



HYBRIS, SCRIPTING LANGUAGE

Interessante questo linguaggio di scripting. Progetto di alcuni giovani appassionati di programmazione, tra i quali nomino il Team Leader: Simone Margaritelli. A quanto pare tutto è nato per gioco, ma poi, il gioco si è trasformato in realtà!

Vediamo assieme di che si tratta..

Hybris sta per: **hybrid** scripting. E' un linguaggio di programmazione "ibrido" di vari linguaggi che prende la parte migliore di ognuno. E' interpretato (cioè che non viene compilato).

E' un linguaggio **procedurale**: contiene dunque quelle che possiamo definire procedure, funzioni. Con questa logica di programmazione, il programma che si intende sviluppare lo si può suddividere in problemi più piccoli.

Vediamo subito un esempio riportato sul sito ufficiale. Date quattro variabili di tipo intero in ingresso che rappresentano le coordinate cartesiane tra due punti, ne calcola la distanza euclidea definita come:


$$(x_a - x_b)^2 + (y_a - y_b)^2$$

```
function euclide( xa, ya, xb, yb ){
    return pow( xa - xb, 2 ) + pow( ya - yb, 2 );
}

xa = 5;
ya = 10;

xb = 2;
yb = 3;

println( "\tP( ".xa.", ".ya." ) - P( ".xb.", ".yb." ) = ".euclide( xa, ya, xb, yb ) );
```



E' stata definita ed utilizzata la funzione "euclide".



Hybris, oltre ad essere procedurale, è anche **typeless**. Questo significa che non richiede una definizione del tipo delle variabili. In base al valore con cui inizializziamo la variabile, imposta automaticamente la caratteristica, quindi il tipo, della variabile. Anche il PHP è “typeless”, il C ad esempio non lo è.

Variabile dichiarata in C:
char stringa[] = { "Hello World !" };

Variabile dichiarata in Hybris:
stringa = "Hello World !";

Gli sviluppatori ci tengono a sottolineare un **concetto fondamentale**:

“Non esiste il linguaggio migliore di tutti, esiste solo il linguaggio più adatto in determinate circostanze...immaginate un linguaggio di programmazione come ad uno strumento di un falegname ... tra tutti gli strumenti di certo non potrà dire quale è il migliore, ma solo quale è quello più adatto per un determinato scopo”.

Le caratteristiche di Hybris si possono riepilogare così:

- Semplice nella scrittura del codice
- Grammatica estremamente intuitiva, ordinata, pulita e facile da apprendere.
- Il codice è riutilizzabile, vista la forte astrazione rispetto a linguaggi non-typeless
- Supporto a tipologie ibride per dati strutturati
- Funzionalità principali integrate nell'interprete stesso

Tutto è open e gratuito. E' sicuramente un progetto complesso, e ha bisogno di essere ottimizzato da bug, infatti il Team è in cerca di collaboratori che possano aiutarli a correggere errori e proporre migliorie.

Script che legge un feed rss ogni 10 minuti e quando trova nuovi elementi mostra un message box all'utente chiedendo se vuole aprirli nel browser.

```
#!/usr/bin/hybris

last_feed = map();
// ciclo principale
while(1){
    // richiedo il feed rss
    xml = http_get( "www.sito_di_esempio.com", "/feed.xml" );
    // parserizzo il feed
    feed = xml_parse(xml);

    // se la variabile last_feed non è stata ancora inizializzata vuol dire che lo script è stato appena avviato
    if( last_feed ){
        // confronta i due feed per determinare la presenza di nuovi elementi
        if( feed != last_feed ){
            // loopa ogni elemento del feed
            foreach( item of feed["channel"][0]["item"] ){
                // controlla se il singolo elemento è nuovo
                if( contains( last_feed["channel"], item ) == 0 ){
                    // mostra il prompt all'utente
                    button = exec( "xmessage \ "Nuova notizia :\n\n".item["title"][0][ "<data>" ].
                        "\n\nAprire nel browser ?\ " -buttons Si:1,No:0 -default Si -center" );
                    if( button != 0 ){
                        exec( "firefox ".item["link"][0][ "<data>" ]." &" );
                    }
                }
            }
        }
    }
    // aggiorna la variabile
    last_feed = feed;
    // pausa per 10 minuti (60 secondi * 10)
    sleep( 600 );
}
```

Script che accetta come argomenti sulla riga di comando username, password di Twitter più una stringa con lo stato che volete inviare al vostro account di twitter.

```
#!/usr/bin/hybris

if( argc < 4 ){
    println( "Usage: ".$0." <username> <password> <status>" );
    exit();
}

// prendo gli argomenti dalla riga di comando
username = $1;
password = $2;
status    = $3;

// imposto gli header necessari per la richiesta http
head = map( // autenticazione http per la pagina delle api di twitter
    "Authorization" -> "Basic ".base64encode( username.":".password ),
    // twitter ha qualche problema con l'header expect, così lo annullo
    "Expect"        -> "" );

// imposto i dati per la richiesta POST
data = map( "status" -> status );

// eseguo la richiesta
resp = http_post( "www.twitter.com", "/statuses/update.json", data, 0, head );

// controllo il response http del server
if( resp =~ /\\"error\\"/m ){
    println( "Errore di autenticazione, controlla username e password !" );
}
else{
    println( "Hai appena postato lo status '".status.'" ." );
}
}
```

Gli esempi di codice che ho riportato li potrete trovare sul sito ufficiale <http://www.hybris-lang.org/> dove troverete anche documentazione, downloads vari e molto altro.

Davide La Pignola



BREVE PROFILO DELL'AUTORE

Consulente IT

Programmatore Java con competenze in piattaforma FileNet

Corsi frequentati:

Programmazione con C

Java base e avanzato, Hibernate

Internetworking con TCP/IP

Html, Javascript, Sicurezza e firewall