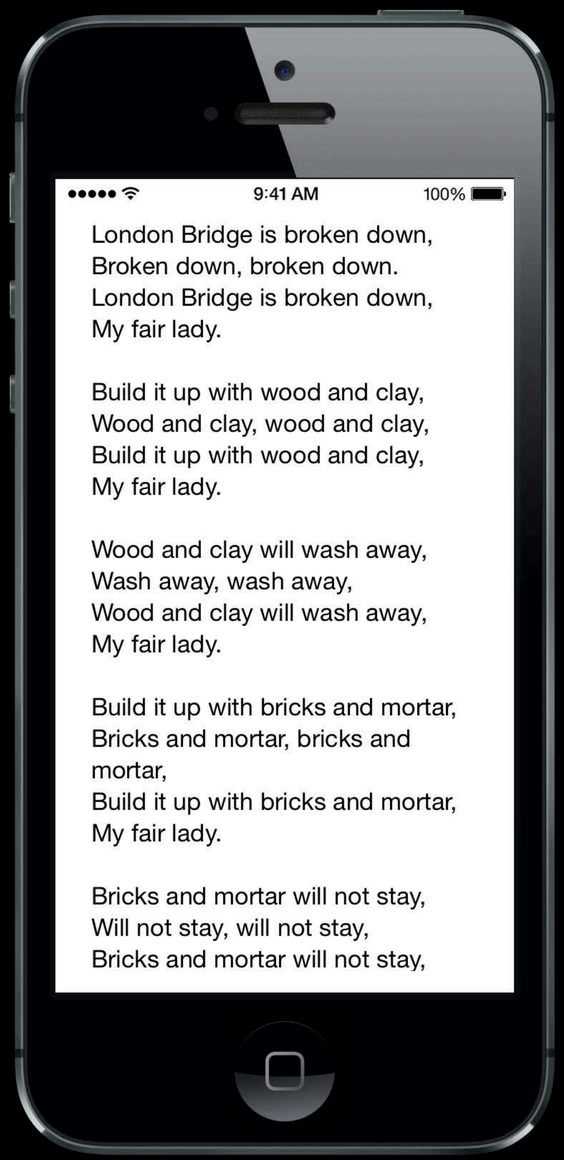
Text

Text Kit

- Objective-C API
- Inspired by the Cocoa text system from OS X
- Wraps Core Text

