

## Capitolo 1

# Introduzione a Swift e allo sviluppo di app per iOS

## Introduzione

Bene arrivati nel mondo dello sviluppo di app per iOS! Speriamo che leggere e usare *Sviluppare applicazioni iOS con Swift* si tradurrà in un'esperienza istruttiva, stimolante, divertente e appagante.

Questo libro si rivolge a programmatori di qualche esperienza che abbiano lavorato con un linguaggio basato su C e orientato agli oggetti come Objective-C®, C++, Java™ o C#. Non è richiesta conoscenza preesistente del linguaggio di programmazione Swift di Apple e del framework Cocoa Touch®, perché l'esame delle app per iPhone e iPad proposte in questo libro, assieme alla trattazione dei concetti fondamentali e alla presentazione del codice, permetterà di acquisire molte nozioni in merito.

## Approccio basato sulle app

Useremo un *approccio basato sulle app*, dove ogni app – di suo completa e funzionante – presenta nuove funzioni, illustrate nel loro contesto di applicazione, un'app per ogni Capitolo dal 2 all'8. Alcune delle app sono progettate come *universal* così che funzionino su iPhone, iPad e iPod touch. Per ciascuna app iniziamo a descriverla, per poi provarla. Successivamente, esaminiamo in breve le parti fondamentali dell'ambiente integrato di sviluppo Xcode®, del linguaggio Swift e del framework Cocoa Touch che utilizziamo per implementare l'app. Ci occupiamo della progettazione dell'interfaccia utente (UI) di ogni app dal punto di vista visivo. Poi forniamo il codice sorgente in forma

## In questo capitolo

- **Introduzione**
- **Dati di vendita di iPhone e iPad**
- **Gesti**
- **Sensori**
- **Accessibilità**
- **iPhone 6 e iPhone 6 Plus**
- **Storia del sistema operativo iOS e delle sue funzioni**
- **iOS 8**
- **Apple Watch**
- **App Store**
- **Objective-C**
- **Swift: il linguaggio di programmazione del futuro di Apple**
- **Posso usare esclusivamente Swift?**
- **Il framework Cocoa Touch® di iOS**
- **L'ambiente integrato di sviluppo Xcode 6®**
- **La programmazione orientata agli oggetti**
- **Prova dell'app Tip Calculator nei simulatori di iPhone e iPad**
- **Prova dell'applicazione completa tramite il simulatore di iPhone**
- **Come nasce una grande app?**
- **iOS e la sicurezza**
- **Documentazione e forum iOS**
- **Riepilogo**

completa con numerazione delle righe ed evidenziazione in grassetto per enfatizzare le porzioni più importanti del codice stesso. Si potranno vedere anche una o più schermate dell'app in funzione. Percorriamo a questo punto il codice in forma dettagliata, dando enfasi ai nuovi concetti di programmazione che vengono presentati nell'app. Il codice sorgente di tutte le app del libro si trova alla pagina <http://www.deitel.com/books/iOS8FP1>. La Prefazione contiene i dettagli sulla licenza d'uso del codice.

## Dati di vendita di iPhone e iPad

iPhone e iPad sono piattaforme in crescita notevole, che creano enormi opportunità per gli sviluppatori di app per iOS.

- iPhone di prima generazione: è stato presentato a giugno 2007, con successo pressoché immediato e vendite in aumento a ogni nuova versione. Apple ha dichiarato 6,1 milioni di iPhone di prima generazione venduti nei primi cinque trimestri (<http://www.apple.com/pr/library/2009/07/21results.html>).
- iPhone 3G: la seconda generazione di iPhone incorporava il GPS ed è arrivata a luglio 2008; ha totalizzato 6,9 milioni di unità vendute nel solo primo trimestre.
- iPhone 3GS: la terza generazione di iPhone vantava la bussola; lanciata nel giugno del 2009, ha venduto 5,2 milioni di pezzi nel primo mese.
- iPhone 4: ha debuttato a giugno 2010 vendendo oltre tre milioni di esemplari nelle prime tre settimane.
- iPhone 4s: presentato a ottobre 2011, ha superato i quattro milioni di pezzi venduti nei primi tre giorni (<http://www.apple.com/pr/library/2011/10/17iPhone-4S-First-Weekend-Sales-Top-Four-Million.html>). Nei primi tre mesi del 2012 Apple ha venduto 35,1 milioni di iPhone, con profitti aziendali quasi raddoppiati rispetto allo stesso trimestre dell'anno precedente (<http://money.cnn.com/2012/04/25/technology/apple-supplier-stocks/index.htm>).
- iPhone 5: nei primi tre giorni di vendita, a settembre 2012, ha toccato i cinque milioni di esemplari (<http://www.apple.com/pr/library/2012/09/24iPhone-5-First-Weekend-Sales-Top-Five-Million.html>).
- iPhone 5s e iPhone 5c: posti sul mercato contemporaneamente a settembre 2013, hanno totalizzato complessivamente oltre nove milioni di modelli venduti nei primi tre giorni (<https://www.apple.com/pr/library/2013/09/23First-Weekend-iPhone-Sales-Top-Nine-Million-Sets-New-Record.html>). A metà gennaio 2014, China Mobile – il più grande operatore mobile del mondo – ha iniziato a vendere iPhone per la prima volta nella Cina continentale. Si sono stimati venti milioni di iPhone venduti da China Mobile nel 2014 (<http://www.forbes.com/sites/connieguglielmo/2013/12/23/apple-inks-important-china-mobile-deal-could-sell-17-milioni-iphones-in-first-year/>).
- iPhone 6: presentato a settembre 2014 nelle due vesti di iPhone 6 e iPhone 6 Plus, ha registrato quattro milioni di preordini in un solo giorno, il doppio di iPhone 5 (<http://techcrunch.com/2014/09/15/apple-sells-4m-iphone-6-and-6-plus-pre-orders-in-opening-24-hours/>). Apple ha dichiarato dieci milioni di unità vendute nel primo weekend di disponibilità di iPhone 6 (<http://www.apple.com/pr/library/2014/09/22First-Weekend-iPhone-Sales-Top-10-Million--Set-New-Record.html>).

Le vendite di iPad non sono da meno. Per Gartner le vendite globali di tablet saliranno da 207 milioni nel 2013 a 321 milioni a fine 2015 (<http://www.gartner.com/newsroom/id/2791017>). iPad ne ha merito diretto per gran parte.

- iPad di prima generazione: lanciato ad aprile 2010, ha venduto tre milioni di unità nei primi ottanta giorni e un totale di oltre 40 milioni in tutto il mondo a settembre 2011 (<http://www.statista.com/statistics/180656/sales-of-tablets-and-ipads-in-the-us-until-2012/>).
- iPad 2: più sottile, più leggero e più veloce, iPad 2 è stato lanciato in marzo 2011 e ha venduto un milione di pezzi nel solo primo finesettimana di disponibilità. A fine 2011, iPad era responsabile del 58% del venduto mondiale di tablet (<http://finance.yahoo.com/news/why-google-android-tablet-market-185500797.html>).
- Nuovo iPad: iPad di terza generazione è stato immesso sul mercato a marzo 2012; in tre giorni se ne sono venduti tre milioni (<http://www.apple.com/pr/library/2012/03/19New-iPad-Tops-Three-Million.html>). Le vendite di iPad nel primo trimestre del 2012 si sono conteggiate in 11,8 milioni di unità, con un aumento del 151% sul medesimo trimestre dell'anno precedente.
- iPad mini di prima generazione e iPad di quarta generazione: le versioni solo Wi-Fi della prima generazione di iPad mini – con schermo da 7,9 pollici – e della quarta generazione di iPad sono state presentate a novembre 2012, con un risultato combinato di tre milioni di pezzi venduti nel primo finesettimana (<https://www.apple.com/pr/library/2012/11/05Apple-Sells-Three-Million-iPads-in-Three-Days.html>).
- iPad mini di seconda generazione (anche noto come iPad mini con display Retina®) e iPad Air (iPad di quinta generazione) sono arrivati a novembre 2013. Nel solo primo trimestre del 2014, Apple ha stabilito un record vendendo 26 milioni di iPad (<http://www.theverge.com/2014/1/27/5350106/apple-q1-2014-earnings>).

## Gesti

Lo schermo Multi-Touch dei dispositivi di Apple permette di controllare l'apparecchio con gesti che partono da uno o più tocchi simultanei (Tabella 1.1). Impareremo come far riconoscere i gesti al nostro codice e predisporre le risposte appropriate.

## Sensori

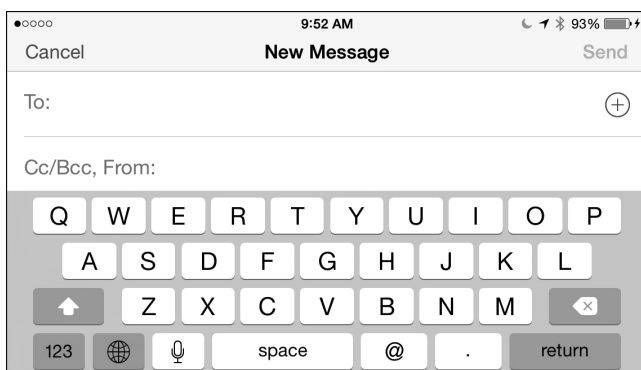
iPhone e iPad comprendono vari sensori.

- L'*accelerometro* permette al device di captare e reagire ad accelerazioni su/giù, destra/sinistra e avanti/indietro. L'esempio pratico più classico è la rotazione dell'apparecchio da *portrait* a *landscape* (da verticale a orizzontale) per cambiare orientamento a immagini, messaggi, pagine web e altro. L'accelerometro consente di controllare giochi scuotendo o inclinando l'apparecchio. Possiamo scuotere il dispositivo per ascoltare un brano musicale scelto casualmente o girarlo in modo che mostri la *tastiera orizzontale*, più comoda per digitare (Figura 1.1). Usiamo l'accelerometro

nell'app *Doodlz* del Capitolo 7, dove permettiamo all'utente di scuotere il device per cancellare il disegno sullo schermo.

**Tabella 1.1** Gestii di Phone, iPod touch e iPad.

Gesto	Azione	Scopo
Tocco	Toccare lo schermo.	Aprire un'app, fare clic su un pulsante.
Doppio tocco	Toccare lo schermo due volte.	Selezionare testo da tagliare o copiare e incollare.
Tocco prolungato ( <i>long press</i> )	Toccare lo schermo e mantenere il dito in posizione.	Spostare il cursore in posta e SMS, spostare icone di app e altro, compreso selezionare testo da tagliare o copiare e incollare.
Trascinamento ( <i>drag</i> )	Toccare e spostare il dito mantenendolo a contatto dello schermo.	Spostare un cursore, spostare la parte visibile di una mappa o una pagina web.
Spazzata ( <i>swipe</i> )	Toccare lo schermo, spostare il dito in una direzione e togliere il dito.	Scorrere elementi di una serie, come foto o cover di album musicali. Una spazzata si ferma automaticamente all'elemento successivo.
Scorsa ( <i>flick</i> )	Toccare e spostare rapidamente il dito in una direzione.	Scorrere lungo una <i>Table View</i> (per esempio <i>Contatti</i> ) o una <i>Picker View</i> (come date e orari nel <i>Calendario</i> ). A differenza di uno <i>swipe</i> , un <i>flick</i> non si ferma automaticamente. È possibile scorrere e mancare il bersaglio desiderato.
Pizzico ( <i>pinch</i> )	Toccare con due dita e simulare un pizzico, o il suo opposto (allontanando le dita).	Zoomare (per esempio per allargare o ridurre testo e immagini).
Scuotimento ( <i>shake</i> )	Scuotere l'apparecchio.	Annullare o ripristinare un'azione (per esempio testo digitato).



**Figura 1.1** Tastiera orizzontale.

- Il *giroscopio a tre assi* (presentato con iPhone 4) collabora con l'accelerometro e affina la sensibilità dell'apparecchio agli spostamenti grazie all'abilità di cogliere la rotazione attorno ai tre assi  $x$ ,  $y$  e  $z$  (sinistra/destra, su/giù e avanti/indietro). Il giroscopio

aiuta l'app *Fotocamera* a stabilizzare le immagini per migliorare foto e video, favorisce la precisione dei comandi dei giochi e molto altro (<http://www.zdnet.com/blog/apple/inside-the-iphone-4s-vibrational-giroscopio/7410>). Usiamo il giroscopio nel Capitolo 7.

- La *bussola* digitale (compresa da iPhone 3GS in poi e su iPad) permette di orientare le mappe in modo che puntino nella direzione indicata dal dispositivo.
- Il  *sensore di luce ambiente*  determina la quantità di luce attorno al dispositivo e regola la luminosità dello schermo per preservare la batteria.
- Il  *sensore di prossimità*  di iPhone determina se il dispositivo si trova vicino al volto (per esempio a causa di una telefonata). Lo schermo si spegne quando iPhone è vicino al volto e si riaccende appena l'apparecchio si allontana. Questo sensore è assente su iPad o iPod touch.
- Il  *sensore magnetico*  determina se una smart cover iPad – che aderisce al dispositivo magneticamente – è aperta o chiusa, il che accende o spegne lo schermo.
- Il  *sensore GPS*  legge i dati di geoposizionamento satellitare per le app di mappe e dipendenti dalla posizione geografica.
- Il  *sensore Touch ID* , inaugurato con iPhone 5s, è un sistema di autenticazione via impronta digitale incorporato nel pulsante *Home*. Serve a sbloccare l'apparecchio ed effettuare acquisti da App Store. Gli sviluppatori lo possono sfruttare da iOS 8 per rafforzare la sicurezza in-app.
- Il  *sensore NFC* , novità di iPhone 6 e iPhone 6 Plus, al momento è usato solo per il servizio di pagamento Apple Pay. Questo e NFC sono presentati nella Tabella 1.2.

## Accessibilità

iOS è completo di una varietà di funzioni di *accessibilità* a favore degli utenti con limitazioni sensoriali e fisiche. *VoiceOver* è un lettore di schermate basato su gesti disponibile in molte lingue, tra le quali l'italiano. Permette a utenti con problemi di vista di interagire con oggetti sullo schermo e comprendere il loro contesto. Per esempio, gli utenti possono toccare lo schermo per ascoltare una descrizione dell'elemento appena toccato e spostare il dito per udire descrizioni del contenuto che sta attorno. *VoiceOver* si usa anche con la tastiera per la pronuncia di ciascun carattere toccato, oppure ogni parola completa. A partire da iOS 7, è integrato con *Mappe*. Le funzioni di riconoscimento vocale consentono di usare comandi a voce per accedere a funzioni del telefono, come effettuare chiamate e riprodurre musica. Gli ipovedenti possono anche abbinare il proprio device con uno schermo Braille in collegamento Bluetooth.

