

# **Scrivere una applicazione html, css e javascript per Android**

Prima di tuffarsi dentro , vorrei stabilire rapidamente il campo di gioco. In questo capitolo, definisco termini chiave, confrontare i pro e i contro dei due approcci di sviluppo più comuni, e dare un corso accelerato sulle tre tecnologie web di base usati in questo libro.

## ***Web Apps Versus applicazioni native***

In primo luogo , vorrei definire cosa intendo per web app e app nativa e prendere in considerazione i loro pro e cons .

### **Che cosa è una Web App ?**

Per me , una web app è fondamentalmente un sito web che è specificamente ottimizzato per l'uso su un smartphone. Il contenuto del sito può essere qualsiasi cosa, da un opuscolo standard di piccola impresa sito di una mutui per un calorico giornaliero inseguitore - il contenuto è irrilevante . gli caratteristiche distintive di una web app è che l'interfaccia utente (UI) è costruito con il web tecnologie standard , è disponibile a un URL ( pubblica, privata , o forse dietro un login) , ed è ottimizzato per le caratteristiche di un dispositivo mobile . Una web app non è installata sul telefono , non è disponibile nel mercato Android , e non è scritto con Java

### **Che cosa è una applicazione nativa ?**

Al contrario, le applicazioni native sono installate sul telefono Android , che hanno accesso al hardware (altoparlanti , accelerometro , fotocamera , ecc ) , e sono scritte con Java . La caratteristica distintiva di una app nativa , tuttavia , è che è disponibile in Android Play Store - una caratteristica che ha catturato l'immaginazione di un orda di imprenditori software in tutto il mondo , me compreso.

## ***Pro e contro***

Applicazioni diverse hanno esigenze diverse. Alcune applicazioni offrono una migliore vestibilità con le tecnologie web di altri. Conoscere i pro e i contro di ogni approccio vi aiuterà a prendere una decisione migliore su quale percorso è appropriato per la situazione .

### **Qui ci sono i pro di sviluppo app nativa :**

- Milioni di proprietari registrati di carte di credito sono un click
- È possibile accedere a tutte le caratteristiche hardware fredde del dispositivo

### **Qui ci sono i contro di sviluppo app nativa :**

- Devi pagare per diventare uno sviluppatore Android
- La vostra applicazione funziona solo su telefoni Android
- È necessario sviluppare l'utilizzo di Java
- Il ciclo di sviluppo è lento ( sviluppare , compilare, distribuire , ripetere )

## Qui ci sono i pro di sviluppo web app:

- Gli sviluppatori Web possono utilizzare i loro strumenti di authoring attuali
- È possibile utilizzare il web design attuale e lo sviluppo delle competenze
- L'app funziona su qualsiasi dispositivo dotato di un browser web
- È possibile correggere i bug in tempo reale
- Il ciclo di sviluppo è veloce

Qui ci sono i contro di sviluppo web app:

- Non è possibile accedere a tutte le caratteristiche hardware fredde del telefono
- Si deve rotolare il proprio sistema di pagamento se si vuole far pagare per l'applicazione
- Può essere difficile da ottenere effetti dell'interfaccia utente sofisticate

## Quale approccio è giusto per voi?

Ecco dove diventa eccitante. La natura sempre - in linea del telefono Android crea un ambiente in cui i confini tra una web app e una applicazione nativa diventano sfocate . Ci sono anche alcune caratteristiche poco note del browser web di Android ( vedi Capitolo 6) che permettono di prendere una web app offline , se vuoi. Cosa c'è di più , molti di terze parti progetti - di cui PhoneGap è la soluzione più notevole , sono attivamente in via di sviluppo che consentono agli sviluppatori web di prendere una web app e pacchettizzare come applicazione nativa per Android e altre piattaforme mobili . Per me , questa è la miscela perfetta . Posso scrivere nella mia lingua madre, rilasciare un prodotto come una web app pura ( per Android e tutti gli altri dispositivi che dispongono di un browser moderno ) , e utilizzare la stessa base di codice per creare una versione avanzata nativa che può accedere al hardware del dispositivo e potenzialmente essere venduto nel mercato Android.

Questo è un ottimo modo per creare un modello "fremium" per la tua app-consentire l'accesso gratuito alla web app e la versione a pagamento per la versione più ricca di funzionalità native.

## Programmazione Web

Le tre principali tecnologie che useremo per costruire applicazioni web sono HTML , CSS e Java Script. Parleremo presto di ciascuno per assicurarsi che siamo tutti sulla stessa pagina prima di passare a cose fantasiose.

