

PRINCIPI DI PROGRAMMAZIONE WML

(aggiornato alla versione 1.2)

di Alessandro Demontis - 06/05/2002

Cosa é WML?

Il *Wireless Mark-up Language*, comunemente chiamato *WML*, é un linguaggio descrittivo per il protocollo WAP operante nelle comunicazioni web-based per la tecnologia mobile. In pratica, se HTML é il linguaggio con cui vengono create le pagine web per computers, WML é il linguaggio usato per la navigazione internet dei cellulari.

WML nasce nel 1998 dal consorzio WAP forum su base di due precedenti linguaggi, l' HDML della Openwave e il TTML della Nokia; é un linguaggio figlio di SGML tramite la specifica XML. Attualmente la versione standard di WML é la 1.3 anche se esiste la versione 2.0 definita dal Wap forum nel 2001 ma non ufficialmente rilasciata né mai standardizzata.

Dal punto di vista linguistico, il WML si presenta come una versione elementare di HTML 2, essendo priva dunque di tecnologie quali embed, frame, iframe, layer e CSS. E' capace però di interagire tramite moduli di input con script WMLscript, CGI, Perl, PHP e altri linguaggi server-based i quali restituiscono dati alla pagina WML originante.

La piattaforma per WML: il protocollo WAP e i browser WML

Il linguaggio WML é un linguaggio per pagine navigabili tramite dispositivi cellulari abilitati, e, in quanto tale, risente delle limitazioni hardware e software della tecnologia cellulare. Le pagine WML sono genericamente scarse, con immagini in bianco/nero, con minimo contenuto; pertanto i 'siti WML' sono divisi in più 'pagine' codificate in uno o più listati.

Ma come si accede ad una pagina WML? Allo stato attuale sono pochi i fornitori di servizi di telefonia cellulare con connessione WAP flat, mentre é ancora abbondante l' utilizzo di tariffe 'a consumo' che renderebbe sconveniente lo sviluppo e la consultazione a scopo didattico di pagine WML. Fortunatamente, esistono una serie di Browser WML installabili sul pc di casa, rilasciati da software house generalmente dipendenti o associate ai servizi di telefonia mobile o ai produttori di telefoni cellulari.

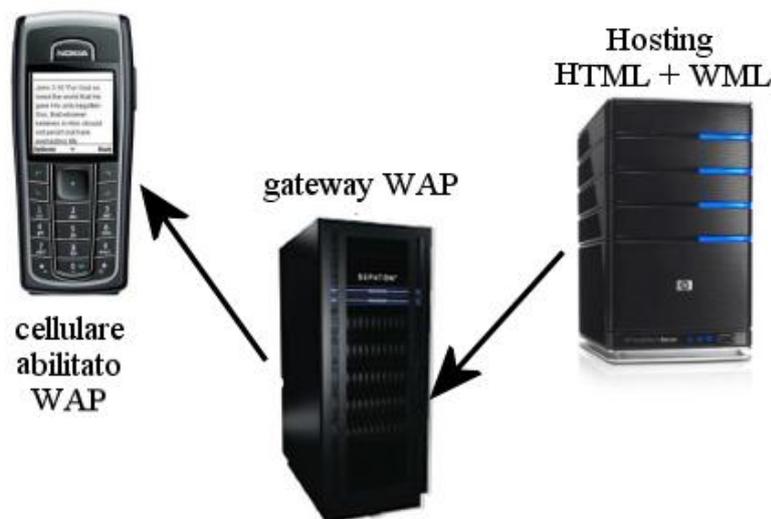
Per la stesura di questo manuale il browser di riferimento é stato il Winwap for Windows 4.2, rilasciato dalla Winwap Technologies, che può essere scaricato liberamente al seguente indirizzo:

http://www.winwap.com/desktop_applications/winwap_for_windows

Per quanto riguarda invece la produzione di pagine WML, ovviamente queste devono essere collocate su uno spazio web che offra agli utenti l'accesso WAP e che sia in comunicazione con i gateway di telefonia cellulare; alternativamente, quegli spazi web con solo supporto del WML lasciano il compito della comunicazione col gateway al dispositivo che si connette. Attualmente tutte le marche produttrici di telefoni cellulari hanno implementato questo secondo tipo di gestione, rendendo possibile consultare siti WML collocati negli stessi spazi web ospitanti siti HTML. Ciò torna utile anche a chi deve fornire servizi wap perchè potrà scegliere un hosting unico per i contenuti HTML e WML. La piattaforma server deve comunque supportare la trasmissione tramite il protocollo WAP. Vediamo ora come questo funziona.

Il WAP (Wireless Access Protocol) é il protocollo per l'acceso internet tramite telefonia mobile, standardizzato dal WAP forum nato nel 1997. Il protocollo WAP attualmente utilizzato é il 2.0, evoluzione del precedente WAP 1.1, rispetto al quale presenta notevoli modifiche e migliorie, prima fra tutte il supportare la specifica XHTML Mobile Profile, capace di visualizzare sui dispositivi di nuova generazione anche i fogli di stile tramite il linguaggio WAP-CSS.

Una piattaforma WAP si presenta strutturalmente in questo modo:



Come abbiamo detto, allo stato attuale, per chi sviluppa pagine WML é sufficiente scegliere un hosting che fornisca il supporto WML, e il resto é lasciato all'operatore di telefonia utilizzato dal cellulare WAP compatibile.

Per chi volesse settare il proprio hosting WEB/WAP sul proprio computer di casa, consiglio vivamente il serve Apache versione 1.3.19 il quale ha pieno supporto WAP/WML e al quale basterà aggiungere i seguenti MIME types:

- Text/vnd.wap.wml = .WML
- Text/vnd.wap.wmlscript = .WMLS

- Image/vnd.wap.wbmp = .WBMP

qualora si intendesse utilizzare sul lato server anche le applicazioni WAP e non solo pagine WML si devono aggiungere questi altri 2 MIME:

- Application/vnd.wap.wmlc = .WMLC
- Application/vnd.wap.wmlscriptc = .WMLSC

una volta configurato Apache in questo modo sarà possibile accedere con il cellulare o l' emulatore WAP alle pagine ospitate sul proprio server.

Sviluppo di pagine WML

Ovviamente, per sviluppare pagine WML, la prima cosa da fare é apprendere le basi di questo linguaggio. Per chi ha già qualche nozione di HTML sarà davvero un gioco da ragazzi, ma l' apprendimento di WML, se condotto nel modo giusto, é facile anche per chi é digiuno di programmazione web. Ho visto numerosi tutorial di WML scaricabili da internet, e dall' esperienza tratta da amici e conoscenti, posso dire che spesso questi tutorial risultano ostici ai non addetti ai lavori perchè generalmente presentano i siti WML paragonandoli ai siti HTML, un paragone non proprio azzeccato che a volte confonde e solleva interrogativi. In questo manuale utilizzerò un approccio diverso, che é poi quello su cui mi son basato per l' apprendimento, e che mi ha permesso di padroneggiare WML in meno di 3 giorni.

Per prima cosa dunque é necessario capire cosa sia una pagina WML e come sia costituita. Chiariamo subito che il termine 'pagina' nel caso del WML é usato impropriamente. A differenza dell' HTML, nel quale effettivamente un documento rappresenta una e una sola pagina di un sito, il linguaggio WML permette, con un unico documento, di contenere l' intero sito.

Per intenderci, se noi navighiamo un sito internet creato con HTML costituito da 3 pagine, avremo 3 documenti distinti, mentre se navighiamo un sito scritto con WML possiamo avere un solo documento.

Questo perchè un documento WML é più simile a una presentazione PowerPoint che a una pagina Web, e, proprio come una presentazione PowerPoint può contenere una o più slides, così il documento WML può contenere una o più CARDS.

L' elemento fondamentale delle pagine WML viene chiamato CARD, e l' insieme di una o più cards é detto DECK. Ogni card corrisponde a una pagina web, e un singolo deck corrisponde a un insieme di pagine web o a un intero sito. L' immagine seguente mostra un confronto tra le strutture di una presentazione Powerpoint, un sito web, e un deck WML.

Slides	Edit Browse	Edit Browse
<p>1</p>  <p>Title</p>	<pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE>Untitled</TITLE> </HEAD> <BODY> pagina 1 pagina 2 pagina 3 </BODY> </HTML></pre>	<pre><?xml version="1.0"?> <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml"> <wml> <card id="card1" title="pagina 1"> <p>questa é pagina 1</p> vai a pagina 2 vai a pagina 3 </card> <card id="card2" title="pagina 2"> <p>questa é pagina 2</p> </card> <card id="card3" title="pagina 3"> <p>questa é pagina 3</p> </card> </wml></pre>
<p>2</p>  <p>Long-term Goal</p>		
<p>3</p>  <p>Customer Wishes</p>		

HTML
un documento iniziale rimanda
a tre distinte pagine, un totale
di 4 documenti html

WML
un documento iniziale (deck)
rimanda a 2 pagine distinte (cards),
un totale di un solo documento wml

Chiarita la struttura dei documenti WML, possiamo passare ora ad impararne la composizione e la stesura. Come per tutti i linguaggi di mark-up, anche nel caso di WML bisogna utilizzare comandi chiamati TAG, i quali in genere sono accoppiati in 'Tag di apertura' e 'Tag di chiusura'. Fanno eccezione solo pochi tags che non hanno bisogno di chiusura.