Altre possibilità a disposizione sono adottare l'impostazione *Testo più grande* per maggiore leggibilità, impostare *Inverti colori* e *Aumenta contrasto* per dare risalto ai comandi difficili da leggere, o usare *Zoom* per ingrandire lo schermo dal 100% al 500% (compresa la schermata home, le app e tutto il resto). Per ingrandire lo schermo si effettua un doppio tocco con tre dita simultaneamente e si trascinano le dita in su per ingrandire e in giù per ridurre. Le impostazioni di accessibilità si trovano su *Impostazioni > Generali > Accessibilità*. In presenza di problemi di udito, iOS mette a disposizione funzioni di sottotitolazione, avvisi visivi e con vibrazione, chiamate video *FaceTime* e molto altro. In presenza di disabilità fisica, *AssistiveTouch* abilita l'inserimento di gesti Multi-Touch con un dito

solo o con uno stilo (da acquistare separatamente). Ancora, Siri® – l’assistente personale digitale di iOS – consente di impartire via voce un gran numero di comandi di sistema. Per aiutare gli utenti affetti da autismo, deficit di attenzione e disordini sensoriali, *Accesso Guidato* può limitare l’apparecchio a usare un’app, disabilitare il tocco su specifiche aree del sistema o per tutto lo schermo, controllare gli accessi all’app *Impostazioni*, spegnere i sensori di movimento e molto altro. Per impostare restrizioni, andiamo in *Impostazioni* > *Generali* > *Accessibilità* > *Accesso Guidato*.

Il complesso delle funzioni di accessibilità viene presentato alla pagina <http://www.apple.com/it/accessibility/>.

Per visionare la *Accessibility Programming Guide for iOS*, l’indirizzo da visitare è <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/iPhoneAccessibility/Introduction/Introduction.html>.

## iPhone 6 e iPhone 6 Plus

iPhone 6 e il più grande iPhone 6 Plus sono stati annunciati a settembre 2014. La Tabella 1.2 ne elenca alcune funzioni chiave.

**Tabella 1.2** Funzioni principali di iPhone 6 e iPhone 6 Plus.

Funzione	Descrizione
Processore A8 a 64 bit	Il nuovo processore aumenta la potenza di elaborazione del 25% e le prestazioni grafiche del 50% rispetto al suo predecessore.
Display Retina HD	Il nuovo display Retina HD ha maggiore luminosità e colori più definiti rispetto ai modelli precedenti. Lo schermo rinforzato per ionizzazione è più robusto e resiste meglio a graffi e rotture. iPhone 6 ha uno schermo da 4,7 pollici per 1334 × 750 pixel; iPhone 6 Plus arriva a 5,5 pollici con risoluzione 1920 × 1080 pixel propria dell’alta definizione (HD). Mostra i giochi con risoluzione più elevata di molte console.
Memoria	Gli iPhone sono disponibili nei tagli di memoria da 16 GB, 64 GB e il nuovo 128 GB. C’è una buona probabilità che questo libro venga letto all’indomani di una nuova generazione di iPhone, per la quale una novità sarebbe il taglio minimo di memoria elevato a 32 GB.
Barometro	Il sensore barometrico determina l’elevazione sul livello del mare calcolandola in base alla pressione atmosferica. Si usa molto nelle app per il benessere fisico, in quanto utile per tracciare con precisione la corsa di un individuo, il numero di scalini percorsi e altro.
Fotocamera	iPhone è la fotocamera più usata al mondo ( <a href="http://www.apple.com/it/iphone-6/">http://www.apple.com/it/iphone-6/</a> ). iPhone 6 e iPhone 6 Plus montano una nuova fotocamera iSight da otto megapixel. Il sensore comprende pixel di messa a fuoco – prima disponibili solo nelle fotocamere professionali – che velocizzano e migliorano l’autofocus. È possibile registrare video in alta definizione 1080 p a 60 fotogrammi al secondo e video al rallentatore a 720 p per 240 fotogrammi al secondo. Una migliore stabilizzazione automatica delle riprese perfeziona il risultato di foto e video. Per trovare altre informazioni sulle nuove fotocamere, si veda <a href="http://www.apple.com/it/iphone-6/cameras/">http://www.apple.com/it/iphone-6/cameras/</a> .
Near-Field Communication	Novità di iPhone 6 e iPhone 6 Plus è uno standard di comunicazione wireless a corto raggio che unisce dispositivi distanti pochi centimetri. Attualmente NFC è usabile solo con Apple Pay per i pagamenti mobile, e gli sviluppatori non possono inserirlo nelle proprie app.

Apple Pay	Tecnologia di pagamento wireless che usa NFC e autenticazione TouchID per offrire ai dettaglianti pagamenti one-touch ottimali sul lato della sicurezza. La carta di credito principale legata al servizio può essere linkata all'account iTunes ed essere sola oppure in compagnia di altre carte, che vengono aggiunte all'app <i>Passbook</i> . I numeri delle carte non vengono memorizzati e i dati di Apple Pay critici per le transazioni vengono cifrati sopra un chip dedicato a scopo di sicurezza. Quando paghiamo, selezioniamo la carta da usare nell'app <i>Passbook</i> , tocchiamo il pulsante <i>Home</i> per autenticarci e simultaneamente teniamo iPhone vicino a un lettore NFC fornito dal dettagliante. La transazione avviene generando un numero di carta di credito usa-e-getta, per cui il dettagliante non vede dati sensibili dell'utente come numeri di carta di credito o altro. Apple Pay può essere utile per i pagamenti online, eliminando il bisogno di comunicare dati della carta di credito e altro a siti di e-commerce. Per altre informazioni, visitate <a href="https://www.apple.com/iphone-6/apple-pay/">https://www.apple.com/iphone-6/apple-pay/</a> .
-----------	---

## Storia del sistema operativo iOS e delle sue funzioni

In questo paragrafo riassumiamo la storia e le funzioni principali delle varie versioni del sistema operativo mobile iOS. Progettato originalmente per iPhone, iOS funziona anche su iPod touch, iPad, Apple TV e, per quanto porti il nome ufficiale di Watch OS, anche su Apple Watch. È un sistema operativo proprietario – con un nucleo disponibile con licenza open source – strettamente controllato da Apple e disponibile solamente sui dispositivi di Apple. Il sistema operativo Android di Google è open source (con alcune restrizioni di fatto) e disponibile su apparecchi di costruttori indipendenti da Google. Per informazioni sulle librerie e i progetti open source di iOS, invitiamo alla consultazione della pagina <http://opensource.apple.com>.

### Il sistema operativo di iPhone

Il sistema operativo di iPhone (denominato a un certo punto iPhone OS e infine iOS) ha debuttato nel 2007 con iPhone di prima generazione. Il sistema operativo comprendeva alcune app di serie tra le quali *iPod* (media player incorporato), *Messaggi* (per gli SMS; originalmente si chiamava *Text*), *Calendario*, *Fotocamera*, *Foto* (inizialmente *Immagini*) e *Mappe*.

### iPhone OS 2: app indipendenti e App Store

iPhone OS 2 e iPhone 3G – presentati nel 2008 – hanno aperto i cancelli alle app indipendenti. Gli sviluppatori potevano creare app per iPhone e iPod touch con iPhone SDK. Approfittando dei framework a disposizione, gli sviluppatori potevano costruire app capaci di accedere ad alcune delle funzioni incorporate del telefono, come i *Contatti*. Fu lanciato App Store, come luogo deputato al download di app gratuite e a pagamento.

### iPhone OS 3

iPhone OS 3.0 ha visto la luce nel 2009 e conteneva molte funzioni nuove, tra le quali

- tagliare, copiare e incollare testo, anche tra app diverse;
- tastiera orizzontale;

- registrare memo vocali con il microfono incorporato;
- inviare foto e video tramite l'app *Messaggi*;
- cercare in Spotlight tra le informazioni in posta, contatti, calendari, note e musica;
- accedere direttamente a iTunes da iPhone;
- scegliere la lingua di sistema tra trenta diverse;
- trasferire dati tra telefoni via connessione peer-to-peer Bluetooth.

## iOS 4

iOS 4 e iPhone 4 sono arrivati nel 2010. Una funzione particolarmente degna di nota era il *multitasking*, che permetteva ad alcuni tipi di app di funzionare in sottofondo (per esempio la riproduzione musicale). iOS 4 aggiungeva anche numerosi framework per sviluppatori, atti all'integrazione di funzioni di base del sistema dentro le app. Per esempio, il framework Event Kit regolava l'accesso a eventi nell'app *Calendario* app, mentre il framework Core Motion sostituiva, rimpiazzandole in forma più ricca, le funzioni iniziali per la lettura dei dati provenienti da sensori come accelerometro, giroscopio e magnetometro. iOS 4 aggiungeva anche *Grand Central Dispatch* (GCD) e il suo modello di *programmazione asincrona* più efficiente del multithreading tradizionale prima praticato. Nella Tabella 1.3 sono descritte alcune funzioni essenziali di iOS 4.

**Tabella 1.3** Funzioni essenziali di iOS 4 (<http://arstechnica.com/apple/2010/06/ars-reviews-ios-4-whats-new-and-notable/>).

Funzione	Descrizione
Multitasking	Certi tipi di app (per esempio GPS e audio) funzionano in parallelo ad altre ed è possibile passare da un'app a un'altra senza interrompere il funzionamento né perdere dati.
FaceTime	Videochiamate tramite le fotocamere frontale e dorsale del dispositivo. Si seleziona un contatto dai <i>Contatti</i> e si tocca il pulsante <i>FaceTime</i> oppure, se è già in corso una conversazione, si tocca il pulsante <i>FaceTime</i> per passare alla videochiamata. Sul dispositivo del destinatario compare un invito alla videochiamata che, se accettato, lascia il passo alla conversazione video.
iAd	Piattaforma pubblicitaria mobile che permette di monetizzare un'app grazie a banner interni. Mentre molte pubblicità in-app, se toccate, portano l'utente nel browser facendolo uscire dall'app, iAd apre le pubblicità – a pieno schermo e interattive – dentro l'app; dopo aver visionato il contenuto, gli utenti possono chiudere l'inserzione e continuare a usare l'app. Apple si prende cura della vendita delle inserzioni e della loro erogazione. Gli sviluppatori che implementano iAd nelle proprie app ricevono il 70% del fatturato di iAd. Mentre scriviamo, iAd funziona in Francia, Germania, Giappone, Italia, Spagna, Regno Unito e Stati Uniti. Sono presenti più informazioni su iAd nel Capitolo 9.
Notifiche push	Permette alle app di ricevere notifiche, anche quando non sono funzionanti. Il servizio può essere usato quando per esempio è disponibile una nuova versione dell'app da scaricare oppure dobbiamo inviare notizie e messaggi all'utente.



Foto <i>High Dynamic Range</i> (HDR)	Consente di applicare la migliore esposizione alle foto. Un'immagine HDR viene creata dallo scatto di tre foto in rapida successione con esposizioni differenti: bassa, normale e alta. Un algoritmo fonde le tre foto in una sola, nella quale i toni sono stati ottimizzati in tutte le aree. Nel dispositivo vengono salvate la foto HDR e quelle usate per generarla.
Game Center	Le API di Game Center permettono di creare giochi sociali e multiplayer. Diventa possibile giocare con amici o trovare altri avversari in giro per il mondo, entrare in classifiche mondiali di punteggi e lanciare sfide su giochi specifici.
Noleggio di serie TV su iTunes	Serie TV senza pubblicità, fruibili per 0,99 dollari a puntata.
Cartelle	Organizza app in cartelle trascinando icone di app una sull'altra.
E-mail migliorata	Casella di ingresso unificata per più account, organizzazione dei messaggi in conversazioni ( <i>thread</i> ), controllo ortografico, ricerca dei messaggi e altro.
<i>iBooks</i>	Download di libri elettronici dal negozio <i>iBooks</i> per leggere su iPhone, iPad o iPod touch.
Playlist	Creazione di scalette musicali personalizzate direttamente sul dispositivo.
Controllo ortografico	Funzionante in <i>Mail</i> , <i>Note</i> , <i>Messaggi</i> e altre app.
Supporto di tastiere wireless	È possibile apparentare al dispositivo una tastiera wireless Bluetooth.

## iOS 5

iOS 5 ha portato numerose migliorie, declinate in oltre un migliaio di novità nelle API e negli strumenti (Tabella 1.4). Per un elenco dettagliato, si veda <https://developer.apple.com/library/ios/releasenotes/General/iOS5APIDiff/>.

**Tabella 1.4** Funzioni di iOS 5 (<https://developer.apple.com/library/ios/releasenotes/General/WhatsNewIniOS/Articles/iOS5.html>).

Funzione	Descrizione
iCloud	iCloud consente di memorizzare dati come musica, foto e video, documenti e posta su server virtuali ("nel cloud") e sincronizzarli con tutti i dispositivi iOS che condividono lo stesso account. Le API iCloud Storage permettono di creare app che scrivono e memorizzano nel cloud i dati degli utenti, che diventano accessibili e modificabili da qualsiasi device Mac o iOS senza bisogno di trasferimenti o sincronizzazioni manuali.
<i>Game Center</i>	Aggiunta di foto al proprio profilo <i>Game Center</i> e memorizzazione di tutti i propri punteggi. Sfide con amici registrati oppure con avversari calibrati sul proprio livello e in base ai giochi praticati.
<i>Centro Notifiche</i>	Centralizzazione delle notifiche di messaggi, posta, richieste di amicizia, quotazioni azionarie, meteo e altro. Il <i>Centro Notifiche</i> si apre con uno swipe verso il basso dalla sommità dello schermo.
<i>Promemoria</i>	Elenchi di impegni sincronizzati automaticamente con le app <i>Calendario</i> e <i>Mail</i> e con l'account iCloud. Avvisi geolocalizzati ricordano di completare un'attività al momento di raggiungere o di lasciare un luogo specifico.
<i>Edicola</i>	Una cartella per le app di quotidiani e periodici, che vengono caricate automaticamente in <i>Edicola</i> appena esce un nuovo numero. I framework <i>Newsstand Kit</i> e <i>Store Kit</i> consentono di creare app che effettuano il push (invio automatico) del contenuto delle testate verso i device degli utenti.

