

Esercizi

Lezione 3: le funzioni e i require/include in PHP, richiami di HTML e CSS

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

Esercizio 1

scrittura di funzioni che manipolano dati

Scrivere una funzione che, date due stringhe verifichi se le stesse sono identiche usando un approccio *C-like* (ovvero confrontando carattere per carattere); nel caso in cui queste fossero identiche restituire 0, altrimenti restituire una stringa contenente i primi due caratteri diversi

Suggerimento: le due stringhe devono essere passate per valore (onde evitare di fare danni su quelle originali che **non** devono essere modificate), inoltre le due stringhe non devono essere per forza controllate per intero: nel caso in cui un carattere fosse diverso possiamo restituire direttamente i due caratteri frutto di una concatenazione di due stringhe monoelemento.

Esercizio 2

numeri primi

Scrivere una funzione che, dato un numero, verifichi se lo stesso è primo oppure no; restituire un booleano (**true** per primo o **false** per non primo). Anche in questo non è lecito usare funzioni di libreria; bisogna ragionare sulla caratteristica matematica dei numeri primi e codificarla in codice.

Suggerimento: non viene richiesto di calcolare il numero di divisori perciò possiamo abortire i controlli al primo controllo indicativo fallito; per farlo è opportuno ragionare sul range di confronti da applicare.

Esercizio 3

modifica di vettori

Scrivere all'interno di uno script PHP denominato **funzioni.php** una funzione che, dati due vettori passati per riferimento, vada a modificarne i contenuti scambiando l'elemento *i*-esimo del primo vettore con l'elemento *i*-esimo del secondo.

Per esempio dati i vettori `$dispari = [1,3,5,7,9]` e `$pari = [2,4,6,8,10]` al termine dell'esecuzione della funzione dati gli stessi vettori quali parametri si dovranno avere i vettori di prima pari a `$dispari = [2,4,6,8,10]` e `$pari = [1,3,5,7,9]`.

Modificare in seguito la funzione di modo che presenti un *TypeError* se i parametri non sono del tipo **array**.

Suggerimento: i vettori devono essere della stessa dimensione perciò è bene verificarlo all'inizio dell'esecuzione della funzione; non essendo specificato un valore di ritorno la funzione non dovrà presentare un tipo di ritorno quando la *irrobustiamo* nella parte del *TypeError*.

Esercizio 4 - svolto

ordinamento di vettori C-like

Scrivere all'interno di uno script PHP denominato **SelectionSort.php** una funzione che, dato un vettore, lo ordini applicando l'algoritmo di ordinamento Selection Sort.

La funzione riceverà come secondo parametro un valore booleano che indicherà se la funzione dovrà operare per valore (restituendo un vettore ordinato) o per riferimento (modificando il vettore originario).

Suggerimento: non dovremo definire una sola funzione ma due ovvero:

- una che si occupi del capire cosa ha scelto l'utente e reindirizzare verso l'ordinamento per riferimento oppure definire un vettore copia, ordinarlo e restituirlo

- una che ordini per riferimento un vettore mediante SelectionSort

Una volta fatto ciò irrobustire la funzione mediante il controllo dei tipi. È possibile effettuarlo sul tipo di ritorno delle varie funzioni?