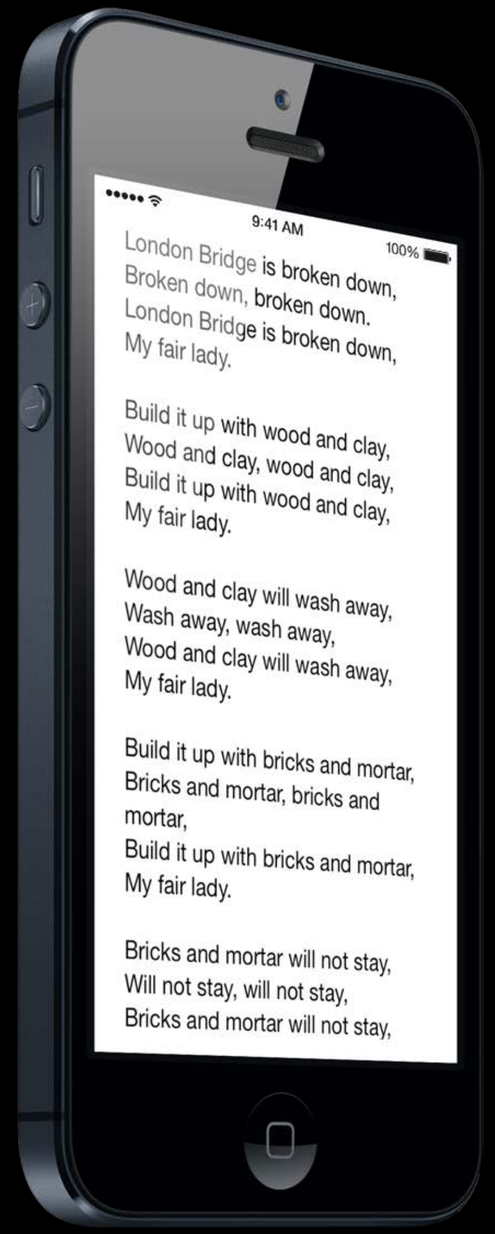
Text

Classes



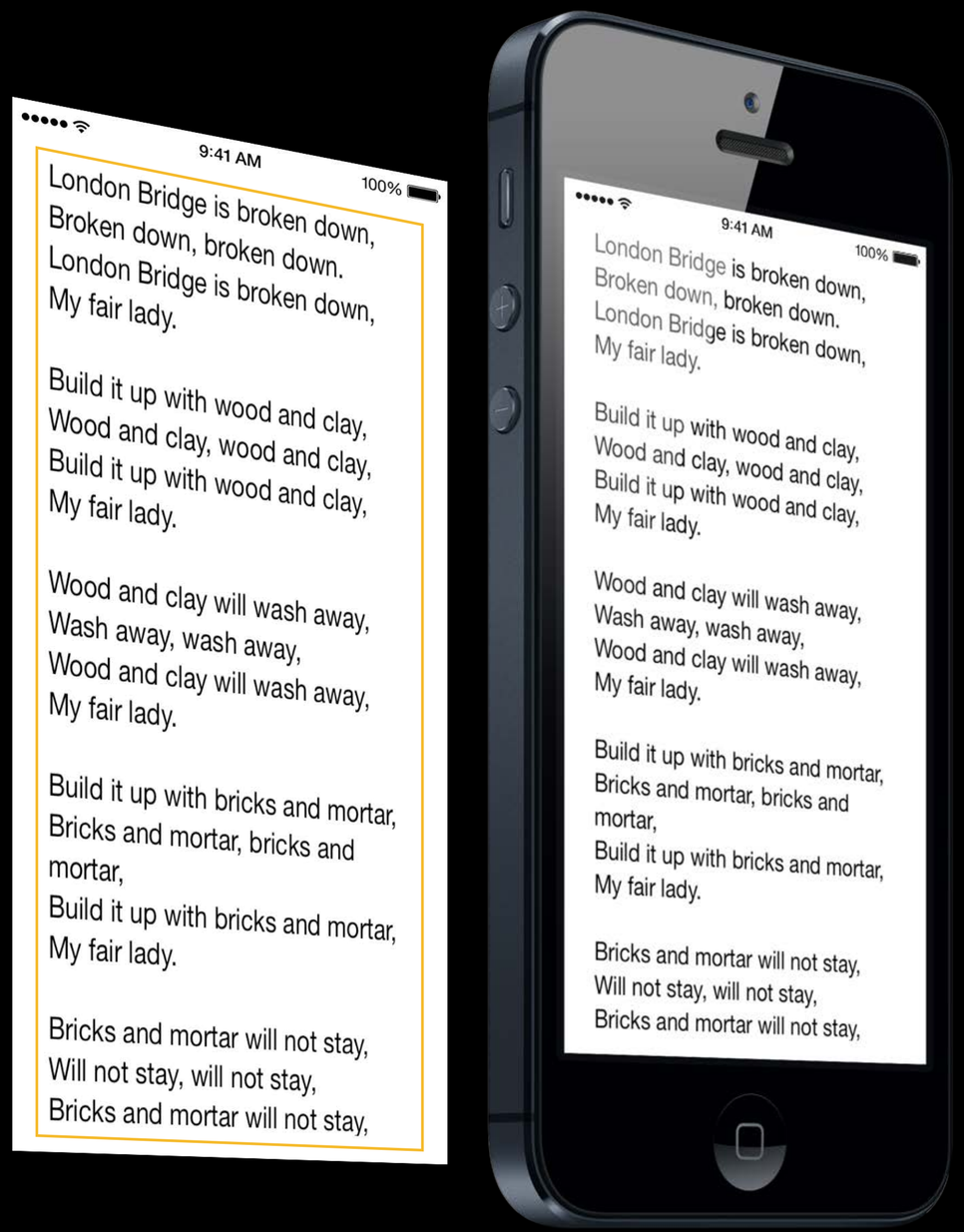
Text

Classes



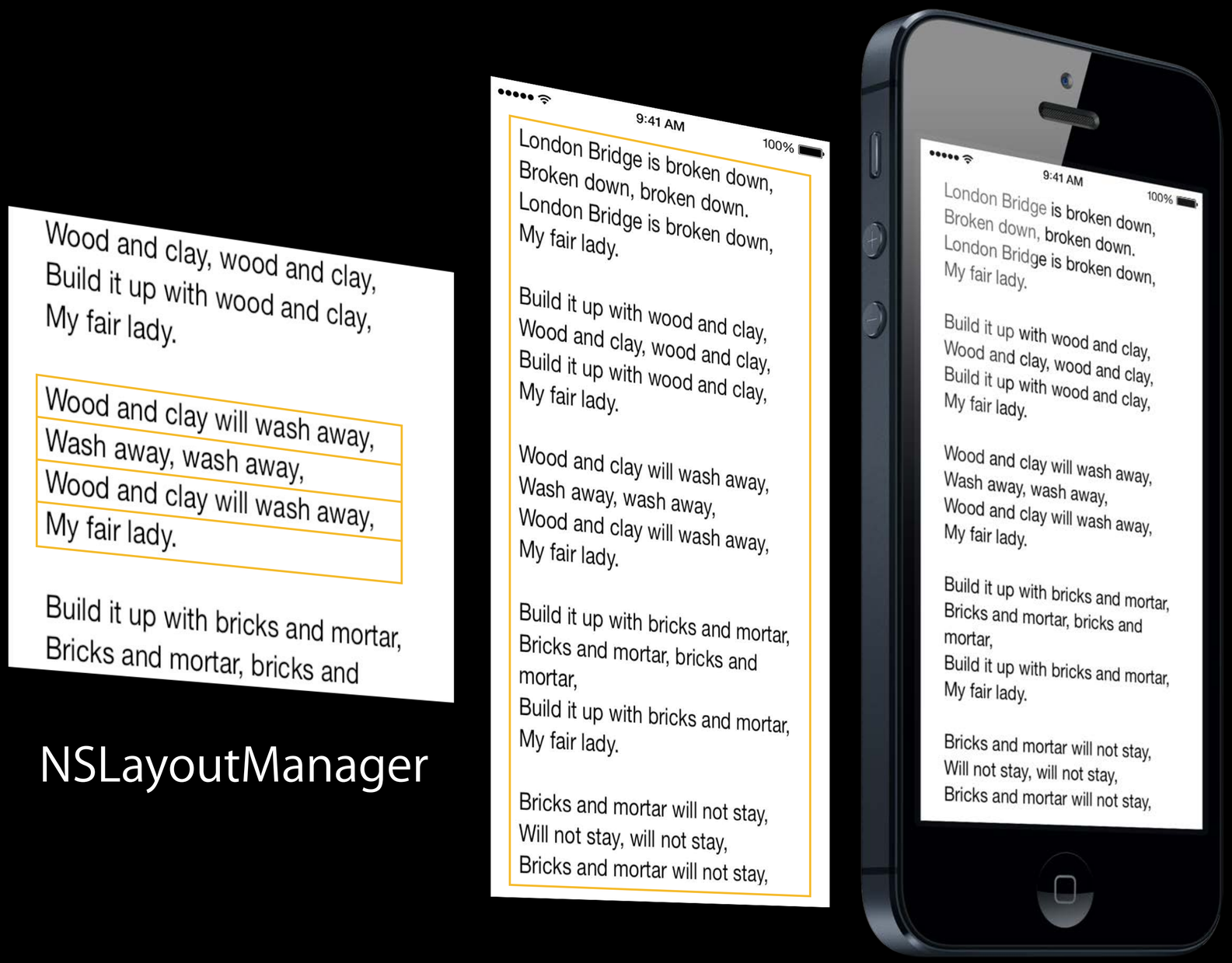
