



Domande e risposte
dell'interrogazione di laboratorio
di informatica 4E - 1 trimestre
(settembre - dicembre)

[Luca Belloni-4E](#)

WWW.LUCABELLONI.ALTERVISTA.ORG

Sommario

Cos'è il Php?.....	3
Cos'è un linguaggio Server Side?	3
Cos'è un Server Web?.....	3
Cos'è l'ambiente Altervista?	4
Cos'è Internet?	4
Cos'è una rete Lan?	4
Cos'è la posta elettronica?	5
Cos'è un blog?.....	5
Cos'è un linguaggio lato client?.....	6
Cos'è l'Html?	6
Descrivere i principali tag del linguaggio Html	7
ELEMENTI DI BASE	7
(tutti i documenti HTML devono contenere questi tag)	7
PROPRIETA' DELLA PAGINA: SFONDI E COLORI	8
FORMATTAZIONE DEL TESTO	8
COLLEGAMENTI (link)	8
IMMAGINI.....	8
DIVISORI.....	8
ELENCHI	8
TABELLE	8
Spiegare il modulo form	9
Differenza da \$Get e \$Post.....	9
Come si effettuano i controlli dei campi di un modulo form?	10
CONTROLLO CAMPI FORM IN PHP	10
CONTROLLO CAMPI FORM IN JAVASCRIPT.....	11
CONTROLLO CAMPI FORM CON JQUERY	11
Come si crea l'area riservata?.....	13
Cosa sono i vettori associativi in Php?.....	16
Cos'è una sessione?.....	16
Cosa sono i cookies?.....	17
Come si mantiene una sessione attiva?	17
Cos'è un file (in generale)?	18
Cos'è un archivio dati?.....	18
Quali sono i parametri per aprire un file in Php?	18
Cosa vuol dire aprire un file?.....	19

Qual è l'istruzione in Php per aprire il file?	19
Cosa vuol dire leggere i records di un file?.....	19
Qual è l'istruzione che permette di leggere i records?	19
Descrivere il ciclo Foreach	20
Spiegare la funzione Explode()	20
Cosa vuol dire scrivere (memorizzare) i dati in file di testo?	21
Qual è l'istruzione di scrittura in un file txt (file dati)?.....	21
Cosa vuol dire chiudere il file?.....	22
Qual è l'istruzione di chiusura del file?.....	22
Saper spiegare riga per riga il seguente codice	22

Cos'è il Php?

PHP (acronimo ricorsivo per *PHP: Hypertext Preprocessor*) è un linguaggio di scripting general-purpose open source molto utilizzato, è specialmente indicato per lo sviluppo web e può essere integrato nell'HTML.

Un linguaggio di scripting, in informatica, è un linguaggio di programmazione interpretato destinato in genere a compiti di automazione del sistema operativo o delle applicazioni, o a essere usato nella programmazione web all'interno delle pagine web ed è definito general-purpose in quanto il php è dotato di grande versatilità. Inoltre è definito open-source, cioè non protetto da copyright e quindi modificabile da tutti gli utenti.

Ciò che distingue PHP da altri linguaggi di scripting del tipo client-side JavaScript è che il codice viene eseguito nel server, generando HTML che sarà dopo inviato al client. Il client dovrebbe ricevere i risultati dell'esecuzione dello script, ma non potrà conoscere qual'è il codice eseguito.

Il linguaggio PHP è definito anche un linguaggio di scripting server side. PHP è un linguaggio di programmazione che permette di creare pagine dinamiche, tramite l'aiuto di database esterni o il passaggio dati fra le stesse pagine. Una delle caratteristiche più importanti e significative di PHP è infatti la possibilità di supportare una completa gamma di databases (MySQL, PostgreSQL, Sql Server, Oracle, SyBase, Access e altri).

Cos'è un linguaggio Server Side?

Nelle reti informatiche, l'espressione lato server (server-side in inglese) fa riferimento a operazioni compiute dal server in un ambito client-server contrapponendosi a tutto ciò che viene eseguito sul client (lato client).

Di solito un server è un programma software (ad es. un server web), che gira su una macchina remota (chiamata per estensione "server") rimanendo in ascolto su determinate porte e raggiungibile da un computer client. Alcune operazioni devono essere compiute dal lato server perché richiedono l'accesso a informazioni o funzionalità non disponibili sul client, o richiedono misure di sicurezza che sarebbero inaffidabili se eseguite lato client.

Nell'ambito della programmazione Web, si definiscono linguaggi lato server quei linguaggi di programmazione che vengono interpretati ed elaborati dal server il quale, successivamente, invia i risultati al client (il browser dell'utente). I linguaggi lato server più diffusi sono il PHP, l'ASP e jsp (java server page). Un programma scritto con questo tipo di linguaggi viene sempre elaborato sul server e mai reso disponibile all'utente, il quale può visualizzare solo il risultato o output del programma (tipicamente codice HTML).

Cos'è un Server Web?

In informatica un server web (o web server) è un'applicazione software che, in esecuzione su un server, è in grado di gestire le richieste di trasferimento di pagine web di un client, tipicamente un web browser.

La comunicazione tra server e client avviene tramite il protocollo HTTP, che utilizza la porta TCP 80 (o 8080), o eventualmente la versione sicura HTTPS, che utilizza invece la 443.

Un web server deve poter dialogare con una gran varietà di client (PC, smartphone, smart TV, tablet, ecc.) e per farlo quindi deve utilizzare dei protocolli riconosciuti da tutti. Quello principale per la navigazione web è l'HTTP (Hyper Text Transfer Protocol).

