

Introduzione

- E' importante sapere come dialogare con i dispositivi domotici e i controllori dal mondo esterno.
- Tutti i linguaggi possono interfacciarsi con le periferiche che i bus rendono disponibili verso le reti classiche.
- Spesso è una questione di supervisione e controllo, mai di vera programmazione dell'impianto.

PHP - Introduzione

- E' un preprocessore per ipertesti
- Nasce per rispondere alla richiesta di siti dinamici
- Ha applicazioni anche nella cli, ma viene usato principalmente come linguaggio server side per applicazioni web
- Nasce con intenti molto più ridotti rispetto al mercato che poi è riuscito a muovere
- Molto ben integrato con Apache e MySQL già dalle prime release

PHP – Introduzione (cont'd)

- Un po' di cifre:
 - End 1998: 250.000 server web
 - Mid 2000: 2.000.000 server web
 - End 2000: 4.800.000 server web (grazie a Zend)
 - End 2002: 10.000.000 server web
 - Oggi: 65% market share nella installazione LAMP
- PHP 5 focalizza sul paradigma a oggetti
- Può girare su qualsiasi server Web, su qualsiasi sistema operativo e consente di interagire con qualsiasi tipo di database

PHP – Lato client e Server

- Il funzionamento si basa sulle richieste dei client verso il server
- Ciò che viene inviato al client non ha codice php, mascherando al client le operazioni eseguite
- E' lato server ciò che JavaScript è lato client
- Il server web riconosce le pagine PHP dall'estensione

PHP - Caratteristiche

- PHP è di alto livello -> nasconde meccanismi complessi come la gestione delle sessioni
- Sintassi nata dal C e dal Perl
- Debole tipizzazione
- Dalla versione 5 ha una buona struttura ad oggetti
- Supporta le maggiori tecnologie per la comunicazione via TCP/IP
- Dalla versione 5 ha un db SQLite embedded
- Molte librerie mirate ai servizi web e internet

PHP – Caratteristiche (cont'd)

- Molto semplice e adattabile agli stili di programmazione
- Strumenti potenti:
 - PEAR
 - PECL
 - Smarty
- Due versioni tuttora in sviluppo attivo: 4 e 5
- Forte estendibilità, supporta l'aggiunta dinamica
- Manca del supporto per Unicode (previsto in PHP 6)

PHP e HTML

- Come fa il server a distinguere dove inizia un pezzo di codice php?
 - Tag di apertura e chiusura `<?php ?>`
- `<html>`
- `<head><title>`
- `<?php echo "Pagina di prova PHP"; ?>`
- `</title></head>`
- `<body>`
- `<?php echo "Buona giornata!"; ?>`
- `</body>`
- `</html>`

PHP e HTML (cont'd)

- Echo accetta diverse stringhe e caratteri speciali
 - echo "Buongiorno a tutti!", "
\n", "È una bellissima giornata";
- Può fare la differenza usare il \n in posti sensati, rendendo il codice HTML generato più leggibile

PHP – I commenti

- Non influenzano la pesantezza dell'HTML inviato al client
- Ne supporta diversi tipi:
 - Stile C++: `//`
 - Stile Perl: `#`
 - Multilinea: `/* */`

PHP - Le variabili

- Rappresentate dal \$
- Case-sensitive
 - \$variabile = 1;
 - \$Variabile = "Questa è una variabile";
- Esistono variabili dette superglobali (\$_GET, \$_GLOBALS, etc.), hanno a che fare con i dati del server, la gestione delle sessioni e del sistema.

PHP – Le costanti

- Contenitori immutabili per valori semplici
- Non hanno bisogno del \$
- Per impostarle si usa l'istruzione `define()`
- Per sapere se è definita si usa `defined()`
- Le costanti autodefinitive da PHP sono di solito riferite a specifiche librerie importate.