(continua)

*(segue)*

<i>Fotocamera</i>	Accesso velocizzato all'app <i>Fotocamera</i> dalla schermata di blocco e pressione del tasto di aumento del volume per scattare. <i>Streaming Foto</i> scarica automaticamente le foto su Mac e altri dispositivi iOS mediante l'account iCloud.
Integrazione di Twitter	Tweet diretti da <i>Fotocamera</i> , <i>Foto</i> , YouTube, Safari o <i>Mappe</i> e memorizzazione nei <i>Contatti</i> dei nomi utente Twitter. La API iOS Twitter account consente l'integrazione di Twitter nella propria app.
<i>Safari</i>	Prestazioni migliorate più nuove funzioni come navigazione a schede su iPad e un <i>Elenco lettura</i> per memorizzare pagine da leggere in differita su qualsiasi dispositivo iOS collegato a iCloud.
PC Free	Attivazione e aggiornamento dei dispositivi iOS via Wi-Fi senza bisogno di connessione diretta al computer.
<i>AirPrint</i>	Stampa wireless dalle app iOS sulle stampanti compatibili <i>AirPrint</i> .
<i>Accessibilità</i>	Funzioni come il lampeggio dei LED e impostazioni di vibrazione personalizzata che permettono agli utenti di vedere o sentire le chiamate in arrivo, supporto per schermi Braille Bluetooth, avvisi audio, lettura del testo evidenziato e altro.
<i>Mail</i>	Testo corsivo, grassetto, sottolineato e indentato. Evidenziazione di messaggi, aggiunta e cancellazione di cartelle, ricerca nel corpo di un messaggio e altro. L'account di posta gratuito di iCloud si sincronizza con altri dispositivi iOS.
<i>Siri</i>	Presente inizialmente su iPhone 4s, è un assistente personale che permette di usare la voce per eseguire numerose attività sul device. Possiamo dire a Siri di chiamare al telefono, dettare e inviare SMS e posta elettronica, mettere eventi e appuntamenti in agenda, cercare sul Web, trovare un luogo sulle mappe, consultare il meteo e molto altro.

## iOS 6

iOS 6, annunciato durante l'Apple World Wide Developer Conference (WWDC) nel 2012, vantava approssimativamente 200 nuove funzioni. La Tabella 1.5 riassume alcune delle novità più rilevanti per gli sviluppatori.

**Tabella 1.5** Funzioni fondamentali di iOS 6 per gli sviluppatori (<https://developer.apple.com/library/ios/releasenotes/General/WhatsNewIniOS/Articles/iOS6.html>).

Funzione	Descrizione
<i>Game Center</i> e il framework <i>Game Kit</i>	<p>Diverse novità e aggiornamenti, tra cui le seguenti.</p> <ul style="list-style-type: none"> <li>• Sfide e inviti ai propri amici a conseguire un <i>achievement</i> (un traguardo specifico dentro un gioco) o un punteggio.</li> <li>• Raggiungimento simultaneo di più traguardi.</li> <li>• Le interfacce utente di <i>achievement</i>, classifiche e richieste di amicizia sono presenti in una scheda nell'interfaccia utente di <i>Game Center</i>.</li> <li>• Maggiore controllo sull'autenticazione in locale del giocatore.</li> <li>• Supporto del timeout nell'attività di gioco. Quando un giocatore in elenco non rispetta il proprio turno di gioco, viene sollecitato il giocatore seguente.</li> <li>• Supporto migliorato del <i>matchmaking</i>, la scelta programmatica di avversari che corrispondono al livello dell'utente. I giocatori possono inviare e ricevere inviti.</li> <li>• Supporto di soprannomi dei giocatori.</li> <li>• Individuazione del giocatore con la migliore connessione verso <i>Game Center</i>.</li> </ul>

Framework Social	Sostituisce il framework Twitter. Il framework Social consente la realizzazione di app che accedono agli account social dell'utente – tra i quali Facebook, Twitter e Sina Weibo (il maggiore sito social della Cina) – per inviare aggiornamenti di stato e immagini. Viene usato nell'app <i>Twitter® Searches</i> del Capitolo 4.
Mappe	Le migliori all'app <i>Mappe</i> e al framework Map Kit comprendono lancio di <i>Mappe</i> dall'interno di un'altra app per mostrare indicazioni stradali o luoghi di interesse, registrazione di app che forniscono itinerari e permettono ad altre app – compresa <i>Mappe</i> – di utilizzarli e nuove interfacce per le app che desiderano chiedere a <i>Mappe</i> itinerari e luoghi di interesse.
Pass Kit	I <i>pass</i> sono sostituti digitali di biglietti (per concerti, voli aerei, ingressi a musei o teatri), tessere fedeltà, coupon e così via, beni normalmente stampati e con una funzione nel mondo fisico, non online. I pass contengono informazioni utili all'utente (dettagli di un evento, descrizione di un'offerta e così via) e, se necessario, un codice a barre, un codice QR o altri dati per la fruizione. I pass sono contenuti e organizzati nell'app <i>Passbook</i> . Un servizio web crea il pass e lo eroga all'utente tramite l'app, la posta o Safari.
Acquisto in-app	Memorizza il contenuto acquistabile in-app sui server Apple invece che obbligare il programmatore a organizzarsi un hosting. L'utente può inoltre acquistare contenuti iTunes (app, musica, libri) dall'interno di un'app indipendente.
iAd	Banner più grandi progettati per le app su iPad.
Promemoria	Le app possono creare e accedere a promemoria che appaiono nell'app omonima. I promemoria possono essere temporizzati o geoidentificati dall'arrivo o dalla partenza da un luogo.
Collection view	Personalizzano il layout dei dati in un'app, comprendono contenuti animati, facilitano la creazione e l'organizzazione di celle e viste e inseriscono/spostano/cancellano elementi in gruppo.
Auto Layout	Impostazione di linee guida per la disposizione di elementi dell'interfaccia utente.
Conservazione di stato	Le app possono registrare e ripristinare lo stato dell'interfaccia utente com'era quando l'utente ha smesso di usare l'app.

iOS 6 presentava numerose novità e aggiornamenti anche per gli utenti, sintetizzate nella Tabella 1.6.

**Tabella 1.6** Funzioni rilevanti di iOS 6.

Funzione	Descrizione
Integrazione di Facebook	Consente attività come invio di aggiornamenti da <i>Centro Notifiche</i> , visione delle informazioni sugli amici nei <i>Contatti</i> e degli eventi in <i>Calendario</i> e invio di foto da <i>Foto</i> o <i>Fotocamera</i> , punteggi di giochi da <i>Game Center</i> e luoghi da <i>Mappe</i> .
Siri	Nuove funzioni, tra cui le seguenti. <ul style="list-style-type: none"> <li>• Punteggi sportivi, medie di battuta (baseball), statistiche individuali e classifiche di squadra.</li> <li>• Informazioni cinematografiche da Rotten Tomatoes, recensioni di locali da Yelp!, prenotazioni al ristorante su Open-Table.</li> <li>• Lancio di app.</li> <li>• Modifica dello stato di Facebook e tweet su Twitter.</li> <li>• Lettura degli itinerari stradali.</li> <li>• Funzione Eyes Free per la guida, che consente all'autista di chiedere indicazioni, cambiare stazione radio e altro. Compatibile con vari modelli fabbricati da Audi, BMW, General Motors, Mercedes-Benz e Toyota.</li> </ul>

(continua)

*(segue)*

<i>FaceTime</i>	<i>FaceTime</i> è disponibile su connessione cellulare e consente videochiamate tra iPhone, iPad e Mac. <i>iMessage</i> funziona in modo analogo.
<i>Passbook</i>	Memorizza biglietti, carte di imbarco, coupon e tessere fedeltà in un unico contenitore. I suoi servizi basati su orari e luoghi mostrano i pass quando è il momento; i codici a barre possono essere scanditi direttamente dai dispositivi iOS. Per esempio, coupon, tessere fedeltà e carte regalo vengono mostrati quando l'utente visita il negozio appropriato, mentre le carte di imbarco appaiono all'arrivo in aeroporto.
<i>Mappe</i>	Nuove funzioni, tra cui le seguenti. <ul style="list-style-type: none"> <li>• Navigazione turn-by-turn, aggiornamenti istantanei sul traffico e stima dell'orario di arrivo.</li> <li>• Siri può accedere a <i>Mappe</i>, aiutare nella ricerca di un luogo e pronunciare le indicazioni da seguire.</li> <li>• Ruotare e inclinare il dispositivo cambia la vista della mappa.</li> <li>• Flyover mostra in alta definizione varie aree metropolitane nel mondo.</li> <li>• Ricerca locale di esercizi commerciali.</li> </ul>
Condivisione foto	Si possono inviare immagini dall'app <i>Foto</i> ad amici che usano iCloud. Le immagini condivise appaiono nella loro app <i>Foto</i> su iOS e su Mac, sul Web e via Apple TV. Per tutti è possibile aggiungere commenti.
<i>Telefono</i>	L'impostazione <i>Non disturbare</i> blocca le chiamate in modo totale oppure personalizzato. È possibile rispondere a una chiamata con un messaggio e impostare un promemoria di richiamata.

## iOS 7

iOS 7 è apparso nel settembre del 2013. Ha mostrato un'interfaccia utente riprogettata e numerose nuove funzioni per facilitare la creazione di giochi agli sviluppatori, come un nuovo sistema di animazione, multitasking migliorato e altro (Tabella 1.7).

**Tabella 1.7** Alcune funzioni rilevanti di iOS 7 (<https://developer.apple.com/library/ios/releasenotes/General/WhatsNewIniOS/Articles/iOS7.html>).

Funzione	Descrizione
Framework Sprite Kit	Un sistema di animazione con accelerazione hardware che permette di creare più facilmente giochi. Comprende un motore di simulazione della fisica, un supporto di grafica e animazione e riproduzione dell'audio.
Framework Game Controller	Permette a un'app di usare controller per giochi, apparecchi hardware collegati a iPhone o iPad fisicamente oppure via Bluetooth.
<i>Game Center</i>	iOS 7 ha portato alcune aggiunte a <i>Game Center</i> tra le quali un numero maggiore di graduatorie per app, scambi di turno tra giocatori nei giochi multiplayer e condizioni per vincere una sfida.
<i>Mappe</i>	Tra le migliori figurano mappe 3D con supporto di grafica sovrainposta, richiesta di itinerari da Apple TV e altro.
AirDrop	Condivisione di documenti tra dispositivi nel raggio dello standard Wi-Fi, anche se si trovano su reti differenti.
Audio Inter-App	Condivide l'audio tra più app sullo stesso dispositivo.
Framework Multipeer Connectivity	Collegamento tra apparecchi fisicamente vicini tramite Bluetooth, rete Wi-Fi o Wi-Fi peer-to-peer.

Framework Media Accessibility	Gestione della sottotitolazione nei media interni a un'app.
Framework Store Kit potenziato	È possibile verificare gli acquisti in-app sul dispositivo dell'utente.
framework Message UI potenziato	Permette agli utenti di inviare allegati assieme ai messaggi.
<i>CarPlay</i>	Aggiunta di iOS 7.1 disponibile su alcuni modelli di auto usciti nel 2014, <i>CarPlay</i> permette di effettuare chiamate telefoniche, ottenere indicazioni di guida e ascoltare e dettare messaggi e riprodurre musica da iPhone. È sufficiente collegare iPhone per accedere alle funzioni tramite i controlli incorporati nell'auto (come pulsanti o schermi touch) o per mezzo dei comandi vocali di Siri.

## iOS 8

iOS 8 ha fatto la propria comparsa durante la WWDC di giugno 2014 ed è stato distribuito a settembre 2014 per iPhone 4s e versioni seguenti. Contiene numerose nuove API che facilitano la creazione di giochi 3D, UI sofisticate, un nuovo sistema di animazione, multitasking migliorato e altro (Tabella 1.8). Le nuove opzioni di *continuity* rendono possibile iniziare un'attività su un dispositivo iOS e completarla su un altro. Si può persino rispondere a una chiamata telefonica e inviare e ricevere messaggi diretti a un iPhone usando un iPad o un Mac (che abbia installato OS X 10.10 o superiore) collegato alla stessa rete Wi-Fi di iPhone. È possibile che in futuro altri device, oltre a iPhone, iPod touch, iPad e Apple Watch – per esempio Apple TV – possano eseguire app iOS. Per questo, Xcode ora abilita alla progettazione di interfacce utente che si autoregolano secondo la dimensione e l'orientamento di un dispositivo. Il simulatore di iOS 8 comprende inoltre iPhone e iPad “virtuali”, di dimensioni modificabili, che permettono di collaudare le interfacce utente su schermi di dimensioni arbitrarie. Per avere altre informazioni sulle novità di iOS 8, consigliamo la visita della pagina <https://developer.apple.com/library/ios/releasenotes/General/WhatsNewIniOS/Articles/iOS8.html>.

**Tabella 1.8** Funzioni rilevanti di iOS 8 (<https://developer.apple.com/library/ios/releasenotes/General/WhatsNewIniOS/Articles/iOS8.html>).

Funzione	Descrizione
Estensioni di app	Permettono di mettere a disposizione di altre app funzioni e contenuti presenti in una nostra app. Alcuni esempi: condivisione di contenuto tra app, modifica di foto e filmati, visione o modifica di contenuto proveniente da altre app (come voci di <i>Calendario</i> ), effettuare attività nella vista <i>Oggi</i> del <i>Centro Notifiche</i> , memorizzare documenti e condividerli con altre app, creare tastiere personalizzate utilizzabili da tutto il sistema.
Document Picker	Permette agli utenti di condividere, leggere e modificare documenti da più app.
Cloud Kit	Condivisione di dati tra app e iCloud, persino in assenza di un account iCloud dell'utente. I dati sono immagazzinati in un repository associato all'app. Si può usare il pannello di controllo di Cloud Kit per monitorare e gestire i dati dell'utente.

(continua)

*(segue)*


---

Handoff	Consente la creazione di app che permettono agli utenti di iniziare un'attività su un dispositivo iOS e proseguire l'attività su un altro (che usi il medesimo ID Apple). Per esempio, un utente potrebbe iniziare un gioco su iPad e poi proseguire la partita su un iPhone o Mac. Maggiori informazioni sulle funzioni di continuity si trovano nella pagina <a href="http://www.apple.com/it/ios/whats-new/continuity/">http://www.apple.com/it/ios/whats-new/continuity/</a> .
Framework Health Kit	Permette la creazione di app che tengono traccia delle informazioni sulla salute dell'utente e le condividono con altri apparecchi connessi, come fitness tracker, misuratori della pressione arteriosa e altro. L'utente conserva il controllo delle informazioni condivise dall'app, che sono registrate in modo sicuro in un contenitore centralizzato. In questo modo l'utente può reperire facilmente tutte le proprie informazioni sanitarie e di benessere dall'app <i>Health</i> .
Framework Home Kit	Crea app che controllano o comunicano con gli apparecchi connessi a casa dell'utente (antifurto, termostati, elettrodomestici, illuminazione, riscaldamento, climatizzazione e così via).
Framework Photos	Comprende nuove API per reperire, visualizzare e modificare foto e video, nonché per lavorare con gli album condivisi via iCloud e altro.
Framework PhotosUI	Aiuta nella creazione di estensioni per l'app <i>Foto</i> .
Autenticazione Touch ID (framework Local Authentication)	Migliora la sicurezza delle app autenticando l'utente via Touch ID con le sue impronte digitali, invece che tramite nome utente e password.
SceneKit	Facilita la creazione di giochi e interfacce utente 3D. Comprende un motore fisico 3D che consente di riconoscere le collisioni, simulare la gravità e altro ancora.
SpriteKit	Comprende nuove funzioni per la creazione di effetti di gioco avanzati, tra i quali riconoscimento delle collisioni preciso al pixel (normalmente si applica a un rettangolo che circonda l'oggetto), integrazione di SceneKit, nuove animazioni, nuovi effetti fisici e altro.
Storyboard unificato	Creazione di un solo storyboard per app universali, invece che storyboard distinti per iPhone e iPad. Ciò permette il progetto di una singola interfaccia utente, alla quale aggiungere in un secondo momento le personalizzazioni desiderate per schermi e orientamenti specifici.