Ogni volta che scriviamo un indirizzo internet, il nostro browser la traduce in una richiesta http e la consegna al server indicato. Riceve poi da questo una risposta http e un file HTML che viene tradotto (si parla di rendering) sotto forma di testo e immagini per mostrarcelo così a video.

Cos'è l'ambiente Altervista?

Altervista è un servizio di hosting che ti permette di creare un sito o blog in modo gratuito, ma funziona anche da database. Il database di AV è MySQL. Il compito principale del database è quello di salvare i dati inserite da un'utente.

Cos'è Internet?

Le reti geografiche nascono negli anni Sessanta del Novecento, finanziate dal Dipartimento della difesa americano per collegare le varie basi militari. Passato il pericolo di un'invasione diretta, il progetto è passato alle università che hanno realizzato una rete in grado di collegare i vari istituti. Questa prima rete(Arpanet), nata senza fini di lucro, è cresciuta nel tempo collegando siti anche in altre parti del mondo fino a diventare una rete mondiale di reti (Internet).

Internet può essere definita come un sistema di reti di computer interconnessi che facilità l'utilizzo di servizi di comunicazione dati come la posta elettronica, il Word Wide Web (WWW), il trasferimento di file e i gruppi di discussione.

Dopo aver realizzato l'accesso a internet, è possibile accedere alle risorse di digitando l'url (Uniform ResourceLocator). Si tratta dell'home associato a un server che si trova sulla rete internet e usa il protocollo http per inviare le risposte al client. Due host sulla rete internet utilizzare il protocollo TCP/IP e comunicano attraverso gli indirizzi IP. Ogni server esistente su internet a un indirizzo IP.

Il protocollo TCP (Trasmission control protocoll) suddivide i dati in più pacchetti che possono essere gestiti dalla rete in modo efficiente. Invece IP (internet protocol) si occupa di impacchettare e indirizzare i dati definendo quanti ne possono essere inseriti in un solo pacchetto. L' IP è ciò che identifica i diversi nodi sulla rete.

Conoscere tutti gli indirizzi IP di internet è troppo complesso e per gli uomini è complicato ricordare una serie di numeri, mentre è più semplice ricordare i nomi. Per questo motivo gli indirizzi Web sono pensati come dei nomi e sono chiamati nomi di dominio. Per esempio all'indirizzo Google.it è associato l'indirizzo ip 64.233.167.99.

Cos'è una rete Lan?

In informatica e telecomunicazioni una Local Area Network (LAN) (in italiano rete in area locale, o rete locale) è una rete informatica di collegamento tra più computer, estendibile anche a dispositivi periferici condivisi, che copre un'area limitata, come un'abitazione, una scuola, un'azienda o un complesso di edifici adiacenti.

Le principali proprietà e caratteristiche delle reti locali sono:

- elevate velocità. Infatti le reti LAN più diffuse oggi operano a velocità da 10Mbit/sec a 1 Gbit/s;

- basse probabilità di errore.Le reti LAN, a causa delle estensioni ridotte, possono consentire di raggiungere velocità di trasmissione molto basse, per cui non sono generalmente utilizzate tecniche di controllo degli errori;
- elevata affidabilità.Le reti locali, se opportunamente progettate, possono continuare ad operare anche in presenza di guasti o malfunzionamenti;
- espansibilità.Le reti locali possono essere progettate in modo da crescere nel tempo secondo le esigenze dell'utente senza significativi cambiamenti nella rete;
- basso costo.Le reti locali hanno ormai raggiunto una elevata diffusione in tutti gli ambienti e presentano per questo motivo un costo complessivo abbastanza modesto.

Cos'è la posta elettronica?

La posta elettronica (e-mail o email, dall'inglese electronic mail) è un servizio Internet grazie al quale ogni utente abilitato può inviare e ricevere dei messaggi utilizzando un computer o altro dispositivo elettronico (es. palmare, smartphone, tablet, ecc.) connesso in rete attraverso un proprio account di posta registrato presso un provider del servizio. È uno dei servizi Internet più conosciuti e utilizzati assieme alla navigazione web.

Il servizio di posta elettronica, chiamato anche "e-mail" (electronic mail) consente a ogni utente che abbia accesso ad un computer che possa connettersi ad Internet di inviare "messaggi" (testi ma anche, più in generale, "oggetti" memorizzati in formato elettronico, sotto forma di file, come programmi, immagini, suoni, etc.) ad un qualsiasi altro utente che disponga di un indirizzo di posta elettronica e che lavori su un qualsiasi altro computer, ovunque collocato, purché raggiungibile tramite connessioni in rete.

Non è necessario che i computer dei due corrispondenti siano contemporaneamente o permanentemente connessi alla rete perché lo scambio di informazioni sia possibile: i messaggi vengono recapitati su caselle di posta elettronica ospitate da appositi server; l'utente che voglia verificare l'arrivo di messaggi a lui indirizzati potrà contattare il server e solamente nell'intervallo in cui avviene la transazione tra il computer dell'utente ed il server è necessario che la connessione di rete sia attiva.

Cos'è un blog?

Nel Web 2.0 e nel gergo di Internet, un blog è un particolare tipo di sito web in cui i contenuti vengono visualizzati in forma anti-cronologica (dal più recente al più lontano nel tempo). In genere il blog è gestito da uno o più blogger, o bloggista, che pubblicano, più o meno periodicamente, contenuti multimediali, in forma testuale o in forma di post, concetto assimilabile o avvicinabile ad un articolo di giornale.

