

Progettazione di Servizi Web e Reti di Calcolatori

Politecnico di Torino – Prof. Antonio Lioy

AA 2018-2019, esercitazione di laboratorio n. 8

Per scrivere file HTML/CSS/JS/PHP occorre usare un normale editor di testo (es. Notepad++). Per file HTML si può avere indifferentemente estensione “.htm” o “.html”. Per CSS “.css”. Per Javascript “.js” e per PHP “.php”.

Se le pagine sono di tipo statico e sono scritte correttamente con link relativi, allora è possibile posizionarle in una qualunque cartella del filesystem e navigarle direttamente aprendole con un qualunque browser.

Le pagine PHP di questa esercitazione sono invece di tipo dinamico lato server. Per usarle correttamente bisogna accedervi attraverso un server HTTP integrato con un interprete PHP (es. tramite XAMPP), e *non* accedere direttamente al corrispondente file su disco locale come fatto nelle esercitazioni precedenti. Un esempio di accesso corretto sarebbe:

<http://localhost/pagina.php> (notare “http” ad inizio link)

mentre un esempio errato sarebbe:

<file:///C:/PWR/lab8/pagina.php> (notare “file” ad inizio link, chiaro indice di **errore!!!**).

Ricordarsi di avviare il server HTTP prima di svolgere l'esercitazione (es. tramite il pannello di controllo XAMPP).

E' anche buona norma provare ad accedere non solo localmente ma anche via rete alle pagine sviluppate. Ad esempio, ipotizzando che il nodo su cui viene attivato XAMPP abbia indirizzo 192.168.1.33 e che ospiti le pagine sviluppate per l'esercizio 8.1 (i cui nomi siano lab8_1.html e lab8_1.php), si usi il browser su un altro nodo e ci si colleghi alla pagina di partenza:

http://192.168.1.33/lab8_1.html

Una volta introdotti i dati nel form e premuto il pulsante Invia deve essere visualizzata la pagina:

http://192.168.1.33/lab8_1.php

Negli esercizi che prevedono l'introduzione di numeri, verificarne il corretto funzionamento non solo introducendo dati validi (es. “5”, “-3”, “+7”, “2.3”, “5e3”) ma anche agendo volutamente in modo scorretto:

- introducendo dati errati (es. “5mila”, “cinquemila”, “5 mila”, “2,3”);
- inviando il form senza aver inserito tutti i dati richiesti.

Si suggerisce di sviluppare ogni esercizio prima in forma base (ossia senza particolari controlli di errore) e quindi facendo le necessarie modifiche per trattare anche il caso di introduzione di dati errati.

Si ricorda che per validare il codice HTML delle pagine contenenti elementi Javascript client-side è necessario usare un validatore installato all'interno del browser. A questo scopo, si consiglia l'uso del plugin “HTML validator” (come nelle precedenti esercitazioni).

Lo stesso concetto e *modus operandi* si applicano anche alla validazione delle pagine PHP, visto che il codice HTML viene generato dall'esecuzione dello script e non è quindi disponibile sul server ma solo sul cliente che riceve il risultato dell'esecuzione dello script.

Per ciascun esercizio, si suggerisce di realizzare l'invio dei dati dei form prima tramite il metodo il metodo GET (particolarmente utile per fare il debug del sito perché visualizza all'interno della URL i parametri passati ed i relativi valori) e poi tramite il metodo il metodo POST (procedura più professionale perché nasconde i dati all'utente e non lascia traccia dei parametri e dei valori nel log del server HTTP).

Per ogni esercizio in cui ciò sia possibile e sensato, verificare lato client, prima dell'invio al server, che i dati inseriti rispettino le specifiche indicate nel testo dell'esercizio (es. siano numeri interi) ed in caso contrario i dati non siano inviati al server ma sia visualizzato un messaggio di errore.

N.B. Questo controllo "lato client" *non* implica la possibilità di non sviluppare gli opportuni controlli anche lato server. Infatti la pagina PHP potrebbe essere acceduta con intenti malevoli, ricevendo quindi valori errati o contraffatti; ciò rende quindi indispensabili i controlli lato server.

Esercizio 8.1

Creare una pagina HTML contenente un form con un campo di input testuale e due pulsanti, rispettivamente *Cancella* e *Invia*.

Realizzare quindi una pagina PHP che riceva i dati dal form e crei una pagina di risposta che abbia come titolo di primo livello "Testo ricevuto" seguito da un paragrafo col contenuto del campo di input testuale.

Esercizio 8.2

Realizzare una pagina HTML contenente un form con due campi di input testuale (per inserire due numeri interi) e due pulsanti, rispettivamente *Cancella* e *Somma*.

Alla pressione del pulsante *Somma* i due numeri introdotti devono essere inviati ad una pagina PHP. La pagina PHP deve creare una pagina di risposta contenente la somma dei due numeri nella forma " $A + B = S$ " ed un link per tornare alla pagina iniziale, associato al testo "Torna alla pagina iniziale".

Alla pressione del pulsante *Cancella* i due campi devono invece essere ripuliti da eventuali numeri che siano stati introdotti.

Esercizio 8.3

Realizzare una pagina HTML contenente un form con un menù a tendina per scegliere un numero intero N compreso tra 1 e 10 ed un pulsante "Crea tabella".

Realizzare quindi una pagina PHP che sia attivata quando viene premuto il pulsante, riceva i dati del form e crei una pagina di risposta contenente la tabella dei quadrati e dei cubi da 1 al valore N introdotto nel form (ogni riga della tabella deve contenere un numero K, il suo quadrato ed il suo cubo).

Esercizio 8.4

Realizzare una pagina HTML contenente un form con:

- un campo di testo per introdurre il prezzo netto di un prodotto (come numero intero oppure reale, con al massimo due cifre dopo la virgola)
- un radio button per scegliere tra tre diverse aliquote IVA (minima, ridotta, ordinaria)
- un pulsante per inviare i dati al server, e far calcolare il prezzo IVA inclusa, presentato con due cifre dopo la virgola

Realizzare quindi una pagina PHP che calcoli e visualizzi il prezzo IVA inclusa, presentato con due cifre dopo la virgola. Il calcolo dell'IVA verrà fatto in base alla corrispondenza: minima=4%, ridotta=10%, ordinaria=22%.