PHP – Tipi di dato

- Ha dei tipi di dato primitivi
- Integer:
 - `$a = 2; //` Possono assumere altre notazioni
- Float (numeri in virgola mobile):
 - `$b = 9.876e3;`
- String:
 - `$string = "questa è una stringa";`
 - I doppi apici permettono l'espansione del contenuto
 - Esistono caratteri speciali da usare con l'escape
 - L'apice singolo non permette l'espansione

PHP – Tipi di dato (cont'd)

- Array:
 - Sia scalari che associativi
 - Può essere definito in due modi:
 - `$a = array ("abc", "def", "ghi");`
 - Oppure:
 - `$a[0] = "abc";`
 - `$a[1] = "def";`
 - `$a[2] = "ghi";`
 - Supporta l'accodamento implicito:
 - `$a[] = "lmn";`
 - `$a[] = "opq";`

PHP – Tipi di dato (cont'd)

- Array associativi
 - Basati sulla coppia nome-valore
 - `$a = array(`
 - `"nome" => "Mario",`
 - `"cognome" => "Rossi",`
 - `"email" => "mario@rossi.com",`
 - `);`
- Tutti gli array supportano l'annidamento per la costruzione di array multidimensionali
- Object (li vedremo meglio in seguito)

PHP – Operatori di base

- Supporta tutti quelli supportati classicamente da altri linguaggi:
 - Addizione $\$a + \b
 - Sottrazione $\$a - \b
 - Moltiplicazione $\$a * \b
 - Divisione $\$a / \b
 - Resto della divisione $\$a \% \b
 - Assegnazione $\$a = \text{“prova”}$
 - Altri operatori (and, or, uguaglianza, comparazione, $?:$, incremento e decremento)

PHP – Strutture di controllo

- Supporta le classiche
 - If
 - Else
 - Elseif
 - While
 - For
 - Switch
 - Foreach
 - Solo per le matrici
 - `foreach(array_expression as $value)`
 - `foreach(array_expression as $key => $value)`
 - Agisce su una copia (da PHP5 anche sull'originale)

PHP - funzioni, crittazione e FTP

- PHP supporta nativamente (built-in) moltissime funzioni, rimando al sito www.php.net per vederle tutte, ma coprono tutti i campi immaginabili nella programmazione
- La crittazione è legata alla libreria mcrypt
 - Vengono supportati molti tipi di crittazione diversi, non tutti utili
- PHP supporta nativamente FTP con funzioni a basso livello, ma anche ad alto livello

PHP - Estensioni

- Oltre ad avere centinaia di funzioni built-in, esiste la possibilità di estendere il linguaggio
- Le estensioni riconosciute devono essere listate in `php.ini`
- Servono a non appesantire inutilmente il motore

PHP – OOP

- Finalmente al livello di potenza di Java
- Ora supporta un costruttore con nome diverso dal nome della classe (`__construct`)
- Supporta i distruttori (richiamati dal garbage collector)
- Supporta la visibilità degli attributi e dei metodi
- Supporta proprietà e metodi statici (accessibili via `::`)
- Supporta l'ereditarietà

PHP – OOP (cont'd)

- Ogni definizione di classe inizia con la parola chiave class
- Viene resa disponibile la pseudo variabile `$this` se il metodo viene chiamato all'interno di un contesto di oggetto (non viene definita per i metodi di classe)

PHP – OOP (cont'd)

- Supporta l'autoloading (utile per caricare classi non ancora definite)
- Supporta l'overloading (membri e metodi)
- Supporta le classi astratte e le interfacce
- Classi built-in
 - Sono state aggiunte potenti classi built-in (Iteratori, ArrayObject)

PHP – Lanciare programmi

- Operatore di esecuzione: ``
 - Restituisce l'output (equivale a `shell_exec`)
 - In modalità sicura si possono lanciare solo i programmi nella cartella definita in `php.ini`
- Altre funzioni:
 - `Exec`: esegue un comando e restituisce solo l'ultima linea (si può usare un parametro aggiuntivo per ottenere ogni riga in un array)
 - `Passthru`: come `exec`, ma rende disponibile tutto l'output del comando
 - `System`: uguale alla funzione C

PHP - Socket

- Primitive di basso e alto livello per le socket (php è un linguaggio ad alto livello)
- Multipiattaforma
- Gestione degli errori molto importante
- Normalmente creeremo sempre dei client (domotica)

PHP – Socket (cont'd)

- Socket di alto livello: Dominio INET o UNIX
- Supporta connessioni sicure
- Si lavora come sui file
- Gestiscono direttamente gli errori (piccolo protocollo per interpretarli)
- Default bloccanti