## Introduzione a HTML

Quando si sta navigando sul web , le pagine che stai visualizzando sono solo documenti di testo salvate sul computer di qualcun altro. Il testo in una pagina web tipica è avvolto in HTML tag , che raccontano al browser la struttura del documento. Con queste informazioni, il browser può decidere come visualizzare le informazioni in un modo che abbia senso.

Si consideri il frammento di pagina web mostra l'Esempio 1-1. Sulla prima riga, la stringa "Benvenuto!" è avvolto in un paio di tag **h1**. Si noti che il tag di apertura e il tag di chiusura sono leggermente diversi: il tag di chiusura ha una barra (/ ) come secondo elemento, mentre il tag aperto non ha una barra.

La disposizione del testo nel tag h1 dice al browser che le parole racchiuse sono una voce, che farà sì che venga visualizzata in dimensione piu' grande e

in grassetto sulla riga. Ci sono anche **h2** , **h3** , **h4** , **h5** , **h6** e tag di titolo . Più basso è il numero, più importante l'intestazione , in questo modo il testo avvolto in un tag **h6** sarà più piccolo (cioè , meno importante -looking) rispetto al testo avvolto in un tag h3 .

Dopo il tag h1 nell'esempio 1-1, ci sono due linee avvolte nei tag **p** . Questi sono chiamati tag di paragrafo. I browser visualizzano ogni paragrafo sulla propria riga. Se il paragrafo è sufficiente a superare la larghezza della finestra del browser, il testo scenderà giù a proseguire sulla riga successiva. In entrambi i casi , una riga vuota verrà inserito dopo l' paragrafo per separarlo dal successivo elemento della pagina.

Esempio 1-1 . snippet HTML

```
<h1> Ciao ! </h1>
<p> Grazie per aver visitato la mia pagina web. </p>
<p> Spero che vi piaccia . </p>
```

Si può anche mettere tag HTML all'interno di altri tag HTML .

Esempio 1-2 mostra una lista non ordinata dei tag ( ul ) che contiene tre elementi della lista (li). In un browser , questo appare come un elenco puntato che riporta con ogni voce su una riga. Quando si dispone di un tag o tag all'interno di un altro tag, il tag interni sono chiamati elementi figli, o figli, del tag genitore. Quindi in questo esempio , i tag li sono figli del genitore ul.

Esempio 1-2 . elenco non ordinato

```
<ul>
<li> Pizza < / li>
<li> Birra < / li>
<li> cani < / li>
< / ul >
```

I tag coperti finora sono tutti i tag di blocco. La caratteristica che definisce i tag di blocco è che vengono visualizzati su una linea propria, senza elementi a destra o sinistra. Ecco perché il titolo , paragrafi , e voci di elenco progrediscono verso il basso la pagina anziché attraverso di esso. L'opposto di un tag di blocco è un tag inline, che, come il nome implica , può apparire in fila. Il tag enfasi (EM) è un esempio di un tag inline, e assomiglia a questo:

```
<p> <em> davvero </em> spero vi piaccia . </p>
```

Il nonno dei tag inline - e forse la più bella caratteristica di HTML è il tag a. La "a" sta per l'ancora, ma a volte si riferisce ad essa come un link o collegamento ipertestuale.

Il testo avvolto in un tag di ancoraggio è cliccabile, in modo che cliccando su di essa fa sì che il browser per caricare una nuova pagina HTML. Per dire al browser che nuova pagina deve essere caricata, dobbiamo aggiungere quello che si chiama un attributo al tag. Gli attributi sono denominati dai valori che si inseriscono in un tag aperto. In un'ancora, è possibile utilizzare l'attributo href per specificare la posizione della pagina di destinazione. Ecco un link alla home page di Google :

`<a href="http://www.google.com/"> Google < / a>`

Questo potrebbe sembrare un po' un guazzabuglio, se non siete abituati a leggere HTML , ma si dovrebbe essere in grado di individuare l'URL per la home page di Google. Starò a vedere un sacco di un tag e gli attributi href in tutto il libro , in modo da prendere un minuto per ottenere la testa intorno questo caso non ha senso a prima vista.

Ci sono un paio di cose da tenere a mente per quanto riguarda gli attributi. I diversi Tag HTML consentono diversi attributi.

È possibile aggiungere più attributi a un tag aperto separandoli con spazi.

Non si può mai aggiungere attributi a un tag di chiusura.