Ogni documento WML deve iniziare con una dichiarazione che identifica al browser il tipo di linguaggio che lo compone, e la versione utilizzata. In linea di massima la dichiarazione é sempre:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

nel caso si voglia utilizzare la versione 1.1 di WML, oppure:

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
```

nel caso si voglia utilizzare la versione 1.2 di WML. Il contenuto del tag DOCTYPE, che qui nel manuale va a capo per esigenze di impaginazione, va scritto in una unica riga nel nostro editor WML. Subito dopo la dichiarazione, il primo tag da utilizzare é quello identificante l' apertura del deck WML, ed é sempre bene, all' inizio della stesura, scriverlo assieme al suo tag di chiusura

associato:

```
<wml>  
</wml>
```

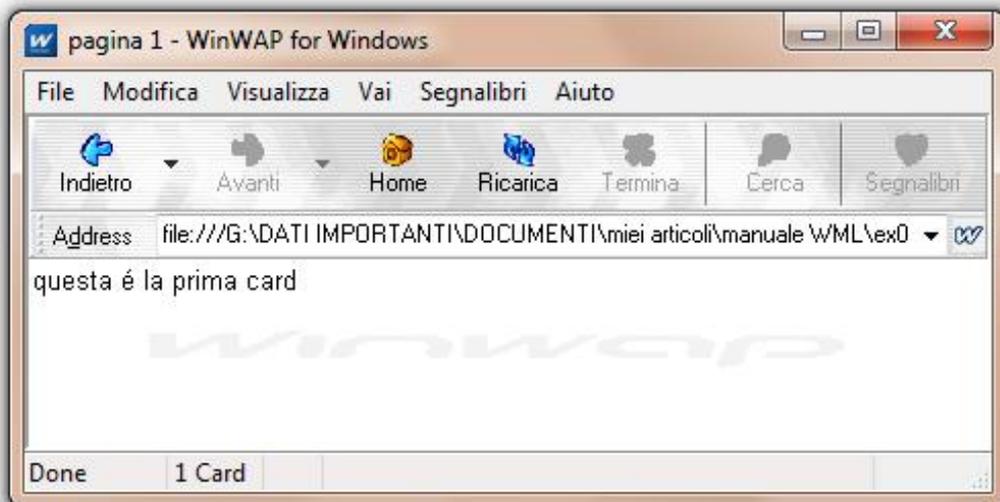
A questo punto siamo pronti per creare la nostra prima card tramite il tag CARD e il suo relativo tag di chiusura. Se si ha in mente di inserire più cards in un unico deck, ognuna di queste andrà identificata tramite l' attributo ID che dovrà contenere un nome univoco scritto tra virgolette e senza spazi. A ogni card si può anche associare un titolo tramite l' attributo TITLE:

```
<wml>  
  <card id="card1" title="pagina 1">  
    questa é la prima card  
  </card>  
</wml>
```

questo breve listato crea un deck WML costituito da una sola card, identificata dall' ID 'card1', e la pagina mostrata avrà titolo: 'pagina 1'.

Il contenuto della pagina sarà la semplice frase "questa é la prima card".

Il risultato é visibile nell' immagine seguente:



Aggiungiamo adesso al deck altre 2 cards, in modo da strutturare un sito composto di 3 pagine, ognuna con un contenuto diverso. Per farlo seguiamo queste indicazioni:

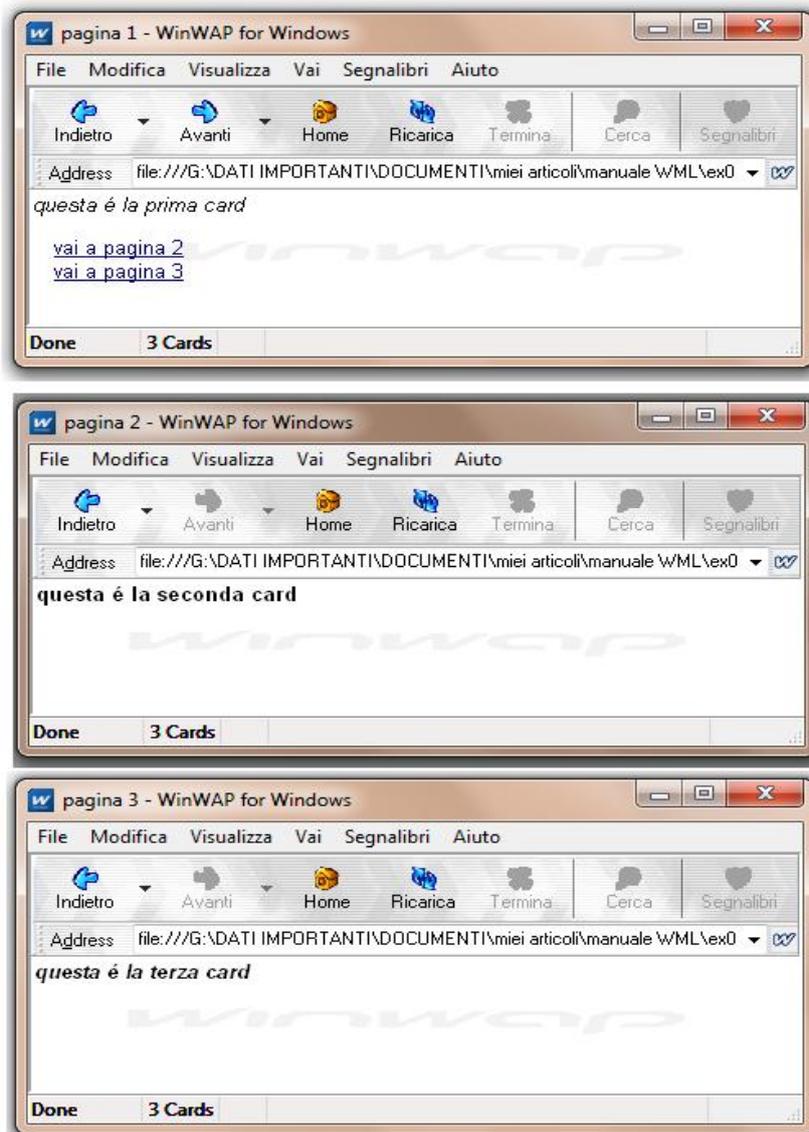
- Nella prima card, quella appena creata, formattiamo la frase col corsivo, tramite l' utilizzo del tag EM e relativa chiusura, e aggiungiamo i links a tutte le altre 3 cards;
- Nella card n°2 scriviamo una frase simile a quella di card1 ma con il testo grassetto, utilizzando il tag STRONG e relativa chiusura;
- L' ultima card, card3, conterrà la stessa frase ma stavolta formattata sia con EM che con STRONG.

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="pagina 1">
    <EM>questa é la prima card</EM>
    <P>
    <A HREF="#card2">vai a pagina 2</A><br />
    <A HREF="#card3">vai a pagina 3</A><br />
  </card>
  <card id="card2" title="pagina 2">
    <STRONG>questa é la seconda card</STRONG>
  </card>
  <card id="card3" title="pagina 3">
    <EM><STRONG>questa é la terza card</STRONG></EM>
  </card>
</wml>

```

e il risultato visivo delle 3 cards sarà il seguente:



In questo listato ho dovuto introdurre, oltre ai tags di formattazione per il corsivo e il grassetto, anche quelli responsabili della fine riga. Il tag BR, che é un tag ad auto-chiusura e si utilizza nella forma
 con uno spazio tra la R e la /, serve per andare a capo senza lasciare righe vuote, mentre il tag P va a capo costituendo un nuovo paragrafo dunque lasciando una riga vuota. La chiusura del tag P é opzionale.

Quando si utilizzano in sequenza più tags, é bene ricordarsi che essi vanno chiusi dal più interno verso il più esterno; nel listato in questione per esempio abbiamo prima aperto il tag EM, poi il tag STRONG, e al momento di chiuderli abbiamo chiuso prima STRONG e poi EM. Questa regola va rispettata in maniera ferrea per non incorrere in errori da parte dell' interprete WML.

Le tabelle

WML supporta le tabelle in maniera molto diversa e più elementare rispetto a quanto succede con HTML. Scordiamoci le tabelle annidate, le spaziature e i bordi personalizzabili nelle celle, lo sfondo raster etc; tutto ciò che le tabelle WML permettono è la definizione della struttura dichiarando anticipatamente il numero di colonne, e un allineamento diverso per ogni colonna. L' allineamento di default nelle celle WML è a sinistra, ma utilizzando l' attributo XML:LANG possiamo modificarlo (si pensi per esempio a quelle lingue che hanno la scrittura da destra a sinistra come l' ebraico o l' arabo). Il tag TABLE permette di associare alla tabella un ID e una CLASS, di definire il numero di colonne tramite l' attributo COLUMNS, e l' allineamento tramite l' attributo ALIGN; gli elementi ID e CLASS sono stati introdotti nel linguaggio con l' intenzione di permettere, nelle successive versioni, la fusione di WML con CSS, ma le versioni fino a WML 1.2 non ne fanno utilizzo.

