

Lezioni PHP

Pianificazione delle videolezioni per argomenti; ogni lezione ha una durata di circa 15 minuti e deve includere uno o più esercizi per la durata complessiva di 40-45 minuti di lavoro.

Lezione 0

In questa lezione si dovrà introdurre il linguaggio PHP e l'ambiente su cui gli studenti lavoreranno. In particolare si tratteranno:

Prima parte: lo sviluppo lato server

- lo sviluppo web: richiami a HTML, CSS e JavaScript
- siti web dinamici: cosa sono e quanto sono diffusi
- i linguaggi lato server: React JS, Java, Python
- il linguaggio PHP
 - perchè studiarlo: PHP never dies (con meme), principali siti web scritti usando PHP, importanza didattica (è molto facile ma estremamente potente)
 - come funziona: logica degli script e possibilità di inserire il codice nelle pagine HTML per farlo caricare prima delle stesse

Seconda parte: il setup di lavoro

- breve presentazione dei server Apache2 e NGINX
- breve presentazione del software phpMyAdmin (peraltro scritto in PHP)
- breve presentazione del software Visual Studio Code
- come lavorare sui siti web e sulle basi di dati in laboratorio (setup con Ubuntu)

Nel video saranno anche presenti dei rimandi ad altri video circa il setup del server su Linux, Windows e macOS per poter lavorare anche da casa.

Lezione 1

In questa lezione si tratteranno alcuni aspetti del linguaggio PHP, in particolare si discuterà della sintassi e dei primi passi nell'uso del linguaggio

Teoria

- come costruire uno script PHP, apertura di Visual Studio Code nella cartella e creazione del primo file index.php
- la sintassi di base: tag `<?php ?>`, dichiarazione di variabili (scalari), operazioni aritmetiche di base e confronti mediante gli `if` `else if` `else`
- la funzione `echo` e la stampa di testo HTML nella pagina
- creazione di una pagina HTML con alcuni testi generati lato server: osservazione della natura *pre-interpretata* del linguaggio

Esercizi in aula

- variabili e operazioni aritmetiche:
 - realizzazione delle quattro operazioni date due variabili `$a` `$b`
 - realizzazione dei possibili incrementi delle due variabili sopra citate
 - date due stringhe `$str1` e `$str2` concatenarle
- confronti:
 - confrontare due variabili `$a` `$b` e stampare mediante la funzione `echo` il valore maggiore tra quelli contenuti, se il primo è divisibile per il secondo e viceversa, se il primo o il secondo (o entrambi) sono pari o dispari
 - confrontare due stringhe `$str1` e `$str2` e dire se sono uguali, diverse, vuote

Lezione 2

In questa lezione si affronteranno i cicli `while` e `do while` tipici del linguaggio C. In particolare la loro trattazione avverrà a partire da semplici incrementi fino a strutture più complesse (con annidazione).

Teoria

- logica dei cicli: ripetizione delle operazioni per un numero non noto di volte, logica precondizionale e postcondizionale
- il ciclo **while** con esempi e analisi di funzionamento
- il ciclo **do while** con esempi e analisi di funzionamento

Esercizi in aula

- stampare i primi 100 numeri positivi e negativi (da 0 a 100 e da 0 a -100)
- stampare i primi 50 numeri pari e dispari
- stampare i primi 20 multipli di 15 e di 27
- dalla pagina wikipedia dell'algoritmo di ordinamento Shell Sort ([link](#)) ricavare l'espressione della successione di Knuth e calcolarne i primi 20 termini (soluzione: l'n-esimo elemento è tre volte il precedente incrementato di uno)

Lezione 3

In questa lezione si affronteranno due elementi fondamentali della programmazione (in generale); i vettori (o *array*) e i cicli **for**, a questi andremo ad affiancare il **foreach** tipico di PHP e di altri linguaggi ad alto livello.