Classificare i blog non è un'operazione semplicissima. Si parla di social blog, blog collettivo, blog aziendale, blog tematico, photoblog, blog vetrina, blog politico, m-blog, video blog, blog letterario, blog didattico, audioblog, blog personale, blog novel, splog (finti blog riempiti di testi riprodotti solo per dare la sensazione che un determinato argomento sia particolarmente rilevante). Una suddivisione più semplificata (e attendibile) si basa su tre categorie:

- blog diario

- blog tematico (inclusi i blog giornalistici)
- blog letterario

I blog diario costituiscono il 59% del totale, i blog tematici il 31% mentre i blog letterari il 10%. Questi dati percentuali sono risultati dall'analisi fatta su un campione di 100 blog presi in esame nell'aprile del 2007 e ricercati casualmente su Google Blog Search.

Nel blog personale (diario) l'autore scrive le proprie esperienze giornaliere, racconti, desideri. I commenti da parte dei lettori sono sempre molto apprezzati.

I blog tematici (o giornalistici) sono i blog dedicati ad un argomento specifico e per questo sono accostabili agli articoli di giornale. Particolarmente in voga i blog monotematici su makeup (trucco), cinema, cucina e viaggi, che hanno reso molto famose persone comuni.

Genericamente i post di tali blog sono caratterizzati dalla presenza, al loro interno, del lessico letterario che si manifesta in quantità maggiori rispetto ai quotidiani. Da una ricerca di Tavosanis (2006) basata sull'analisi di 72 verbi identificati come "letterari" oppure "obsoleti" nel vocabolario De Mauro (2000), infatti, è risultato che nei blog la presenza di tali verbi è di circa quattro volte superiore rispetto al sito de "la Repubblica". (Non tutte le 72 forme all'infinito dei verbi sono state ritrovate, ma solo 25). In questa categoria possono rientrare anche le fan fiction ossia blog dedicati ai romanzi, ai cartoni animati, ai telefilm e ai film.

Cos'è un linguaggio lato client?

In informatica, nell'ambito delle reti di calcolatori, il termine lato client (client-side in inglese) indica le operazioni di elaborazione effettuate da un client in un'architettura client-server. Rappresenta dunque il front-end di un sistema informatico e di un'applicazione web con architettura multi-tier.

Un rapporto tipico di questo tipo è quello effettuato da un'applicazione, come un web browser, che avvia una connessione ad un server per poter funzionare. Le operazioni effettuate in modo client-side di solito vengono effettuate in questo modo poiché richiedono risorse che non possono essere prese sul server, ma che lo sono sul client, perché l'utente deve poter inserire dati da cui avere risposte od osservare il funzionamento dello script, oppure perché al server manca la potenza di elaborazione per poter far funzionare lo script su tutti i client.

Un'operazione tipica client-side ad esempio è quella del caricamento di un'interfaccia grafica utente, ad esempio il front-end di un'applicazione web o di un comune sito web: il browser client riceve i dati HTML e JavaScript dal server ed il motore di rendering li elabora visualizzandoli sul monitor dell'utente. Sotto quest'ottica la programmazione web lato client è una parte dell'intera programmazione web.

Cos'è l'Html?

L'HTML fu inventato nel 1990 da uno scienziato chiamato Tim Berners-Lee. Lo scopo finale era quello di semplificare l'accesso alle pubblicazioni scientifiche tra scienziati di diverse università. Il progetto riscosse un successo molto più grande di quanto avesse mai

immaginato lo stesso Tim Berners-Lee. Con l'invenzione dell'HTML stese le fondamenta per il web come lo conosciamo noi oggi.

In informatica l'HyperText Markup Language (HTML; traduzione letterale: linguaggio a marcatori per ipertesti) è un linguaggio di markup, ovvero un insieme di regole che descrivono i meccanismi di rappresentazione (strutturali, semantici o presentazionali) di un testo. Nato per la formattazione e impaginazione di documenti ipertestuali disponibili nel web 1.0, oggi è utilizzato principalmente per il disaccoppiamento della struttura logica di una pagina web (definita appunto dal markup) e la sua rappresentazione, gestita tramite gli stili CSS per adattarsi alle nuove esigenze di comunicazione e pubblicazione all'interno di Internet.

HTML è l'abbreviazione di "HyperText Mark-up Language" ed è proprio il suo nome a spiegare il suo funzionamento:

- **Hyper** è l'opposto di lineare. Tempi addietro - quando il topo era qualcosa a cui il gatto dava la caccia - i programmi per computer giravano linearmente: quando il programma aveva eseguito un'azione passava alla linea successiva, quindi alla linea successiva e così di seguito. Ma l'HTML è diverso - tu puoi andare in qualsiasi posto in qualsiasi momento. Per esempio, non è necessario visitare MSN.com prima di visitare HTML.net.
- **Text** è auto-esplicativo.
- **Mark-up** è quello che fai con il testo. Tu marchi il testo così come fai con un programma di editing quando usi le intestazioni, i punti elenco, il grassetto per il testo, e così via.
- **Language** è quello che è l'HTML, che utilizza molte parole inglesi.

Descrivere i principali tag del linguaggio Html

I tag sono le istruzioni dell'html che compongono la struttura di una pagina, e presentano alcune caratteristiche fondamentali che è bene conoscere sin dall'inizio:

- Un tag è un comando racchiuso tra i segni <(minore) e >(maggiore), chiamati anche parentesi angolari; per esempio: , , <HEAD>, <BODY>ecc. Quando il browser incontra questi simboli in un documento, "capisce" che si tratta di istruzioni da eseguire e non di testo da visualizzare sullo schermo.
- La maggior parte dei tag (ma non tutti) funziona "in coppia", secondo questo schema: <TAG_DI_INIZIO>testo o elemento a cui si applica l'istruzione </TAG_DI_FINE>
- I tag, in generale, possono contenere attributi che ne definiscono determinate proprietà. Ogni attributo può assumere differenti valori.