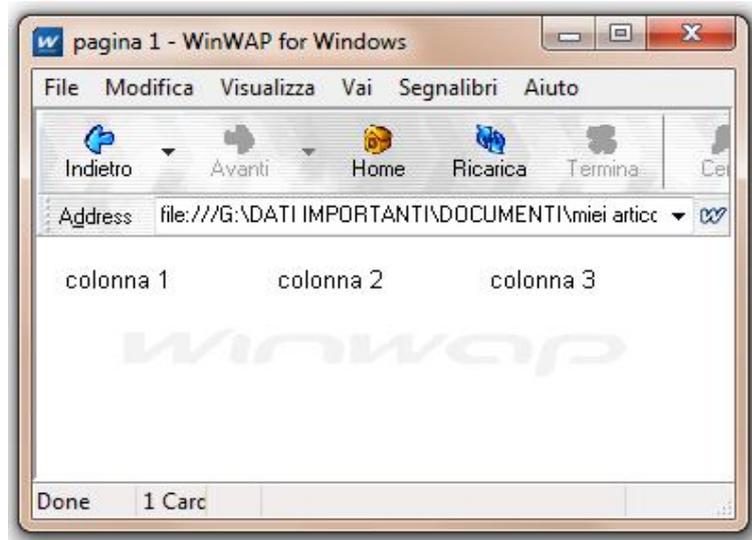
Costruiamo adesso la nostra prima tabella:

```
<TABLE columns="3" align="LCR">
<TR>
  <TD>colonna 1</TD>
  <TD>colonna 2</TD>
  <TD>colonna 3</TD>
</TR>
</TABLE>
```

integrandola nelle dichiarazioni base che costituiscono un deck WML, il listato completo sarà:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="pagina 1">
    <P>
      <TABLE columns="3" align="LCR">
        <TR>
          <TD>colonna 1</TD>
          <TD>colonna 2</TD>
          <TD>colonna 3</TD>
        </TR>
      </TABLE>
    </P>
  </card>
</wml>
```

Visualizzando il documento con Winwap avremo il seguente risultato:

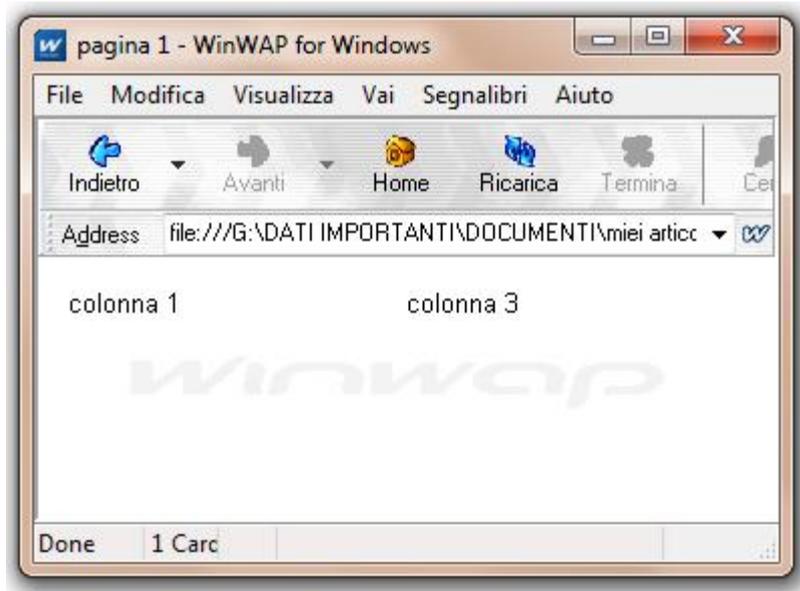


Come possiamo notare, la tabella viene mostrata senza i bordi che ne costituiscono l'inteleiatura. Questo tipo di visualizzazione dipende dall'interprete WML adottato dal software del server e dal dispositivo cellulare utilizzato. La definizione `ALIGN="LCR"` indica all'interprete l'allineamento del testo delle tre colonne: la prima è allineata a sinistra (L per Left), la seconda al centro (C per Center), e la terza a destra (R per Right).

Il linguaggio WML richiede che ogni colonna vuota in una tabella venga comunque dichiarata, quindi se per esempio volessimo scrivere solo nella prima e nella terza cella della riga, il listato andrà modificato come segue:

```
<TABLE columns="3" align="LCR">
<TR>
  <TD>colonna 1</TD>
  <TD></TD>
  <TD>colonna 3</TD>
</TR>
</TABLE>
```

il cui risultato sarà il seguente:



Nel caso le nostre celle contengano frasi lunghe, a seconda della capacità di visualizzazione del dispositivo cellulare o dell' emulatore WAP queste verranno tagliate di netto per non stravolgere la struttura della tabella; per evitare ciò, ed imporre che ogni cella mandi a capo i contenuti, dobbiamo assicurarci che la nostra tabella sia inserita all' interno di un paragrafo tramite il tag P e relativa chiusura, al quale valorizzeremo l' attributo MODE con il valore NOWRAP:

```
<P align="center" mode="nowrap">  
<TABLE columns="3" align="LCR">  
<TR>  
  <TD>colonna 1 molto lunga il testo deve essere mandato a  
capo</TD>  
  <TD></TD>  
  <TD>colonna 3</TD>  
</TR>  
</TABLE>  
</P>
```

il cui risultato é:



In linea di massima, il numero di colonne dichiarate e il numero di lettere presenti nella dichiarazione dell' allineamento dovrebbero essere congruenti, cioè se utilizziamo `COLUMNS="2"` anche in `ALIGN` dovremmo mettere 2 lettere; se ne dichiariamo solo una, qualunque essa sia, questa verrà attribuita alla prima colonna, mentre la seconda sarà assunta normalmente come allineata a sinistra o a destra a seconda della lingua utilizzata nelle impostazioni del computer.

Un altro fattore da tenere ben presente è che teoricamente secondo le specifiche WML se dichiariamo un certo numero di colonne (es: `COLUMNS="2"`) ma inseriamo invece all' interno di `TR` un numero diverso di colonne tramite `TD`, il contenuto di quelle eccedenti sarà aggiunto all' ultima colonna dichiarata. Questo però non è sempre vero, infatti alcuni interpreti in caso di discrepanza ignorano l' attributo `COLUMNS` e creano tante colonne quante sono le dichiarazioni `TD`.

Possiamo concludere qui la parte riguardante le tabelle ed iniziare a parlare dei contenuti grafici.

Le immagini

Per tanti anni l' utilizzo di immagini nelle pagine WML è stato limitato ai loghi, specialmente a causa del fatto che il protocollo WAP utilizza il formato WBMP graficamente molto scarno. La maggior parte dei programmi di grafica ormai permette di salvare immagini in questo formato: tenete presente che si tratta di un formato bianco/nero e che le immagini generalmente non devono superare i 150 pixel di grandezza in orizzontale.

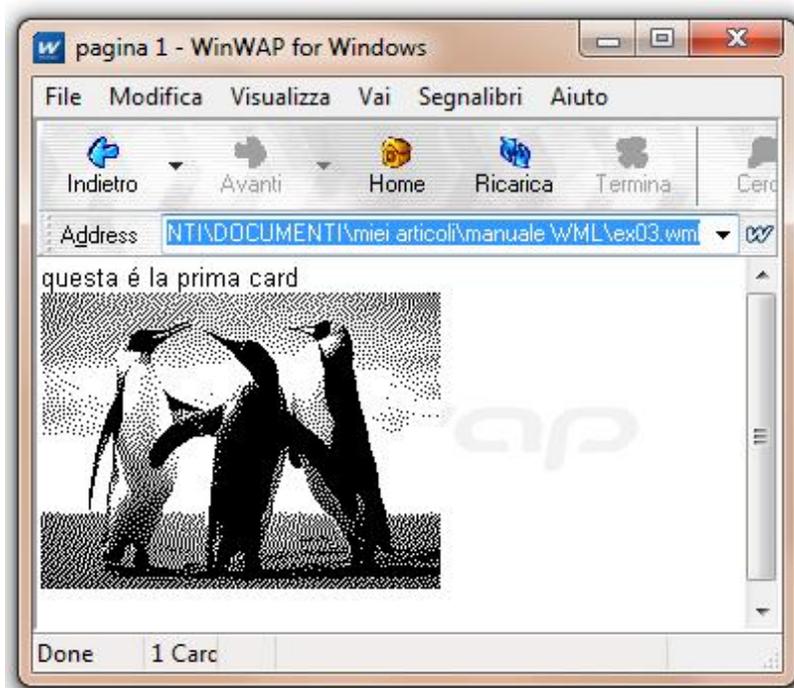
Il tag per inserire le immagini è `IMG`, ed è un tag senza chiusura, il cui attributo `SRC` permette di specificare il percorso e il nome dell' immagine da caricare. Come per l' HTML, questo percorso può essere locale o una URL. Tramite l' attributo `ALT` si può specificare un testo alternativo da mostrare nel

caso l' immagine non sia disponibile. Sono disponibili anche gli attributi ALIGN per indicare l' allineamento verticale dell' immagine rispetto al testo nella pagina (top, middle, bottom), HEIGHT per indicare quanti pixel dovrà essere alta, e WIDTH per indicare quanto dovrà essere larga. Se questi 3 attributi non vengono valorizzati l' interprete mostrerà l' immagine in basso rispetto al testo, con le sue dimensioni native.

Vediamo un esempio:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="pagina 1">
    questa é la prima card<br />
    <IMG SRC="Penguins.wbmp" ALT="pinguini">
  </card>
</wml>
```

avrà come risultato:



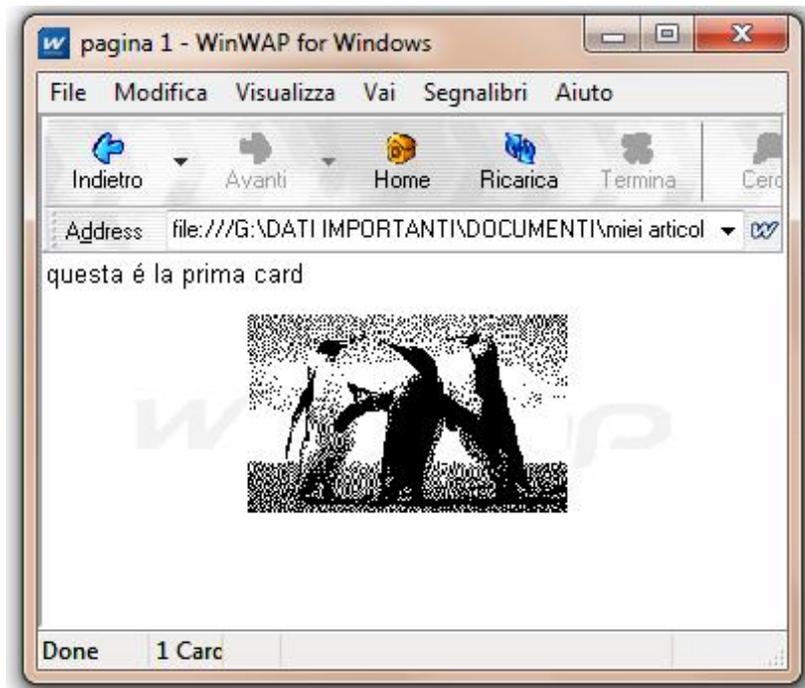
L' immagine é allineata a sinistra e ha le sue dimensioni originali, 200px x 150px. Purtroppo, il parametro ALIGN in WML permette solo un allineamento verticale. Come fare per allineare la foto al centro nella pagina? Bisogna utilizzare un escamotage includendo l' immagine in un paragrafo tramite il tag P al quale daremo l' attributo ALIGN="center". Se vogliamo allinearla al centro e diminuirne le dimensioni modifichiamo dunque il codice così:

```

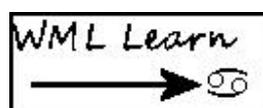
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="pagina 1">
    questa é la prima card<br />
    <P align="center">
      <IMG SRC="Penguins.wbmp" WIDTH="160" HEIGHT="100"
ALT="pinguini">
    </p>
  </card>
</wml>

```

ed ecco il risultato:



Bene, adesso che abbiamo imparato ad usare le tabelle e le immagini possiamo iniziare a sviluppare un contenuto più complesso. Immaginiamo di creare una intera homepage in WML per un ipotetico sito chiamato "WML Learn", il cui logo é questa immagine:



Vogliamo mostrare il logo in alto a destra nella pagina e, sotto di esso, una frase di benvenuto.

Sotto di questa, allineata a centro pagina, una tabella che comprenda 3 colonne:

- la prima colonna deve contenere del testo
- la seconda colonna deve contenere l' immagine dei pinguini
- la terza colonna deve contenere altro testo

sotto la tabella, un' altra riga di testo anche questa centrata.

Per allineare il logo a destra ad inizio pagina dovremo usare il tag P con attributo ALIGN impostato su "right" e prima di chiudere il paragrafo dovremo andare a capo con BR e scrivere la nostra frase di benvenuto:

```
<P align="right">  
<IMG SRC="wmllogo.wbmp"> <BR />  
Benvenuti su WML Learn!  
</P>
```

Dopo di ciò dovremo comporre il paragrafo centrato contenente la tabella. Si noti che siccome utilizziamo l' allineamento centrato nel paragrafo, a seconda del software dell' interprete questo regna sull' allineamento delle singole colonne della tabella. E' bene comunque inserire l' allineamento delle colonne nella dichiarazione della tabella, di modo che se un interprete é in grado di distinguere le due formattazioni possa farlo:

```
<P mode="nowrap" align="center">  
<TABLE columns="3" align="RCL">  
<TR>  
<TD>Questo sito vi insegnerà a programmare in WML senza  
fatica</TD>  
<TD><IMG SRC="Penguins.wbmp" BORDER=0></TD>  
<TD>Unire tabelle testo e immagini è davvero molto  
semplice</TD>  
</TR>  
</TABLE>  
</P>
```

Rimane solo la frase finale, anche essa in un paragrafo centrato:

```

<P align="center">
Tornate presto a visitarci!
</P>

```

Ed ecco il listato finale:

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="pagina 1">
    <P align="right"> <IMG SRC="wmllogo.wbmp"><BR /> Benvenuti
su WML Learn!</P>
    <P mode="nowrap" align="center">
    <TABLE columns="3" align="RCL">
    <TR>
    <TD>Questo sito vi insegnerà a programmare in WML senza
fatica</TD>
    <TD><IMG SRC="Penguins.wbmp" BORDER=0></TD>
    <TD>Unire tabelle testo e immagini è davvero molto
semplice</TD>
    </TR>
    </TABLE>
    </P>
    <P align="center">Tornate presto a visitarci!</P>
  </card>
</wml>

```

Il cui risultato sarà:



La nostra homepage é completa, ma per poter essere utilizzata come 'prima pagina' di un sito, manca ancora qualcosa: collegarla ad altre pagine. Questo si può fare con i link e con i pulsanti di azione, una particolarità del WML che permette di agire sulla pagina direttamente con i tasti del dispositivo cellulare.

Link e Azioni

Abbiamo già visto nei primi esempi di codice l' utilizzo dei tags per linkare le varie cards tra di loro tramite il tag A e il suo attributo HREF, metodo del tutto uguale a quello che utilizzeremmo in una pagina HTML. Il risultato di un link creato con questo metodo é una frase di testo generalmente sottolineata sulla quale si deve clickare per andare all' indirizzo voluto, sia esso una nuova card o un link esterno. Nel mondo del WAP e del WML, però, concepiti per gestire navigazioni effettuate con dispositivi cellulari, é bene utilizzare un altro tipo di link e cioè il metodo dei pulsanti.

Questo metodo consiste nell'uso di due tag:

- il tag GO anzi che A per creare il vero e proprio link - é un tag senza chiusura come BR
- il tag DO che definisce il tipo e la label del pulsante - é un tag con chiusura

Eccone un esempio pratico:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1" title="pagina 1">
  <P align="center"><EM>questa é la prima card</EM>
  <BR />
  <DO TYPE="accept" LABEL="vai a pagina 2">
  <GO HREF="#card2" />
  </DO></P>
</card>
<card id="card2" title="pagina 2">
  <STRONG>questa é la seconda card</STRONG>
</card>
</wml>
```

il cui risultato é:



I pulsanti possono essere di diverso tipo, tra cui i più importanti sono:

- pulsante ACCEPT per accettare l' opzione proposta dal pulsante, gestita dal successivo tag GO
- pulsante PREV che permette di tornare alla pagina precedente, azione gestita o da un tag GO o dal tag PREV (senza chiusura)
- pulsante OPTIONS che permette di visualizzare una serie di opzioni (che vanno programmate in una apposita card e vi si accede tramite il tag GO)

Ma se tutti questi tipi di bottoni hanno bisogno di un tag che specifichi il collegamento, ci si può chiedere: non basterebbe avere un unico tipo ACCEPT e poi gestire tramite GO tutte le azioni, mandando a una pagina voluta, alla pagina precedente, o a una pagina di opzioni?

La domanda sarebbe legittima in una normale navigazione web, ma il tag DO é stato concepito per essere associato ai tasti di navigazione dei cellulari. Cosa significa in pratica?