Text

Classes



NSTextContainer

Text
Classes



NSLayoutManager

NSTextContainer

Text

Classes

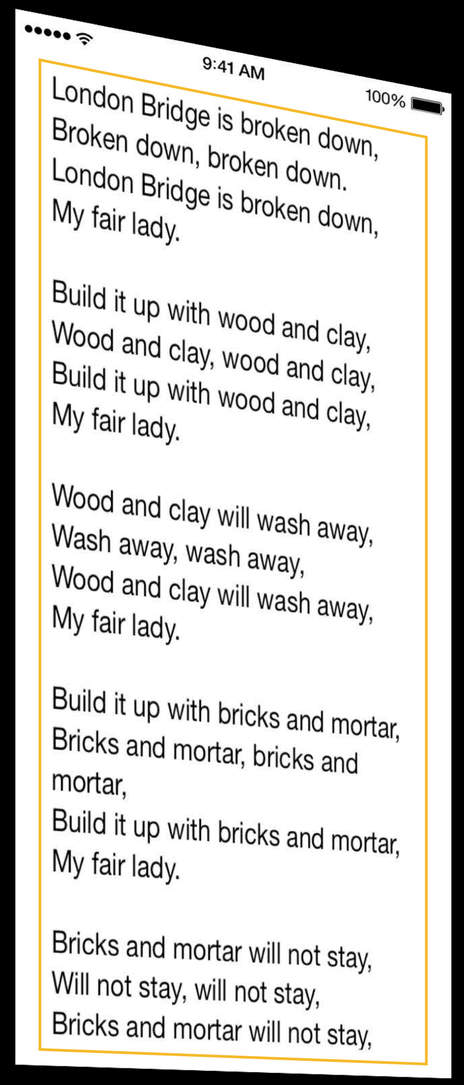