Ci sono un'infinita di tag HTML, ma andiamo a descrivere i principali.

ELEMENTI DI BASE

(tutti i documenti HTML devono contenere questi tag)

Tipo documento	<HTML></HTML>	(segnano l'inizio e la fine del file)
Titolo	<TITLE></TITLE>	(titolo della pagina, da inserire all'interno della testata)
Testata	<HEAD></HEAD>	(informazioni descrittive; come il titolo)
Corpo	<BODY></BODY>	(contenuto della pagina)

PROPRIETA' DELLA PAGINA: SFONDI E COLORI

Immagine di sfondo	<BODY BACKGROUND="URL">
Colore di sfondo	<BODY BGCOLOR="#*****">
Colore del testo	<BODY TEXT="#*****">
Colore dei collegamenti	<BODY LINK="#*****">
Colore dei collegamenti visitati	<BODY VLINK="#*****">
Colore del collegamento selezionato	<BODY ALINK="#*****">

FORMATTAZIONE DEL TESTO

Neretto		
Corsivo	<I></I>	
Sottolineato	<U></U>	
Apice		
Pendice		
Centrato	<CENTER></CENTER>	(sia per testo che immagini)
Font: dimensioni		(valori da 1 a 7)
Font: modifica misura		(valori da -3 a +3)
Font: colore		
Font: tipo		

COLLEGAMENTI (link)

Link a un documento	
Link a indirizzo e-mail	

IMMAGINI

Visualizzazione immagine	
Immagine: allineamento	
Immagine: testo in alternativa	
Immagine: dimensioni (pixel)	
Immagine: dimensioni (% d. largh. e alt. d. pagina)	
Immagine: bordo (pixel)	

DIVISORI

Paragrafo	<P></P>
Interruzione riga (Singolo ritorno a capo)	
Riga orizzontale	<HR>

ELENCHI

Elenco puntato		(prima di ogni elemento)
----------------	---------------	-------------------------------

TABELLE

Inizio e fine tabella	<TABLE></TABLE>
Tabella: allineamento	<TABLE ALIGN=LEFT RIGHT CENTER>
Tabella: spessore bordo	<TABLE BORDER=?></TABLE>
Larghezza tabella (in pixel)	<TABLE WIDTH=?>
Larghezza % tabella	<TABLE WIDTH=?%>
Tabella: colore di sfondo	<TABLE BGCOLOR="#*****"></TABLE>
Riga	<TR></TR>
Cella	<TD></TD>

Spiegare il modulo form

Un form HTML è una finestra contenente vari elementi di controllo che consentono al visitatore di inserire informazioni. Una volta inseriti, i dati vengono inviati ad uno script che li elabora. Sintassi:

```
<form action="<action="[URL dello script][script]" method="method" method="[GET o POST]"="[POST]">>
<input type="<type="[elemento di controllo][controllo]" name="nome" name="[nome]"="[nome]" ...>"...>
<input type="<type="[elemento di controllo][controllo]" name="nome" name="[nome]"="[nome]" ...>" ...>
...>
.....
</form>
```

Immaginiamo di avere la seguente form HTML:

```
<FORM action="http://localhost/password.php" method="post">
<INPUT type="TEXT" name="utente">
<INPUT type="SUBMIT" name="Invia" value="Spedisci">
</FORM>
```

L'attributo action serve per specificare l'URL dello script.server.

L'attributo method serve per specificare la modalità di invio delle serve informazioni. Può essere GET o o POST. Con GET le informazioni vengono inserite nel "indirizzo URL, dunque sono visibili nella barra degli indirizzi del browser, ma sono vincolate dalla lunghezza

massima di un URL, che è di 256 caratteri. Con POST i dati vengono scritti sullo "standard input" del l'applicazione destinataria, dunque non sono visibili ma soprattutto non ci sono limiti sulla quantità di dati inviata. Si può accedere alle informazioni inviate anche utilizzando gli array super globali \$_GET e \$_POST (a seconda del metodo usato).

Differenza da \$Get e \$Post

Con il metodo GET i dati vengono passati direttamente all'interno dell'indirizzo web (URL) della pagina, il quale si presenterà accompagnato da un punto di domanda (?) seguito dai dati organizzati in coppie nome/valore (qualora vi siano diverse coppie queste saranno legate tra loro dal simbolo &).

Il metodo POST, invece, è quello più utilizzato per inviare i dati ad una applicazione PHP tramite i form (moduli html).

Diversamente dal metodo GET, il metodo POST spedisce i dati in maniera non direttamente visibile per l'utente, attraverso la richiesta HTTP che il browser invia al server.

Come si effettuano i controlli dei campi di un modulo form?

I controlli permettono di segnalare all'utente che compila il modulo che i dati inseriti sono errati o mancanti e per questo viene richiesto di inserire i dati. I controlli di un campo form possono essere effettuati tramite php, javascript o JQuery.