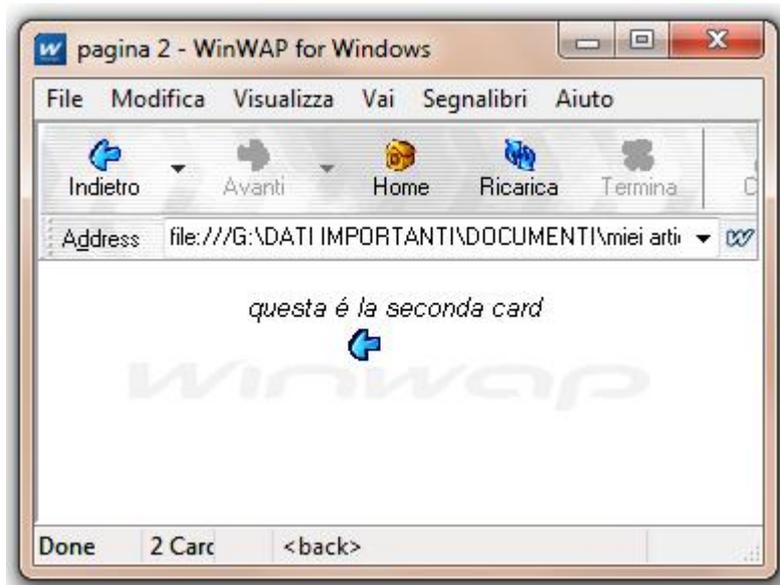
Che gli interpreti del software del cellulare, quanto trovano un tipo ACCEPT, lo associano a un tasto funzione del cellulare, quando trovano il tipo PREV lo associano a un altro tasto del cellulare, e quando trovano OPTIONS lo associano a un terzo tasto (esiste anche il tipo HELP associato ancora a un altro tasto). Il modo in cui gli interpreti fanno queste associazioni e i tasti coinvolti variano da cellulare a cellulare, anche se in generale il tipo ACCEPT é indirizzato al tasto che normalmente nei menu si utilizza per OK, e il tipo PREV é indirizzato al tasto che normalmente nei menu si utilizza per INDIETRO o

CANCELLA. Questa però non è una regola universale.

Nel listato seguente vengono mostrati i due tipi di pulsanti ACCEPT e PREV, uno per pagina. Riguardo al tipo PREV bisogna specificare che, quando utilizzato, esso causa la pulizia della cache di navigazione, per questo non è possibile l'implementazione di un tipo che permetta di ritornare alla pagina dalla quale si è usciti con PREV.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1" title="pagina 1">
  <P align="center"><EM>questa é la prima card</EM>
  <BR />
  <DO TYPE="accept" LABEL="vai a pagina 2">
  <GO HREF="#card2" />
  </DO></P>
</card>
<card id="card2" title="pagina 2">
  <P align="center"><EM>questa é la seconda card</EM>
  <BR />
  <DO TYPE="prev">
  <PREV />
  </DO></P>
</card>
</wml>
```

il risultato della prima card lo abbiamo già visto poco sopra, mentre per la seconda card, quella che coinvolge il tipo PREV, sarà questo:



Gestire gli eventi

WML ha nel proprio arsenale una serie di quattro gestori di eventi elementari. Si tratta di timers, riconoscimento di particolari tasti del dispositivo cellulare, e la scelta tra più opzioni.

I nomi dei gestori di eventi sono:

- ONTIMER
- ONENTERBACKWARD
- ONENTERFORWARD
- ONPICK

Analizziamoli uno ad uno.

Il più semplice degli eventi è quello chiamato ONTIMER e, come dice il nome, gestisce eventi collegati ad un timer settato dalla card WML stessa.

L' evento ontimer si può dichiarare all' interno della definizione della card, e la sua sintassi è questa:

```
<card id="card1" title="pagina 1" ontimer="image.wml"
newcontext="true">
  tra 5 secondi verranno caricate le immagini
  <timer name="clock1" value="5">
</card>
```

questo codice appena visto mostrerà un avviso, e dopo 5 secondi verrà caricato il file esterno chiamato images.wml

Ma possiamo usare i timer anche all' interno dello stesso deck WML; per esempio all' interno di una card chiamata card1, possiamo mettere un bottone che inviti ad andare alla card2. Clickando sul bottone, verremo mandati a una card-intermezzo nella quale apparirà un messaggio di attesa e, dopo 3 secondi, verremo portati alla card2. Realizziamo questo con il codice seguente:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.xml">
<wml>
  <card id="card1" title="pagina 1">
```

```

    <P align="center"><EM>questa é la prima card</EM>
    <BR />
    <DO TYPE="accept" LABEL="vai a pagina 2">
    <GO HREF="#card-intermezzo" />
    </DO></P>
</card>
<card id="card-intermezzo" title="attesa" ontimer="#card2"
newcontext="true">
state per essere reindirizzati...
<timer name="clock1" value="3">
</card>
<card id="card2" title="pagina 2">
    <P align="center"><EM>questa é la seconda card</EM>
</P>
</card>
</wml>

```

L' attributo NEWCONTEXT, settato al valore TRUE, ha una funzione ben specifica. I timer WML terminano al valore zero, quindi una volta passati alla card successiva, se premessimo il tasto 'indietro' nel telefonino torneremmo alla pagina che contiene il contatore ma questo avrebbe ancora valore zero e quindi compirebbe l' azione per cui é definito. Nell' esempio del listato sopra, da card2 premendo il tasto 'back', andremmo alla card-intermezzo e subito verremmo rimandati a card2. Questo viene evitato dal valore "true" di NEWCONTEXT che fa ripartire il timer resettandolo di nuovo al valore di VALUE.

Il timer può essere utilizzato anche con un' altra sintassi, che ci permette di evitare il passaggio attraverso la card-intermezzo. Modifichiamo il codice cosí:

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card id="card1" title="pagina 1">
    <P align="center"><EM>questa é la prima card</EM>
    <BR />
    <DO TYPE="accept" LABEL="vai a pagina 2">
    <GO HREF="#card2" />
    </DO></P>
    <onevent type="ontimer">
    <go href="#card2"/>
    </onevent>
    <timer value="5"/>
</card>
<card id="card2" title="pagina 2">
    <P align="center"><EM>questa é la seconda card</EM>

```

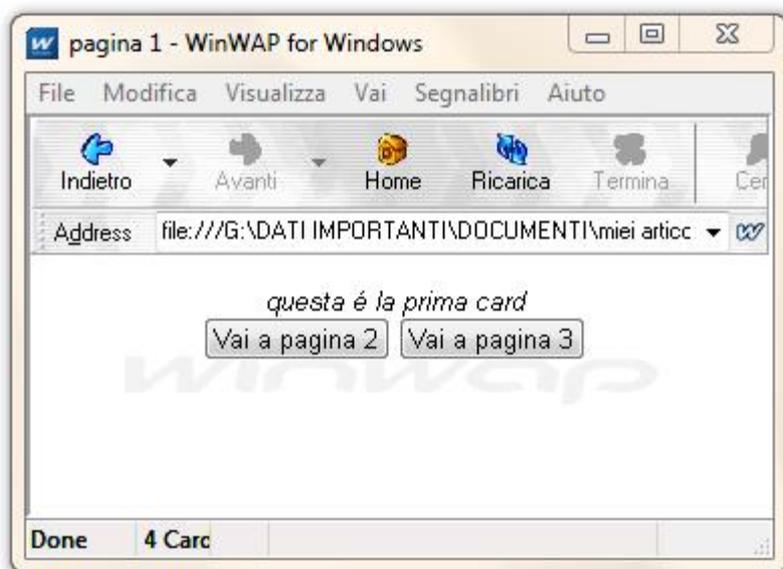
```
</P>  
</card>  
</wml>
```

In questo caso, il timer é nella card1 che contiene anche il bottone per passare a card2. Ciò vuol dire che, se non clickiamo sul bottone entro 3 secondi, sarà il timer a condurci direttamente alla card destinazione. L' utilità di questa funzione é lampante se si pensa alle situazioni in cui si vuole dare all' utente una scelta a tempo, per esempio andare a 2 diverse card, ma se non si effettua una scelta entro un dato lasso di tempo sarà il programma ad andare ad una destinazione preimpostata.

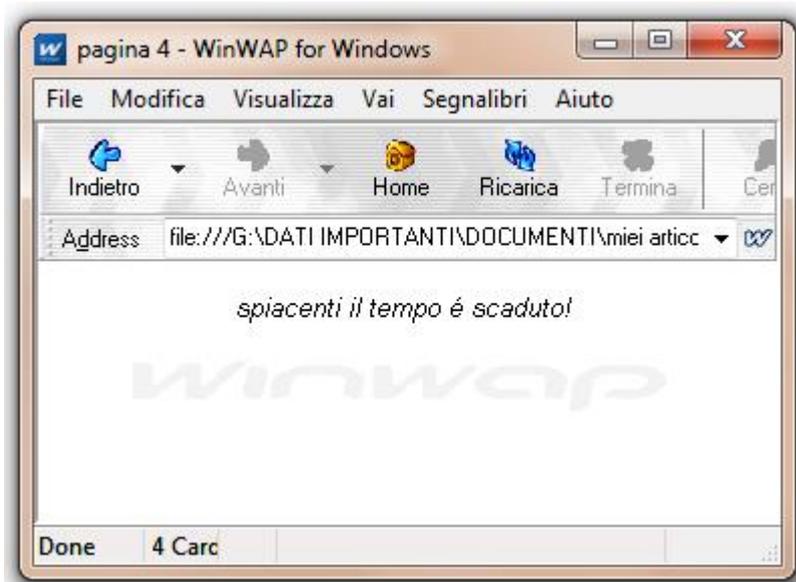
Modifichiamo le cards in questo modo:

```
<?xml version="1.0"?>  
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"  
"http://www.wapforum.org/DTD/wml12.dtd">  
<wml>  
<card id="card1" title="pagina 1">  
  <P align="center"><EM>questa é la prima card</EM>  
  <BR />  
  <DO TYPE="accept" name="button1" LABEL="vai a pagina 2">  
  <GO HREF="#card2" /> </DO>  
  <DO TYPE="accept" name="button2" LABEL="vai a pagina 3">  
  <GO HREF="#card3" /> </DO>  
  </P>  
  <onevent type="ontimer">  
  <go href="#card4"/>  
  </onevent>  
  <timer value="15"/>  
</card>  
<card id="card2" title="pagina 2">  
  <P align="center"><EM>questa é la seconda card</EM> </P>  
</card>  
<card id="card3" title="pagina 3">  
  <P align="center"><EM>questa é la terza card</EM> </P>  
</card>  
<card id="card4" title="pagina 4">  
  <P align="center"><EM>spiacenti il tempo é scaduto!</EM> </P>  
</card>  
</wml>
```

Come risultato la prima card mostrerà questa situazione:



E, se non facciamo una scelta per tempo, verremo reindirizzati alla card4:



Lasciamo da parte i timer e occupiamoci di gestire gli eventi relativi ai tasti del dispositivo cellulare. I due eventi coinvolti gestiscono rispettivamente:

- la pressione del pulsante BACK
- la pressione del pulsante FORWARD (OK in alcuni dispositivi)

Il primo caso é gestito dall' evento ONENTERBACKWARD che, come per i timer, può essere dichiarato all' interno della definizione della card o attraverso il tag ONEVENT. Utilizzando il secondo metodo, possiamo anche rendere 'globale' la gestione dell' evento, cioè renderla disponibile per tutte le

card. Basterà dichiarare l' evento prima della definizione delle singole card.

Vediamo alcuni esempi di codice:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<onevent type="onenterbackward">
<go href="#card3"/>
</onevent>
<card id="card1" title="pagina 1">
  <P align="center"><EM>questa é la prima card</EM>
  <BR />
  <DO TYPE="accept" name="button1" LABEL="vai a pagina 2">
  <GO HREF="#card2" /> </DO>
  </P>
</card>
<card id="card2" title="pagina 2">
  <P align="center"><EM>questa é la seconda card</EM> </P>
</card>
<card id="card3" title="pagina 3">
  <P align="center"><EM>avete premuto il tasto 'back' </P>
</card>
</wml>
```

Questo codice dichiara che se viene premuto il tasto del cellulare al quale, nella navigazione, é associata la funzione di 'Indietro', il browser reindirizza alla card3.

Questo genere di evento può essere utilizzato per rimediare ad una pressione accidentale del tasto, automatizzando il ricaricamento della pagina da cui si proviene per evitare all' utente di doverci tornare appositamente.

Come fare? Basta dichiarare all' interno di ogni card un ONEVENT che rimandi alla card stessa:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card id="card1" title="pagina 1">
  <P align="center"><EM>questa é la prima card</EM>
  <BR />
  <DO TYPE="accept" name="button1" LABEL="vai a pagina 2">
  <GO HREF="#card2" /> </DO>
  </P>
```

```

<onevent type="onenterbackward">
<go href="#card1"/>
</onevent>
</card>
<card id="card2" title="pagina 2">
  <P align="center"><EM>questa é la seconda card</EM> </P>
<onevent type="onenterbackward">
<go href="#card2"/>
</onevent>
</card>
</wml>

```

in questo codice, se dalla card1 premiamo accidentalmente il tasto 'Indietro' del telefonino, verremo riportati alla stessa card1. Lo stesso avviene all' interno della card2.

L' evento ONENTERFORWARD agisce alla stessa maniera ma interviene nel caso si sia premuto il tasto associato alla funzione di navigazione 'Avanti' del cellulare. Ha le stesse caratteristiche di ONENTERBACKWARD ma in più può avvalersi del task PREV.

Guardiamo quest' esempio:

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card id="card1" title="pagina 1">
  <P align="center"><EM>questa é la prima card</EM>
  <BR />
  <DO TYPE="accept" name="button1" LABEL="vai a pagina 2">
  <GO HREF="#card2" /> </DO>
  </P>
<onevent type="onenterforward">
<go href="#card1"/>
</onevent>
</card>
<card id="card2" title="pagina 2">
  <P align="center"><EM>questa é la seconda card</EM> </P>
<onevent type="onenterforward">
<prev />
</onevent>
</card>
</wml>

```

In questo codice abbiamo utilizzato entrambi i metodi, il GO e il PREV, nelle due diverse card. Logicamente, se dalla card 2 si va avanti, il prev porta di nuovo alla card2 perchè era l' ultima caricata.

Veniamo ora all' ultimo evento, chiamato ONPICK, forse il più importante di tutti perchè permette di gestire una scelta intenzionale da parte dell' utente, ad esempio nella selezione tra più voci di un elenco. Per imparare ad usare ONPICK impariamo anche a creare selezioni e opzioni. Immaginiamo di voler dare all' utente una scelta sulla card a cui dirigersi; finora abbiamo usato dei bottoni utilizzando DO oppure dei link con GO o A. Un altro metodo é quello di creare una lista di selezione con i tags SELECT e OPTION:

```
<select name="CardSelect.">
  <option value="1" onpick="#card1"> vai alla Card 1 </option>
  <option value="2" onpick="#card2"> vai alla Card 2 </option>
  <option value="3" onpick="#card3"> vai alla Card 3 </option>
</select>
```

Il tag SELECT crea una struttura di scelta, le cui voci sono dichiarate dai vari OPTION. Ogni opzione deve avere un VALUE unico, e una gestione ONPICK.

Graficamente il risultato delle SELECT varia a seconda dell' interprete WML del dispositivo: può essere:

- una lista di voci con al fianco una casella tonda di selezione, che diventerà 'riempita' al momento della selezione;
- un pulsante che mostra la prima opzione e una 'freccetta' clickando sulla quale si visualizzano le altre voci della lista.

Vediamo un listato intero e il risultato grafico:

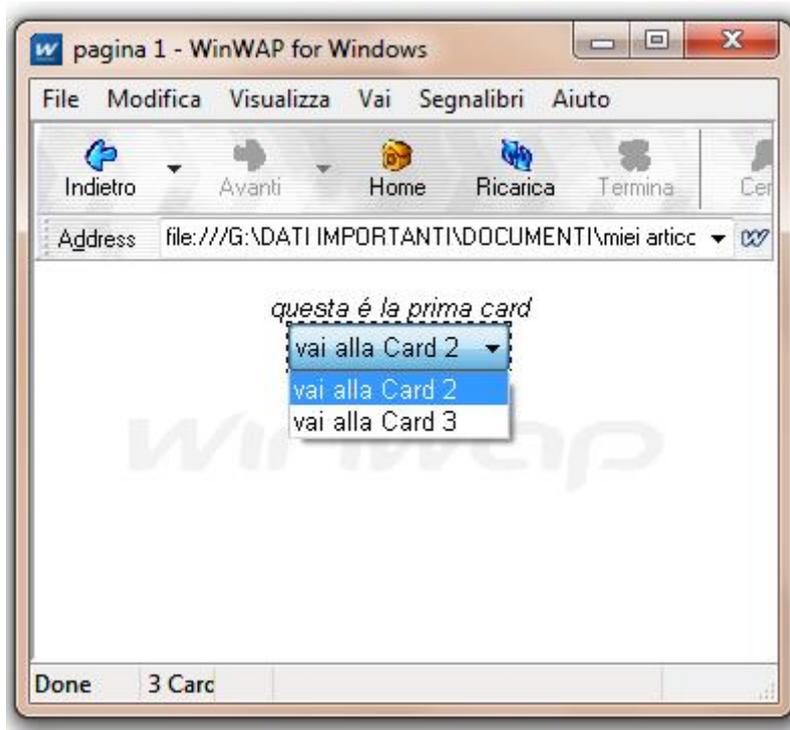
```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card id="card1" title="pagina 1">
  <P align="center"><EM>questa é la prima card</EM>
  <BR />
  <select name="CardSelect">
    <option value="1" onpick="#card2"> vai alla Card 2
</option>
    <option value="2" onpick="#card3"> vai alla Card 3
</option>
  </select>
```

```

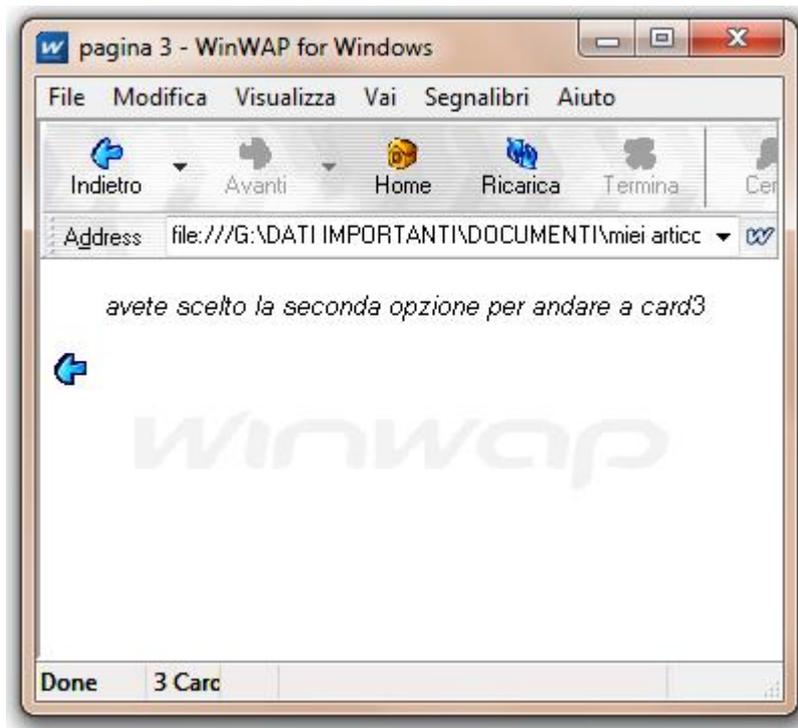
    </P>
  </card>
  <card id="card2" title="pagina 2">
    <P align="center"><EM>avete scelto la prima opzione per andare
a card2</EM> </P>
    <DO TYPE="prev">
      <PREV />
    </DO>
  </card>
  <card id="card3" title="pagina 3">
    <P align="center"><EM>avete scelto la seconda opzione per
andare a card3</EM> </P>
    <DO TYPE="prev">
      <PREV />
    </DO>
  </card>
</wml>