---

## Apple Watch

Nel settembre del 2014, Apple ha annunciato Apple Watch, i cui primi esemplari stanno aparendo negli Apple Store italiani mentre scriviamo queste righe. Apple Watch usa Bluetooth per apparentarsi con un iPhone 5 o superiore equipaggiato con iOS 8. L'orologio segna l'ora con un margine di precisione di 50 millisecondi e costituisce un nuovo sistema per comunicare, monitorare i propri parametri corporei, visionare mappe, ricevere notifiche, effettuare pagamenti con Apple Pay, controllare la posta, consultare il meteo e molto altro. Tim Cook, amministratore delegato succeduto a Steve Jobs, ha definito il nuovo prodotto "l'apparecchio più personale mai creato da Apple".

Apple Watch sfoggia uno display Retina flessibile, sensibile sia al tocco sia alla pressione, quindi in grado di distinguere tra un polpastrello appoggiato e uno che preme. Con uno swipe sulla schermata *Home* possiamo scorrere tra le app e toccarne una per lanciarla. La *Digital Crown* – la corona sul lato destro del device – va ruotata in avanti o all'indietro

per zoomare in avanti e all'indietro, navigare e scorrere tra gli elementi sullo schermo. Premendo la Digital Crown si torna alla schermata *Home* e si accede a Siri. Con uno swipe verso l'alto dal fondo dello schermo si accede ai *Glances* (sguardi) personalizzati, come meteo, musica, Borsa e così via.

Il *Taptic Engine*, un attuatore lineare, fa vibrare leggermente l'orologio quando arriva una notifica (per dire, un'e-mail o un messaggio). Alzare il polso è sufficiente per mostrare la notifica sullo schermo. L'app *Mappe* mostra dove ci troviamo e le aree intorno a noi se eseguiamo uno swipe o zoomiamo. Quando usiamo l'app per avere indicazioni stradali, il *Taptic Engine* invia leggeri tocchi al polso per segnalare il momento di girare a destra o a sinistra. I codici sono diversi, per cui sappiamo in che direzione svoltare senza dover guardare la mappa.

*Digital Touch* – la funzione svolta dal tasto laterale sul lato destro, sotto la Digital Crown – visualizza i nostri *Contatti*. È possibile iniziare una chiamata telefonica, inviare messaggi di testo o comunicare in modo originale quanto può esserlo mostrare in modo istantaneo sullo schermo del destinatario quello che stiamo disegnando con il dito, oppure mostrargli il nostro battito cardiaco.

*WatchKit* permette di estendere la nostra app per iPhone e iOS 8 a Apple Watch. Per esempio, possiamo inviare notifiche all'orologio, usare l'orologio per raccogliere informazioni relative alla salute e all'attività fisica (frequenza cardiaca, distanza percorsa e così via) e altro. La Tabella 1.9 propone alcuni esempi di app progettate per Apple Watch.

**Tabella 1.9** Esempi di app progettate per Apple Watch.

Esempio	Descrizione
America Airlines	Si occupa delle carte di imbarco.
W Hotels (parte della catena Starwood)	Check-in presso l'hotel e apertura della porta della camera semplicemente portando Apple Watch davanti alla serratura.
BMW	Mostra il livello di carica della batteria (nelle auto elettriche) e individua il luogo in cui l'auto è parcheggiata.
MLB	Aggiorna punteggi e classifiche della Major League di baseball.
Termostati Honeywell	Controllo e configurazione a distanza del termostato.
App Fitness	Monitora l'attività fisica durante il giorno.
App Workout	Imposta obiettivi specifici (corsa, ciclismo e così via) e registra i nostri progressi.
Apple TV	Controlla Apple TV.

## App Store

Mentre scriviamo, sono scaricabili oltre 1,3 milioni di app da App Store (<http://techcrunch.com/2015/06/08/itunes-app-store-passes-1-5m-apps-100b-downloads-30b-paid-to-developers/>) e i download finora effettuati di app iOS hanno passato i cento miliardi (<http://www.theverge.com/2015/6/8/8739611/apple-wwdc-2015-stats-update>). Le app per iOS hanno per gli sviluppatori un potenziale enorme. La Tabella 1.10 elenca alcune app molto diffuse in varie categorie. Alcune sono gratuite e altre sono a pagamento. Gli sviluppatori decidono il prezzo di vendita delle proprie app su App Store e ricevono il 70 per cento dell'incasso.

Molte app hanno una versione limitata, gratuita, che mira a incoraggiare la prova e l'acquisto della versione completa. Altri modi di guadagnare dalle app sono l'inserimento di pubblicità oppure la vendita in-app di beni virtuali.

**Tabella 1.10** Alcune celebri app per iPhone e iPad reperibili su App Store.

<b>Categoria</b>	<b>App di esempio</b>
Books	iBooks®, Kindle, Audio Books from Audible, Goodreads, NOOK
Business	TurboScan, Adobe Reader, Job Search, HotSchedules, SayHi Translate
Catalogs	Emoji, Pokedex Pro Elite, My Movies for iPhone Pro, cPRO
Education	Duolingo, Quizlet, Stack the States™, iTunes U, Fit Brains Trainer
Entertainment	iTube Pro, Netflix, Disney Movies Anywhere, Podcasts, WeFollow
Finance	My Weekly Budget, MileBug, Bank of America, PayPal™, Mint
Food & Drink	Starbucks, GrubHub, OpenTable, Craft Check, Clean and Green Eating
Games	Angry Birds, Cooking Academy, Fruit Ninja, Jaws™, Skee-Ball
Health & Fitness	Sleep Cycle, Fitbit, Lose It!, Weight Watchers Mobile, Fitness Buddy
Kids	Tiny Firefighters, Endless Reader, PlayKids, Toca Pet Doctor
Lifestyle	Lockster, Tinder, eBay, Amazon, CARFAX, Cupid Dating
Medical	Baby Connect, Doctor on Demand, SnoreLab, Epocrates, Pregnancy+
Music	Spotify, Shazam, Pandora Radio, Slacker Radio, Beats Music, Rdio
Navigation	Google Maps, Waze, MapQuest®, Trailhead, MotionX™ GPS Drive
News	CNN, NYTimes, USA Today, WSJ, Flipboard, The Blaze, Yahoo!®
Newsstand	Time Magazine, Forbes Magazine, Women's Health Mag
Photo & Video	Instagram, Camera+, Snapchat, YouTube, iMovie, PicPlayPost
Productivity	Gmail, Dropbox, Evernote, Keynote, Pages, Zippy, Lookup+
Reference	Google® Translate, Dictionary.com, Ancestry, WolframAlpha
Social Networking	Facebook®, Pinterest, Twitter, Skype™, LinkedIn®, WhatsApp Messenger
Sports	ESPN® ScoreCenter, NFL Live Football, Bike Repair, MLB.com At Bat
Travel	Uber, TripIt, Google Earth, Yelp®, Gas Buddy, Kayak, Hotels.com
Utilities	Globo, Scan, Chrome, Text Free with Pinger, Hushed, RedLaser
Weather	The Weather Channel®, WeatherBug®, Smoggy Air Quality

## Objective-C

Il linguaggio di programmazione C è stato sviluppato nei primi anni settanta da Dennis Ritchie presso i Bell Laboratories e si affermò rapidamente come linguaggio di sviluppo per il sistema operativo UNIX. Objective-C, creato da Brad Cox e Tom Love presso StepStone primi anni Ottanta, ha aggiunto a C capacità di programmazione orientata agli oggetti. Nel 1988, NeXT ha preso in licenza Objective-C da StepStone e ha sviluppato un compilatore e librerie per realizzare l'interfaccia utente del sistema operativo NeXTSTEP. Ha inoltre sviluppato Interface Builder per creare interfacce utente grafiche con tecniche di *drag and drop*. Apple ha acquisito NeXT nel 1997; da allora Objective-



C si è enormemente diffuso, soprattutto in virtù del suo utilizzo nello sviluppo di app per iPhone e iPad.

## Swift: il linguaggio di programmazione del futuro di Apple

Il linguaggio di programmazione Swift è stato senza dubbio l'annuncio più importante della WWDC 2014. Sviluppato all'interno di Apple dal 2010, era un progetto noto a pochissime persone. Per quanto sia sempre possibile creare app con Objective-C, Apple afferma che Swift è il suo prossimo linguaggio di programmazione per applicazioni e sistemi. Tutte le app di questo libro sono codificate in Swift.

### Funzioni fondamentali

Swift è un linguaggio moderno, con sintassi semplificata rispetto a Objective-C. La Tabella 1.11 elenca alcune delle funzioni di Swift assenti in Objective-C.

**Tabella 1.11** Alcune funzioni chiave di Swift che non si trovano in Objective-C.

Funzione di Swift	Descrizione
Inferenza del tipo	Anche se Swift è un linguaggio fortemente tipizzato, in molti casi non è necessario specificare il tipo di una variabile o di una costante; Swift può inferire il tipo sulla base del valore dell'inizializzatore della variabile o della costante.
Migliorie all'istruzione <code>switch</code>	A differenza delle istruzioni <code>switch</code> di altri linguaggi basati su C, lo <code>switch</code> di Swift può testare valori di qualsiasi tipo. I suoi <code>case</code> sono inoltre molto più flessibili, applicabili a valori singoli, insieme di valori e intervalli valori. Possiamo anche indicare criteri booleani che devono essere <code>true</code> perché vi sia corrispondenza.
Closure	Swift supporta tecniche di programmazione funzionale sotto forma di <i>closure</i> (funzioni anonime che alcuni linguaggi chiamano <i>lambda</i> ). Le closure sono manipolabili al pari di dati: assegnate a variabili, passate a funzioni come argomenti e restituite da funzioni. Molte delle funzioni globali incluse nella Standard Library di Swift ricevono closure come argomenti; per esempio, esiste una versione della funzione <code>sort</code> che riceve una closure per la comparazione di due oggetti al fine di determinare la loro posizione nell'ordinamento.
Tuple	Swift contempla le <i>tuple</i> , raccolte di valori che possono essere dello stesso tipo o di tipi differenti. Il linguaggio fornisce sintassi per comporre (creare) e scomporre (estrarre valori da) una tupla.
Optional	Gli <i>optional</i> consentono di definire variabili e costanti che potrebbero non avere un valore. Il linguaggio fornisce meccanismi per determinare se un optional abbia un valore e, nel caso, ottenerlo. Gli optional funzionano con qualunque tipo in Swift, là dove il concetto corrispondente in Objective-C – un puntatore che punta a un oggetto oppure è <code>nil</code> – funziona solo per i tipi riferimento.
Tipo Dictionary	Il tipo <code>Dictionary</code> di Swift fornisce supporto incorporato per la manipolazione di dati in forma di coppie chiave-valore.

(continua)

*(segue)*


---

Tipi valore Array, String e Dictionary	I tipi Array, String e Dictionary di Swift sono tipi di valore (non di riferimento come ci si potrebbe attendere) implementati come struct. Gli oggetti di tipo valore sono copiati quando li assegniamo a variabili o costanti, li passiamo a funzioni o vengono restituiti da funzioni. Il compilatore Swift ottimizza le operazioni di copia di tipo valore, effettuandole solo quando sia necessario.
Controllo dei confini degli Array	Se accediamo a un elemento che si trova fuori dai confini di un Array, si verifica un errore di runtime.
Affinità con le classi dei tipi valore struct ed enum	I tipi struct ed enum di Swift possiedono varie funzioni associate normalmente alle classi, il che li rende più solidi delle rispettive controparti in Objective-C. Gli oggetti di tipo struct ed enum sono tipi valore.
Funzioni che restituiscono più valori (mediante tuple)	Le funzioni possono restituire più valori, anche di tipo diverso, sotto forma di tuple.
Generics	Invece di scrivere codice separato per effettuare attività identiche su tipi differenti (per esempio sommare un Array di interi e un Array di valori in virgola mobile), i generics consentono di scrivere il codice una volta sola e inserire dei segnaposto in luogo del tipo o dei tipi di dati da manipolare. Il compilatore inserisce i tipi opportuni in sostituzione dei segnaposto. In una chiamata generica di funzione, il compilatore determina i tipi effettivi in base agli argomenti specificati nel codice che chiama la funzione. Per un tipo generico, il compilatore usa il tipo effettivo che specifichiamo quando dichiariamo un oggetto di quel tipo generico. I tipi Array e Dictionary di Swift sono tipi generici e molte delle sue funzioni globali sono funzioni generiche. Quando usiamo funzioni e tipi generici, viene eseguito un controllo di tipo in compilazione per garantire che funzioni e tipi siano stati usati correttamente. Per esempio, creare un Array di interi e poi tentare di inserire una String nell'Array dà luogo a un errore di compilazione.
Overloading degli operatori	Possiamo definire funzioni che effettuano un overload degli operatori esistenti in modo da funzionare con tipi nuovi, e anche definire operatori interamente nuovi, cosa che non è possibile fare in C# o C++.
Controllo dell'overflow nei calcoli interi	Per default, tutti i calcoli con numeri interi verificano un eventuale overflow aritmetico e generano un errore di runtime se sussiste un overflow.
Interpolazione di String	L'interpolazione di String permette di costruire String inserendo valori di variabili, costanti o espressioni dentro i segnaposto direttamente in letterali String.
Tipi nidificati	Possiamo definire tipi nidificati dentro altre definizioni di tipo, operazione comunemente svolta per definire enum o classi di utility e struct nascoste nello scope di un altro tipo.
Funzioni nidificate	Nelle definizioni di funzione se ne possono nidificare altre; per esempio, una funzione nidificata può essere chiamata nello scope della funzione che la contiene e può essere restituita da quest'ultima per essere usata in altri scope.