CONTROLLO CAMPI FORM IN PHP

```
<?
// controllo campi compilati
if ( ( $nome == "" ) || ( $età == "" ) || ( $sesso == "" ) || ( $email == "" ) ) {
echo "<h3>Errore!</h3><br>Deve compilare tutti i campi dal form per continuare<br>
La preghiamo di <A HREF='home.php'>tornare indietro e correggere</A>. Grazie";
exit;
}
```

```
// controllo campo età compilato con numeri
if (!is_numeric($telefono)) {
echo "<h3>Errore!</h3><br>Deve inserire la sua età<br>
La preghiamo di <A HREF='home.php'>tornare indietro e correggere</A>. Grazie";
exit;
}
```

```
// controllo campo email - email valida
if (!ereg("@",$email)){
echo "<h3>Errore!</h3><br>Deve inserire una email valida<br>
La preghiamo di <A HREF='home.php'>tornare indietro e correggere</A>. Grazie";
exit;
}
```

```
// inizio elaborazione dei dati
?>
```

Analizziamo il codice php che effettua il controllo:

```
if ( ( $nome == "" ) || ( $età == "" ) || ( $sesso == "" ) || ( $email == "" ) )
```

Controlla che tutti i campi o meglio le variabili ad essi associati contengano almeno un valore cioè che l'utente abbia compilato tutti i campi come volevamo.

```
if (!is_numeric($telefono))
```

Questo codice controlla che la variabile \$telefono contenga come valore solo numeri o stringi di numeri.

```
if (!ereg("@",$email))
```

Con questo if controllerete che nella variabile \$email ci sia il carattere "@" ovvero controlla la presenza della "@" nel valore della variabile \$email

CONTROLLO CAMPI FORM IN JAVASCRIPT

```
<script language="JavaScript">
function controllo(){
with(document.modulo) {
if(nome.value=="") {
alert("Errore: compilare il campo NOME");
nome.focus();
return false;
}
if(cognome.value=="") {
alert("Errore: compilare il campo COGNOME");
cognome.focus();
return false;
}
}
alert("Controllo effettuato con successo. Il modulo sarà inviato.");
return true;
}
</script>
```

CONTROLLO CAMPI FORM CON JQUERY

Codice html:

```
<html>
<head>
<title>Esempio di validazione con JQuery</title>
<!--Ultima versione di jQuery (minified) -->
<script src="https://code.jquery.com/jquery-2.1.4.min.js"></script>
<!-- Ultima versione di jquery.validate (minified) -->
<script src="http://ajax.aspnetcdn.com/ajax/jquery.validate/1.14.0/jquery.validate.min.js"></script>
<!-- Ultima versione di bootstrap (minified) -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
<!-- codice jQuery -->
<script type="text/javascript" src="app.js"></script>
</head>
<body>
<!-- Form container -->
<div class="main">
<h1>Registrazione</h1>
<!-- form da validare -->
<form action="#" id="form">
<div class="form-group">
<label for="login">Login</label>
<input type="text" name="login" class="form-control">
</div>
<div class="form-group">
<label for="email">Email</label>
<input type="text" name="email" class="form-control">
</div>
```

```
</div>
<div class="form-group">
  <label for="password">Password</label>
  <input type="password" name="password" class="form-control">
</div>
<div class="form-group">
  <input type="submit" value="Registrati" class="submit" class="form-control">
</div>
</form>
</div>
</body>
</html>
```

Grazie alle seguenti stringhe di codice inseriamo nell'html le librerie jquery:

```
<script src="https://code.jquery.com/jquery-2.1.4.min.js"></script>
<script src="http://ajax.aspnetcdn.com/ajax/jquery.validate/1.14.0/jquery.validate.min.js"></script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
```

Attraverso la seguente stringa importiamo il file *app.js* dove vengono definite le regole di validazione per il controllo del form:

```
<script type="text/javascript" src="app.js"></script>
```

Codice JavaScript:

```
$(document).ready(function() {
  // Selezione form e definizione dei metodi di validazione
  $("#form").validate({
    // Definiamo le nostre regole di validazione
    rules : {
      // login - nome del campo di input da validare
      login : {
        // Definiamo il campo login come obbligatorio
        required : true
      },
      email : {
        required : true,
        // Definiamo il campo email come un campo di tipo email
        email : true
      },
      password : {
        required : true,
        // Settiamo la lunghezza minima e massima per il campo password
        minlength : 5,
        maxlength : 8
      }
    },
    // Personalizziamo i messaggi di errore
    messages: {
      login: "Inserisci la login",
      password: {
        required: "Inserisci una password password",
        minlength: "La password deve essere lunga minimo 5 caratteri",
      }
    }
  });
});
```

```
        maxlength: "La password deve essere lunga al massimo 8 caratteri"
    },
    email: "Inserisci la tua email"
},
// Settiamo il submit handler per la form
submitHandler: function(form) {
    form.submit();
}
});
});
```

Validatori predefiniti

Questo plug-in di validazione ci mette a disposizione vari metodi di validazione utili nei i casi più comuni. Di seguito alcuni di questi:

- **required** – Rende un campo obbligatorio.
- **remote** – Richiede una risorsa remota per validare un campo.
- **minlength** – Definisce la lunghezza minima per un campo.
- **maxlength** – Definisce la lunghezza massima per un campo.
- **rangelength** – Definisce un range valido di lunghezza per un campo.
- **email** – Definisce che un campo deve essere in formato email
- **url** – Definisce che un campo deve essere un URL valido
- **date** – Definisce che un campo deve essere una data valida.
- **dateISO** – Definisce che un campo deve essere un data valida nel formato ISO.
- **number** – Definisce che un campo deve essere un numero decimale
- **digits** – Definisce che un campo deve contenere solo numeri
- **creditcard** – Definisce un campo come un campo in formato numero carta di credito
- **equalTo** – Definisce che un campo deve essere uguale ad un'altro campo.

Come si crea l'area riservata?

La pagina login.php viene utilizzata per inserire le credenziali di accesso all'area riservata. Nel form action della pagina login.php form method="post" action="login2.php" se inserisco login2.php, soltanto un utente potrà collegarsi se invece nell'action inserisco login3.php (form method="post" action="login3.php") allora permetterò a tre persone di collegarsi alla pagina riservata.