```

il cui risultato é visibile in queste immagini:



L' immagine sopra mostra come appare la lista una volta che é stata 'aperta' clickando sul triangolino nero verso il basso nel pulsante. Selezionando una delle due voci, veremo reindirizzati alla card corrispondente:



Nella prossima sezione impareremo a gestire le selezioni in modo da farle interagire con input di dati e variabili.

Input di dati, selezioni e variabili

Una pagina WML non deve per forza essere una pagina statica di pura consultazione; esattamente come per il web, anche i siti wap hanno dovuto soddisfare la richiesta dell' utente di poter interagire con le pagine, personalizzare le informazioni, ricevere o fornire dei feedback e dei contenuti personali. Per fare questo, le pagine devono poter gestire l' input di dati e immagazzinarli in variabili gestite poi tramite uno script.

Così, anche col telefonino, ci si può per esempio registrare in un sito immettendo una login ed una password, modificare il proprio profilo e salvare informazioni. Tutto questo processo è distinto in almeno tre fasi:

- 1) la pagina propone una serie di form di input, che possono avere valori preimpostati o possono accettare valori immessi dall' utente;
- 2) i dati immessi vengono racchiusi in variabili, che possono essere gestiti da WML stesso o da uno script esterno;
- 3) viene in genere fornito un feedback all' utente per confermare l' avvenuto immagazzinamento e trattamento dei dati, a volte fornendo un riepilogo.

In questo capitolo vedremo tutte queste fasi grazie ad un listato abbastanza lungo che contiene un intero sito in cui é possibile:

- registrare un nuovo account;
- avere conferma con un riepilogo dei dati inseriti;
- effettuare la login;
- arrivare nella propria homepage.

Costruiamo prima di tutto la parte grafica. Utilizziamo a questo scopo come base il listato che avevamo già creato per la nostra ipotetica homepage di WML Learn del capitolo riguardante le immagini. Al telaio, che era costituito dalle seguenti righe di codice:

```
<card id="card1" title="pagina 1">
  <P align="right">
    <IMG SRC="wmllogo.wbmp"> <br />
    Benvenuti su WML Learn!
  </P>
  <P mode="nowrap" align="center">
    <TABLE columns="3" align="RCL">
      <TR>
        <TD>Questo sito vi insegnerà a programmare in WML senza
fatica</TD>
        <TD><IMG SRC="Penguins.wbmp" BORDER=0></TD>
        <TD>Unire tabelle testo e immagini è davvero molto
semplice</TD>
      </TR>
    </TABLE>
  </P>
  <P align="center">
    Tornate presto a visitarci!
  </P>
</card>
```

approntiamo le seguenti modifiche:

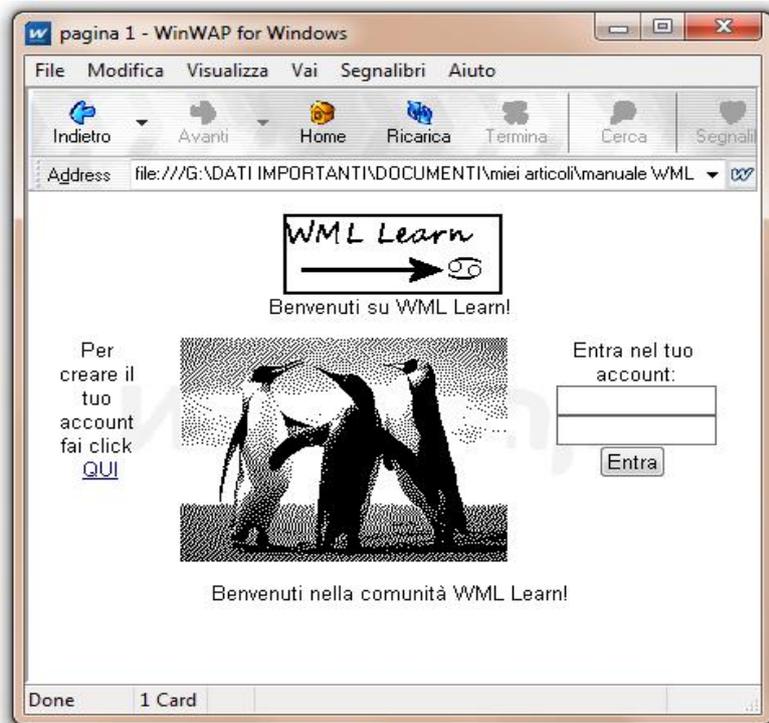
```
<card id="card1" title="pagina 1">
  <P align="center">
    <IMG SRC="wmllogo.wbmp"> <br />
    Benvenuti su WML Learn!
  </P>
  <P mode="nowrap" align="center">
    <TABLE columns="3" align="RCL">
```

```

<TR>
<TD>Per creare il tuo account fai click <br />
<a HREF="#cardregister">QUI</a></TD>
<TD><IMG SRC="Penguins.wbmp" BORDER=0></TD>
<TD> Entra nel tuo account:<br />
<input type="text" name="login" size="8" title="login" />
<br />
<input type="password" name="pass" size="8" title="pass" />
<br />
<anchor>
<do type="accept" label="entra">
<go href="login.php">
<postfield name="login" value="$(login)"/>
<postfield name="pass" value="$(pass)"/>
</go>
</do>
</anchor></TD>
</TR>
</TABLE>
</P>
<P align="center">Benvenuti nella comunità WML Learn!</P>
</card>

```

Il risultato dovrebbe essere una costruzione simile a questa (i diversi dispositivi cellulari, a seconda dell' interprete che implementano, potrebbero allineare diversamente):



La colonna di sinistra che rimanda alla pagina per registrarsi é un semplice link creato con il tag A, quindi esaminiamo punto per punto la colonna di destra con le form per il login al sito.

Le due form sono create con il tag INPUT:

```
<input type="text" name="login" size="8" title="login" />  
<br />  
<input type="password" name="pass" size="8" title="pass" />  
<br />
```

L' attributo TYPE chiarisce all' interprete che tipo di dati verranno immessi e come trattarli; nel primo caso, il valore 'text' comunica all' interprete che si sta immettendo semplice testo, mentre nel secondo caso il valore 'password' comunica all' interprete che ciò che verrà inserito ha valore di segretezza e quindi per ogni lettera immessa verrà visualizzato un asterisco (*).

L' attributo NAME deve essere univoco e costituisce la dichiarazione della variabile in cui verrà salvato il dato inserito nella form. Quando queste variabili devono essere utilizzate per manipolazioni devono essere precedute dal simbolo di dollaro \$, quindi, siccome in questo caso abbiamo usato nei due form i name 'login' e 'pass', le variabili si chiameranno \$login e \$pass.

L' attributo SIZE decide di quanti caratteri deve essere la larghezza del form di input, mentre l' attributo TITLE si comporta come il corrispettivo di tutti gli altri tag e non ha qui nessuna funzione se non di fornire una indicazione.

Una volta inseriti i dati nelle due form, queste devono essere inviate o a una card o a uno script che li gestisca. In questo esempio mandiamo la login e la password ad uno script PHP che verificherà la correttezza dei dati (eventualmente salvati in un database) e si occuperà di mandare alla homepage privata dell' utente. Per fare questo abbiamo creato un link con il tag ANCHOR con relativa chiusura, all' interno del quale abbiamo creato un bottone utilizzando DO e definito il vero e proprio link allo script *login.php* tramite GO. Il tag POSTFIELD si occupa della vera e propria trasmissione delle variabili allo script.

