

# Programmazione distribuita I

(01NVWOV)

## AA 2014-2015, Esercitazione di laboratorio n. 6

Obiettivo: scrivere pagine dinamiche in PHP. Manuale PHP disponibile all'URL <http://php.net>

Per la presente esercitazione si utilizzerà la suite XAMPP, già predisposta in laboratorio in ambiente Windows. Dopo aver lanciato XAMPP, attivare Apache dall'apposito pannello di controllo. La cartella dove depositare i files visibili dal web browser è **c:\xampp\htdocs**. **Creare una sottocartella** e lavorare solo al suo interno per evitare di distruggerne il contenuto (links di xampp ecc.).

Per scrivere pagine PHP l'estensione del file in cui è memorizzata la pagina **deve obbligatoriamente essere** “.php” altrimenti le pagine non vengono riconosciute dal web server apache già predisposto in XAMPP.

Inoltre, per verificare il funzionamento delle pagine PHP, esse **devono essere accedute tramite il server web** (es. <http://localhost/miacartella/pagina.php>) e non accedendo direttamente al corrispondente file su disco locale.

Per ciascun esercizio, si suggerisce di realizzare l'invio dei dati dei form alternativamente tramite il metodo POST o tramite il metodo GET (quest'ultimo è particolarmente utile per fare il debug del sito perché visualizza all'interno della URL i parametri passati ed i relativi valori).

Per debuggare il funzionamento delle pagine PHP, si suggerisce l'utilizzo del pacchetto “Eclipse for PHP Developers”, già predisposta in laboratorio. Impostare come **Workspace C:\xampp\htdocs** cosicché la directory radice del web server coincida con quella dei lavori debuggati tramite Eclipse. Per creare un nuovo progetto: File / New project / PHP, inserire un nome di cartella, e procedere premendo OK. Tramite File / New / PHP file, creare un nuovo file, per es. `<?php phpinfo(); ?>` ed eseguirlo tramite Run / Run as / PHP Web page accettando l'URL proposto. Tale configurazione consentirà di impostare dei breakpoint e debuggare il codice PHP sviluppato.

**NB:** Al solo scopo di **facilitare il debug**, si consiglia di utilizzare l'estensione .php anche per i files che contengono solo codice HTML, cosicché si possa impostarli come punto di partenza di una sessione di debug. Nella “Debug configurations”, verificare che sia impostato “Xdebug” come “Server Debugger”. E' anche utile impostare il flag “Break at first line”.

**NB:** I files posti in **c:\xampp\htdocs** sono visibili a tutti gli utenti che utilizzano il computer, indipendentemente dall'utente che usano per il login. **NON** lasciare files personali o che devono essere protetti, come la soluzione delle proprie prove d'esame di programmazione web!

### Esercizio 6.1.1

Creare una pagina HTML contenente un form con un campo di input testuale e due pulsanti, rispettivamente Reset e Submit.

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

### Esercizio 6.1.2

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

Realizzare quindi una pagina PHP che riceva i dati del form, verifichi la correttezza dell'input e visualizzi la tabella dei quadrati e dei cubi da 1 al valore N introdotto nel form.

### Esercizio 6.1.3

Realizzare una pagina HTML contenente un form per introdurre nome, cognome e sport praticati (scelta multipla tra Basket, Calcio, Nuoto, Sci e Volley).

Realizzare quindi una pagina PHP che riceva i dati del form e produca una tabella formata da due colonne. Nella prima colonna vanno elencati tutti i nomi dei campi del form di input e delle variabili server (presenti in `$_SERVER`) mentre nella seconda colonna vanno elencati i corrispondenti valori.

### Esercizio 6.1.4

Creare una pagina HTML contenente

- un form con due campi di input testuale:
  - nel primo inserire il nome (lunghezza massima 30 caratteri)
  - nel secondo inserire l'età (max 120 anni)
- due pulsanti rispettivamente per il reset di tutti i campi e per l'invio dei dati al server.

Realizzare quindi una pagina PHP che riceva i dati del form e – se i dati rispettano i limiti – risponda con un saluto confidenziale usando il nome del richiedente ed aggiungendo l'appellativo:

“piccolo bimbo” se età < 10

“giovine amico” se  $10 \leq \text{età} < 30$

“gentile signore” se età  $\geq 30$

Ad esempio, inserendo nel form i dati (“Dante” e “56”) la risposta visualizzata dal browser dovrebbe essere “Ciao Dante, gentile signore, tutto bene oggi?”