NSTextStorage

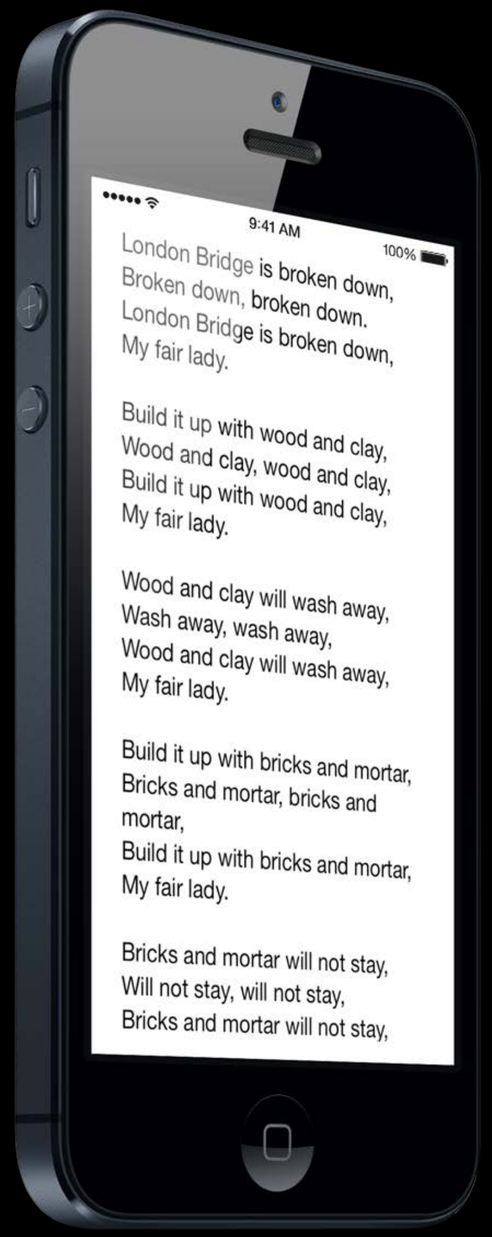
Wood and clay, wood and clay,
Build it up with wood and clay,
My fair lady.

