



Xcode & Swift

iOS-Development for designers.

SCHEDULE

1. Intro

- 1.1. Motivation
- 1.2. A look ahead
- 1.3. Device Capabilities

2. Xcode

- 2.1. Interface Overview
- 2.2. Quick Swift Hands-On
- 2.3. Playgrounds

3. Preparing your Designs

- 3.1. Designing for iOS
- 3.2. Design to Xcode

4. Lunchbreak

5. Coding Session

- 5.1. Project 1 – ParticleBurst
- 5.2. Project 2 – Tossable View
- 5.3. Project 3 – GeoReminder

6. Links/Tips & Tricks

7. Wrap Up

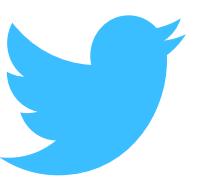
1.

Intro

Arthur Schiller

8. Semester Interface Design

www.arthurschiller.de | www.hyph.me



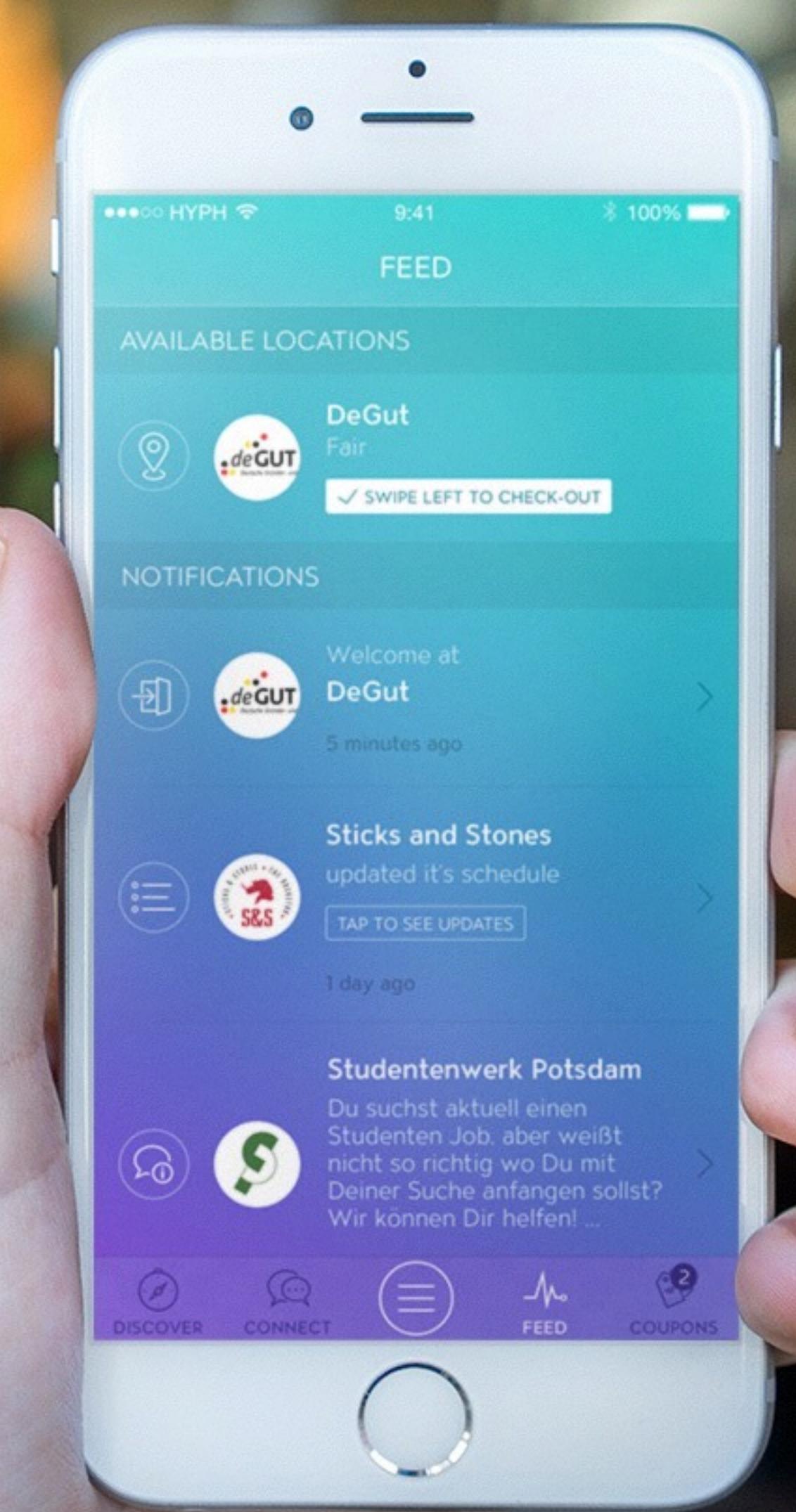
@arthurschiller_



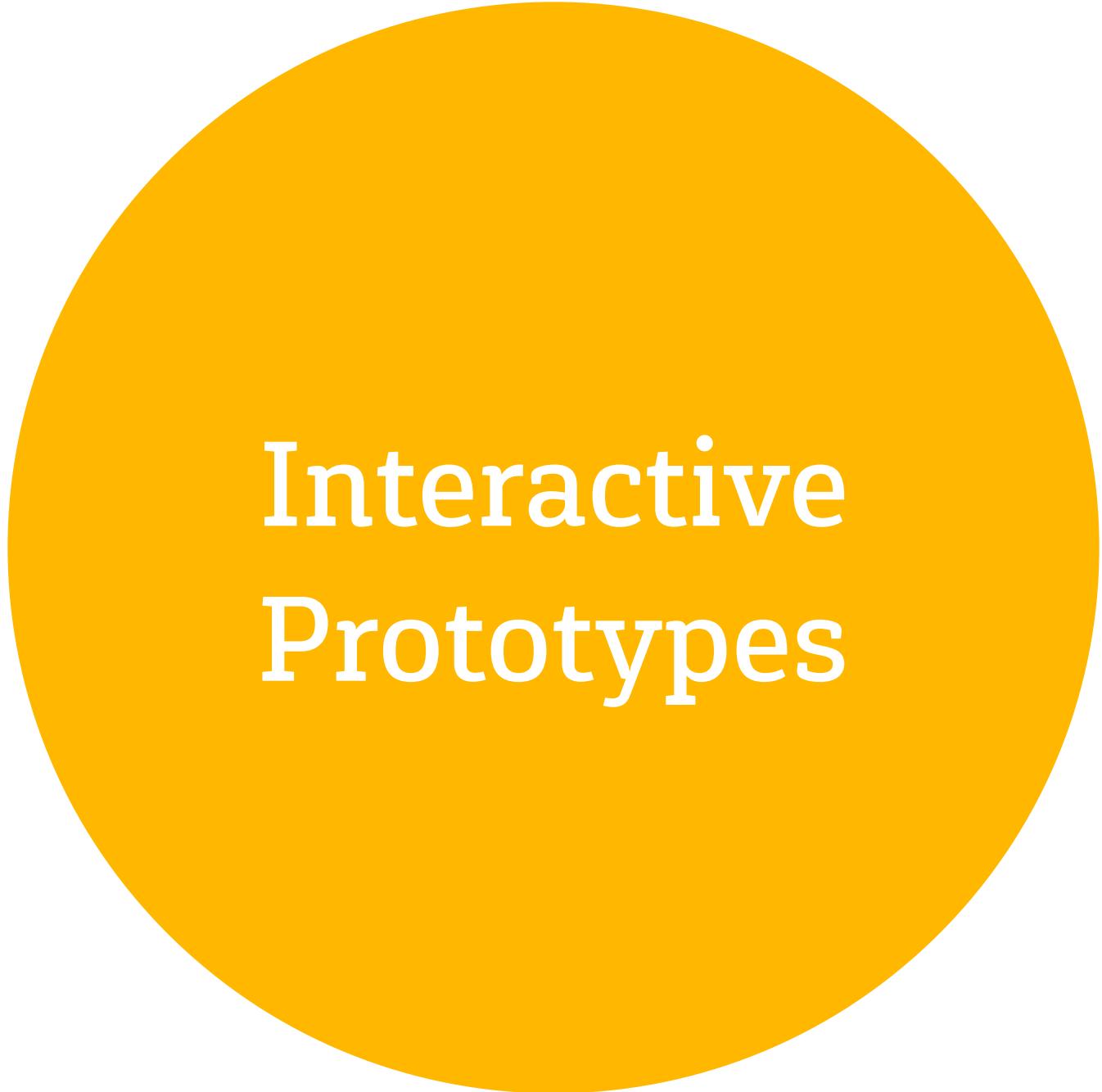
HYPH



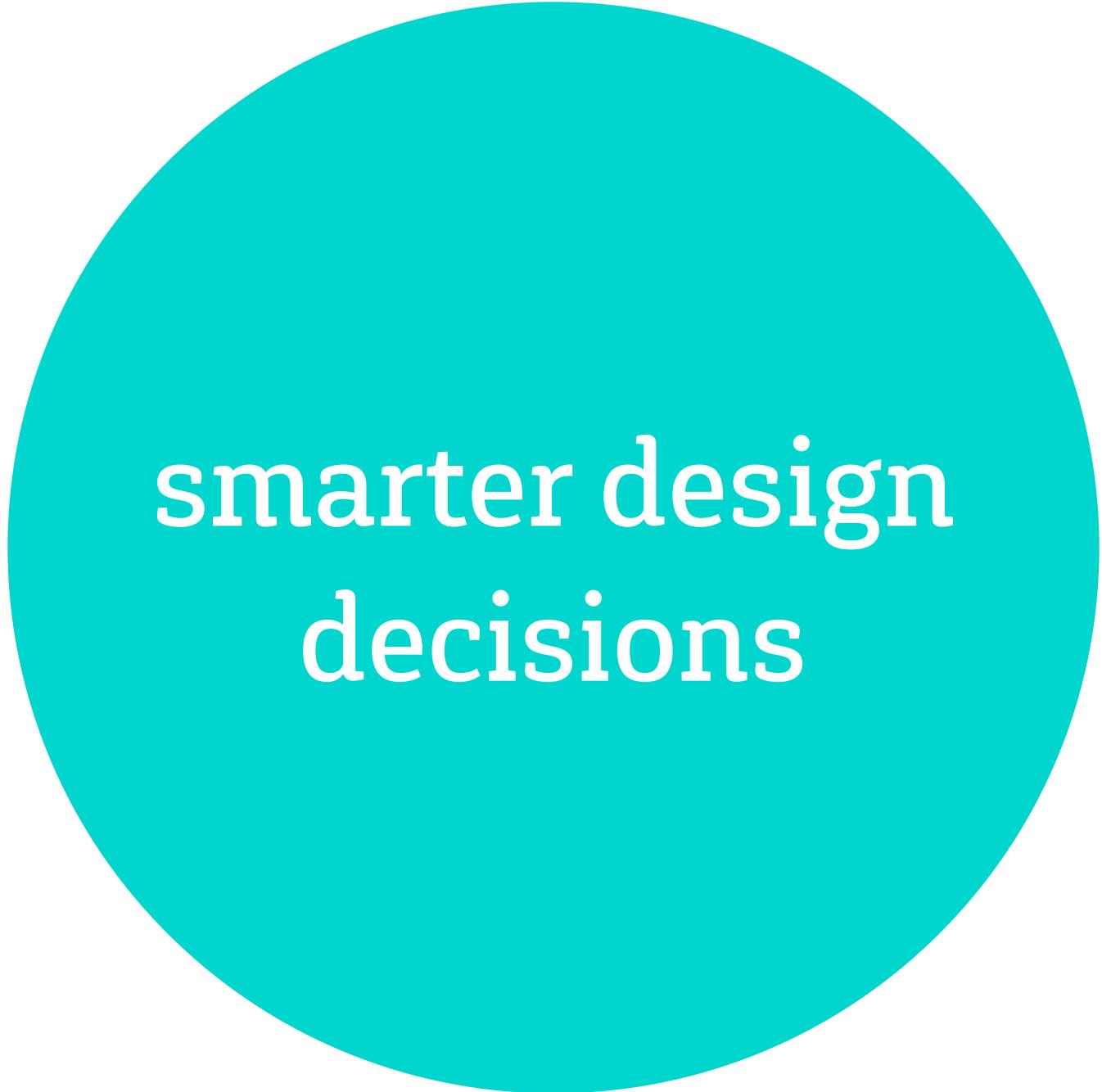




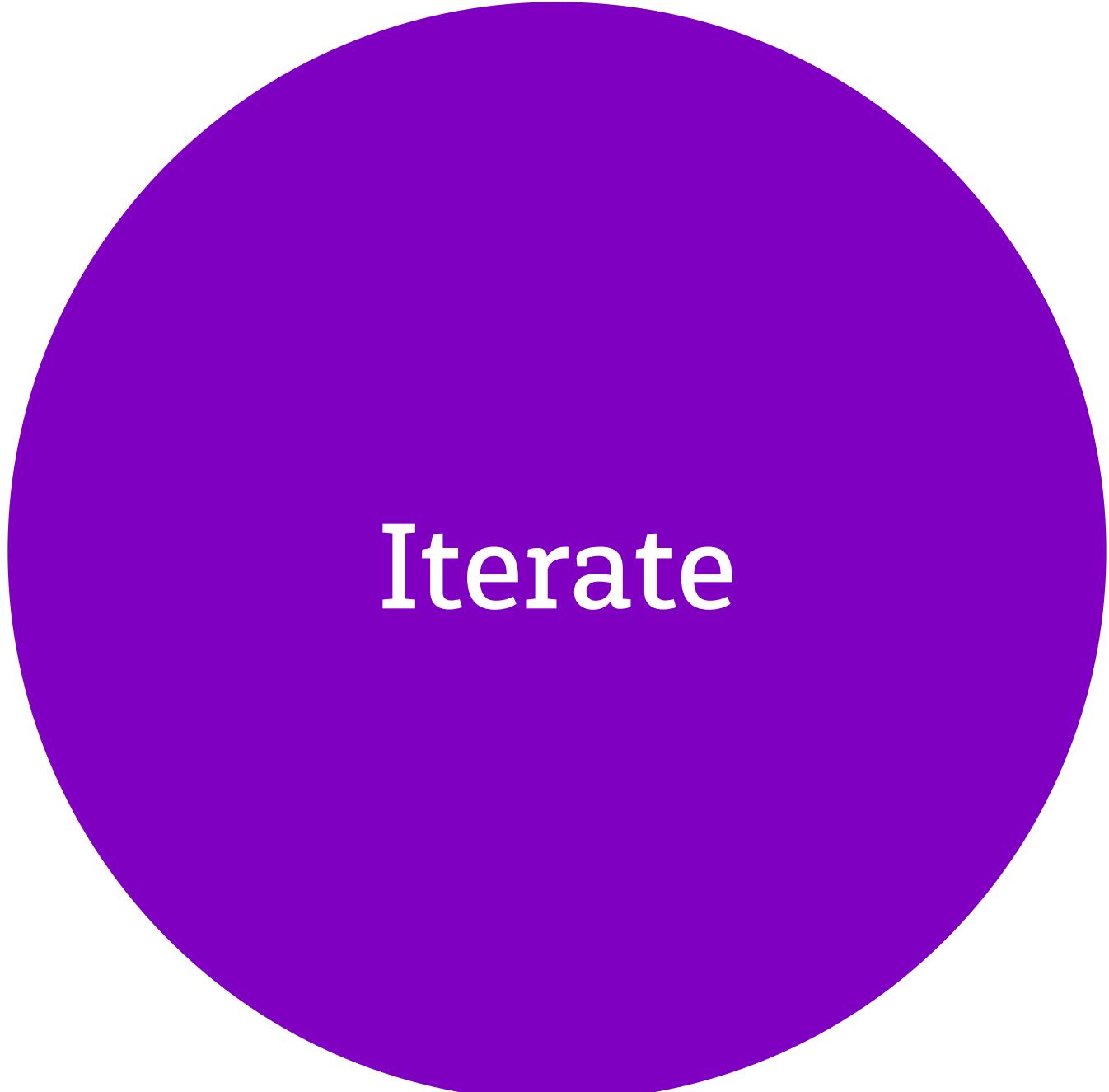
MOTIVATION



Interactive
Prototypes



smarter design
decisions



Iterate



Save money





Demand
& Salary



Tool for own
projects and
studies

A LOOK AHEAD



Mobile is eating the world

- People everywhere are using mobile devices
- changed communication, business, accessing news, entertainment
- most popular way to browse internet
- primary way to get online for younger generation