---

## Prestazioni

Swift è stato progettato per dare migliori risultati di Objective-C sugli odierni sistemi multicore. Durante la presentazione di apertura della WWDC 2014, Apple ha dichiarato il codice Swift come una volta e mezzo più veloce del codice Objective-C. Sebbene

Array, String e Dictionary siano tipi *valore* le cui variabili vengono normalmente *copiate* quando sono passate o assegnate, il compilatore di Swift ottimizza queste operazioni eseguendole solo quando necessario.

## Prevenzione degli errori

Swift elimina molti errori comuni di programmazione e rende il codice più solido e sicuro. Molte delle decisioni adottate – come non fare restituire un valore all’operatore di assegnazione – non sarebbero state praticabili su Objective-C, perché avrebbero provocato incompatibilità con vario codice preesistente.

### Alcune funzioni di Swift che eliminano errori comuni di programmazione

- Sono richieste parentesi graffe ({}), attorno al corpo di ogni istruzione di controllo. Questa cautela contro la dimenticanza delle graffe nei body che contengono più istruzioni.
- A differenza di Objective-C, C e C++, Swift non include i puntatori.
- L’operatore di assegnazione (=) non restituisce un valore. Se = viene usato in una condizione al posto di ==, si verifica un errore di compilazione. Ciò previene la distrazione di quanti potrebbero dimenticarsi di digitare il doppio segno di uguale quando richiesto.
- Il punto e virgola è opzionale, a meno che si intenda separare più istruzioni presenti sulla stessa riga.
- Le parentesi attorno alle condizioni nelle istruzioni di controllo sono opzionali e il risultato è una leggibilità del codice appena migliore.
- Variabili e costanti devono essere inizializzate prima di essere usate, nelle loro definizioni oppure tramite metodi di inizializzazione nelle definizioni di tipo.
- I calcoli interi sono sottoposti al controllo di overflow per default; in caso di overflow, si genera un errore di runtime.
- Swift non ammette conversioni implicite tra tipi numerici.
- Gli indici di Array (subscript) sono controllati agli estremi al momento dell’esecuzione; accedere a un elemento esterno ai confini di un Array genera un errore di runtime.
- La gestione automatica della memoria elimina la maggior parte delle perdite; è ancora possibile mantenere il riferimento a oggetti non più utilizzati ed evitare che il runtime ne reclami la memoria occupata. Swift mette anche a disposizione riferimenti deboli per i casi nei quali uno o più riferimenti circolari tra oggetti impedirebbero normalmente un corretto riutilizzo della memoria impegnata.

## La Standard Library di Swift

La *Standard Library di Swift* contiene i tipi incorporati in Swift (String, Array, Dictionary e i vari tipi numerici interi e in virgola mobile), protocolli (Equatable, Comparable e Printable) e funzioni globali (per esempio quelle di ordinamento e stampa). Discuteremo dei vari elementi man mano che li incontreremo nel libro. Per approfondire la conoscenza di questi elementi, la cosa migliore è studiare la *Swift Standard Library Reference* alla pagina <https://developer.apple.com/library/ios/documentation/General/Reference/SwiftStandardLibraryReference>.

## App scritte in Swift e i framework Cocoa® e Cocoa Touch®

Swift, come Objective-C, può fare uso del framework Cocoa di OS X e del framework Cocoa Touch di iOS. Queste potenti librerie di componenti aiutano a creare app rispondenti ai requisiti di *look and feel* di un'app iOS e OS X formulati da Apple. I framework sono scritti principalmente in Objective-C (alcuni in C); Apple ha dichiarato che i prossimi nuovi framework saranno sviluppati in Swift.

## Interoperabilità tra Swift e Objective-C

Swift e Objective-C possono convivere nella stessa app e non c'è bisogno di riscrivere tutto il codice Objective-C per aggiungere porzioni realizzate in Swift. Gran parte delle API Cocoa Touch sono tuttora scritte in Objective-C e quindi i programmatori Swift creano tipicamente app che interagiscono con codice Objective-C esistente, come faremo anche noi negli esempi di questo libro.

Di fatto passeremo sovente oggetti Swift in metodi di classi scritte in Objective-C, come quelle in Cocoa Touch. I tipi numerici di Swift e i suoi tipi `String`, `Array` e `Dictionary` possono essere utilizzati in contesti dove ci si attende di vedere i loro equivalenti Objective-C. Allo stesso modo, gli equivalenti Objective-C (`NSString`, `NSMutableString`, `NSArray`, `NSMutableArray`, `NSDictionary` e `NSMutableDictionary`), quando vengono restituiti al proprio codice Swift, sono trattati automaticamente come le rispettive controparti Swift. Questo meccanismo – detto *bridging* – è trasparente e non necessita della nostra attenzione. Questi sono gli argomenti principali trattati nel libro di Apple *Using Swift with Cocoa and Objective-C*, disponibile su iBooks Store e alla pagina <https://developer.apple.com/library/ios/documentation/Swift/Conceptual/BuildingCocoaApps>.

- Impostare il progetto di un'app Swift che può usare Cocoa o Cocoa Touch.
- Interoperabilità tra Swift e Objective-C: interagire con le API di Objective-C, creare classi Swift da usare con codice Objective-C e altro.
- Progetti contenenti file Swift e Objective-C.
- Suggerimenti per la migrazione del vecchio codice Objective-C verso Swift.

## Altre risorse di Apple su Swift

Oltre ai documenti menzionati nei paragrafi precedenti, Apple fornisce risorse aggiuntive per apprendere l'uso di Swift.

- Il blog Swift di Apple si trova all'indirizzo <https://developer.apple.com/swift/blog/>.
- Il codice dimostrativo in Swift presentato alla WWDC 2014 è reperibile alla pagina <https://developer.apple.com/wwdc/resources/sample-code/>.
- Il libro di Apple *The Swift Programming Language* presenta Swift e contiene una breve panoramica del linguaggio, la guida al linguaggio (che copre le funzioni chiave di Swift in maggiore profondità) e la guida di riferimento (che presenta la grammatica di Swift e i dettagli di ciascuna funzione). L'opera si trova su iBooks Store e all'indirizzo [https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift\\_Programming\\_Language/](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/).
- Qui invece si trovano i video didattici prodotti durante la WWDC 2014: <https://developer.apple.com/videos/wwdc/2014/>.

## Posso usare esclusivamente Swift?

Questo libro è diretto a differenti fasce di lettori:

- programmatori Objective-C che stanno lavorando in Swift a nuove iOS;
- programmatori Objective-C che stanno migliorando app esistenti con Swift;
- programmatori Java, C++ e C# nuovi allo sviluppo iOS che intendono realizzare app con Swift.

Una delle domande più comuni è probabilmente “posso programmare la mia app iOS interamente in Swift?”. La risposta cambia in funzione del genere di lettore.

### Programmatori Objective-C che stanno sviluppando nuove app iOS in Swift

Apple incoraggia i programmatori Objective-C a passare allo sviluppo con Swift. In Swift si possono creare intere app, sebbene si userà ancora il framework Cocoa Touch, scritto in gran parte in Objective-C.

### Programmatori Objective-C che stanno migliorando nuove app iOS in Swift

Apple incoraggia i programmatori Objective-C anche all’uso di Swift per lavorare alle app esistenti. Anche se le migliorie vengono scritte esclusivamente in Swift, interagiranno estesamente con il codice Objective-C e il framework Cocoa Touch. Invece di convertire un’app esistente in Swift, è possibile salvaguardare l’investimento effettuato riutilizzando il codice Objective-C, che magari nel tempo è stato raffinato, ottimizzato e sottoposto a debugging capillare. Meglio essere a conoscenza dei contenuti del documento Apple *Using Swift with Cocoa and Objective-C* prima di entrare nei dettagli dell’interoperabilità tra Swift e Objective-C.

### Programmatori Java, C++ e C# nuovi allo sviluppo di app iOS

Apple incoraggia all’uso di Swift anche i programmatori Java, C++ e C# che non conoscono Objective-C e vogliono affrontare lo sviluppo per iOS. L’azienda ritiene che Swift possa efficacemente abbassare le barriere di ingresso per i nuovi entranti nel mondo dello sviluppo iOS. Le sue similarità con questi e altri linguaggi di grande diffusione lo renderanno più familiare e avvicinabile di Objective-C. Servirà comunque il framework Cocoa Touch che è scritto in Objective-C, ma si vedrà come Apple fornisca le interfacce Swift necessarie per lo scopo.

### In rapida evoluzione

Swift è così nuovo che la sua evoluzione è in questo momento molto rapida, e prevedibilmente continuerà a crescere a un ritmo sostenuto. Apple ha dichiarato che Swift avrà *compatibilità binaria* con le future versioni di Swift e con Objective-C. Questo non

CoreFoundation	Libreria di interfacce di programmazione che permette a framework e librerie di condividere codice e dati. Presentata nell'app <i>Twitter</i> <sup>®</sup> <i>Searches</i> del Capitolo 4 e usata più volte nel corso del libro.
CoreLocation	Per determinare posizione e orientamento di un iPhone e configurare e temporizzare l'erogazione di eventi georeferenziati.
CoreMedia	Per creare, riprodurre e amministrare audio e video.
Core Motion	Per ricevere e gestire eventi dell'accelerometro e altri dati dai sensori di movimento.
CoreTelephony	Usato per ricevere informazioni sull'operatore telefonico dell'utente.
EventKit	Permette alla nostra app di accedere ai dati nell'app <i>Calendario</i> per aggiungere e modificare eventi.
HealthKit	Per registrare e tracciare dati di salute e benessere.
HomeKit	Per controllare apparecchi compatibili con l'Home Automation Protocol di Apple.
JavaScriptCore	Per eseguire JavaScript in un'app.
MobileCoreServices	Comprende tipi standard e costanti.
MultipeerConnectivity	Consente alle app di scoprire e comunicare con servizi distribuiti da dispositivi iOS nelle vicinanze. La comunicazione avviene tramite reti Wi-Fi, connessioni Wi-Fi peer-to-peer dirette tra device o via reti personali Bluetooth.
NewsstandKit	Permette alle app di scaricare ed elaborare contenuto dell' <i>Edicola</i> (abbonamenti a quotidiani e periodici).
PassKit	Per creare, distribuire, aggiornare e interagire con pass amministrati dall'app <i>Passbook</i> . Su iPhone 6 e iPhone 6 Plus, PassKit si occupa anche delle carte di credito per Apple Pay.
PushKit	Abilita le app alla registrazione per ricevere dati inviati in modalità push da server remoti.
QuickLook	Mostra anteprime dei file persino se sono in formati non direttamente supportati dall'app (esempio classico, i documenti Microsoft Office).
Social	Permette alle app di integrarsi con i servizi di social networking per svolgere attività come pubblicare messaggi e foto in nome e per conto degli utenti.
StoreKit	Supporto degli acquisti in-app per l'elaborazione delle transazioni.
SystemConfiguration	Determina lo stato e la disponibilità della rete su un iPhone.
UIAutomation	Usato per integrare il collaudo automatizzato dell'interfaccia utente.
WebKit	Permette alle app di renderizzare contenuto web direttamente, senza ricorrere al browser Safari.

**Tabella 1.15** Strato dei framework Core OS per accedere al kernel di iOS.

Framework	Descrizione
Accelerate	Per matematica complessa ed elaborazione di immagini. Comprende funzioni per operare su vettori e matrici, elaborare segnali digitali, gestire numeri molto grandi e altro.
CoreBluetooth	Per comunicare con apparecchi <i>Bluetooth Low Energy</i> (Bluetooth LE), come cardiofrequenzimetri, dispositivi per il fitness, sensori di prossimità e così via.

(continua)

*(segue)*

ExternalAccessory	Permette a iPhone di interagire con accessori indipendenti autorizzati, collegati via Bluetooth o con il connettore del dock.
LocalAuthentication	Abilita le app all'autenticazione degli utenti con passphrase o TouchID (per esempio per validare un acquisto).
Security	Per mettere in sicurezza dati utilizzati in un'app.
System	Funzioni del sistema operativo BSD e API POSIX.

## Web service

I *web service* sono componenti software memorizzati su un computer e accessibili via Internet da un'app (o altro componente software) situata su un computer diverso. Con i web service si possono creare *mashup* e sviluppare rapidamente app attraverso la combinazione di web service complementari, spesso creati da organizzazioni eterogenee e magari uniti a flussi di informazioni di vario genere. 100 Destinations (<http://www.100destinations.co.uk>), per esempio, combina foto e tweet da Twitter con le capacità di mappatura di Google Maps per offrire un'esperienza di esplorazione del mondo mediante le foto scattate da altri. Programmableweb (<http://www.programmableweb.com/>) fornisce un elenco di oltre 11.150 API e 7300 mashup, con tanto di guide alla creazione di mashup personalizzati. La Tabella 1.16 elenca alcuni web service di grande diffusione. Secondo Programmableweb, le tre API più usate per creare mashup sono Google Maps, Twitter e YouTube. Google fornisce numerosi web service, che possiamo studiare all'indirizzo <http://code.google.com/apis/gdata/docs/directory.html>.

Nel Capitolo 4 usiamo i web service di Twitter. La Tabella 1.17 elenca alcuni mashup piuttosto diffusi nell'utilizzo.

**Tabella 1.16** Alcuni web service molto usati (<http://www.programmableweb.com/apis/directory/1?sort=mashups>).

Fonte dei web service	Utilizzo
Google Maps	Servizi di mappatura
Twitter	Microblogging
YouTube	Ricerca video
Facebook	Social networking
Instagram	Condivisione di foto
Foursquare	Geolocalizzazione mobile
LinkedIn	Social networking per la professione
Groupon	Social commerce
Netflix	Videonoleggio
eBay	Aste
Wikipedia	Enciclopedia collaborativa
PayPal	Pagamenti
Last.fm	Radio via Internet
Amazon eCommerce	Shopping di libri e mille altri prodotti

Salesforce.com	Customer Relationship Management (CRM)
Skype	Telefonia Internet
Microsoft Bing	Ricerca
Flickr	Condivisione di foto
Zillow	Prezzi degli immobili
Yahoo Search	Ricerca
WeatherBug	Meteo

**Tabella 1.17** Alcuni mashup molto usati sul Web.

URL	Descrizione
<a href="http://twikle.com/">http://twikle.com/</a>	Twikle usa web service di social networking per aggregare notizie condivise online.
<a href="http://trendsmap.com/">http://trendsmap.com/</a>	TrendsMap usa Twitter e Google Maps. Permette di tracciare i tweet per posizione geografica e vederli istantaneamente su una mappa.
<a href="http://www.dutranslation.com/">http://www.dutranslation.com/</a>	Il mashup Double Translation usa i servizi di traduzione di Bing e Google per fornire traduzioni da e per oltre 50 linguaggi e permetterci di confrontare i risultati.
<a href="http://musicupdated.com/">http://musicupdated.com/</a>	Music Updated usa i web service di Last.fm e YouTube. Possiamo usarli per tracciare i nostri artisti preferiti e seguire la pubblicazione di album, concerti e altro.

