Bash e Google API
Sereni e .... "Variabili"

Un metodo comodo per poter sapere che tempo farà il giorno dopo è quello di mettersi distesi sul divano di casa propria, telecomando in mano e cercare di beccare il canale giusto. Tutto questo non è proprio la massima vocazione per chi ha voglia di mettersi in gioco nel mondo dell'informatica, ma magari, direte, per chi usa un computer basterebbe trovare il sito giusto. Comodo, ma non divertente!
La cara Bash ha dato, dà e ci darà sempre una mano per risolvere questo tipo di situazioni e di esempi ne esistono un'infinità.
Vogliamo prendere un caso particolare: le previsioni del tempo! Proviamo a costruire uno script che ci dia la possibilità di poter ottenere determinate informazioni invece che a farlo sia il telecomando.
Per poter ottenere ciò dobbiamo chiedere una mano a Google e precisamente alle sue API (Application Programming Interface) e ancora più precisamente... alle Weather API. Google ci mette a disposizione un'enorme mole di informazioni e le Weather API sono degli strumenti che forniscono dati relativi alle previsioni del tempo (come dice la parola stessa Weather API, API meteorologiche!).
Cominciamo a scrivere lo script considerando un qualsiasi editor di testo che, magari, ci dia la possibilità di poter evidenziare i termini specifici della bash. Come prima riga indichiamo proprio il fatto che stiamo utilizzando la bash stessa e quindi utilizziamo le informazioni fornite da Google: in che modo? Prendiamo in considerazione un browser e facciamo un esempio digitando nella barra degli indirizzi la seguente riga:

http://www.google.com/ig/api?weather=lecce,italy&hl=it

noteremo che viene visualizzata una pagina in formato xml che contiene un certo numero di tag entro i quali sono contenuti un insieme di dati nel formato 'nome=valore'. Nel nostro esempio scegliamo un sottogruppo di questi tag.
Anche nell'indirizzo, dopo il punto interrogativo, sono presenti due coppie '*nome=valore*' ed esattamente il nome della città di cui vogliamo conoscere il tempo meteorologico (nel nostro caso Lecce) e la specificazione della lingua. Grazie alla specificazione di quest'ultima si ottengono informazioni nel formato **SI** (*Sistema Internazionale*).

Occorre, quindi, ottenere la pagina e

```
−<xml_api_reply version="1">
  −<weather module_id="0" tab_id="0" mobile_row="0" mobile_zipped="1" row="0" section="0">
    −<forecast_information>
       <city data="Lecce, Puglia"/>
       <postal_code data="lecce,italy"/>
       <latitude_e6 data=""/>
       <longitude_e6 data=""/>
       <forecast_date data="2010-01-20"/>
       <current_date_time data="2010-01-20 22:15:00 +0000"/>
       <unit_system data="SI"/>
     </forecast_information>
    −<current_conditions>
       <condition data="Parzialmente nuvoloso"/>
       <temp_f data="39"/>
       <temp_c data="4"/>
       <humidity data="Umidità: 87%"/>
       <icon data="/ig/images/weather/partly_cloudy.gif"/>
       <wind_condition data="Vento: N a 0 km/h"/>
     </current_conditions>
```

poterla elaborare costruendo, di fatto, una sorta di parser che filtri ciò che ci interessa. Per ricavare la pagina utilizziamo il comando **wget** indicandogli la pagina su citata. Una volta fatto questo evitiamo di visualizzare eventuali errori inviando questi ultimi in una sorta di "cestino" (*/dev/null*), cioè dirigiamo lo standard error in un file speciale.
Per evitare che wget possa scrivere il file **xml** in un file di tipo ipertestuale utilizziamo l'opzione **-O** e questo fa sì che l'output di wget venga redirezionato verso i comandi successivi attraverso l'utilizzo di pipe in serie. Le pipe filtreranno i dati importanti.
Vediamo, ora, come raccoglie i dati il nostro script commentando la seguente riga:

nella prima pipe incontriamo il

```
wget "http://www.google.com/ig/api?weather=lecce,italy&hl=it" 2>/dev/null -O -| tr '>' '\n' | tr '/' ' ' | tr '"' ' ' | tr '<' ' ' >filetmp.txt
```