Login.php

```
<html>
<head>
<title>Login Area</title>
</head><body>
<?php
if(isset($_GET['error_login']))
{
echo "Error ,please compile correctly the form!";
}
?>
```

```
<form method="post" action="login3.php">
<table>
<tr><td>Username</td></tr>
<tr><td><input type="text" name="user" id="user" /></td></tr>
<tr><td>Password</td></tr>
<tr><td><input type="text" name="pass" id="pass"/></td></tr>
<tr><td><input type="submit" value="Log In" /></td></tr>
</table>
</form>
</body>
</html>
```

Login2.php

(solo un utente può entrare nell'area riservata)

```
<?php
session_start();
$user="ciccio";
$pass="12345";
$page_to="page.php";
if(isset($_POST['user']) && isset($_POST['pass']))
{
if( $_POST['user']==$user && $_POST['pass']==$pass)
{
$_SESSION['isLogged']="true";
header("Location:".$page_to);
}
else
{
header("Location:login.php?error_login=1");
}
}
else
{
header("Location:login.php?error_login=1");
}
?>
```

page3.php

(tre utenti possono loggarsi)

la pagina di destinazione è: \$page_to="page.php";

```
<?php
session_start();
$utenti[0]["user"]="user1";
$utenti[0]["pass"]="pass1";
$utenti[1]["user"]="user2";
$utenti[1]["pass"]="pass2";
$utenti[2]["user"]="user3";
$utenti[2]["pass"]="pass3";
$page_to="page.php";
```

```
$isLogged=false;
if(isset($_POST['user']) && isset($_POST['pass']))
{
for($i=0;$i<count($utenti);$i++) {
if( $_POST['user']==$utenti[$i]['user'] && $_POST['pass']==$utenti[$i]['pass'] )
{
$isLogged=true;
}
}
}
if($isLogged)
{
$_SESSION['isLogged']="true";
header("Location:".$page_to);
}
else
{
header("Location:login.php?error_login=1");
}
}
else {
header("Location:login.php?error_login=1");
}
}
?>
```

Page.php

```
<?php
session_start();
if(!isset($_SESSION['isLogged']))
{
header("Location:login.php?error_login=1");
}
else {
if($_SESSION['isLogged']!="true")
{
header("Location:login.php?error_login=1");
}
}
?>
<html>
<head>
</head>
<body>
<div>
Utente Loggato!
</div>
</body>
</html>
```

Queste pagine usano le sessioni,dopo aver essersi loggato in ogni pagina php deve esserci questo script (è presente nella pagina page.php)

```
<?php
```

```
session_start();
if(!isset($_SESSION['isLogged']))
{
header("Location:login.php?error_login=1");
}
else {
if($_SESSION['isLogged']!="true")
{
header("Location:login.php?error_login=1");
}
}
?>
```

Cosa sono i vettori associativi in Php?

In informatica un array o vettore è una struttura dati complessa, statica e omogenea, usata in molti linguaggi di programmazione e chiaramente ispirata alla nozione matematica di vettore, (o di matrice, nel caso di array bidimensionali). Ciascuna delle celle dell'array è identificata da un valore di indice. L'indice è generalmente numerico e i valori che gli indici possono assumere sono numeri interi contigui che partono da 0 o da 1.

L'array associativo è un array i cui elementi sono accessibili mediante nomi, quindi stringhe anziché indici puramente numerici. Questo non comporta però l'obbligo di utilizzare solo un tipo di indice: alcuni elementi dell'array possono avere un indice numerico, altri un indice di tipo stringa.

Cos'è una sessione?

La sessione è una funzione che ti permette di gestire il contenuto protetto tramite una stringa di caratteri chiamata cookie viene memorizzata nell'aria temporanea del browser, di conseguenza ti permette di entrare in un'area riservata.

SESSION START= ATTIVARE SESSIONE (login)
SESSION DESTROY= CHIUDERE SESSIONE (logout)

A differenza dei cookie le sessioni di PHP non scrivono nulla sul computer del utente, ma operano (quasi esclusivamente) sul server scrivendo, all'interno di un'apposita cartella, degli specifici files dove vengono salvati alcuni dati importanti relativamente alla sessione di navigazione dell'utente.

Il ciclo di sessione comprende tre fasi:

Avvio o apertura: Si avvia uno scambio di informazioni tra l'utente (od anche il client) che intende avvalersi di un servizio e il server che sovrintende tale servizio. La procedura più nota e comune è quella di login. Il server prepara e memorizza un pacchetto di informazioni

permanenti che diventeranno i parametri di sessione e, per l'utente, i suoi dati di lavoro per tutta la sessione (solitamente i dati preparati dal server vengono inseriti nei cookie, che a loro volta vengono scaricati dal web Browser e sfruttati per mantenere la sessione del servizio attivo, senza dover ogni volta reinserire i dati manualmente.)

Lavoro in sessione: Il colloquio prosegue, sempre con lo scambio delle informazioni di sessione. Le tipiche applicazioni per il web prevedono spesso che nell'arco della sessione parte di queste informazioni possano essere variate o aggiunte di nuove.

Chiusura: Su richiesta dell'utente o del client il server cancella le informazioni di sessione. In assenza di una specifica richiesta, nella gran parte delle applicazioni è prevista la chiusura o fine automatica di sessione, dopo un certo tempo in cui l'utente/client non invia alcun messaggio.

Cosa sono i cookies?

I cookie HTTP (più comunemente denominati cookie web, o per antonomasia cookie) sono un tipo particolare di magic cookie, una sorta di gettone identificativo, usato dai server web per poter riconoscere i browser durante comunicazioni con il protocollo HTTP usato per la navigazione web.