Wood and clay will wash away,
Wash away, wash away,
Wood and clay will wash away,
My fair lady.

Build it up with bricks and mortar,
Bricks and mortar, bricks and

NSLayoutManager





NSTextContainer

Text Kit

UITextView and NSTextContainer

```
- (instancetype)initWithFrame:(CGRect)frame
                textContainer:(NSTextContainer *)textContainer;


@property (nonatomic,readonly) NSTextContainer *textContainer;
```

```
@property(nonatomic,readonly) NSLayoutManager *layoutManager;
@property(nonatomic,readonly,retain) NSTextStorage *textStorage;
```

Text Kit

UITextView and NSTextContainer

```
- (instancetype)initWithFrame:(CGRect)frame
                textContainer:(NSTextContainer *)textContainer;


@property (nonatomic,readonly) NSTextContainer *textContainer;
```

```
@property(nonatomic,readonly) NSLayoutManager *layoutManager;
@property(nonatomic,readonly,retain) NSTextStorage *textStorage;
```

Text Kit

NSTextContainer

```
@interface NSTextContainer : NSObject <NSCoding,
NSTextLayoutOrientationProvider>

- (id)initWithSize:(CGSize)size;
@property(assign, NS_NONATOMIC_IOSONLY) NSLayoutManager *layoutManager;
@property(NS_NONATOMIC_IOSONLY) CGSize size;
@property(copy, NS_NONATOMIC_IOSONLY) NSArray *exclusionPaths;
@property(NS_NONATOMIC_IOSONLY) NSLineBreakMode lineBreakMode;
@property(NS_NONATOMIC_IOSONLY) CGFloat lineFragmentPadding;

- (CGRect)lineFragmentRectForProposedRect:(CGRect)proposedRect
          atIndex:(NSUInteger)characterIndex
          writingDirection:(NSWritingDirection)baseWritingDirection
          remainingRect:(CGRect *)remainingRect;
@end
```

Text Kit

NSTextContainer

```
@interface NSTextContainer : NSObject <NSCoding,
NSTextLayoutOrientationProvider>

- (id)initWithSize:(CGSize)size;
@property(assign, NS_NONATOMIC_IOSONLY) NSLayoutManager *layoutManager;
@property(NS_NONATOMIC_IOSONLY) CGSize size;
@property(copy, NS_NONATOMIC_IOSONLY) NSArray *exclusionPaths;
@property(NS_NONATOMIC_IOSONLY) NSLineBreakMode lineBreakMode;
@property(NS_NONATOMIC_IOSONLY) CGFloat lineFragmentPadding;

- (CGRect)lineFragmentRectForProposedRect:(CGRect)proposedRect
          atIndex:(NSUInteger)characterIndex
          writingDirection:(NSWritingDirection)baseWritingDirection
          remainingRect:(CGRect *)remainingRect;
@end
```

Text Kit

NSTextContainer

```
@interface NSTextContainer : NSObject <NSCoding,
NSTextLayoutOrientationProvider>

- (id)initWithSize:(CGSize)size;
@property(assign, NS_NONATOMIC_IOSONLY) NSLayoutManager *layoutManager;
@property(NS_NONATOMIC_IOSONLY) CGSize size;
@property(copy, NS_NONATOMIC_IOSONLY) NSArray *exclusionPaths;
@property(NS_NONATOMIC_IOSONLY) NSLineBreakMode lineBreakMode;
@property(NS_NONATOMIC_IOSONLY) CGFloat lineFragmentPadding;

- (CGRect)lineFragmentRectForProposedRect:(CGRect)proposedRect
          atIndex:(NSUInteger)characterIndex
          writingDirection:(NSWritingDirection)baseWritingDirection
          remainingRect:(CGRect *)remainingRect;
@end
```

