

**Università degli Studi di Camerino**  
**Corso di Laurea in Informatica A.A. 2015/2016**  
**Prova Scritta di Programmazione Web del 21 Giugno 2016 – I Appello**  
**Docenti: Francesco Tiezzi e Luca Tesei**

**Nome e Cognome:** \_\_\_\_\_ **Matricola:** \_\_\_\_\_

**Durata della Prova:** 45 minuti

**Domande a Risposta Aperta:** scrivere nello spazio indicato, in caso di spazio insufficiente continuare sul retro del foglio.

1. (6 punti) Descrivere il ciclo di vita di una Servlet ed i metodi coinvolti nella sua gestione.

*Si vedano le slides "Core servlets: Servlet basics" e Sezione 3.6 del Capitolo 3 del libro di testo.*

2. (6 punti) Descrivere le principali differenze fra le tecnologie Servlet e JSP.

*Si vedano le slides "Introduction to JSP" e Capitolo 10 del libro di testo.*

3. (6 punti) Descrivere le seguenti caratteristiche del linguaggio Javascript: tipi delle variabili, modello di memoria e scope (visibilità) delle variabili.

*La dichiarazione di variabili in Javascript non richiede l'indicazione del tipo. Inoltre, il tipo del valore contenuto in una variabile può cambiare nel corso della vita della variabile stessa. I tipi di valore ammessi sono: null, Numeric (interi o virgola mobile), String, Boolean, Array (con indici numerici) e Object (Collezione di variabili e funzioni con nome).*

*Il modello di memoria di Javascript è composto da tre aree distinte: l'ambiente globale (in cui vengono allocate le variabili globali, non strutturato), l'ambiente locale (in cui vengono allocate le variabili locali, strutturato a pila) e lo heap (in cui vengono allocati array e oggetti, non strutturato).*

*Le variabili dichiarate fuori dalle funzioni o con lo specificatore "global" sono globali; sono visibili ovunque ed esistono fino alla chiusura della pagina web contenente lo script. Le variabili dichiarate all'interno delle funzioni (o i parametri formali delle stesse) sono visibili all'interno della funzione in cui sono dichiarate (o specificate come parametri) e, a cascata, nelle funzioni chiamate da questa (a meno di sovrapposizione di nomi). Esse vengono deallocate quando la funzione in cui sono state dichiarate termina. In caso di sovrapposizione di nomi si applica la regola seguente: un nome si riferisce alla variabile locale più recente dichiarata con quel nome.*

4. (6 punti) Descrivere il meccanismo della definizione e chiamata di funzioni e del passaggio dei parametri nel linguaggio PHP

*In PHP una funzione può essere definita in qualsiasi punto del codice in cui sia possibile iniziare un nuovo comando. La sintassi è la seguente*

```
function nome_funzione($p_1, $p_2, ..., $p_n) {  
// Codice della funzione con eventuale return; o return val; }
```

*dove \$p\_1, ..., \$p\_n è una lista di nomi di variabili (parametri formali), eventualmente vuota e "val" è un valore di un qualsiasi tipo. La chiamata di funzione si ottiene scrivendo:*

```
nome_funzione(t_1, ..., t_n);  
// oppure  
x = nome_funzione(t_1, ..., t_n); //se la funzione ritorna un valore
```

*dove t\_1, ..., t\_n sono termini (variabili o espressioni di variabili e costanti) chiamati parametri attuali il cui valore verrà associato ai nomi dei parametri formali all'inizio della esecuzione della funzione. Il meccanismo di passaggio dei parametri è principalmente per valore, cioè viene creata una copia del valore del parametro attuale t\_i e assegnato alla variabile (parametro) \$p\_i per ogni i. Nel caso di oggetti il valore passato è il riferimento all'oggetto che, quindi, può essere modificato dal codice della funzione. E' anche possibile il passaggio di parametri per riferimento come segue:*

```
function inc( &$i ) { $i = $i + 1; }  
...  
$x = 3;  
inc($x);  
// qui x vale 4
```

**Domande a Risposta Chiusa:** contrassegnare una sola risposta tra le quattro possibili scelte. Ogni domanda vale 1 punto se la risposta è corretta, -0,5 punti se la risposta è errata e 0 punti se non c'è risposta.

1. L'elemento HTML `form` permette di:

- a) definire un modulo
- b) formattare il testo con uno specifico font
- c) definire una formula matematica
- d) importare un foglio di stile

2. In una servlet, per leggere i dati di una form si utilizza:

- a) il metodo `getParameter("name")` dell'oggetto di tipo `HttpServletRequest`
- b) il metodo `getFormData()` dell'oggetto di tipo `Request`
- c) il metodo `readParameter("name")` dell'oggetto di tipo `PrintReader`
- d) l'attributo `formParameters` di un'istanza della classe `HttpServletRequest`

3. L'esecuzione di una pagina JSP richiede:

- a) la traduzione della pagina JSP in servlet
- b) l'interpretazione di un comando alla volta della pagina JSP
- c) la traduzione diretta della pagina JSP in linguaggio macchina
- d) la traduzione diretta della pagina JSP in bytecode

4. In Javascript la funzione `document.getElementById(<parametro>)`:

- a) restituisce una funzione tramite la quale è possibile modificare la variabile indicata dal `<parametro>`
- b) restituisce un riferimento all'elemento della pagina HTML il cui attributo `id` è uguale al `<parametro>`, che deve essere una stringa
- c) restituisce il valore dell'attributo di nome `<parametro>`, che deve essere una stringa, del tag `body` della pagina HTML
- d) aggiunge l'elemento `<parametro>` alla pagina HTML

5. In PHP la variabile superglobal `$_POST`:

- a) Contiene l'indirizzo IP del client che ha effettuato la richiesta della pagina PHP
- b) Contiene l'indirizzo di posta elettronica associato ad un form HTML inviato alla pagina PHP
- c) E' un array associativo (mappa) contenente tutti i valori degli elementi input del form HTML inviato alla pagina PHP tramite metodo POST
- d) E' un array associativo (mappa) che associa un identificatore unico ad ogni form HTML che usa il metodo POST inviato alla pagina PHP

6. In PHP il modo corretto per aprire il file `time.txt` in sola lettura è:

- a) `open("time.txt");`
- b) `fopen("time.txt", "r+");`
- c) `fopen("time.txt", "r");`
- d) `open("time.txt", "read");`