## L'ambiente integrato di sviluppo Xcode 6<sup>®</sup>

Per programmare in Swift ci servirà l'ambiente integrato di sviluppo (IDE, *Integrated Development Environment*) Xcode 6, che comprende il supporto dei linguaggi Swift, Objective-C, C++ e C. L'editor di Xcode fornisce colorazione della sintassi, autoindentazione, autocompletamento e altro. È disponibile gratuitamente su Mac App Store; si veda per i dettagli il capitolo "Prima di iniziare". La Tabella 1.18 elenca numerose funzioni chiave di Xcode 6 ([https://developer.apple.com/library/ios/documentation/DeveloperTools/Conceptual/WhatsNewXcode/Articles/xcode\\_6\\_0.html](https://developer.apple.com/library/ios/documentation/DeveloperTools/Conceptual/WhatsNewXcode/Articles/xcode_6_0.html)), più alcune importanti aggiunte delle versioni più recenti. Nel corso del libro faremo più volte riferimento a Xcode 6 come Xcode.

**Tabella 1.18** Funzioni fondamentali in Xcode 6 e altre versioni recenti.

Funzione	Descrizione
<b>Nuove funzioni chiave in Xcode 6</b>	
Playground	Un <i>playground</i> è una finestra di Xcode nella quale possiamo digitare codice Swift che viene eseguito mentre lo si digita, cosa che permette di correggere immediatamente gli errori e vedere i risultati del codice (output testuali, grafici, animazioni e altro). Non è più necessario passare dalla compilazione e dall'esecuzione per procedere al debug, risparmiando così tempo di sviluppo. I playground forniscono inoltre funzioni interessanti come le timeline, che permette di esaminare nel tempo l'esecuzione di un algoritmo, per esempio usando un cursore per visionare i risultati di ciascuna iterazione di un ciclo oppure ogni fotogramma di un'animazione.

(continua)



*(segue)*

<i>Read-Eval-Print-Loop (REPL)</i>	Il <i>Read-Eval-Print-Loop</i> (REPL) è uno strumento di debugging per interagire con un'app in funzione. Al pari di un playground, lo si può usare per scrivere istruzioni che vengono eseguite immediatamente, ma REPL non dispone delle capacità del playground di occuparsi di grandi quantità di codice o renderizzare risultati complessi (come grafica e animazioni), per quanto si possa usarlo per lanciare file Swift. REPL è direttamente utilizzabile in Xcode 6 o nell'app Terminale di OS X.
Interface Builder, progettazione per storyboarding e Auto Layout adattativo	Interface Builder di Xcode consente di progettare le interfacce utente delle nostre app mediante tecniche di drag and drop. Tipicamente creiamo dentro Storyboard interfacce utente che mappano graficamente i percorsi che un utente può intraprendere nella nostra app, compresa ogni schermata e le transizioni tra le schermate. Per default, i nuovi storyboard per iOS usano le funzioni di Auto Layout di Interface Builder, le quali permettono di creare interfacce utente responsive che si adeguano ai cambiamenti di orientamento, alle dimensioni di schermo e alle azioni dell'utente. Prima di Xcode 6, occorre preparare più storyboard a supporto di varie dimensioni (per esempio iPhone e iPad) e orientamento ( <i>portrait</i> e <i>landscape</i> , cioè verticale e orizzontale) del dispositivo. Xcode 6 aggiunge l'Auto Layout adattativo, che permette di usare un solo storyboard per l'interfaccia utente della nostra app per tutte le dimensioni di schermo e gli orientamenti supportati.
Simulatore di iOS 8	Il simulatore di iOS 8 è uno strumento per il collaudo su Mac di un'app iOS. Fornisce supporto di vari iPhone e iPad. Comprende anche nuovi iPhone e iPad "elastici", che permette di testare progetti di Auto Layout adattativo ridimensionando la finestra del simulatore.
Resa grafica istantanea in Interface Builder	Quando progettiamo la nostra interfaccia utente (compresi componenti personalizzati) e scriviamo codice che la gestisce, Interface Builder renderizza istantaneamente il risultato e ci dà modo di vedere come funzionerà la nostra app.
View debugger	Il view debugger permette di trovare e correggere i problemi di interfaccia utente. Quando lo usiamo per mettere in pausa la nostra app, visualizza la resa grafica in modalità 3D per aiutarci a individuare l'errore e ad agire sul codice corrispondente.
Funzioni di progettazione di giochi	Xcode ora supporta la progettazione di giochi 2D e 3D così come un editor di particelle utilizzabile per creare animazioni complesse come fuoco, fumo, acqua in movimento e fuochi artificiali.

**Altre funzioni di Xcode**

Compilatore LLVM	LLVM ( <i>Low Level Virtual Machine</i> ; <a href="http://llvm.org">llvm.org</a> ) – sviluppato primariamente da Chris Lattner, creatore di Swift – è un compilatore veloce e open source per Swift, Objective-C e molti altri linguaggi pienamente integrato in Xcode.
Fix-it	Segnala errori e suggerisce correzioni al codice mentre digitiamo; non c'è bisogno di compilare. È una funzione propria anche dei playground.
Debugger LLDB	Comprende un veloce ed efficiente motore di debugging multicore.
Editor Assistant	Quando lavoriamo con l'editor di Xcode e la finestra divisa in due sezioni, l'editor Assistant anticipa altri file di cui potremmo avere bisogno. Per esempio, se stiamo lavorando su un'interfaccia utente in Interface Builder, l'editor Assistant mostra il file di codice corrispondente oppure, se stiamo definendo una nuova classe Swift che eredita da una superclasse, l'editor Assistant visualizza quest'ultima.
Simulazione di posizione geografica	Ora possiamo selezionare nel simulatore un elenco di posizioni geografiche allo scopo di collaudare app che usano il framework Core Location, che consente di determinare posizione e direzione di spostamento del dispositivo.

EditorVersion	Se usiamo sistemi di controllo del codice sorgente come Git, l'editorVersion di Xcode può mostrarci più versioni del codice fianco a fianco per un confronto, per visionare un log degli eventi passati e altro.
Instruments	Instruments aiuta a testare la nostra app, monitorare l'allocazione di memoria, tracciare grafici di prestazioni con OpenGL ES, tracciare l'interazione dei processi di sistema, individuare ed eliminare colli di bottiglia prestazionali e altro.
XCTest	XCTest è uno strumento di unit testing. Gli unit test aiutano a garantire che i componenti software di un'app funzionino come previsto. Possiamo usare XCTest per automatizzare il collaudo delle funzioni, visionare i risultati dei test, controllare se si sono verificati problemi di qualsiasi tipo e correggerli. Si può sapere di più sullo unit testing visitando <a href="http://bit.ly/TestingWithXcode">http://bit.ly/TestingWithXcode</a> .

## Il simulatore di iOS

Il simulatore di iOS permette la verifica funzionale di app iOS su un Mac, per non obbligare all'acquisto di un dispositivo iOS. Se intendiamo distribuire la nostra app, il test su apparecchi reali è essenziale. Non tutte le funzioni dei vari device sono disponibili nel simulatore. Per esempio, la fotocamera – usata di frequente nelle app iOS – non funziona nel simulatore. In compenso possiamo riprodurre molti dei gesti a tocco singolo e multiplo, per mezzo di tastiere e mouse Mac (Tabella 1.19).

**Tabella 1.19** Gesti sul simulatore di iOS (<http://bit.ly/gesti-simulatore-ios>).

Gesto	Azione del simulatore
Tocco ( <i>tap</i> )	Un clic del mouse.
Doppio tocco ( <i>double tap</i> )	Due clic del mouse.
Tocco prolungato ( <i>touch and hold</i> )	Clic prolungato tenendo il dito sul mouse.
Trascinamento ( <i>drag</i> )	Clic prolungato tenendo il dito sul mouse e trascinando.
Spazzata ( <i>swipe</i> )	Clic prolungato tenendo il dito sul mouse, spostamento nella direzione del puntatore e rilascio del mouse.
Colpetto ( <i>flick</i> )	Clic prolungato tenendo il dito sul mouse, spostamento nella direzione del gesto e rilascio rapido del mouse.
Pizzico ( <i>pinch</i> )	Pressione prolungata sul tasto Opt. Appaiono due cerchi che simulano i due tocchi. Spostiamo i cerchi nella posizione di partenza e, con un clic prolungato del mouse, trasciniamoli nella posizione di arrivo.
Trascinamento con due dita ( <i>two-finger drag</i> )	Posizionamento del puntatore dove si verificherà il trascinamento. Pressione prolungata sul tasto Opt e spostamento dei cerchi nella posizione iniziale. Pressione prolungata del tasto Maiusc e spostamento dei cerchi nella posizione centrale, seguito dal rilascio del tasto Maiusc. Pressione prolungata del tasto Maiusc e del mouse con spostamento dei cerchi nella direzione del trascinamento con due dita, e rilascio di entrambi.
Rotazione ( <i>rotate</i> )	Posizionamento del puntatore dove si verificherà la rotazione. Pressione prolungata del tasto Opt e spostamento dei cerchi nella posizione di partenza. Mantenendo premuto Opt, pressione prolungata del tasto Maiusc. Spostamento dei cerchi nella posizione centrale della rotazione e rilascio del tasto Maiusc. Pressione prolungata del mouse, rotazione dei cerchi nella posizione finale e rilascio del tasto Opt.

Nonostante il simulatore di iOS possa imitare cambiamenti di orientamento tra orizzontale e verticale e il gesto di scuotimento, non possiede un sistema per simulare le letture dell'accelerometro e di vari altri sensori. Possiamo naturalmente installare l'app su un iPhone o iPad; i requisiti per farlo sono descritti nel Capitolo 2. Da Xcode 7 in avanti, non è più necessario essere membri paganti del programma sviluppatori di Apple per installare un'app su un apparecchio a scopo di collaudo. (Il capitolo "Prima di iniziare" di questo libro contiene ulteriori informazioni.)

## La programmazione orientata agli oggetti

C'è sempre maggiore richiesta di software nuovo e potente, ma realizzare app in modo veloce, corretto ed economico rimane un obiettivo sfuggente. Gli *oggetti* o, più precisamente, le *classi* da cui provengono gli oggetti, sono essenzialmente componenti software *riutilizzabili*. Esistono oggetti data, orario, audio, video, automobile, persona e così via. Quasi qualsiasi *sostantivo* può essere ragionevolmente rappresentato come un oggetto software in termini di *attributi* (per esempio nome, colore, dimensioni) e *comportamenti* (per esempio calcolo, spostamento e comunicazione). I gruppi di sviluppatori software possono adottare un approccio di progettazione e implementazione modulare, orientata agli oggetti, per migliorare la produttività rispetto alle tecniche precedentemente diffuse come la programmazione strutturata. I programmi orientati agli oggetti sono spesso più facili da capire, correggere e modificare.

### L'automobile come oggetto

Supponiamo di voler aumentare la velocità di un'auto premendo il pedale dell'acceleratore. Che cosa deve avvenire perché si possa fare? Prima di poter guidare un'auto, qualcuno deve averla progettata. Un'auto nasce di solito come una serie di disegni tecnici, simili a quelli di un'abitazione. Questi disegni comprendono il progetto di un pedale dell'acceleratore. Il pedale nasconde all'autista i complessi meccanismi che effettivamente aumentano la velocità dell'auto, come il pedale del freno nasconde i meccanismi che la riducono e lo sterzo nasconde i meccanismi che permettono le svolte. Tutto questo permette a persone prive di competenze tecniche su motori, freni e sterzi di guidare ugualmente un'auto e di farlo con facilità.

Così come è impossibile cucinare usando il progetto di una cucina, è impossibile guidare i progetti di un'auto. Qualcuno deve costruirla a partire da quei progetti. Un'automobile autentica possiede un vero pedale dell'acceleratore, il che tuttavia non è ancora abbastanza; un'auto non accelera da sola (non ancora!) e dunque l'autista deve premere il pedale per accelerare.

### Metodi e classi

L'esempio dell'auto ci aiuta a mostrare alcuni concetti base della programmazione orientata agli oggetti. Eseguire un compito in un programma richiede un *metodo*. Il metodo contiene le istruzioni che eseguono il compito e nasconde queste istruzioni all'utente, proprio come l'acceleratore nasconde il meccanismo che aumenta la velocità dell'auto. In Swift, creiamo una unità di programma detta *classe* per contenere l'insieme dei metodi

che eseguono il compito cui la classe è addetta. Per esempio, una classe che rappresenta un conto bancario potrebbe contenere un metodo per depositare denaro, uno per ritirare soldi e un terzo per interrogare la banca allo scopo di ottenere il saldo. Una classe è concettualmente simile ai disegni tecnici dai quali nasce un'auto, che contengono gli schemi del pedale dell'acceleratore, dello sterzo, dei freni e così via.

## Instanziazione

Proprio come l'auto va costruita prima di poter essere guidata, bisogna costruire un *oggetto* di una classe prima che un programma possa eseguire i compiti definiti dai metodi della classe in questione. La procedura per farlo si chiama *instanziazione*. Ci si riferisce a un oggetto come a un'*istanza* della sua classe.

## Riutilizzo

I disegni tecnici dei componenti di un'auto possono essere *riutilizzati* più volte per costruire più auto e allo stesso modo si può riutilizzare una classe più volte per costruire più oggetti. Il riutilizzo delle classi al momento di costruirne di nuove fa risparmiare tempo e fatica. Inoltre porta alla costruzione di sistemi più affidabili e efficaci, perché spesso le classi che già esistono e da cui si derivano classi nuove sono state approfonditamente sottoposte a testing, debugging e ottimizzazione delle *prestazioni*. Proprio come le parti intercambiabili sono state fondamentali per la Rivoluzione Industriale, le classi riutilizzabili sono cruciali per l'evoluzione del software favorita dalla tecnologia a oggetti.

## Messaggi e chiamate a un metodo

Premere il pedale dell'acceleratore invia un *messaggio* all'auto perché svolga un compito, nel caso andare più veloce. Analogamente, si inviano messaggi a un oggetto. Ciascun messaggio è implementato come una *chiamata di un metodo*, che ordina al metodo dell'oggetto di eseguire la propria attività. Per esempio, un programma potrebbe chiamare il metodo `deposito` per incrementare il saldo di un oggetto `conto bancario`.

