

Progettazione di Servizi Web e Reti di Calcolatori

Politecnico di Torino – Prof. Andrea Atzeni

AA 2023-2024 esercitazione di laboratorio n. 10

Per scrivere file HTML/CSS/JS/PHP occorre usare un normale editor di testo, presente localmente o direttamente l'ambiente di sviluppo "remote Code Server", disponibile nel cloud del corso alla URL (specifica per ciascuno studente) avente la seguente forma generale:

https://<nome_server>/<sMatricola>/dev

I file HTML possono avere indifferentemente estensione ".htm" o ".html", i file CSS ".css", quelli Javascript ".js" e quelli 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.

Questa esercitazione richiede invece la creazione anche di pagine dinamiche lato server in PHP. Per usarle correttamente, bisogna accedervi tramite un interprete PHP, che è integrato nel server HTTP usato dall'ambiente cloud del corso. E' errato accedere direttamente al corrispondente file su disco locale, come fatto nelle esercitazioni precedenti le pagine statiche HTML. Invece, è sempre necessario passare attraverso un server web che esegua il codice richiamando l'interprete PHP.

In alcuni ambienti di sviluppo moderni, come Visual Studio Code, questo può essere fatto usando un'estensione dell'ambiente di sviluppo che contenga un web server e l'interprete PHP (ad esempio, con VSC è possibile installare l'estensione "PHP Server").

Il codice PHP inserito nel nostro ambiente viene presentato alla URL:

https://<nome_server>/<sMatricola>/website/<percorso_file>

Un esempio di URL corretta (da inserire come indirizzo della pagina nel browser) sarebbe:

<https://195.231.2.153.sslip.io/s12345/website/lab9/pagina.php>

(notare "https" ad inizio link) mentre un esempio errato sarebbe:

<file:///C:/PWR/lab9/pagina.php>

(notare "file" ad inizio link, indice di errore, in quanto assente il passaggio mediante server HTTP).

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 *modus operandi* si applica anche alla validazione delle pagine PHP: poiché il codice HTML viene generato dall'esecuzione dello script lato server, non è sensato validare con un validatore HTML direttamente la pagina PHP, in quanto il codice HTML finale non è disponibile sul server ma solo sul client (dopo l'esecuzione dello script server side appunto).

Per ciascun esercizio, si suggerisce di realizzare l'invio dei dati dei form prima tramite 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 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.

Per verificare le espressioni regolari si possono usare gli strumenti offerti da <http://regex101.com/>.

Esercizio 10.1

Usando il concetto di sessione implementato in PHP, realizzare una pagina (PROD) contenente una serie di beni presentati con una tabella in cui:

- la prima colonna contenga il nome del prodotto (CD, DVD, memoria SD, memoria USB);
- la seconda colonna contenga una breve descrizione del prodotto;
- la terza colonna contenga il prezzo del prodotto (rispettivamente 0,5 Euro, 1 Euro, 7.5 Euro e 5 Euro);
- nella quarta colonna sia possibile impostare la quantità desiderata di ciascun prodotto tramite un campo testuale (posto inizialmente a zero).

Premendo un pulsante apposito "aggiungi al carrello" si deve accedere ad una seconda pagina (BASKET) che presenti la quantità totale ed il prezzo complessivo dei beni selezionati.

All'interno della stessa sessione, accedendo nuovamente alla pagina PROD, nella quarta colonna devono essere pre-impostate (ma modificabili) le quantità selezionate all'ultimo accesso.

Esercizio 10.2

Dopo aver svolto l'esercizio precedente, identificare quale sia stato il valore dell'identificativo di sessione usato (e scelto automaticamente dal server) per la sessione precedente

Esercizio 10.3

Realizzare un sito web per "acquistare" una serie di prodotti, strutturato su diverse pagine.

La prima pagina (PROD) presenta la lista dei beni come nell'esercizio 10.5 ma il pulsante "aggiungi al carrello" è presente al fianco di ciascun prodotto (quinta colonna della tabella) ed è relativo solo a quel prodotto. Cliccando sul nome del prodotto si accede invece ad una specifica pagina contenente una breve descrizione del prodotto ed il suo prezzo.

Da ogni pagina deve essere possibile raggiungere (tramite un apposito link) una pagina di riepilogo (BASKET), che rappresenti il "carrello" dei beni selezionati sino a quel momento, elencando solo i prodotti selezionati, la rispettiva quantità ed il costo totale. Da questa pagina è possibile tornare alla pagina PROD per aggiungere altri prodotti oppure, tramite un pulsante di conferma ("compra"), procedere con "l'acquisto" visualizzando una pagina (CHECKOUT) che svuota il carrello e conferma che l'ordine è andato a buon fine.

Esercizio 10.4

A partire dall'esercizio precedente, Inserire un contatore "statistico" che tenga traccia, per la sessione corrente, del numero medio di pagine visitate dagli utenti prima di procedere ad un

acquisto (cioè quante pagine vengono visitate, prima di procedere ad un checkout, corrispondente idealmente ad un acquisto).

Esercizio 10.5

Realizzare un sito costituito da una pagina iniziale (FIRST) che permette di scegliere se visualizzare informazioni su Torino o su Milano, a cui devono corrispondere due pagine separate. Ogni pagina deve essere disponibile sia in inglese sia in italiano, deve essere possibile cambiare lingua in qualsiasi momento da una qualunque delle pagine e la scelta deve essere ricordata per tutta la sessione anche se l'utente visita altre pagine esterne al sito.