Mobile is eating the world

- constant growth
- by 2020 70% of world population will use smartphones
- businesses demand mobile developers
- mobile development ranked as one of the hottest jobs in 2016
- good time to learn skill



Why learn this skill

- iOS & Android make up 96% of smartphone OS market
- both lucrative
- Android more market share
- iOS users tend to spend more money(4× as much)
- startups often choose iOS first until reach of critical mass



Why learn this skill

- Android apps more expensive to build: *less sophisticated tools, fragmentation, more users with old OS versions*
- iOS probably easier for beginners

DEVICES OVERVIEW

INTRO

DEVICES OVERVIEW





Main features

- 64-bit CPU
- 12-megapixel camera
- 4K video recording
- motion coprocessor
- 3D touch
- GPS, WiFi, Bluetooth 4.0(iBeacon)



TouchID Fingerprint sensor

- unlock device with fingerprint
- authorize to buy stuff, validate,...



Barometer

- measure air pressure
- indirectly measure current height
- e.g used in Health-App to determine how many floors you walked



Gyroscope

- three axis sensor
- when combined with accelerometer gives device six axes to operate on
- makes device more sensitive, responsive, powerful for gaming





Motion sensor/Accelerometer

- detects how you hold the device
- switch from portrait to landscape



Proximity Sensor

- determine how far iPhone is from your face
- turn screen on/off during calls
- prevent accidental button taps



Ambient light sensor

- determine how much light is available in surrounding area
- automatically adjust brightness

2.

Xcode



2.2.

Swift

Closures



ing

uples

vyntax

ound

Multiple return types

History

- released in 2014
- successor to Objective-C
- focus on performance
- invented to match ever-improving hardware processing speeds
- within first year ranked in top-20 programming languages
- open source since December 3rd 2015

Closures



ing

uples

yntax

ound

Multiple return types

Features

- safe
- simple, easy to follow
- playgrounds
- no semicolon
- one class file
- needs less code
- emoji support 😎😍

Closures



ing

uples

syntax

ound

Multiple return types

Features

A screenshot of the Xcode interface. The left sidebar shows a file structure for an 'ObjCProject' with targets for 'iPhone 6'. It includes files like 'MyObjClass.h', 'MyObjClass.m', 'MyClass.swift', 'AppDelegate.h', 'AppDelegate.m', 'ViewController.h', 'ViewController.m', 'Main.storyboard', 'Images.xcassets', 'LaunchScreen.xib', and 'Supporting Files'. The right pane shows a Swift file with the following code:

```
// Use this file to import your Swift code into your桥接文件.
// #import "MyObjClass.h"
```

Closures



ing

uples

syntax

ound

Multiple return types

Glimpse & Comparison

Declaring a variable

```
var str = "Hello, playground" // Swift  
NSString *str = @“Hello, playground”; // Objective-C
```

Using dots instead of brackets

```
UIColor.redColor() // Swift  
[UIColor redColor]; // Objective-C
```

Closures



ing

uples

vntax

ound

Multiple return types

Playgrounds

- interactive Swift coding environment build into Xcode
- allow to quickly test code snippets
- real time results
- no need to create full project
- learn, explore, prototype
- rich comments, pages, navigation