Teoria

- array: cosa sono e perché sono utili
- gli array “classici” e gli array associativi, dichiarazione in PHP e il loro scorrimento
- le funzioni:
 - `array_exist_key()`
 - `array_search()`
 - `array_keys()` e `array_values()`
 - `array_combine()`
- i cicli **for** visti per iterare matematicamente e per scorrere un array
- il ciclo **foreach** per scorrere un vettore “classico” e un vettore associativo

Esercizi in aula

- costruire un array secondo le seguenti specifiche mediante l'uso di un ciclo **for**:
 - ha dimensione $N = 25$
 - alla posizione 0 ha valore N
 - alla posizione i ha valore $N-i$
 - alla posizione $N-1$ ha valore 0
- dichiarare il seguente array associativo secondo le seguenti specifiche:
 - il nome è **\$province**
 - le chiavi sono delle stringhe di due caratteri maiuscoli che rappresentano la sigla della provincia
 - i valori sono le corrispondenti province espressi come stringhe formattate con la prima lettera maiuscola e la restante parte minuscola
 - la dimensione del vettore è 5
- iterare mediante un ciclo sul vettore **\$province** stampando una struttura HTML a piacere (tabella, elenco puntato, ...) che riporti la sigla della provincia e il nome della stessa
- dichiarare ora un altro vettore chiamato **\$province2** che a differenza del precedente vettore **\$province** non è formattato usando una struttura **\$chiave => \$valore** ma è semplicemente un vettore “classico” al cui interno sono presenti anche alcune province non inserite precedentemente (rispettando sempre la dimensione pari a 5); a questo punto iterare mediante un ciclo **foreach** sui valori di **\$province2** e, mediante la funzione `array_key_search()`, stabilire se ognuna delle province è presente anche nell'array **\$province2**; in tal caso stamparne la sigla, altrimenti stampare una dicitura di errore.

Lezione 4

In questa lezione si ultimerà la parte relativa alla pura programmazione in PHP; a seguito di questa le lezioni verteranno sulla scrittura di codice PHP all'interno di progetti che integrano HTML, CSS, JavaScript e MySQL. In questa lezione verranno trattate le funzioni e le istruzioni **require** e **require_once** per integrare il codice presente in più pagine.

Teoria

- cos'è una funzione e perché è utile
- come dichiarare e usare una funzione
- parametri e valori restituiti da una funzione
- le variabili globali
- l'istruzione `require` con esempi
- il problema del riuso e l'istruzione `require_once`
- la funzione `header()`

Esercizi in aula

- 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
- scrivere una funzione che, dato come parametro un numero, restituisca la somma dello stesso per un altro numero definito come variabile globale al di fuori della funzione
- scrivere una funzione che, dato un numero, verifichi se lo stesso è pari o dispari; restituire un booleano (`true` per pari o `false` per dispari)
- mediante l'istruzione `require` chiamare il file `richiesto.php` appositamente scritto per stampare una frase a piacere; una volta fatto ciò dichiarare dentro `richiesto.php` una variabile e verificare che la stessa sia accessibile dal file da cui è stato richiesto.

Lezione 5

In questa lezione si tratteranno i richiami alla programmazione *front-end*, in particolare verrà spiegato come integrare il codice PHP all'interno di una pagina HTML eventualmente comprensiva di codice CSS per la grafica e di parti JavaScript per la programmazione lato client.

Teoria

- come possiamo usare PHP oltre che come linguaggio di programmazione classico
- richiami di HTML, CSS e JavaScript (molto generali)
- integrare il codice PHP nella pagina HTML (logica della pre-interpretazione del codice)
- spiegazione iniziale dell'esercizio in aula (array annidati)

Esercizio in aula

*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`, ...)
- `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 è possibile seguire l'immagine presente nella cartella `lezione5/demo/` di questo repository.

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.

Lezione 6

In questa lezione si approfondirà il legame tra HTML e PHP, aggiungendo anche qualche elemento di JavaScript

Teoria

- il passaggio di dati mediante GET e POST e i relativi vettori
- passare dei parametri mediante GET e POST, l'integrazione con i *form* HTML
- richiami di JavaScript e uso di Swal2 per le notifiche
- alcuni elementi di programmazione a oggetti in PHP (sarà utile per la lezione successiva circa l'uso delle query SQL)
- suggerimenti per svolgere l'esercizio in aula

Esercizio in aula

Creare un sito web sfruttando HTML, CSS, JavaScript e PHP che contenga tre pagine:

- una pagina **index.php** (ovvero la *homepage* del sito) che mostri al suo interno due **form** HTML identici che richiedano i seguenti dati al visitatore:
 - Cognome
 - Nome
 - Data di nascita
 - Luogo di nascita
 - Indirizzo email
 - Numero di Telefono

Questi due **form** avranno come differenza grafica i bottoni di invio che saranno rispettivamente **GET** e **POST**

- una pagina chiamata *schedaGET.php* e una chiamata *schedaPOST.php* che riportino in maniera identica i dati dell'utente formattati in una tabella e due tasti **+** e **-** che rispettivamente aprano una notifica Swal2 di successo e di errore (possiamo intendere il loro significato come approvazione o disapprovazione del profilo come per una candidatura per un posto di lavoro)

Sebbene, a livello di front-end, le pagine siano identiche ciò che cambia è il back-end delle pagine; in particolare la prima pagina riceverà i dati come parametro GET mentre la seconda come parametro POST. Deduciamo perciò ora la necessità di avere due **form** separati nella pagina iniziale.

Si noti quanto segue nella progettazione del back-end:

- entrambe le pagine dovranno verificare che tutti i dati siano stati recapitati; per farlo un approccio può essere quello di usare un vettore di comodo contenente il nome dei campi richiesti e iterare sullo stesso mediante un **foreach** che controlli con la funzione **array_key_exists()** sui campi dei vettori **\$_GET** e **\$_POST**
- se dovessero mancare l'utente dovrà essere rimandato alla pagina iniziale (*feature extra*: sarebbe bello se l'utente ricevesse anche una notifica mediante Swal2 ma non è fondamentale, farlo solo se avanza tempo; in tal caso un approccio che si può seguire è quello di controllare la presenza di un parametro GET quale potrebbe essere **notify** fissato a un valore e, in tal caso, mediante **echo** stampare del codice JavaScript nella pagina)
- il fatto che non sia specificata la necessità di una pagina di appoggio contenente i dati usati da entrambe le pagine non implica che questa non possa essere utilizzata; non è comunque un errore scrivere tutto il codice nelle due pagine *schedaGET.php* e *schedaPOST.php*

Lezione 7

In questa lezione viene introdotto il legame tra PHP e basi di dati; in particolare si effettua un generale richiamo circa le basi di dati e su come effettuare le query da PHP.

Teoria

- richiami di basi di dati: cosa sono e a cosa servono
- perché è importante che le richieste vengano elaborate lato server
- alcuni strumenti per gestire le basi di dati, presentazione di *phpMyAdmin* e spiegazione di alcune sue funzioni fondamentali
- come si effettua una query (richiami di sintassi *MySQL*)
- la classe *mysqli* con particolare riferimento ai metodi e alle funzioni:
 - *num_rows*
 - *mysqli_fetch_row*
 - *query*
- consigli per risolvere l'esercizio in aula

Esercizio in aula

Costruire una base di dati utile a gestire i volumi contenuti in una biblioteca; in particolare la base di dati dovrà contenere una tabella contenente le seguenti informazioni relative ai libri:

- titolo
- autore
- ISBN
- data di restituzione

Dove la data di restituzione è un campo che può assumere due tipologie di valori: una data se il libro è attualmente in prestito e *NULL* se il libro è disponibile.

Una volta costruita tale base di dati costruire usando HTML, CSS e PHP una pagina web che mostri in una tabella i vari volumi riportando le informazioni sopra salvate e, nel caso in cui il volume sia in prestito, la data di restituzione.

Si noti che per verificare se la data di restituzione è un valore o è *NULL* è possibile utilizzare la funzione `is_null()` di PHP.

Per connettersi alla base di dati è possibile copiare all'interno della cartella *server* il file `database.php` presente nella cartella *lezione7* di questo repository; ovviamente tale file andrà richiesto dalla pagina PHP.

Lezione 8

In questa lezione verranno ultimate le informazioni relative alla costruzione di siti web PHP mediante l'uso delle sessioni (e delle relative variabili) e alcuni cenni relativi alle funzioni di hash.