Tale riconoscimento permette di realizzare meccanismi di autenticazione, usati ad esempio per i login; di memorizzare dati utili alla sessione di navigazione, come le preferenze sull'aspetto grafico o linguistico del sito; di associare dati memorizzati dal server, ad esempio il contenuto del carrello di un negozio elettronico; di tracciare la navigazione dell'utente, ad esempio per fini statistici o pubblicitari.

Date le implicazioni per la riservatezza dei naviganti del web, l'uso dei cookie è categorizzato e disciplinato negli ordinamenti giuridici di numerosi paesi, tra cui quelli europei, inclusa l'Italia. La sicurezza di un cookie di autenticazione dipende generalmente dalla sicurezza del sito che lo emette, dal browser web dell'utente, e dipende dal fatto che il cookie sia criptato o meno. Le vulnerabilità di sicurezza possono permettere agli hacker di leggere i dati del cookie, che potrebbe essere usato per ottenere l'accesso ai dati degli utenti, o per ottenere l'accesso (con le credenziali dell'utente) al sito web a cui il cookie appartiene.

I cookie, e in particolare i cookie di terza parte, sono comunemente usati per memorizzare le ricerche di navigazione degli utenti; questi dati sensibili, possono essere una potenziale minaccia alla privacy degli utenti; proprio questo ha indotto le autorità europee e degli Stati Uniti a regolamentarne l'uso mediante una legge nel 2011. Infatti la legislazione europea impone a tutti i siti degli stati membri, di informare gli utenti che il sito utilizza certe tipologie di cookie.

Come si mantiene una sessione attiva?

Normalmente effettuando il logout da una sessione o chiudendo il browser essa si chiude, Se si vuole mantenere aperta la sessione bisogna che il sito in questo metta a disposizione dell'utente un cookie che prolunghi la durata di stazionamento ed anche se chiude il browser la sessione rimane attiva.

Esempio per far rimanere una sessione attiva per 7 giorni:

```
setcookie("nome","valore",time()+60*60*24*7);
```

Cos'è un file (in generale)?

Il termine inglese, in informatica, viene utilizzato per riferirsi a un contenitore di informazioni/dati in formato digitale, tipicamente presenti su un supporto digitale di memorizzazione opportunamente formattato in un determinato file system.

Le informazioni scritte/codificate al suo interno sono leggibili solo tramite uno specifico software in grado di effettuare l'operazione.

Tecnicamente, i dati di un file sono codificati/organizzati in una sequenza di byte in linea con il linguaggio binario usato dal computer, immagazzinati come un solo elemento su una memoria di massa attraverso il file system (sistema di archiviazione) esistente su quella data memoria. Il sistema, quando riceve una richiesta dall'utente (esempio: aprire un file), provvede poi all'operazione inversa di decodifica e visualizzazione, facendo apparire sul video parole e numeri.

Cos'è un archivio dati?

Un archivio è: un insieme organizzato di dati che possono essere utilizzati da utenti diversi con programmi diversi.

Le caratteristiche che gli archivi di dati devono avere sono: - Deve esistere un nesso logico tra le informazioni contenute; - Devono essere rappresentate in un formato che renda possibile la loro interpretazione; - Devono essere registrate su un supporto che ne renda possibile la scrittura e la rilettura anche a distanza di tempo; - Devono essere organizzate in modo da permettere una facile consultazione.

La gestione di questi archivi viene realizzata attraverso le seguenti operazioni: - La creazione dell'archivio stesso; - La consultazione o interrogazione, cioè il reperimento delle informazioni, all'interno dell'archivio; - L'inserimento di nuovi dati dopo che l'archivio è stato creato; - La modifica o l'aggiornamento dei dati già presenti nell'archivio; - La cancellazione delle informazioni che non si vogliono più conservare; - L'ordinamento dei dati secondo un determinato criterio; - La fusione tra due o più archivi, cioè la creazione di un nuovo archivio utilizzando i dati contenuti negli archivi di partenza.

Fondamentali sono le funzioni per la sicurezza di questi archivi, che evitano la perdita di dati. Il BACKUP consiste nel creare una copia dell'archivio in modo da evitare che venga perduto; Il RESTORE è il procedimento inverso del Backup, cioè consente il recupero dei dati da una copia precedentemente salvata.

Quali sono i parametri per aprire un file in Php?

In PHP, un file si apre con il comando fopen(filename, mode), dove il mode può essere uno dei seguenti:

Mode	Descrizione
'r'	Solo lettura, puntatore all'inizio del file

'r+'	Lettura e scrittura, puntatore all'inizio del file
'w'	Solo scrittura, puntatore all'inizio del file. Se il file non esiste, tenta di crearlo
'w+'	Lettura e scrittura, puntatore all'inizio del file. Se il file non esiste, tenta di crearlo
'a'	Solo scrittura, puntatore alla fine del file. Se il file non esiste, tenta di crearlo
'a+'	Lettura e scrittura, puntatore alla fine del file. Se il file non esiste, tenta di crearlo

Cosa vuol dire aprire un file?

L'apertura di un file avviene in questo modo: il programma segnala al sistema operativo che ha necessità di accedere a un certo file. Il sistema operativo controlla che il file esista e che non sia già usato o bloccato, da esso o da un altro programma; crea alcune strutture dati per gestire le operazioni successive e riserva una certa quantità di memoria RAM, detta buffer, per memorizzare i dati in transito da e per il file; restituisce al programma un simbolo (o un handle oppure un indirizzo di memoria fisica) a cui il programma farà riferimento nelle successive operazioni sul file; gli altri programmi possono accedere allo stesso file solo in modo limitato (solo lettura) o non possono accedervi affatto.