PLAYGROUNDS HANDS-ON

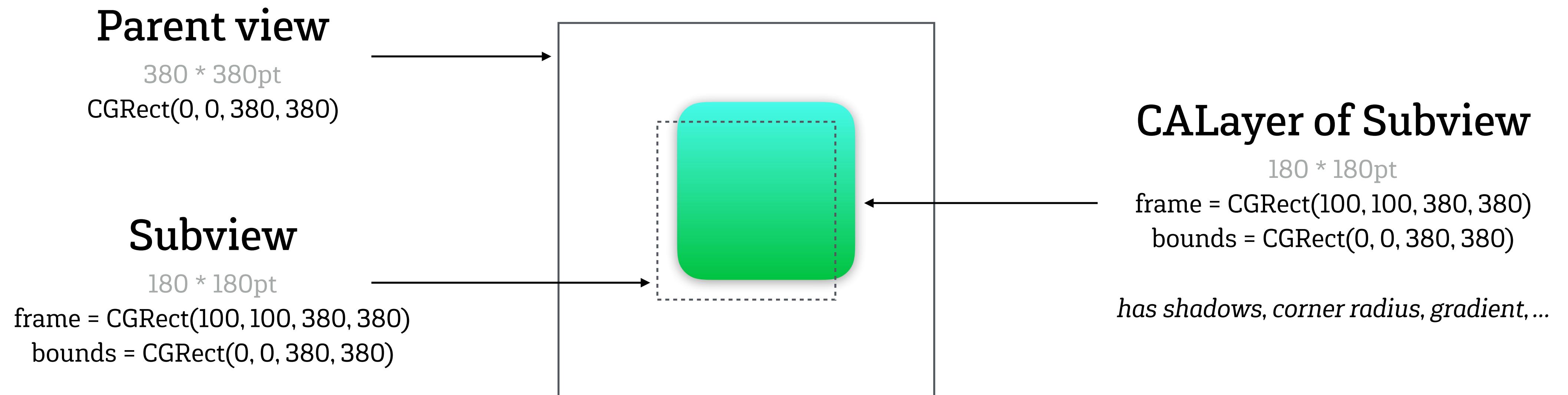
UIKit

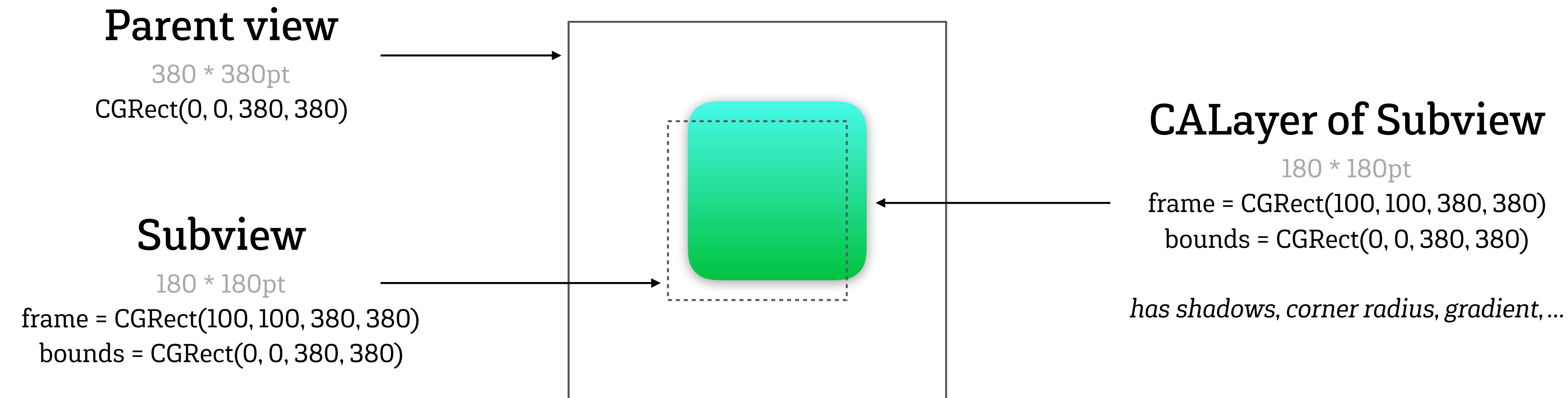
UIView

- rectangular area on the screen and the interfaces for managing the content in that area
- provides basic behavior for filling its rectangular area with a background color
- More sophisticated content can be presented by subclassing UIView

CALayer

- the base of every UIView
- manages image-based content and allows you to perform animations on that content
- can also be used without a view to display content
- main job: *manage the visual content*



