

**Università degli Studi del Piemonte Orientale**  
**Dipartimento di Scienze ed Innovazione Tecnologica**  
**DiSIT**

**Esercizio 2: Find nerds**  
**(PHP, FORMS e FILES)**

**Testo dell'esercizio**

Per questo esercizio 2, vi verrà chiesto di creare un sito costituito da più pagina PHP allo scopo di trovare gli "accoppiamenti ideali" tra nerds. Dovrete utilizzare PHP per processare le informazioni che l'utente invierà tramite form HTML.

Il nome del sito è **NerdLuv**, ed il suo scopo è quello di consentire a nerd e geeks di incontrarsi in base ad alcuni parametri di affinità elettive. Il sito è costituito dalle seguenti pagine:

- **[signup.php](#)**  
Una pagina contenente un form che l'utente può compilare per richiedere la registrazione di un nuovo account
- **[signup-submit.php](#)**  
La pagina che riceve i dati inviati da [signup.php](#) e che effettivamente registra l'account del nuovo utente.
- **[matches.php](#)**  
Una pagina con un form per un utente già registrato che ritorna sul sito e vuole verificare chi sono le persone a lui/ lei affini.
- **[matches-submit.php](#)**  
La pagina che riceve i dati inviati da [matches.php](#) e che mostra chi sono le persone a lui/lei affini.

Sono forniti alcuni file di supporto sulla pagina DIR del corso. Il primo è la versione completa della home page, [index.php](#). Questa pagina contiene semplicemente dei link verso le altre pagine. Gli altri file che sono forniti nella loro forma definitiva sono [top.html](#) e [bottom.html](#), che contengono il codice HTML delle parti comuni a tutte le pagine, ovvero l'header ed il footer. Questi file sono da includere nelle altre pagine. Inoltre, viene fornito un file CSS ([nerdluv.css](#)) completo di tutte le regole di stile necessarie per il sito. Occorrerà inserire dall'interno di tutte le tue pagine un link a questo file CSS, per poi usare le sue regole all'interno del codice HTML. Non si ha bisogno di altro codice css se non quello contenuto dentro il file **nerdluv.css**.

## Pagina principale (index.php)

Il file index.php, mostra un logo nell'header, dei link ai file [signup.php](#) e [matches.php](#), e contiene delle note e delle immagini nel footer. Non è necessario modificare questo file, ma occorre limitarsi ad inserirlo nella stessa cartella insieme agli altri files contenuti all'interno della cartella DIR.

Il link "Sign up" consente l'apertura e l'esecuzione del relativo codice php della pagina [signup.php](#) (a sinistra in basso) mentre se si seleziona "Check matches" si aprirà la pagina [matches.php](#) (a destra in basso)

**nerdLuv<sup>tm</sup>**  
where meek geeks meet

New User Signup:

**Name:**

**Gender:** Male ☐ Female ☒

**Age:**

**Personality type:**  (Don't know your type?)

**Favorite OS:**

**Seeking age:**  to

**Favorite gender:**

**Sign Up**

This page is for single nerds to meet and date each other! Type in your personal information and wait for the nerdly luv to begin! Thank you for using our site.

Results and page (C) Copyright NerdLuv Inc.

[Back to front page](#)

**nerdLuv<sup>tm</sup>**  
where meek geeks meet

Returning User:

**Name:**

**View My Matches**

This page is for single nerds to meet and date each other! Type in your personal information and wait for the nerdly luv to begin! Thank you for using our site.

Results and page (C) Copyright NerdLuv Inc.

[Back to front page](#)



Quando inviate i dati con "View My Maches", i risultati saranno visualizzati come nella seguente figura

### Matches for Lara Croft



Anakin Skywalker

**gender:** M  
**age:** 27  
**type:** INTJ  
**OS:** Linux



Nostalgia Critic

**gender:** M  
**age:** 28  
**type:** ENTJ  
**OS:** Linux

## Pagina di registrazione (signup.php)



Welcome to NerdLuv, Elain Bons!

Now [log in to see your matches!](#)

This page is for single nerds to meet and date each other! Type in your personal information and wait for the nerdly luv to begin! Thank you for using our site.

Results and page (C) Copyright NerdLuv Inc.

 [Back to front page](#)



La pagina signup.php, incorpora l'header ed il footer come al solito, poi contiene un form per creare un nuovo account.

