**PROCEDURA PER LA CREAZIONE DI UN FILE .CGI**

Ogni file cgi deve iniziare con l’indicazione del percorso in cui è situato il vostro eseguibile perl.

Se usate xampp su windows di solito il percorso è il seguente

C:\xampp\perl\bin\perl.exe

Ogni pagina cgi deve essere composta in questo modo:

**Prima parte: indirizzo dell’eseguibile perl e inizializzazione delle variabili di collocamento**

In primo luogo, dunque, scriviamo le indicazioni per raggiungere l’eseguibile PERL. Nel mio caso è

#!"C:\xampp\perl\bin\perl.exe"

All’inizio del file si troveranno due variabili

*$titolo*

*$where*

$titolo è il titolo della pagina, se io voglio che il titolo della pagina sia Home - ARSR dovrò scrivere

*$titolo=”Home”;*

$where indica la parola che compare sull subheader (diciamo la barra lunga blu che c’è sotto l’header.   
Se io mi trovo in home la variabile $where deve essere settata così

*$where=’<span lang=”en”>Home</span>’;*

Questa variabile serve anche al menu per capire su quale pagina ci troviamo e se porla come attiva o meno.

Vi sono poi le variabili di stampa, sono le variabili dove si salveranno le cose che vengono poi stampate a video:

*$header;*

*$menu;*

*$footer;*

*my $htmlprint;*

Le prime tre andranno aggiunte, una volta inizializzate, in coda a $htmlprint. Basta metterle dentro “ “ perché perl interpreta anche le variabili dentro ai doppi apici. Le variabili che si usano solo in un documento vanno create ponendo un my davanti, quelle che invece passano ad altri documenti, ad esempio menu e header, non devono avere il my davanti, se no non vengono passate!

In questa parte verrà incluso il file con dentro le variabili che contengono gli indirizzi di tutte le pagine. Tale file è in preparazione da Nicolò.

**Seconda parte: il template**

Di seguito vengono riportate testualmente queste righe di codice:

*print "content-type: text/html\n\n";*

*require ("header.cgi");*

e poi se si è su lato client:

*require ("menu.cgi");*

se si è su lato amministratore:

*require ("admin\_menu.cgi");*

e infine

*require ("footer.cgi");*

i require non fanno altro che salvare header, menu e footer nelle rispettive variabili.

**Terza parte: il codice vero e proprio**

a questo punto viene la vostra parte.

bisogna capire come utilizzare le librerie xml per inserire gli xml nel codice perl.

**Quarta parte: mettere tutto insieme**

Man mano che si producono linee di html vanno aggiunte in $htmlprint. ad esempio.  
*$htmlprint=”$header$menu<div>$contenuto</div>\n$footer”;*

inserire gli \n al punto giusto è importante per creare un codice html leggibile.

alla fine di tutto dare il comando di stampa:

*print $htmlprint;*

**I Caso particolare - Parsing XML**

per fare il parsing xml è necessario includere una libreria che si chiama XML::LibXML. Per farlo è consigliato installare Strawberry Perl, fare quindi partire il file CPAN così installato.

Si aprirà una consolle, quando il prompt sarà pronto si digiti install e si prema invio.

A questo punto la libreria dovrebbe essere installata, ma non nel vostro file perl di xampp, quanto in quello di strawberry. Quindi il percorso da indicare a inizio file sarà quello del file bin di perl installato con strawberry. Nel mio caso

*#!"C:\strawberry\perl\bin\perl.exe"*

per effettuare il parsing si includa la libreria di cui sopra con il comando

*use XML::LibXML;*

Si crei quindi un oggetto parser:

*$parser = XML::LibXML->new();*

Si apra un file e lo si dia in pasto al parser:

*my $file='../data/file.xml';*

*my $doc = $parser->parse\_file($file);*

Per ottenere il nodo radice si digiti:

*my $root = $doc->getDocumentElement;*

per ottenere un array che contiene tutti gli “oggetti” dentro alla collezione si digiti:

*my @items= $root->getElementsByTagName('item');*

su questo array si può iterare come vedete nell’esempio scritto sotto, con un foreach

*foreach $item(@items);*

*{*

*}*

All’interno del foreach per accedere a un elemento specifico basta digitarne l’xpath all’interno del metodo find contando che si parte comunque dall’indirizzo dell’oggetto iterato.

es

*foreach $item(@items)*

*{*

*$imagesource=$item->find(‘image/source’);*

*print $imagesource;* #stampa il testo contenuto nel campo <source> all’interno di <image> dentro all’oggetto <item> della collezione.

*}*

**Esempio**

Ho provato a farlo con il file “esempio.cgi” che Matteo aveva già abbozzato.

Per ora non ho messo il menu lato admin perché ancora deve essere creato quindi ho incluso il menu lato client.

Questo è il risultato, come vedete è molto pulito:

*#!"C:\strawberry\perl\bin\perl.exe"*

*use XML::LibXML;*

*use CGI::Carp qw(fatalsToBrowser);*

*$parser = XML::LibXML->new();*

*$title = 'Inserisci Attrezzatura';*

*$where = '<span lang="en">administrator</span>';*

*$header;*

*$menu;*

*$footer;*

*my $htmlprint;*

*print "content-type: text/html\n\n";*

*require ("header.cgi");*

*require ("menu.cgi");*

*require ("footer.cgi");*

*$htmlprint= "$header$menu<div id=\"content\">";*

*my $fileattr='../data/attrezzature.xml';*

*my $doc = $parser->parse\_file($fileattr);*

*#controllo se il file è aperto*

*if(!$doc){$ERRORE=1;}*

*#trovo il nodo radice*

*my $root = $doc->getDocumentElement;*

*@attr = $root->getElementsByTagName('attrezzatura');*

*$id=0;*

*foreach $nod (@attr){*

*my $src=$nod->find('img/source');*

*my $alt=$nod->find('img/alt');*

*$htmlprint="$htmlprint<div class=\"attrezzatura\">\n";*

*$immagine="\n<div>\n<img class=\"attr\_img\" id=\"attr\_img$id\" src=\"$src\" alt=\"$alt\"/></div>\n";*

*my $nome=$nod->find('nome');*

*my $codice= $nod->find('codice\_prodotto');*

*my $descrizione=$nod->find('descrizione');*

*my $prezzo=$nod->find('prezzo');*

*$htmlprint= "$htmlprint$immagine\n*

*<div>\n*

*<h3>$nome</h3>\n*

*<p>Codice: $codice</p>\n*

*<p>Descrizione: $descrizione</p>\n*

*<p>Prezzo: $prezzo</p>\n*

*</div>\n*

*</div>\n";*

*$id++;*

*}*

*$htmlprint="$htmlprint</div>\n$footer";*

*print $htmlprint;*

Consiglio di inserire il comando

*use CGI::Carp qw(fatalsToBrowser);*

Che permette di visualizzare a browser gli errori del codice che scrivete, così non diventate scemi a cercare di capire cosa non va.

Io confido che alla Gaggi questo vada bene.

Tante buone cose,

Giovanni