Text Kit

NSTextContainer

```
@interface NSTextContainer : NSObject <NSCoding,
NSTextLayoutOrientationProvider>

- (id)initWithSize:(CGSize)size;
@property(assign, NS_NONATOMIC_IOSONLY) NSLayoutManager *layoutManager;
@property(NS_NONATOMIC_IOSONLY) CGSize size;
@property(copy, NS_NONATOMIC_IOSONLY) NSArray *exclusionPaths;
@property(NS_NONATOMIC_IOSONLY) NSLineBreakMode lineBreakMode;
@property(NS_NONATOMIC_IOSONLY) CGFloat lineFragmentPadding;

- (CGRect)lineFragmentRectForProposedRect:(CGRect)proposedRect
          atIndex:(NSUInteger)characterIndex
          writingDirection:(NSWritingDirection)baseWritingDirection
          remainingRect:(CGRect *)remainingRect;
@end
```

Text Kit

NSTextContainer

```
@interface NSTextContainer : NSObject <NSCoding,
NSTextLayoutOrientationProvider>

- (id)initWithSize:(CGSize)size;
@property(assign, NS_NONATOMIC_IOSONLY) NSLayoutManager *layoutManager;
@property(NS_NONATOMIC_IOSONLY) CGSize size;
@property(copy, NS_NONATOMIC_IOSONLY) NSArray *exclusionPaths;
@property(NS_NONATOMIC_IOSONLY) NSLineBreakMode lineBreakMode;
@property(NS_NONATOMIC_IOSONLY) CGFloat lineFragmentPadding;

- (CGRect)lineFragmentRectForProposedRect:(CGRect)proposedRect
          atIndex:(NSUInteger)characterIndex
          writingDirection:(NSWritingDirection)baseWritingDirection
          remainingRect:(CGRect *)remainingRect;
@end
```

Text Kit

NSLayoutManager global options

```
@interface NSLayoutManager : NSObject <NSCoding>
…
@property(NS_NONATOMIC_IOSONLY) BOOL showsInvisibleCharacters;
@property(NS_NONATOMIC_IOSONLY) BOOL showsControlCharacters;
@property(NS_NONATOMIC_IOSONLY) CGFloat hyphenationFactor;
@property(NS_NONATOMIC_IOSONLY) BOOL usesFontLeading;
@property(NS_NONATOMIC_IOSONLY) BOOL allowsNonContiguousLayout;
@property(readonly, NS_NONATOMIC_IOSONLY) BOOL hasNonContiguousLayout;
…
@end
```

Text Kit

NSLayoutManager features

- Invalidation

```
- (void)invalidateLayoutForCharacterRange:(NSRange)charRange
                     actualCharacterRange:(NSRangePointer)actualCharRange;
```

- Glyphs and glyph properties

```
@property(readonly, NS_NONATOMIC_IOSONLY) NSUInteger numberOfGlyphs;

- (CGGlyph)glyphAtIndex:(NSUInteger)glyphIndex
           isValidIndex:(BOOL *)isValidIndex;

- (NSUInteger)getGlyphsInRange:(NSRange)glyphRange
                        glyphs:(CGGlyph *)glyphBuffer
                    properties:(NSGlyphProperty *)props
                characterIndexes:(NSUInteger *)charIndexBuffer
                     bidiLevels:(unsigned char *)bidiLevelBuffer;
```

Text Kit

NSLayoutManager features

- Invalidation

```
- (void)invalidateLayoutForCharacterRange:(NSRange)charRange
                     actualCharacterRange:(NSRangePointer)actualCharRange;
```