## Attributi e proprietà

A parte saper svolgere dei compiti, un'auto possiede *attributi*, come colore, numero di porte, quantità di carburante nel serbatoio, velocità attuale e totale dei chilometri percorsi). Anche gli attributi dell'auto sono rappresentati come parte del progetto ingegneristico che, per esempio, prevederà un contachilometri e un indicatore del livello di carburante. Nel guidare un'auto vera, questi attributi viaggiano assieme all'auto e sono propri di questa auto e di nessun'altra. Ogni auto sa dire quanto carburante vi sia nel proprio serbatoio, ma non quanto ve ne sia nei serbatoi di altre auto.

Similmente, un oggetto possiede attributi che porta con sé durante l'uso in un programma. Questi attributi sono specificati come parte della classe dell'oggetto. Per esempio, un oggetto `conto bancario` dispone di un attributo `saldo` che rappresenta la quantità di denaro nel conto. Ogni oggetto `conto bancario` conosce il proprio saldo ma non i saldi di altri conti. Gli attributi sono specificati dalle proprietà della classe.

## Incapsulamento e occultamento delle informazioni

Le classi (e i loro oggetti) *incapsulano* le loro proprietà e i loro metodi. Le proprietà e i metodi di una classe (e quelli del suo oggetto) sono intimamente correlati. Gli oggetti possono comunicare con altri oggetti, ma normalmente non è loro permesso di sapere come gli altri oggetti sono stati implementati; questi dettagli sono nascosti dentro gli oggetti. Questo occultamento di informazioni, come vedremo, è cruciale per una buona ingegneria del software.

## Ereditarietà

Si può creare rapidamente una nuova classe di oggetti per *ereditarietà*; la nuova classe (chiamata *sottoclasse*) nasce con le caratteristiche di una classe esistente (la *superclasse*), per poi personalizzarsi con caratteristiche distintive. Nell'analogia dell'auto, un oggetto di classe *cabrio* è certamente un oggetto della classe più generale *automobile* ma, più nello specifico, il tettuccio gode di certe proprietà.

## Protocolli

Swift supporta anche un concetto noto come *protocollo*, il quale descrive un insieme di metodi che può essere chiamato su un oggetto, qualcosa di simile al concetto di *interfaccia* presente in altri linguaggi di programmazione. Per default, i metodi di un protocollo devono essere implementati da una qualunque classe che dichiara di essere *conforme* (cioè di implementare) il protocollo stesso. I protocolli pensati per l'uso in codice Swift e Objective-C possono anche contenere metodi opzionali; in questi casi, una classe può implementare un *sottoinsieme* dei metodi del protocollo. Per default, è necessario implementare tutti i metodi presenti nei protocolli Swift.

Le classi possono essere conformi a un numero di protocolli arbitrario, proprio come un'auto implementa interfacce separate per guidare, azionare la radio, regolare la climatizzazione e così via. Esattamente come i costruttori implementano in ciascun modello le stesse funzioni in modi differenti, le classi possono conformarsi ai metodi di un protocollo in modi diversi. Per esempio, un sistema software può prevedere un protocollo "backup" che offre i metodi *registra* e *ripristina*. Le classi possono implementare questi metodi in modi differenti, a seconda di che genere di dati viene trattato, come programmi, testo, audio, video e così via, e il genere di apparecchi su cui questi elementi sono contenuti.

## Schemi di progettazione

Gli *schemi di progettazione* sono architetture collaudate e riutilizzabili che i programmatori impiegano per risolvere problemi ricorrenti durante lo sviluppo di software orientato agli oggetti. Nello sviluppo di app iOS, gli schemi di progettazione portano un vocabolario condiviso di progettazione tra gli sviluppatori di app iOS. Chi aderisce a schemi ben noti per iOS, può abbreviare la fase di progettazione dell'app e sfruttare le potenti funzioni fornite dalle API di iOS. (Tra i libri fondamentali sugli schemi di progettazione vi sono *Design Patterns: Elements of Reusable Object-Oriented Software*, di Gamma, Helm, Johnson e Vlissides, ©1994, Addison-Wesley, e *Cocoa Design Patterns*, di Buck e Yacktman, ©2010, Addison-Wesley.)

La nozione di schema di progettazione è nata in architettura. Gli architetti usano nella progettazione un insieme di elementi consolidati e collaudati, come archi e colonne per la costruzione di edifici resistenti; questi elementi possono allora essere visti come schemi di progettazione architettonici.

Lo schema di progettazione più comune che useremo nello sviluppo di app si chiama *Model-View-Controller* (MVC), che separa i dati dell'app (contenuti nel *model*) dai componenti per la presentazione grafica (la *vista*) e dalla logica di elaborazione (il *controllore*). Consideriamo un'app di rubrica contatti. Quando l'utente utilizza l'interfaccia grafica per aggiungere un contatto, il controllore dell'app aggiorna il model, cioè memorizza il contatto in un database o file. Quando il model cambia, avvisa il controllore il quale, a sua volta, aggiorna la vista, in questo caso per mostrare l'elenco contatti aggiornato. Nello scrivere app iOS, useremo un insieme esteso di schemi di progettazione comuni in Cocoa Touch. Presenteremo vari schemi man mano che li incontriamo nella progettazione e nell'implementazione delle app di questo libro. Per una panoramica di quelli più comuni usati nello sviluppo per iOS e OS X, si può visitare la pagina <http://bit.ly/iOSDesignPatterns>.

## Prova dell'app Tip Calculator nei simulatori di iPhone e iPad

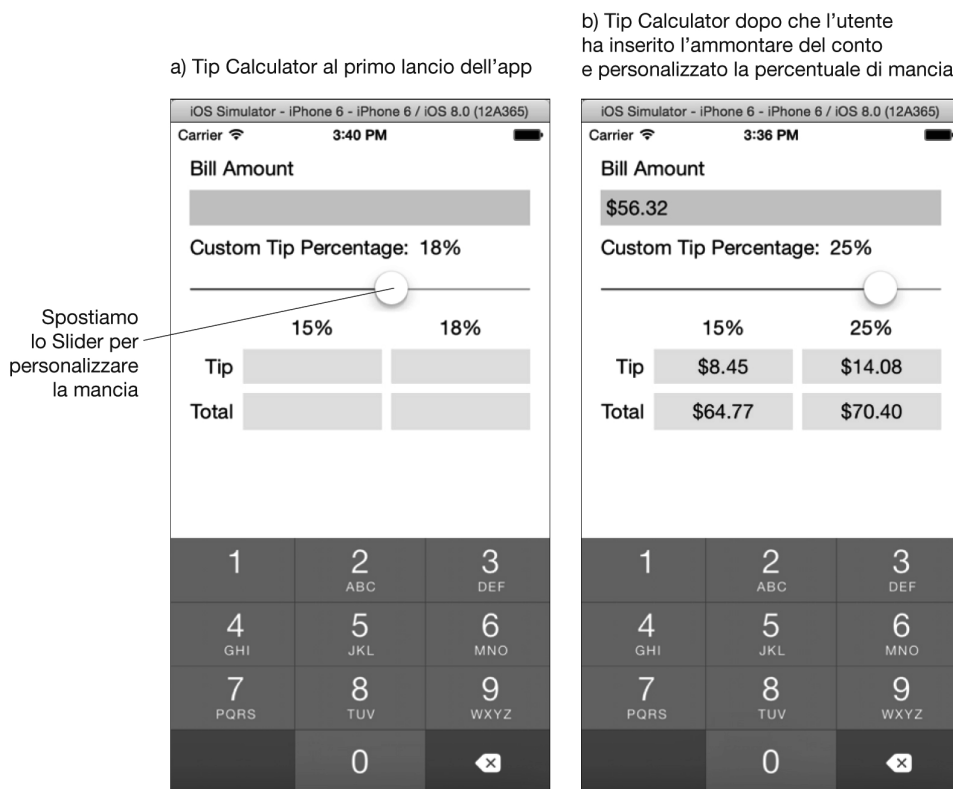
In questo paragrafo, lanceremo la nostra prima app iOS e interagiremo con essa attraverso i simulatori di iPhone e iPad. *Tip Calculator* (Figura 1.2a) – che realizzeremo nel Capitolo 3 – calcola e mostra le mance possibili dato un conto da pagare al ristorante. Mentre digitiamo ogni singola cifra del conto attraverso il tastierino numerico, l'app calcola e visualizza la mancia in base a una percentuale (tipica americana) del 15% e a una seconda percentuale personalizzata (Figura 1.2b). La percentuale personalizzata viene specificata spostando uno *Slider*, il che aggiorna le *Label* relative e visualizza le mance standard e personalizzata nella colonna a destra delle *Label* gialle, sotto lo *Slider* (Figura 1.2b). Abbiamo scelto il 18% come percentuale personalizzata di default perché molti ristoranti americani aggiungono questa percentuale per tavolate di sei o più persone.

## Prova dell'applicazione completa tramite il simulatore di iPhone

I passi seguenti mostrano come provare l'app.

1. Controllare la configurazione. Il capitolo “Prima di iniziare” di questo libro ci consente di capire se abbiamo impostato correttamente il computer.
2. Individuare la cartella dell'app. Apriamo una finestra del *Finder* e portiamoci sulla cartella *Documenti* > *Examples* o su quella dove abbiamo copiato gli esempi di programmazione.
3. Aprire il progetto *TipCalculator*. Apriamo la cartella *TipCalculator* e facciamo doppio clic sul file *TipCalculator.xcodeproj* per aprire il progetto in Xcode.
4. Lanciare l'app *Tip Calculator*. In Xcode, facciamo clic sul selettore *Scheme* a destra dei pulsanti *Run* e *Stop* nell'angolo superiore sinistro dell'IDE di Xcode (Figura 1.3), poi selezioniamo il simulatore *iPhone 6*. Facciamo clic sul pulsante *Run* (o premiamo *Cmd+R*) per lanciare l'app nel simulatore.

5. Inserire il totale di un conto, per esempio la cifra 56.32, usando il tastierino numerico. Se facciamo un errore, premiamo il pulsante di cancellazione (⌫) nell'angolo inferiore destro del tastierino per cancellare l'ultima cifra digitata. Il tastierino non contiene un separatore decimale. L'app legge i caratteri digitati, li trasforma in numero, lo divide per 100 e mostra il risultato nella *Label* blu con due decimali, per poi calcolare il valore delle mance. Questa app usa il formato di valuta locale specifico di iOS per mostrare i valori monetari nel formato della valuta dell'utente. In Italia verranno visualizzati durante la digitazione i valori € 0,05, € 0,56, € 5,63 e € 56,32.
6. Selezionare una percentuale personalizzata di mancia attraverso lo *Slider*; le *Label* nella colonna a destra sotto lo *Slider* mostrano la mancia corrispondente e il totale finale del conto. Trasciniamo il cursore dello *Slider* verso destra fino a raggiungere una percentuale personalizzata del 25%. Il valore dello *Slider* cambia all'istante mentre trasciniamo il cursore e l'app aggiorna tutti i valori, fino a quando rilasciamo il cursore (Figura 1.2b).
7. Chiudiamo l'app con un clic sul pulsante *Home* del simulatore o con un clic sul pulsante *Stop* di Xcode, o ancora selezionando *iOS Simulator > Quit iOS Simulator* dalla barra dei menu.



**Figura 1.2** Tip Calculator al primo lancio dell'app e quando l'utente inserisce l'ammontare del conto e la percentuale personalizzata di mancia.



**Figura 1.3** I pulsanti Run e Stop e il selettore Scheme.

## Prova dell'applicazione completa tramite il simulatore di iPad

Per provare l'app usando il simulatore di iPad, facciamo clic sul selettore *Scheme* e selezioniamo il simulatore *iPad Air*. Un clic sul pulsante *Run* lancia l'app nel simulatore.

## Prova delle app del libro

Per avere un'idea generale delle funzioni che impareremo a codificare in questo libro, proviamo come descritto le app presentate nei Capitoli da 2 a 8.

## Come nasce una grande app?

Con oltre un milione e mezzo di app su App Store, come creare un'app iOS che il pubblico vorrà cercare, scaricare, usare e raccomandare ad altri? Dobbiamo pensare a che cosa rende un'app divertente, utile, interessante, attrattiva e durevole. Un nome indovinato, un'icona suggestiva e una descrizione intrigante potrebbero aumentare le possibilità di vedere scelta la propria app su App Store. La Tabella 1.20 elenca alcune caratteristiche irrinunciabili di una grande app.

**Tabella 1.20** Caratteristiche di una grande app.

---

### Caratteristiche di una grande app

---

#### Grandi giochi

- Intrattiene e diverte.
- Sfida.
- Livelli a difficoltà progressiva.
- Mostra i punteggi e ricorda quelli più elevati.
- Fornisce un feedback sonoro e visivo.
- Offre versioni single-player, multiplayer e di gioco in rete.
- Contiene animazioni di alta qualità.
- Innova con l'uso della realtà aumentata, proiettando informazioni virtuali sull'ambiente del mondo fisico. Si usa soprattutto nelle app che elaborano video.

#### Utility indispensabili

- Funzioni di vera utilità e informazioni accurate.
  - Aumenta la produttività personale e professionale.
  - Rende più comodo il lavoro (per esempio per organizzare un elenco di attività o tenere sotto controllo le spese).
- 

(continua)



(segue)

- Rende più informato l'utente.
- Fornisce informazioni essenziali (Borsa, notizie, meteo, traffico).
- Fornisce servizi locali georeferenziati (coupon di attività locali, prezzi più convenienti, consegna a domicilio).

#### **Caratteristiche generali**

- È aggiornata alle ultime funzioni di iOS.
  - Accede alle informazioni personali dell'utente solo se necessario.
  - Non richiede l'iscrizione a social media.
  - Funziona come dichiarato.
  - I bug vengono corretti prontamente.
  - Segue le *iOS Human Interface Guidelines* di Apple (<http://bit.ly/iOSMobileHIG>).
  - Supporta i gesti classici di iOS in modo standard.
  - Parte in fretta.
  - È reattiva.
  - Economizza con intelligenza memoria, banda e batteria.
  - Usa icone di alta qualità su App Store e al proprio interno.
  - È creativa e per qualche verso inedita.
  - Viene usata con regolarità.
  - Usa media di qualità.
  - Sfrutta le caratteristiche del dispositivo.
  - È intuitiva e facile da usare.
  - È accessibile alle persone con disabilità (si veda l'*Accessibility Programming Guide for iOS*).
  - Dà agli utenti una ragione e i mezzi per pubblicizzarla (per esempio dando loro la possibilità di passare un punteggio straordinario su Facebook o Twitter).
  - Offre contenuto ricco per il contesto dell'app.
  - Parla la lingua dell'utente, nel testo, nell'audio e nella grafica.
  - È un'app universale che funziona al meglio su tutti i dispositivi iOS. Nel libro creeremo diverse app universali.
- 

## **iOS e la sicurezza**

È responsabilità di un bravo sviluppatore iOS tenere al sicuro le informazioni personali dell'utente dentro l'app e quando vengono trasmesse su Internet. La nostra app dovrebbe richiedere e memorizzare solo le informazioni necessarie. Per esempio, se non fornisce servizi georeferenziati, non dovrebbe chiedere la posizione dell'utente. La Tabella 1.21 sintetizza le considerazioni di sicurezza su iOS trattate nella *Security Overview* della iOS Developer Library.