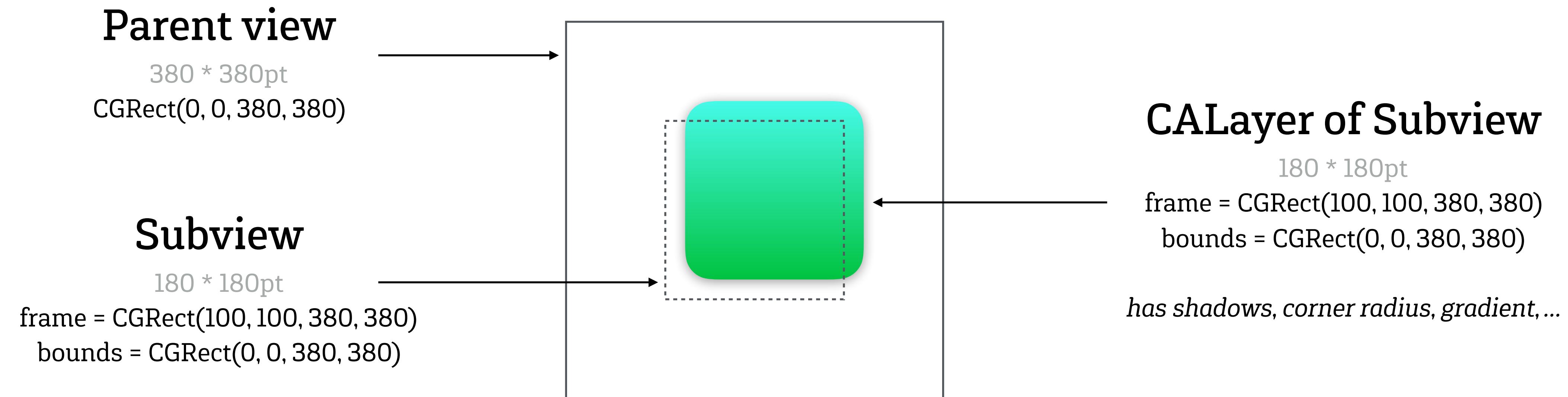


UIView

- rectangular area on the screen and the interfaces for managing the content in that area
- provides basic behavior for filling its rectangular area with a background color
- More sophisticated content can be presented by subclassing UIView

CALayer

- the base of every UIView
- manages image-based content and allows you to perform animations on that content
- can also be used without a view to display content
- main job: *manage the visual content*