Teoria

- concetto di sessione: cos'è e perché è utile
- le variabili di sessione, concetto generale
- avviare una sessione e concluderla in PHP
- le variabili di sessione in PHP, dichiarazione, uso e distruzione
- le funzioni di hash: presentazione della loro utilità e della loro logica di funzionamento
- la funzioni di hash **MD5** e **SHA256** e il loro uso nel codice PHP
- consigli per la risoluzione dell'esercizio in aula

Esercizio in aula

Costruire una base di dati che contenga una tabella in grado di salvare le seguenti informazioni degli utenti registrati:

- nome
- cognome
- indirizzo mail
- password (salvata mediante un hash che segue l'algoritmo **SHA256**)

Una volta creata tale base di dati costruire un sito web usando HTML, CSS e PHP in grado di mostrare all'utente un form di login (questo comprenderà dunque un campo per l'indirizzo mail e uno per la password) e una pagina denominata *areaRiservata.php* all'interno della quale possa vedere i propri dati; tale pagina mostrerà anche un tasto per disconnettersi e tornare alla pagina di login.

Si noti che le pagine da creare saranno in totale quattro:

- la prima sarà la pagina *index.php* contenente il form di login
- la seconda sarà una pagina senza parte grafica chiamata *nuovaSessione.php* che, ricevendo mediante POST i dati inseriti nella pagina *index.php* effettuerà una query di estrazione dei dati e reindirizzerà l'utente alla pagina *index.php* nel caso in cui il login fallisca oppure alla pagina *areaRiservata.php* nel caso in cui il login avvenga con successo
- la terza sarà la pagina *areaRiservata.php*
- la quarta sarà una pagina chiamata *distruggiSessione.php* che distruggerà la sessione e reindirizzerà alla pagina *index.php*

Ovviamente sarà necessario predisporre una connessione alla base di dati mediante *mysqli*; per farlo è possibile usare il file *database.php* presente nella cartella *lezione8* di questo repository richiedendolo dalla pagina *nuovaSessione.php*.

Possibili esercizi aggiuntivi

I seguenti esercizi sono da ritenersi di puro approfondimento personale; richiedono infatti molto più tempo e risorse per essere realizzati ma consentono, a chi interessa, una maggiore focalizzazione degli elementi trattati.

Per svolgerli è naturalmente possibile seguire per la programmazione PHP un approccio *C-like* oppure, per approfondire meglio la conoscenza generale della programmazione e nello specifico del linguaggio PHP, programmando a oggetti. Per aiutarsi con la programmazione a oggetti può essere utile recuperare la lezione 6 (nello specifico l'ultima parte in cui vengono posti alcuni elementi di OOP) e la lezione aggiuntiva 1 (che verte sui concetti base della programmazione a oggetti con la loro codifica in linguaggio PHP).

Biblioteca

Recuperare l'esercizio in aula della lezione 6 della biblioteca e creare un portale web realizzato con i linguaggi HTML, CSS, JavaScript, MySQL e PHP in grado di memorizzare i dati relativi ai volumi e quelli relativi ai bibliotecari.

Il portale dovrà dare la possibilità ai bibliotecari di gestire i prestiti inserendoli o eliminandoli e i volumi in maniera analoga.

I bibliotecari dovranno effettuare il login in un'apposita pagina e dovranno poter effettuare il logout. Come per l'esercizio in aula i dati circa i libri dovranno essere accessibili a tutti i visitatori del sito web in formato tabellare che includa l'eventuale prestito.

Registro elettronico

Creare un portale web mediante i linguaggi HTML, CSS, JavaScript e PHP in grado di gestire alcuni studenti con le relative valutazioni; le valutazioni, salvate in una tabella della base di dati dovranno essere modificabili, cancellabili e estensibili dai docenti. Gli studenti, effettuando il login, potranno vedere le proprie valutazioni.

Blog

Creare un portale web che mostri agli utenti collegati degli articoli pubblicati sul sito (e salvati in un'apposita base di dati). Alcuni utenti (autori) potranno pubblicare o cancellare articoli (non è per ora richiesta la modifica).

Gli articoli saranno mostrati mediante un'apposita pagina che riceve come parametro GET l'id univoco (chiave primaria) dell'articolo; dovrà essere disponibile una pagina che contenga l'indice degli articoli (generato dinamicamente mediante PHP).