Il codice HTML per il form deve contenere i seguenti campi con etichetta (label):

- **Name**  
Un campo di testo di 16 caratteri per il nome e cognome dell'utente
- **Gender**  
I radio button che consentono all'utente di scegliere il genere (Male o Female). Quando l'utente seleziona il testo accanto al radio button, quel pulsante risulterà selezionato; inizialmente la selezione di default sarà Female.
- **Age**  
Un campo di testo lungo 6 caratteri per indicare l'età anagrafica. Il campo permetterà di inserire fino ad un massimo di 2 caratteri.
- **Personality type**  
Un campo di testo lungo 6 caratteri per il tipo di personalità (ad esempio ISTJ o ENFP). Si potranno inserire fino ad un massimo di 4 caratteri. L'hyperlink a destra contiene il link a [www.humanmetrics.com/cgi-win/JTypes2.asp](http://www.humanmetrics.com/cgi-win/JTypes2.asp)
- **Favorite OS**  
Un menù a tendina (select) che consentirà all'utente di specificare il proprio sistema operativo. Le scelte sono: {Windows, macOS, Linux}. Inizialmente "Linux" è selezionato.
- **Seeking age:** due campi di testo lunghi 6 lettere ciascuno per indicare l'intervallo di età preferito per il "partner". I campi permetteranno di inserire fino ad un massimo di due caratteri. Per default, i campi avranno dei placeholder "min" e "max" rispettivamente. Quando l'utente inizia a digitare, il placeholder sparirà.
- **Sign Up:** Quando il pulsante viene selezionato, il form invierà i dati ricevuti come descritto di seguito.

## Inviare ed elaborare i dati di registrazione (signup-submit.php)

Quando l'utente invierà i dati, i dati del form saranno elaborati. L'invio deve essere effettuato tramite un metodo POST alla pagina [signup-submit.php](#). (I nomi che corrisponderanno ai parametri che saranno passati a [signup-submit.php](#) potete sceglierli voi). Il vostro codice PHP dovrebbe leggere i dati passati attraverso i parametri della query e memorizzarli come descritto in basso. La pagina risultante avrà i soliti header e footer ed il testo che vedete a sx che ringrazia l'utente

Il testo **"log in to see your matches!"** contiene un link a [matches.php](#). I dati dell'utente saranno memorizzati in un file [singles.txt](#), che si troverà nella stessa cartella degli altri file. Su moodle troverete una versione iniziale di questo file. Esso contiene i dati dei singoli utenti registrati, ognuno corrispondente ad una riga, seguendo il seguente formato che specifica il nome completo dell'utente, il genere (M o F), l'età, il tipo di personalità, il sistema operativo, i valori min/max per l'età del partner cercato, tutti separati da virgola:

**Angry Video Game Nerd,M,29,ISTJ,Mac OS X,1,99**  
**Lara Croft,F,23,ENTP,Linux,18,30 Seven of**  
**Nine,F,40,ISTJ,Windows,12,50**

Il vostro codice dentro [signup-submit.php](#) dovrà inserire alla fine del file [singles.txt](#) una nuova riga che conterrà i dati dell'utente appena registrato. Potrebbe tornarvi utile la funzione PHP `file_put_contents` vista a lezione. In ogni pagina, assumete che vi arrivino dal form dei dati validi (la corretta validazione dell'input sarà oggetto di una lezione futura). Per esempio, nessun campo dovrà rimanere vuoto o contenere caratteri non validi (ad esempio la virgola). Altra cosa da assumere: nessun utente registrerà un nome già registrato e presente nel file [singles.txt](#).

## Cercare le persone più affini (matches.php)

La pagina [matches.php](#) contiene i soliti header e footer, oltre ad un form che consente il login. Dovete scrivere il codice HTML per il form che vedete a dx, che include un solo campo:

**Name:** Esso è costituito da un'etichetta ed un campo testo lungo 16 caratteri per consentire all'utente di digitare il proprio nome completo. Inizialmente sarà vuoto. L'invio al server avverrà tramite passaggio di parametri ed una query string creata a partire dal nome associato al campo testo.

Quando l'utente selezionerà il pulsante "View My Matches," il form invierà i dati

**nerdLuv<sup>tm</sup>**  
where meek geeks meet


Returning User:

Name:

**View My Matches**

This page is for single nerds to meet and date each other! Type in your personal information and wait for the nerdy luv to begin! Thank you for using our site.

Results and page (C) Copyright NerdLuv Inc.