Per approfondire il tema dello sviluppo di app iOS sicure, si legga in dettaglio la guida *iOS Security* all'indirizzo [https://www.apple.com/business/docs/iOS\\_Security\\_Guide.pdf](https://www.apple.com/business/docs/iOS_Security_Guide.pdf).

## **Trova il mio iPhone e la cancellazione remota**

iPhone e iPad contengono l'app *Trova il mio iPhone*, che aiuta in caso di smarrimento o furto. Deve essere stato impostato iCloud nelle impostazioni. Per rintracciare il dispositivo, ci si collega a iCloud da qualunque computer, seguendo l'indirizzo <http://www.icloud>.

com/find. Apparirà una mappa della posizione approssimativa del device: possiamo far emettere un suono all'apparecchio o fargli mostrare un messaggio per aiutare la persona che lo avesse rinvenuto. Se il dispositivo non è recuperabile, la funzione di cancellazione a distanza elimina tutti i dati personali e ripristina le impostazioni di fabbrica. Queste funzioni possono richiedere l'indicazione dell'ID Apple e della password relativa.

**Tabella 1.21** Gli argomenti della OS Security Overview (<http://bit.ly/iOSSecurityOverview>).

Argomento	Descrizione
Risk Assessment and Threat Modeling	Definizione e valutazione dei rischi, con determinazione delle minacce potenziali e i sistemi per mitigarle.
Code Security	Tratta di come individuare falle di sicurezza nel nostro codice, la firma del codice per autenticazione, il sandboxing delle app perché eseguano solo le attività che dichiarano di eseguire e il principio del privilegio minore (dividere l'app in più parti e a ciascuna assegnare solo i permessi richiesti per il suo compito specifico).
Authentication and Authorization	Discute autenticazione (verifica dell'identità di utente e rete) e autorizzazione (concedere all'utente o a un server i permessi per svolgere un compito preciso e limitato).
Cryptographic Services	Illustra la comunicazione sicura e la memorizzazione dei dati cifrata, così che i dati possano essere letti o modificati solo dalle persone autorizzate.

## Documentazione e forum iOS

La Tabella 1.22 elenca parte della documentazione disponibile sul sito iOS Developer di Apple. La maggior parte di questi documenti è disponibile gratuitamente a partire dalla pagina [developer.apple.com](http://developer.apple.com). Man mano che si approfondiscono le tematiche di sviluppo per iOS e nascono domande, problemi, dubbi o richieste, il sito offre forum dove confrontarsi e scoprire dettagli o novità non presenti nella documentazione (Tabella 1.23).

**Tabella 1.22** Documentazione online (perlopiù presso [developer.apple.com](http://developer.apple.com)) per sviluppatori iOS.

Titolo	Descrizione
<i>iOS App Programming Guide</i>	Novità di Xcode
<i>iOS Human Interface Guidelines</i>	Panoramica su Xcode
<i>Getting Started</i>	Introduzione a Cocoa
<i>Programming with Objective-C</i>	Linee guida per il codice Cocoa
<i>Objective-C Runtime Programming Guide</i>	Guida per la compatibilità di SDK
<i>Accessibility Programming Guide for iOS</i>	Codice di esempio
<i>Game Center Programming Guide</i>	Novità di iOS 8
<i>Social Framework Reference</i>	Guida alla libreria Standard di Swift
<i>The Swift Programming Language</i>	Il linguaggio di programmazione Swift

**Tabella 1.23** Forum iOS.

URL	Descrizione
<a href="https://devforums.apple.com/community/ios">https://devforums.apple.com/community/ios</a>	Forum dove porre domande e trovare risposte da parte di altri sviluppatori e ingegneri Apple.
<a href="http://stackoverflow.com/questions/tagged/ios">http://stackoverflow.com/questions/tagged/ios</a>	Sito di programmazione contenente innumerevoli domande e risposte su iOS.
<a href="http://iphonedevsdk.com/">http://iphonedevsdk.com/</a>	Forum sullo sviluppo per iPhone, tutorial, codice dimostrativo e altro.
<a href="http://www.raywenderlich.com/forums/viewforum.php?f=18">http://www.raywenderlich.com/forums/viewforum.php?f=18</a>	Contiene un form molto attivo sullo sviluppo per iOS oltre a consigli, trucchi e tutorial.
<a href="http://iosdeveloperforums.com/">http://iosdeveloperforums.com/</a>	Conversazioni su sviluppo per iOS, trucchi e segreti di Xcode, sviluppo di giochi, tutorial.
<a href="http://forums.macrumors.com/forumdisplay.php?f=135">http://forums.macrumors.com/forumdisplay.php?f=135</a>	Contiene un forum sulla programmazione per iPhone e iPad.

## Riepilogo

Questo capitolo presenta il mondo di iOS. Abbiamo illustrato le caratteristiche del sistema operativo, con link a varie app di grande diffusione, gratuite e a pagamento, disponibili su App Store. Abbiamo imparato i gesti a tocco singolo e multiplo, nonché come effettuarli sui dispositivi veri e sul simulatore. Si sono esaminate le disponibilità di sensori per ogni apparecchio così come le funzioni di accessibilità di serie in iOS. Abbiamo presentato il linguaggio di programmazione Swift ed elencato i framework di Cocoa Touch per iOS che permettono un rapido sviluppo di app. Abbiamo introdotto anche alcune funzioni essenziali di Xcode. Si è parlato velocemente dei concetti base della programmazione orientata agli oggetti, come classi, oggetti, metodi, proprietà e altro. Abbiamo provato l'app *Tip Calculator* e riflettuto su che doti possieda una grande app, oltre a introdurre il documento *Security Overview* per iOS. Abbiamo infine fornito link alla documentazione online fondamentale e a siti e forum utili per entrare in contatto con la community degli sviluppatori, per porre domande e ricevere risposte. Nel prossimo capitolo realizzeremo la nostra prima app per iOS usando solo tecniche di programmazione visiva, senza scrivere del codice. L'app mostrerà testo e due immagini. Scopriremo anche diverse nozioni sull'accessibilità di iOS e l'internazionalizzazione delle app.

assicura la compatibilità del codice sorgente tra le versioni nuove e quelle vecchie. Apple conta di fornire convertitori di codice che aiutino ad aggiornare il codice sorgente Swift esistente con le funzioni che verranno aggiunte al linguaggio.

## Mescolare codice Swift e Objective-C

Riteniamo che nella maggior parte dei casi le app iOS esistenti diventeranno un misto di Swift e Objective-C, invece che essere convertite integralmente. In questo libro tutto il codice applicativo è scritto in Swift e usiamo il framework Cocoa Touch quando è necessario.

## Il framework Cocoa Touch® di iOS

Il framework Cocoa si è evoluto partendo dai progetti di NeXT. OpenStep è stato sviluppato come API di programmazione orientata agli oggetti per la realizzazione di un sistema operativo. Dopo che Apple ha acquisito NeXT, OpenStep si è evoluto in Rhapsody e molte delle librerie di base divennero parte della API Yellow Box. Rhapsody e Yellow Box hanno finito per diventare rispettivamente il sistema operativo OS X e Cocoa. Cocoa Touch è la versione di Cocoa per dispositivi iOS, tipicamente vincolati nelle risorse di memoria, processore e potenza delle batterie rispetto alle controparti desktop e portatili e differenti nelle funzioni. I tre framework principali per lo sviluppo su OS X e iOS sono Foundation, AppKit e UIKit, che presentiamo di seguito.

### Foundation

Il framework Foundation – in Cocoa e Cocoa Touch – comprende la classe `NSObject` per la definizione del comportamento di un oggetto. Foundation possiede anche classi per tipi base, per memorizzare dati, elaborare testo e stringhe, calcolare differenze tra date e orari, lavorare con le notifiche tra app e molto altro.

### AppKit

Il framework AppKit di Cocoa serve a sviluppare le interfacce utente di un'app OS X. AppKit fornisce controlli (finestre, menu, pulsanti, pannelli, campi testo, finestre di dialogo), gestione degli eventi, supporto dei gesti e altro. Supporta anche la condivisione del contenuto tra servizi, integrazione con iCloud, stampa, accessibilità (per utenti disabili), notifiche push, grafica e altro.

### UIKit

Il framework UIKit di Cocoa Touch è simile ad AppKit, però ottimizzato per lo sviluppo di interfacce utente per app da caricare su dispositivi iOS. UIKit comprende controlli di interfaccia Multi-Touch, gestione degli eventi basati sul movimento, elaborazione dei dati provenienti dai sensori (prossimità, movimento, accelerazione, luce ambiente, giroscopio) e altro.

## Altri framework Cocoa Touch

Il framework Cocoa Touch permette di accedere comodamente alle funzioni di iOS e incorporarle nella nostra app. I framework aiutano a creare app che rispecchiano il look and feel unico di iOS. Le Tabelle 1.12–1.15 elencano i framework e ne forniscono brevi descrizioni. Altre informazioni si possono ricavare dalla sezione *Frameworks* della *iOS Developer Library* alla pagina <http://developer.apple.com/library/ios/navigation/index.html#section=Frameworks>.

**Tabella 1.12** Lo strato del framework Cocoa Touch che genera app grafiche basate su eventi.

Framework	Descrizione
AddressBookUI	Mostra informazioni sui contatti contenute nell'app omonima.
EventKitUI	Modifica, crea e mostra eventi del calendario dall'interno della nostra app.
GameKit	Funzioni vocali, di rete Bluetooth e altre da usare in giochi o altre app.
MapKit	Aggiunge immagini cartografiche e satellitari ad app georeferenziate. Permette annotazioni sulle mappe, identifica aree su una mappa usando l'overlay e altro.
MessageUI	Crea messaggi di posta dall'interno di un'app, dalla qualche crea e invia anche SMS.
NotificationCenter	Mostra informazioni dell'app in <i>Centro Notifiche</i> e permette agli utenti di compiere alcune attività, come rispondere a un messaggio o aggiornare lo stato di un social network.
PhotosUI	Incorpora foto, video e comandi di editing nella nostra app.
Twitter	Aggiunge alla nostra app la possibilità di pubblicare tweet.
UIKit	Classi per creare e gestire un'interfaccia utente, compresi eventi, disegno, finestre, viste e controlli Multi-Touch. Ne parliamo nell'app <i>Welcome</i> del Capitolo 2 e se ne vedranno diversi usi nel corso del libro.
iAd	Framework per la pubblicità in-app usato per inserire annunci pubblicitari in un'app.

**Tabella 1.13** Strato dei framework per aggiungere audio, video, grafica e animazioni alla nostra app.

Framework	Descrizione
AVFoundation	Interfaccia di registrazione e riproduzione audio (simile ad Audio Toolbox). Comprende gestione e modifica degli asset mediali, acquisizione e riproduzione di video, gestione dei brani audio, organizzazione dei metadati dei media, panning stereofonico, sincronizzazione sonora e un'interfaccia Objective-C per determinare formato, qualità di campionamento e numero di canali dei file audio. Comprende anche classi per riprodurre una sequenza di media e registrare i campionamenti in un file di dati. Usato nell'app <i>Cannon</i> del Capitolo 6.
AssetsLibrary	Framework per accedere alle librerie mediali dell'utente, comprendenti foto e video caricati sul dispositivo o memorizzati nell'app <i>Foto</i> . Permette anche alla nostra app di registrare negli album fotografici nuove foto e nuovi filmati.
AudioToolbox	Interfaccia di registrazione e riproduzione audio per sonoro in streaming e avvisi.

(continua)

*(segue)*

AudioUnit	Interfaccia per utilizzare i plug-in di elaborazione audio di iOS.
CoreAudio	Framework per la dichiarazione dei tipi di dato e delle costanti usati da altre interfacce di Core Audio.
CoreGraphics	API per disegnare, renderizzare immagini, regolare il colore e i gradienti, trasformare le coordinate negli spazi colore e gestire documenti PDF. Usato nell'app <i>Cannon</i> del Capitolo 6.
CoreMidi	Permette alle app di interagire con apparecchi MIDI ( <i>Musical Instrument Digital Interface</i> ) come sintetizzatori.
CoreText	API per la disposizione del testo e la manipolazione dei font.
CoreVideo	API basate su C per riproduzione, modifica ed elaborazione di video.
GLKit	Fornisce funzioni che semplificano la creazione di app OpenGL ES (per esempio giochi).
GameController	Consente a un'app di supportare controller esterni per giochi connessi a un dispositivo iOS.
ImageIO	Supporto per la lettura e la scrittura di vari formati di immagine.
MediaAccessibility	Funzioni per accedere alle preferenze di sottotitolazione.
MediaPlayer	Individua e riproduce file audio e video dall'interno di un'app.
Metal	Dà a un'app l'accesso al processore grafico dell'apparecchio per grafica 3D con accelerazione hardware e al trasferimento dalla CPU al chip grafico delle attività di calcolo più intensive.
OpenAL	Una libreria open source per il sonoro tridimensionale.
OpenGL ES	Un sottoinsieme di OpenGL per grafica 2D e 3D.
QuartzCore	Usato per creare animazioni ed effetti renderizzati in hardware per avere le massime prestazioni.
SceneKit	Per aggiungere modelli 3D all'interfaccia utente di un'app. Usato comunemente nei giochi.
SpriteKit	Per giochi 2D basati su sprite. Supporta animazioni, simulazioni fisiche, riconoscimento delle collisioni e gestione degli aventi. Usato nell'app <i>Cannon</i> del Capitolo 6.

**Tabella 1.14** Strato dei framework Core Services.

Framework	Descrizione
Accounts	Consente alle app l'accesso alle informazioni sull'account dell'utente contenute nel database Accounts, al posto di memorizzare le credenziali dell'utente stesso. Gli utenti possono concedere alla nostra app di usare i loro account, evitando così di dover inserire le loro credenziali.
AdSupport	Permette a un'app di pubblicare inserzioni pubblicitarie e determinare la configurazione dell'apparecchio rispetto a esse.
AddressBook	Framework per accedere ai contenuti dei <i>Contatti</i> dell'utente.
CFNetwork	Framework per usare nell'app protocolli di rete per attività come autenticazione di server HTTP e HTTPS, lavoro con server FTP, creazione di connessioni cifrate e altro.
CloudKit	Usato per controllare il trasferimento di dati tra app e iCloud.
CoreData	Framework per attività correlate al ciclo di vita dell'oggetto. Usato nell'app <i>Address Book</i> del Capitolo 8.