Google : http://www.google.it

Bash e Google API

Sereni e .... "Variabili"

comando **tr** utile ad eliminare determinati caratteri come '>', '/', '"' e '<' (un esempio è dato dalla seconda figura).

```
?xml version= 1.0 ?
xml_api_reply version= 1
weather module_id= 0  tab_id= 0  mobile_row= 0  mobile_zipped= 1  row= 0  section= 0
forecast_information
city data= Casarano, Puglia
postal_code data= casarano,italy
latitude_e6 data=
longitude_e6 data=
forecast_date data= 2010-01-20
current_date_time data= 2010-01-20 22:55:00 +0000
unit_system data= SI
 forecast_information
current_conditions
condition data= Parzialmente nuvoloso
temp_f data= 39
temp_c data= 4
humidity data= Umidità: 87%
icon data=  ig images weather partly_cloudy.gif
wind_condition data= Vento: N a 0 km h
 current_conditions
```

Output dello script

```
=========================================
IL TEMPO SU        : Lecce, Puglia
=========================================
Condizione attuale : Chiaro
Data odierna       : 2010-01-21
Temperatura attuale: 4 gradi Celsius
Umidita'           : 93%
Vento              : N a 0 km h

=========================================
* PREVISIONI per i PROSSIMI GIORNI *
=========================================
Giorno             : gio
Temperatura minima : 3
Temperatura massima: 11
Condizione         : Possibilita' di pioggia
-----------------------------------------
Giorno             : ven
Temperatura minima : 1
Temperatura massima: 9
Condizione         : Chiaro
-----------------------------------------
Giorno             : sab
Temperatura minima : 5
Temperatura massima: 8
Condizione         : Possibilita' di pioggia
-----------------------------------------
Giorno             : dom
Temperatura minima : 6
Temperatura massima: 13
Condizione         : Chiaro
=========================================
```

Google Gadget

Salviamo tutto in un file chiamato '**filetmp.txt**'. Quest'ultimo file contiene tutti i valori che ci occorrono, agendo su di esso si possono spulciare quelli a noi graditi individuando le parole chiave. Ad esempio per ottenere il valore dell'umidità lo script sfrutta i comandi **cat** e **grep** cercando la parola humidity nel testo. Individuata la riga, questa viene passata a **cut**, il quale la suddivide in campi (il cui delimitatore è il carattere ':') e con **-f2** viene prelevato il dato che si trova dopo il carattere ':' e cioè il vero e proprio valore dell'umidità. In molti casi il separatore è '=' per questo lo troveremo al posto di ':'. In altri casi si è preferito sfruttare il fatto che determinate informazioni si trovano sempre allo stesso numero di riga per questo per prelevare il dato si è utilizzato il comando head il quale preleva le prime righe (ad esempio head -n 22 le prime 22 righe) di 'filetmp.txt'. L'ultima di tali righe è proprio quella di nostro interesse e per prelevarla si utilizza il comando **tail -n 1**.
Il comando **echo** ci agevola nello specificare meglio quali informazioni si stanno prelevando e per meglio interfacciarle all'utente finale.

Miglioramenti
Lo script fin qui descritto non contempla, però, la possibilità di poter scegliere una città a nostro piacere. Per poter dare quest'ulteriore possibilità sfruttiamo il comando **read** assieme al nome della variabile in cui vogliamo immagazzinare il nome della città (*read paese*). In questo modo lo script, appena dopo il suo avvio si ferma in attesa che l'utente immetta il nome della città su cui effettuare le previsioni. Il valore effettivo della variabile verrà passato alla stringa dell'indirizzo dato in pasto a wget ($paese).

Di seguito troverete il link per scaricare lo script completo, il quale non rappresenta una forma definitiva (anche se pienamente funzionante e testabile da subito), ma un invito al miglioramento e alla personalizzazione seguendo quello che è lo spirito hacker.

Script :

http://www.salug.it/~public/journal/SJ-0x05/weather-script.tar.gz

Google Weather API : http://www.googleapihelp.com/2009/08/google-weather-api.html
Guida avanzata di scripting Bash : http://www.pluto.it/files/ildp/guide/abs/index.html