 [Back to front page](#)



tramite una richiesta GET alla pagina [matches-submit.php](#).

Il nome del parametro dovrà semplicemente "name" ed il suo valore sarà il testo inserito dall'utente e codificato opportunamente per spedire questa informazione al server. Ad esempio, se l'utente cerca i partner ideali per Lara Croft, la URL generata dovrebbe terminare con **matches-submit.php?name=Lara+Croft**

#### Matches for Lara Croft



Anakin Skywalker	
gender:	M
age:	27
type:	INTJ
OS:	Linux



Nostalgia Critic	
gender:	M
age:	28
type:	ENTJ
OS:	Linux

## Mostrare la lista delle persone più affini ([matches-submit.php](#))

Quando vengono mostrati i partner ideali di un dato utente, [matches-submit.php](#) visualizzerà un'area centrale che contiene la lista dei vari risultati. Scrivete il codice PHP che legge il nome dal parametro ricevuto tramite query dalla pagina [matches.php](#) e, quindi, cerca tutti i partner "ideali" a quell'utente in particolare. Gli utenti registrati da confrontare sono contenuti tutti nel file [singles.txt](#) come precedentemente descritto. Potete assumere che il nome passato tramite il parametro corrisponderà ad un utente già contenuto nel file [singles.txt](#). Sotto il banner dell'header, ci sarà un titolo di intestazione "Matches for (name)". Sotto questo testo, sarà presentata la lista dei risultati che

corrispondono alle affinità ideali dell'utente.

Un "partner ideale" è una persona con le seguenti qualità

- Il **genere opposto** a quello dell'utente (NOTA: come parte opzionale del presente esercizio, rendete questo sito più politically correct, consentendo nella fase di registrazione di specificare non solo l'intervallo di età preferita, ma anche il genere del partner cercato, che potrà essere M, F o entrambi)
- **Un'età compatibile**; cioè, ogni persona che ha un'età che rientra nel range [min, max] definito dall'utente in fase di registrazione
- Deve avere lo **stesso sistema operativo** preferito dell'utente in oggetto (su questo punto non si discute, la scelta del OS è discriminatoria!)
- Condivide **almeno una lettera**, nella stessa posizione, della stringa che descrive il tipo di personalità. Per esempio, ISTP e ESFP hanno 2 due caratteristiche in comune (S e P).

Appena avete individuato i partner ideali per quell'utente, inviate in output il codice HTML che mostra i risultati, nello stesso ordine in cui sono stati trovati nel file. Ogni partner usa la stessa immagine [user.jpg](#), il nome della persona, ed una lista non ordinata (ul) che specifica il genere, l'età, il tipo di personalità ed il sistema operativo preferito. Usate il riferimento assoluto all'immagine [user.jpg](#) come indicato sotto:

<http://www.cs.washington.edu/education/courses/cse190m/12sp/homework/4/user.jpg>

## Stile

Gli stili di cui avete bisogno sono già stati definiti nel file [nerdluv.css](#), ma se avete ancora bisogno di definire delle regole per dei tag appositi che userete o per particolari attributi, allora potete modificare il file che vi abbiamo messo a disposizione su moodle. State attenti in particolare agli stili sui form: cercate il più di seguire le regole di usabilità che abbiamo introdotto a lezione. Cercate di ricalcare le immagini che vedete in questo testo per capire quali valori usare per width, colori, font, bordi, etc. Cercate di fare in modo che i vari campi e controlli nel siano tutti ben allineati tra di loro in colonne, così che ogni label abbia proprietà float: left e che abbiano dimensione del font 11em. Mettete prima un paragrafo che contiene l'immagine del "partner ideale" trovato, con un width di 150px. Il nome della persona deve stare a dx. Il paragrafo deve avere un colore di background "light blue". La sezione che conterrà il genere, l'età etc., deve essere rappresentata tramite una lista non ordinata (ul).

## Deploy e test

Caricate tutti i file in una cartella che chiamerete **esercizio2** e non dimenticate di caricare anche i file [top.html](#), [bottom.html](#), [index.php](#) e [nerdluv.css](#) anche se non li avete modificati. Ricordatevi che in locale, è possibile controllare il vostro sito usando l'URL

**`http://localhost/esercizio2/index.php`**

## Strategie di sviluppo e consigli...

Usate il file `index.php` come riferimento per scrivere `matches.php` e `matches-submit.php` in modo che funzioni correttamente con gli utenti già esistenti.