Analizziamo prima di tutto la struttura dei link. Finora abbiamo sempre visto il tag GO nella forma:

```
<GO href="paginadestinazione" />
```

ma questa forma non è valida per l' invio di dati perchè il POSTFIELD deve comparire dentro la dichiarazione di link. Quindi dobbiamo utilizzare GO come un link HTML come se usassimo A con relativa chiusura. Ma ciò, da solo,

non basta, perchè GO utilizzato in questa maniera deve per regola comparire entro una struttura ANCHOR con apertura e chiusura.

Di regola, se dobbiamo mandare dei dati tramite GO, la struttura generale sarà:

```
<anchor>
...
<go href="...">
...
</go>
</anchor>
```

se il link é associato ad un pulsante, la struttura GO dovrà essere inglobata in una struttura DO:

```
<anchor>
<do ....> <go href="...">
...
</go> </do>
</anchor>
```

e veniamo finalmente alla vera e propria trasmissione delle variabili con POSTFIELD.

```
<postfield name="login" value="$(login)"/>
<postfield name="pass" value="$(pass)"/>
```

L' attributo NAME si occupa di acchiappare il medesimo attributo della form dichiarata con INPUT, e l' attributo VALE trasforma la dichiarazione di variabile in variabile vera e propria aggiungendo la \$. A partire da questo momento, lo script di destinazione (in questo caso: login.php) potrà elaborare e gestire le variabili.

Ora che abbiamo finito di esaminare la funzione di login possiamo scrivere uno script PHP che si occupi di gestire questi dati. Per questo scopo utilizziamo un semplice script php in cui sono inseriti già i valori corretti di login e password (un metodo non sicuro e non pratico ma a puro scopo esplicativo) che reindirizza a un nuovo file WML in caso di corretto login.

```
if($login == "alex" && $pass == "wml1" )
{ header('Location: /loginok.wml);
```

```

    }
else {
    print "dati di login sbagliati";
}

```

il nostro file *loginok.wml* ovviamente potrebbe essere una semplice card di benvenuto o qualcosa di più complesso, l' importante é aver capito il meccanismo che regola queste procedure di invio e manipolazione dei dati.

Ora invece creiamo la card di registrazione. Supponiamo di voler far inserire all' utente:

- il nome
- il cognome
- l' età
- il sesso
- la login scelta
- la password scelta

comporremo la card (che avemo chiamato: cardregister) più o meno in questo modo:

```

<card id="cardregister" title="registrati!">
Crea il tuo account compilando i form qui sotto:
  <p>
Nome: <input name="name" size="12"/><br />
Cognome: <input name="surname" size="12"/><br />
Sesso: <select name="sex"><br />
  <option value="male">Maschio</option>
  <option value="female">Femmina</option>
</select>
Età: <input name="age" size="4" format="*N"/>
</p>
<p> scegli i tuoi dati di accesso</p>
Login: <input name="login" size="12"/><br />
Password: <input type="password" name="pass" size="12"/><br
/>

<anchor>
<go method="get" href="process.php">
  <postfield name="name" value="$(name)"/>
  <postfield name="surname" value="$(surname)"/>
  <postfield name="age" value="$(age)"/>
  <postfield name="sex" value="$(sex)"/>
  <postfield name="login" value="$(login)"/>

```

```

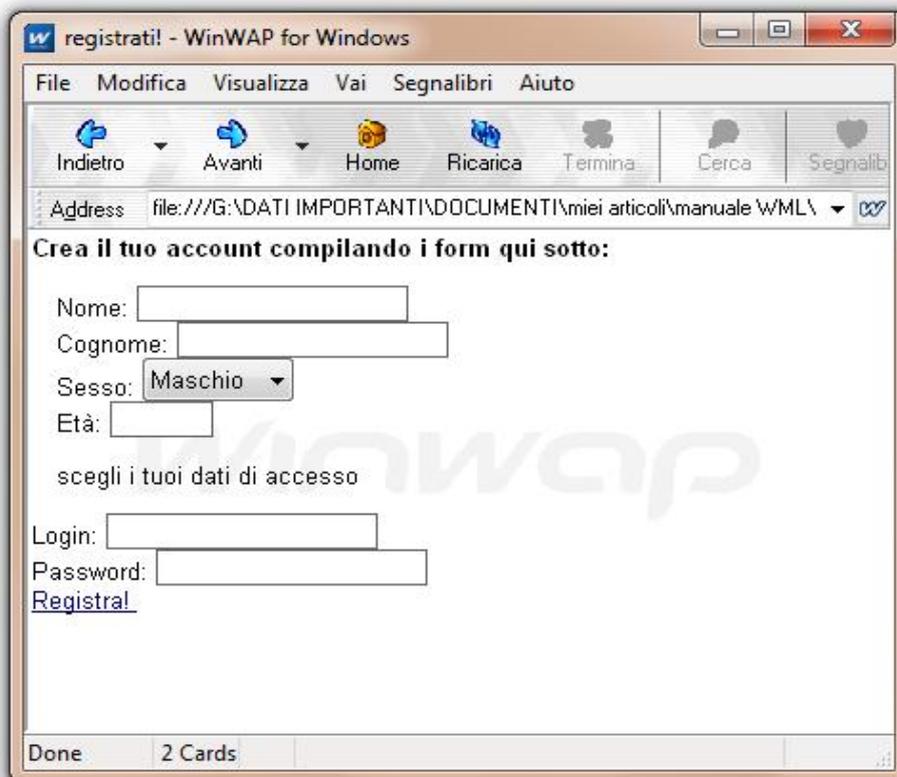
    <postfield name="pass" value="$(pass)"/>
  </go>
  Registra!
</anchor>
</card>

```

Con i vari input e select abbiamo creato i campi di immissione e di scelta (l'utente potrà selezionare il sesso tramite una select) e dichiarato le variabili; su questa parte l'unica cosa nuova da osservare è l'utilizzo dell'attributo FORMAT nell'input relativo all'età. Il valore *N indica all'interprete che deve accettare solo valori numerici di qualsiasi lunghezza. Se avessimo voluto impostare il limite a 2 cifre (già 3 sono troppe, a meno che non si abbia intenzione di vivere 100 anni!!) avremmo usato il valore 2N per l'attributo.

Successivamente all'immissione, con quella sfilza di POSTFIELD e l'utilizzo di GO, abbiamo richiamato e formato le variabili definite negli input e le abbiamo dirette verso il file *process.php* che si occuperà di gestirle, per esempio creando l'account in un database.

Il risultato grafico di questa card è il seguente:



Abbiamo quasi completato la creazione del nostro mini-sito per WML Learn... ci rimane solo una cosa da fare: fornire all'utente il feedback di

avvenuta registrazione.

Per farlo dobbiamo definire il nostro script process.php in modo che questo crei una pagina WML di riassunto. Il file process.php ovviamente dovrà anche contenere le istruzioni di scrittura dei dati nel database, ma questi esulano dallo scopo di questo manuale quindi qui verrà fornita solo la parte relativa alla conferma di ricezione dei dati, che sarà qualcosa del genere:

```
<?php echo 'Content-type: text/vnd.wap.wml'; ?>
<?php echo '<?xml version="1.0"?.'.'>'; ?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>
  <card id="card1" title="WML Response"> <p>
    Complimenti! Ti sei registrato con i seguenti dati:<br/>
    Nome: <?php echo $_GET["name"]; ?><br/>
    Cognome: <?php echo $_GET["surname"]; ?><br/>
    Età: <?php echo $_GET["age"]; ?><br/>
    Sesso: <?php echo $_GET["sex"]; ?><br/>
    e hai scelto i seguenti dati di accesso:
    Login: <?php echo $_GET["login"]; ?><br/>
    Password: <?php echo $_GET["pass"]; ?><br/>
  </p> </card>
</wml>
```

(... parte di codice PHP che scrive i dati nel database...)

Il nostro mini-sito é completo, le funzioni di login e registrazione sono state implementate e, se abbiamo PHP e un wapserver installati nel pc, rimane semplicemente da scrivere i dati nel database per avere un mini portale funzionante.

Conclusioni

Siamo arrivati al termine di questo tutorial, ed abbiamo esaminato e imparato le principali funzionalità del WML fino alla versione 1.2; ovviamente, molto é stato lasciato fuori, per esempio non sono stati esaminati i tag responsabili della creazione di templates, o l' assegnazione delle 'accesskey' cioè le funzioni che possono essere svolte direttamente dal dispositivo cellulare utilizzando i tasti numerici da 0 a 9. La scelta di non trattare questi e altri argomenti di dettaglio è dovuta al fatto che a seconda del dispositivo cellulare il funzionamento dei template e delle accesskey é assai variabile, quindi si sarebbe rischiato di fornire esempi non sempre verificabili. Ciononostante, il tutorial ha trattato abbastanza dettagliatamente tutti quegli aspetti del WML che possono tornare utili in ogni situazione di programmazione delle card. Gli

esempi qui riportati ovviamente sono elementari e non propriamente adatti ad una situazione professionale, ma costituiscono un ottimo materiale di apprendimento dei principi che regolamentano la programmazione in ambiente WML e allo stesso tempo aprono la strada a chi volesse cercare maggiori approfondimenti in rete.