

Esercizi

Lezione 4: PHP, HTML e le richieste GET e POST

Per affrontare i seguenti esercizi potrebbero essere utili le slide ed eventuale documentazione reperita online.

Esercizio 1

passaggio di parametri GET

Costruire uno script denominato *index.php* che svolga le seguenti operazioni:

- controlli se è avvenuta una richiesta di tipo GET
- in caso affermativo controlli se è stato impostato il parametro *azione*
- se è stato impostato il parametro *azione* ne stampi il valore altrimenti stampi *Richiesta GET senza parametri*

Provare lo script appena creato raggiungendolo navigando con il browser all'indirizzo `localhost/index.php?azione=1` o eventualmente modificando il valore del parametro *azione*. Cosa succede modificando il nome del parametro raggiungendo la pagina?

Cosa succedrebbe se effettuassimo una richiesta POST alla pagina? Potremo farlo come facciamo con le richieste GET o dovremo usare altri strumenti più complessi?

Esercizio 2

da vettori associativi a JSON e viceversa

Definire un vettore associativo che associ alle chiavi *sinistra*, *destra*, *davanti*, *indietro* i nomi dei compagni di classe posti in tale direzione sotto forma di un vettore classico. Il vettore finale dovrà dunque essere un vettore associativo che associa alle chiavi precedentemente specificate quattro vettori (al massimo monoelemento o vuoti).

Mediante le funzioni presentate a lezione di teoria codificare ora il vettore come stringa JSON; tale stringa andrà stampata a schermo e memorizzata all'interno di una variabile chiamata `$stringa_json`.

Utilizzare ora il complementare della funzione appena utilizzata per ricreare un vettore associativo. Cosa succede se omettiamo il secondo parametro `true`? Otteniamo lo stesso un vettore associativo o qualcosa di diverso?

E se al posto di quattro vettori classici fossero vettori associativi? La risposta alla domanda precedente sarebbe la stessa?

Esercizio 3

province

*Attenzione: prima di buttarsi a capofitto sulla risoluzione dell'esercizio leggere **attentamente** l'intera consegna; non viene infatti richiesto di creare un sito web statico come siete certamente già capaci ma di creare una base statica che verrà popolata di contenuti grazie al codice PHP*

Costruire un sito web usando HTML, CSS e PHP che contenga le seguenti pagine:

- *index.php*
- generiche pagine *nomeProvincia.php* (per esempio potrebbero esserci le pagine *torino.php*, *milano.php*, ...) [almeno 3]
- *dati.php* che contenga le informazioni delle province sotto forma di array annidato

Nella pagina **index.php** (ovvero la homepage del sito) dovranno comparire i seguenti elementi:

- il titolo **Alcune province italiane**
- un elenco puntato contenente i nomi delle province scelte (tali nomi dovranno essere dei collegamenti ipertestuali a ciascuna pagina *nomeProvincia.php* specificate in precedenza)

Nelle pagine **nomeProvincia.php** dovranno invece essere presenti:

- il titolo contenente il nome della provincia
- le informazioni relative alla provincia quali gli abitanti, la regione, la collocazione geografica, ... formattate in una tabella

Si noti che per il front-end basta creare una pagina per le province e ricopiarla cambiando il nome ogni volta. Per l'aspetto grafico seguire un template a piacere purché si faccia uso del CSS e che siano presenti gli elementi citati in precedenza

Una volta realizzato il front-end si proceda a integrare il codice PHP come segue:

- nella pagina *dati.php* sarà presente un array annidato che associa a ogni nome di provincia un array contenente i dati (organizzati per *chiave parlante* ovvero dove ogni chiave in formato stringa specifica il dato, per esempio alla chiave *Regione* sarà associata la regione che contiene la provincia)
- nella pagina *index.php* l'elenco puntato sarà generato da un'istruzione **foreach** che stamperà gli elementi dell'elenco puntato mediante un'istruzione **echo**
- nelle pagine *nomeProvincia.php* invece i campi della tabella saranno stampati attraverso un **echo** che stamperà ogni campo della tabella

Naturalmente, essendo il vettore annidato dichiarato all'interno del file *dati.php* servirà usare l'istruzione **require** per accedere ai dati contenuti nello stesso dalle altre pagine.

Esercizio 4 - svolto

fiumi italiani

Costruire un sito web usando HTML, CSS e PHP che contenga le seguenti pagine:

- *index.php*
- *fiume.php*
- *dati.php* che contenga le informazioni di alcuni fiumi sotto forma di un vettore associativo

La pagina **index.php** dovrà mostrare il titolo **Fiumi italiani** e un link denominato *fiume nomeFiume* che porti alla pagina **fiume.php** passando come parametro GET il nome del fiume (che è la chiave del vettore associativo definito nel vettore della pagina **dati.php**)