Ci sono centinaia di combinazioni possibili di attributi e tag, ma non preoccupatevi, non ci resta che preoccuparsi di una dozzina in tutto questo libro. Il frammento di codice HTML che stiamo guardando normalmente risiede nella sezione corpo di un documento HTML completo.

Un documento HTML è costituito da due sezioni: la testa ed il corpo.

Il corpo è dove mettere tutto il contenuto che si desidera che gli utenti vedano. La testa contiene informazioni sulla pagina , la maggior parte dei quali è invisibile agli user.

Il corpo e la testa sono sempre avvolti in un elemento HTML.

Esempio 1-3 mostra il frammento nel contesto di un adeguato documento HTML. Per ora la sezione testa contiene un elemento titolo, che dice al browser quale testo da visualizzare nella barra del titolo della finestra.

Esempio 1-3. Un documento HTML corretto

```
<!doctype html>
<html>
<head>
<title> La mia bellissima pagina </ title>
</ head>
<body>
<h1> Ciao! </ h1>
<p> Grazie per aver visitato la mia pagina web. </ p>
<p> Spero che vi piaccia. </ p>
<ul>
<li> Pizza </ li>
<li> Birra </ li>
<li> cani </ li>
</ ul>
</ body>
</ html>
```

Normalmente, quando si utilizza il browser web che stai visualizzando le pagine che sono ospitate su Internet. Tuttavia, i browser sono perfettamente in grado di visualizzare documenti HTML che sono sulla vostra macchina locale. Per mostrarvi cosa intendo, vi invito ad aprire un editor di testo e inserire il codice dell'esempio 1-3.

Una volta terminato l'inserimento del codice di esempio 1-3, salvarlo sul desktop come test.html e poi aprirlo con Chrome trascinando il file sull'icona di

Chrome dell'applicazione o l'apertura di Chrome e selezionare File - Apri file. Facendo doppio clic test.html funzionerà pure, ma potrebbe aprire con un editor di testo o un altro browser, a seconda delle impostazioni.

## **Introduzione ai CSS**

Come si è visto , i browser rendono alcuni elementi HTML con stili diversi ( per esempio , intestazioni sono grandi e in grassetto , i paragrafi sono seguiti da una linea vuota , e così via). Questi stili sono molto semplici e sono principalmente lo scopo di aiutare il lettore a capire la struttura e il significato del documento. Per andare al di là di questo struttura semplice di base, si utilizzano fogli di stile CSS( CSS ).

CSS è un linguaggio di fogli di stile che si utilizza per definire la presentazione visiva di un Documento HTML. È possibile utilizzare i CSS per definire le cose semplici, come il colore del testo, la dimensione e stile (grassetto, corsivo , ecc ), o le cose complesse come il layout di pagina , gradienti , opacità , e molto più.

Esempio 1-4 mostra una regola CSS che indica al browser di visualizzare il testo nell'elemento corpo con il colore rosso. In questo esempio, il corpo è il selettore (questa specifica quale sia colpiti dalla regola) e le parentesi graffe racchiudono la dichiarazione (la regola stessa). La dichiarazione comprende una serie di proprietà e dei loro valori. In questo esempio , il colore è proprietà, e rosso è il valore della proprietà colore .

Esempio 1-4 . Una regola CSS semplice

```
body { color: red; }
```

I nomi delle proprietà sono predefiniti nelle specifiche CSS , il che significa che non è possibile crearli. Ogni proprietà si aspetta un valore appropriato , e non ci può essere un sacco dei valori appropriati e formati di valore per una data proprietà . Ad esempio, è possibile specificare i colori con le parole chiave predefinite come il red, oppure utilizzando Codici di colore HTML, che utilizza una notazione esadecimale:

un simbolo 'cancelletto ( # ) seguito da tre coppie di cifre esadecimali ( 0 - F ) che rappresentano ( da sinistra a destra ) rosso, verde , e il blu ( colore rosso è rappresentato come #FF0000). Proprietà che si aspettano misure può accettare valori come 10px , 75% e 1em . Esempio 1-5 mostra alcuni comuni dichiarazioni di css.

Il codice del colore indicato per il colore di sfondo corrisponde al CSS "Grigio ".

Esempio 1-5 . Alcune dichiarazioni comuni CSS

```
body {  
  colore: rosso ;  
  background-color: # 808080 ;  
  font-size: 12px ;  
  font-style: italic ;  
  font-weight: bold ;  
  font-family: Arial;  
}
```