

Progettazione di Servizi Web e Reti di Calcolatori

Politecnico di Torino – Prof. Andrea Atzeni

AA 2023-2024 esercitazione di laboratorio n. 9

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/>.

Creare una pagina A che visualizza la scritta "Italia!" ed imposta sul browser un cookie con nome "Country" e valore "IT", senza specificare la scadenza del cookie.

Esercizio 9.1

Utilizzando l'esempio sull'uso della variabile `$_SERVER` visto a lezione, realizzare la pagina PHP "server_info.php" che visualizzi le seguenti informazioni:

Benvenuto,
hai richiesto la pagina `<URL>` usando il metodo `<metodo_HTTP>`.
Stai usando il browser `<nome_browser>` dal nodo con indirizzo IP `<IP_client>`.
Sei collegato al server `<nome_server>` (indirizzo IP `<IP_server>` e porta `<porta_server>/TCP`).

Si noti che la `<URL>` deve essere costruita partendo da diverse variabili dell'array `$_SERVER`, ad esempio:

<http://localhost/lamiapagina.php>

Esercizio 9.2

Creare quindi una pagina B che legga il valore del cookie con nome "Country" e lo visualizzi (se il cookie esiste) altrimenti indichi che il cookie è inesistente.

Per verificare il corretto comportamento delle pagine create, svolgere nell'ordine i seguenti passi:

- visitare prima la pagina A e poi la pagina B;
- chiudere quindi il browser e poi ri-avviarlo;
- visitare la pagina B, poi la pagina A e quindi la pagina B.

Esercizio 9.3

Utilizzando il browser, aprire una nuova scheda e visualizzare 3 o 4 siti a piacere (es. <http://www.libero.it>, <http://www.polito.it>). Confrontare il risultato visualizzato dalla pagina creata con quello ottenibile visualizzando i cookie direttamente attraverso il browser (*developer's tools, tab Application, selezionare il server di interesse nel DB cookie*). E' possibile visualizzare i cookie di altri siti web? Perché?

Esercizio 9.4

Modificare la pagina A dell'esercizio 9. impostando come data di scadenza del cookie il 31 dicembre 2024 e quindi ripetere le visite delle pagine A e B come descritto nell'esercizio 9.2: quali cambiamenti si osservano? Perché capitano? E quali cambiamenti si osservano impostando la scadenza del cookie al 31 dicembre 2023?

Esercizio 9.5

Creare una pagina HTML che contenga un form con due campi di input testuali, “nome” e “cognome”, e due pulsanti rispettivamente per il reset (Cancella) e per l’invio dei dati (Invia). I campi “nome” e “cognome” non devono superare i 30 caratteri e devono contenere solo lettere, di cui la prima maiuscola.

Il form invia i dati ad una pagina PHP che oltre al contenuto (es. una breve descrizione su un argomento a piacere), per ogni accesso diretto alla pagina (ossia non tramite il form HTML) successivo al primo, mostri anche un saluto personalizzato per l’utente, ad esempio:

```
Bentornato, caro <nome> <cognome>, nel mio umile sito.
```

Esercizio 9.6

Inerentemente l’esercizio precedente, considerando un ciclo completo di interazioni, composto da

1. Richiesta della pagina web contenente il form
2. Invio dati del form al server web
3. Pagina di risposta del server (contenente breve descrizione)
4. Accesso (in un ipotetico momento successivo) diretto alla pagina contenente la breve descrizione (+ il saluto)

individuare quali campi di intestazione HTTP inerente il trasferimento dei cookie compaiano nei 4 messaggi scambiati, indicando se si tratta di una “request” o una “response” per il protocollo HTTP.

SUGGERIMENTO: possibile investigare lo scambio tramite uno strumento come Wireshark (che fornisce molti dettagli su tutti i livelli di rete, ma in questo caso necessita del superamento di un ostacolo) o tramite i *developer’s tool (tab network)*.