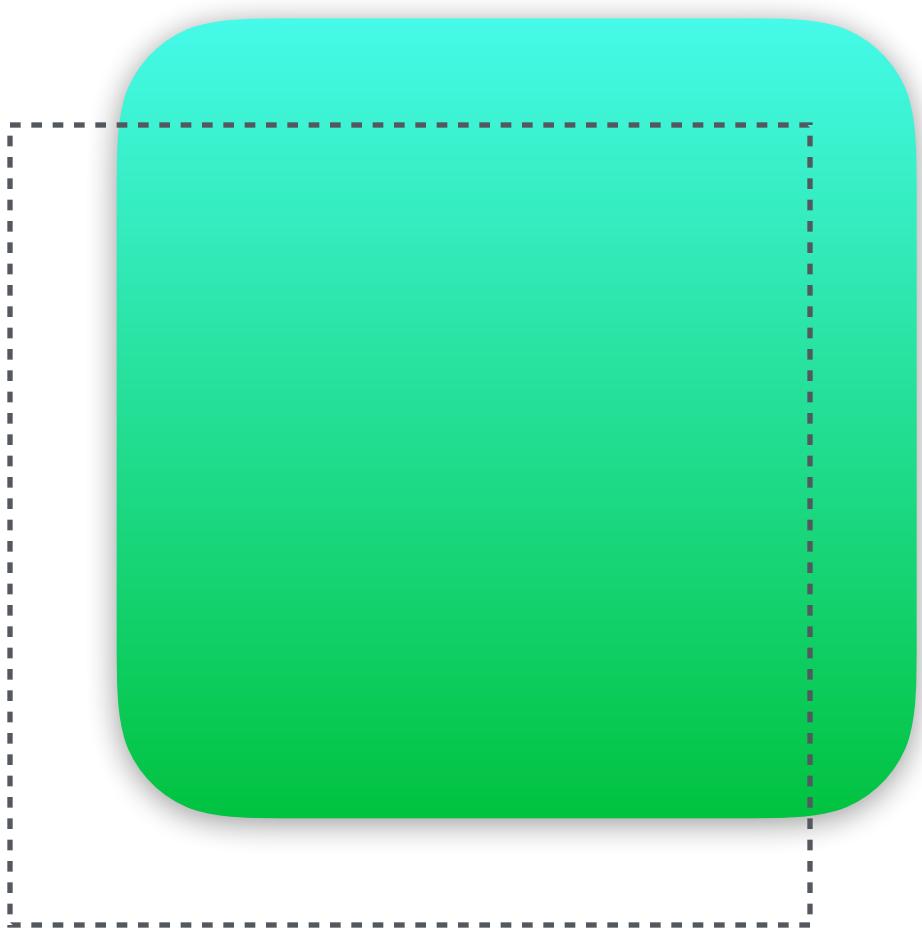


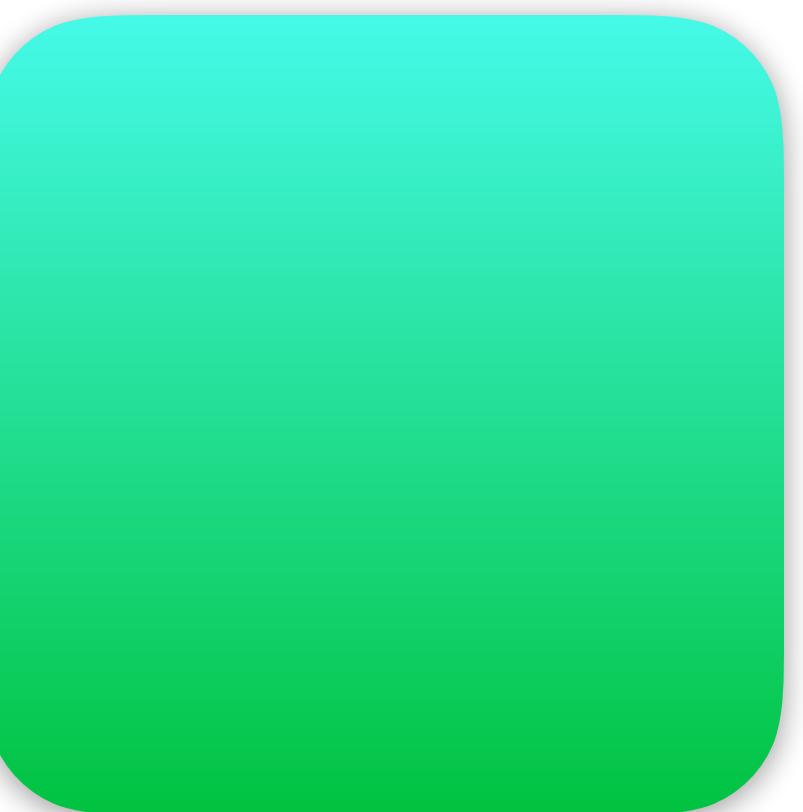
UIView

- rectangular area on the screen and the interfaces for managing the content in that area
- provides basic behavior for filling its rectangular area with a background color
- More sophisticated content can be presented by subclassing UIView

CALayer

- the base of every UIView
- manages image-based content and allows you to perform animations on that content
- can also be used without a view to display content
- main job: *manage the visual content*





Different CALayer Subclasses

- AVCaptureVideoPreviewLayer
- AVPlayerLayer
- AVSampleBufferDisplayLayer
- AVSynchronizedLayer
- CAEmitterLayer
- CAGradientLayer
- CAMetalLayer
- CAReplicatorLayer
- CAScrollLayer
- CAShapeLayer
- CATextLayer
- CATiledLayer
- CATransformLayer



6.

Links

Links

Learn

- <https://www.raywenderlich.com>
- <https://designcode.io>
- <https://www.udemy.com>
- <http://iostuts.io>
- <http://nsscreencast.com>
- <https://www.objc.io>

Resources/Tools

- <https://cocoapods.org>
- <https://github.com/Carthage/Carthage>
- <http://www.swifttoolbox.io>

Newsletters

- <http://ios-goodies.com>

Thanks!