- Scrivete una versione iniziale che crea una lista che include ogni utente registrato, anche quelli che hanno profili "incompatibili". In questo modo, potete iniziare a fare debugging delle funzioni I/O, degli stili, etc. Quindi aggiungete i vincoli sul genere, l'età ed il sistema operativo. Infine concentrativi sul tipo di personalità. Prima preoccupatevi del codice PHP, poi quando tutto è okay, soffermatevi sullo stile e la presentazione (CSS non è il focus di questo esercizio).
- Dopo, scrivete `signup.php` e `signup-submit.php`. Dopo aver terminato le pagine `matches*.php` avrete capito come usare i form, rendendo la scrittura di queste altre pagine più semplice, anche se qui ci sono difficoltà aggiuntive: ci sono più campi nel form e dovrete scrivere in un file.

Usate le istruzioni di debug `print` e `print_r` per tenere traccia del vostro stato e quindi dei vostri bug. Per esempio, puoi usare `print_r($_GET)`; oppure `$_POST` per vedere i parametri inviati tramite query. Visualizzate anche il codice sorgente dal browser per meglio individuare errori nell'HTML generato da PHP.

Ricordate che i controlli in un form devono avere dei nomi. A volte conviene anche aggiungere l'attributo value per inviare correttamente i dati al server.

Potete usare la pagina:

<http://webster.cs.washington.edu/params.php>

come action temporanea per fare il test dei valori passati ai campi del form.

Valutate anche la configurazione di **XDebug** sul vostro IDE. Se non lo avete fatto la volta scorsa, localizzate i file di log secondo la vostra configurazione del server Apache e modulo PHP. In generale, seguite tutti i consigli forniti nel testo dell'esercizio precedente.

## Implementazione e valutazione

L'output HTML per tutte le pagine deve superare il test di validazione W3C (<https://validator.w3.org>). Ovviamente il testo non coinvolge il codice PHP, ma solo l'HTML da esso generato). Non usate tabelle HTML. Dato che stiamo usando form HTML, scegliete i controlli corretti ed i loro attributi di conseguenza. Scegliete GET e POST per spedire i dati al server quando richiesto nel testo dell'esercizio2. Il vostro codice PHP non deve generare warning o errori (possibilmente neanche notice - seguire i consigli delle slides viste durante il corso di laboratorio). Minimizzate l'uso di variabili globali (a volte sono necessarie, ma abituatevi ad usare funzioni parametrizzate), usate correttamente l'indentazione e la spaziatura, evitate linee più lunghe di cento caratteri che rendono il codice di difficile lettura.

Alcune sezioni HTML sono in comune nelle varie pagine, ad esempio quelle presenti nei file top.html e bottom.html. Includete questi file in modo appropriato nelle vostre usando la funzione PHP include.

La ridondanza rende il codice difficile da leggere e complicato da riutilizzare. Come visto a lezione, usate funzioni, parametri, return (nei punti corretti secondo una buona programmazione strutturata!!!), file esterni da includere, cicli, variabili, etc. per evitare ridondanza. Se avete del codice PHP (ad esempio una funzione che usate in diversi punti del sito) considerate la possibilità di inserirlo in un file common.php o functions.php che poi potrà essere incluso nelle altre pagine.

Per fare tutto alla perfezione, riducete la quantità di codice PHP embedded nel codice HTML. Sostituire queste porzioni di codice chiamando delle funzioni da voi definite, il cui codice può essere riportato in fondo alla vostra pagina o all'inizio (quindi dopo il tag di chiusura HTML o prima del doctype - se fate così dovete modificare ovviamente bottom.html o top.html in modo opportuno). Ricordatevi di limitare il più possibile le istruzioni PHP print ed echo: usate piuttosto le espressioni blocco `<?= ... ?>`, così come abbiamo visto a lezione durante il corso di laboratorio.

Un altro aspetto importante è legato all'uso dei commenti PHP. Ad esempio, all'inizio di ogni file, di ogni funzione, e di ogni blocco PHP, ricordatevi di mettere dei commenti descrittivi.

Cercate di strutturare il vostro codice in un modo simile agli esempi visti durante le lezioni frontali, facendo particolare riferimento ai diversi casi di studio che abbiamo analizzato al termine di ogni lezione. Usate spazi bianchi ed intestazioni in modo appropriato. Non mettete più di un elemento blocco per ogni linea della vostra pagina HTML.