### Esercizio 6.1.5

Realizzare una pagina HTML contenente un form (con nome “calcolatrice”), con i seguenti campi:

- due input testuali con valore numerico (operando1 e operando2)
- tipo di operazione (un valore a scelta tra +, -, \*, /)
- due pulsanti rispettivamente per il reset di tutti i campi e per l'invio dei dati al server.

Realizzare quindi una pagina PHP che indichi l'operazione svolta ed il suo risultato.

### Esercizio 6.2.1

Creare una pagina A che visualizza la scritta “Italia!” ed imposta sul browser un cookie con nome “Country” e valore “IT”.

Creare quindi una pagina B che legga il valore del cookie con nome Country e lo visualizzi.

Per verificare il corretto comportamento delle due 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 6.2.2

Realizzare una pagina PHP che visualizzi una tabella contenente tutti i cookie che il server web riesce a leggere dal browser che visita la pagina. In particolare nella prima colonna vanno elencati i nomi dei cookie, mentre nella seconda colonna vanno elencati i corrispondenti valori.

Confrontare il risultato visualizzato dalla pagina con quello ottenibile visualizzando i cookie direttamente dal menù del browser (es. in Firefox: Strumenti / Opzioni / Privacy / Cookies).

### Esercizio 6.2.3

Modificare la pagina A dell'esercizio 6.2.1 impostando come data di scadenza del cookie il 31 dicembre 2010 e quindi ripetere le visite delle pagine A e B come descritto nell'esercizio 6.2.1.

### Esercizio 6.2.4

Creare una pagina HTML che contenga un form con due input testuali, "nome" e "cognome", e due pulsanti rispettivamente per il reset e per l'invio dei dati al server.

Il form deve far riferimento ad una pagina PHP contenente una (breve) descrizione su un argomento a piacere, ed in più, per ogni accesso alla pagina successivo al primo, anche un saluto personalizzato per l'utente, ad esempio

*Bentornato, caro <nome> <cognome>, nel mio umile sito*

### Esercizio 6.2.5

Creare una pagina (UNO) dotata solo di un bottone ENTRA che serve ad accedere ad un'altra pagina (DUE) che contiene il testo di una barzelletta ed un pulsante ESCI che fa tornare alla pagina iniziale.

L'accesso alla pagina DUE deve essere permesso solo se non si è già richiesto la visione della barzelletta più di due volte. Si suppone che il termine della visione si abbia quando viene premuto il pulsante ESCI.

Nota: per simulare l'accesso multiplo alla pagina basta collegarsi alla pagina DUE con diverse finestre o tab del browser.

### Esercizio 6.2.6

Realizzare una pagina PHP contenente una serie di beni organizzati in 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;
- nella quarta colonna sia possibile impostare la quantità desiderata tramite un campo testuale.

Premendo un pulsante apposito si deve accedere ad una pagina che presenti la quantità totale ed il prezzo complessivo dei beni selezionati.

Accedendo nuovamente alla pagina iniziale nella quarta colonna devono essere presenti le quantità selezionate all'ultimo accesso.

NB: Per assicurarsi che tutto funzioni, non usare il tasto back ma ricaricare la pagina di partenza scrivendone l'URL o premendo il bottone Reload.

### Esercizio 6.2.7

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

La prima pagina presenti la lista dei beni acquisibili, e per ogni bene contenga una casella di input che permetta di specificarne la quantità che si desidera acquistare.

Cliccando sul nome del prodotto si accede ad una diversa pagina con una breve descrizione del prodotto contenente anche il prezzo di tale prodotto.

Da ogni pagina deve essere possibile raggiungere una pagina di riepilogo, che rappresenti il “carrello” dei beni acquisiti sino a quel momento, rappresentando i prodotti che si desiderano acquistare e la rispettiva quantità selezionata. Da questa pagina è possibile tornare alla pagina iniziale, oppure, tramite un pulsante di conferma (“compra”), procedere con “l'acquisto”, o infine modificare le quantità scritte nell'elenco (in apposite caselle di testo) e farle aggiornare mediante un pulsante “aggiorna”.

Premendo il pulsante di conferma si visualizza una pagina di riepilogo con tutti i prodotti scelti, il totale relativo alle varie tipologie di prodotti ed il totale complessivo di tutti i prodotti acquistati (in questa pagina le quantità non sono modificabili). Si metta inoltre a disposizione un pulsante che richieda di confermare il proprio desiderio di acquisto prima di concludere l'ordine.