- Glyphs and glyph properties

```
@property(readonly, NS_NONATOMIC_IOSONLY) NSUInteger numberOfGlyphs;

- (CGGlyph)glyphAtIndex:(NSUInteger)glyphIndex
           isValidIndex:(BOOL *)isValidIndex;

- (NSUInteger)getGlyphsInRange:(NSRange)glyphRange
                        glyphs:(CGGlyph *)glyphBuffer
                    properties:(NSGlyphProperty *)props
                characterIndexes:(NSUInteger *)charIndexBuffer
                     bidiLevels:(unsigned char *)bidiLevelBuffer;
```

Text Kit

NSLayoutManager features

- Drawing primitives

```
- (void)drawBackgroundForGlyphRange:(NSRange)glyphsToShow
                            atPoint:(CGPoint)origin;

- (void)drawGlyphsForGlyphRange:(NSRange)glyphsToShow
                        atPoint:(CGPoint)origin;

- (void)drawUnderlineForGlyphRange:(NSRange)glyphRange
                     underlineType:(NSUnderlineStyle)underlineVal
                    baselineOffset:(CGFloat)baselineOffset
                  lineFragmentRect:(CGRect)lineRect
            lineFragmentGlyphRange:(NSRange)lineGlyphRange
                   containerOrigin:(CGPoint)containerOrigin;
```

Text Kit

Additional classes

- NSTextAttachment
- NSTextStorage

Introducing Text Kit	Presidio Wednesday 2:00PM
Advanced Text Layouts and Effects with Text Kit	Mission Thursday 2:00PM
Using Fonts with Text Kit	Presidio Friday 9:00AM

More New Features

Multipeer Connectivity

- Local network discovery
- Session management
- Encrypted sessions
- File transfers

Nearby Networking with Multipeer Connectivity

Mission
Wednesday 10:15AM

SpriteKit

- iOS
- OS X
- High-performance sprite-based game framework
- Image atlas support
- UIKit and AppKit integration

Introduction to SpriteKit	Presidio Wednesday 11:30AM

Game Controllers

- Buttons
- Analog joysticks
- Multiple controllers

Integrating with Game Controllers

Pacific Heights
Tuesday 3:15PM

MapKit

- Directions
- 3D cameras
- Map tile overlays
- Map snapshots
- Geodesic polylines

What's New in MapKit	Presidio Thursday 9:00AM

CoreLocation

- Bluetooth LE beacons
  - Advertising
  - Ranging
- New region types
- Region monitoring

What's New in Core Location	Presidio Thursday 11:30AM

Accessibility

- Guided Access API

Accessibility in iOS	Pacific Heights Tuesday 9:00AM

GameCenter

- New turn-based game API
  - Turns tabs
  - Mode for bidding
- Leaderboard improvements
- System integrity features

What's New in Game Center

Mission
Wednesday 3:15PM

Apple Evangelists

Contact information

John Geleynse
Director, Technology Evangelism
geleynse@apple.com

Mike Stern
User Experience Evangelist
stern@apple.com

Jake Behrens
UI Frameworks Evangelist
behrens@apple.com

Paul Marcos
Application Services Evangelist
pmarcos@apple.com

Apple Evangelists

Contact information

Allan Schaffer
Graphics and Game Technologies Evangelist
aschaffer@apple.com

Dave DeLong
App Frameworks and Developer Tools Evangelist
delong@apple.com

Paul Danbold
Core OS Evangelist
danbold@apple.com

Eryk Vershen
Media Technologies Evangelist
evershen@apple.com

Ernie Prahbakar
Developer Forums Evangelist
ernest@apple.com

Apple Evangelists

Contact information

David Harrington
Senior Manager, Hardware Evangelism
david@apple.com

Stephen Chick
iPhone Evangelist
chick@apple.com

Craig Keithley
MFi and I/O Technologies Evangelist
keithley@apple.com

Mark Tozer-Vilchez
Desktop Technologies Evangelist
tozer@apple.com

WWDC2013