Qual è l'istruzione in Php per aprire il file?

```
<?php
$variabile_associata_al_file= fopen ($nomefile, "r");
?>
```

Cosa vuol dire leggere i records di un file?

Nella fase di lettura il programma richiede dei dati dal file. Il sistema operativo li legge, li memorizza nel buffer di cui comunica l'ubicazione al programma richiedente.

In fase di "lettura" è possibile far ricorso a diverse funzioni che tuttavia presentano ognuna delle specifiche particolarità.

Quella che in caso di database testuali risulta essere maggiormente utile è file().

La funzione file() accetta come unico parametro il percorso al file che si desidera leggere ed esegue una lettura del file rigo per rigo restituendo un array in cui ciascun elemento è costituito da un rigo.

Ovviamente, avendo ottenuto un array, questo potrà essere letto con un classico ciclo foreach.

Qual è l'istruzione che permette di leggere i records?

```
<?php
include('setting.php');
$array_righi = file($my_database_txt);
foreach($array_righi as $key => $articolo){
```

```

list( $titolo, $autore, $prezzo, $codice, $categoria, $casa, $copie) = explode("|",
$articolo);
    echo '
    <p>
        <table width="100%">
            <tr">
                <td width="17%"> <b>Titolo:</b> ' .$titolo. ' </td>
                <td width="17,5%"> <b>Autore:</b> ' .$autore. ' </td>
                <td width="10%"> <b>Prezzo:</b> ' .$prezzo. ' </td>
                <td width="15,5%"> <b>Codice:</b> ' .$codice. ' </td>
                <td width="20%"> <b>Categoria:</b> ' .$categoria. ' </td>
                <td width="12,5%"> <b>Casa:</b> ' .$casa. ' </td>
                <td width="12,5%"> <b>Copie:</b> ' .$copie. ' </td>
            </tr>
        </table>
        <a href="action.php?delete=' .$key. '">Elimina</a> - <a
href="form_update.php?row=' .$key. '">Modifica</a>
    </p>
    <hr />;
}
?>

```

Descrivere il ciclo Foreach

Il ciclo foreach è stato pensato specificatamente per essere utilizzato per la gestione degli array: infatti il ciclo si ripeterà tante volte quanti sono il numero di elementi che compone un array.

La sua sintassi è molto semplice:

```
foreach ( array as nuova_variabile_valore) { azione }
```

Il foreach richiede come unico parametro l'array che si intende "ciclare", mentre il secondo parametro sarà la variabile che dovrà, ad ogni ciclo, contenere il valore dell'elemento dell'array.

L'azione contenuta nelle parentesi graffe verrà ripentuta per quanti sono gli elementi dell'array

Spiegare la funzione Explode()

Scorrendo il nostro array otterremo un singolo rigo per ogni ciclo; per isolare i singoli dati presenti all'interno di tale rigo si farà ricorso alla funzione explode().

La funzione explode di PHP ha il compito di suddividere una stringa sulla base di un dato separatore. La sintassi è la seguente:

```
explode(separatore, stringa);
```

La funzione explode() riceve due parametri obbligatori: il primo è il carattere separatore; il secondo è una stringa da trasformare in un array.

Cosa vuol dire scrivere (memorizzare) i dati in file di testo?

Memorizzare dei dati in un file di testo vuol dire scrivere essi all'interno del file. Per memorizzare i dati prima bisogna aprire il file e poi scriverci all'interno. La scrittura avviene in questo modo:

Il programma vuole scrivere dei dati in un file. Per prima cosa memorizza i dati in questione in un buffer, di cui poi comunica l'ubicazione al sistema operativo.

Per poter scrivere su un file di testo occorre anzitutto assicurarsi che questo abbia i permessi di scrittura: a questo scopo può essere utile la funzione `is_writable()`. Essa prende come parametro il percorso al file e restituisce un valore booleano, TRUE se il file può essere scritto. La scrittura del file avverrà con le funzioni `fopen()`, `fwrite()` e `fclose()`.

La funzione `fopen()` serve per aprire il collegamento con la risorsa (il file da scrivere). Essa prevede due parametri obbligatori: il percorso al file e una stringa che ci indicherà modalità con la quale si vorrà operare sul file (si rimanda al manuale per maggiori dettagli).

La funzione `fwrite()` esegue la scrittura sul file e prevede due parametri: la risorsa e la stringa da scrivere.

Infine, la funzione `fclose()` esegue la chiusura del file e prevede come unico parametro la risorsa.

Qual è l'istruzione di scrittura in un file txt (file dati)?

```
<?php
include("setting.php");

/*****
SCRITTURA
*****/
if(isset($_POST['scrivi']))
{
    if(!is_writable($my_database_txt)){
        exit("il file non ha i permessi di scrittura!");
    }
    // riceviamo i dati e li filtriamo
    $bad_char = array("|", "\r\n", "\r", "\n");
    $titolo = str_replace($bad_char, "", $_POST['titolo']);
    $autore = str_replace($bad_char, "", $_POST['autore']);
    $prezzo = str_replace($bad_char, "", $_POST['prezzo']);
    $codice = str_replace($bad_char, "", $_POST['codice']);
    $categoria = str_replace($bad_char, "", $_POST['categoria']);
    $casa = str_replace($bad_char, "", $_POST['casa']);
    $copie = str_replace($bad_char, "", $_POST['copie']);
    // apriamo il file
    $open = fopen($my_database_txt, "a+");
    // scriviamo i dati separati dal carattere separatore
```

```
fwrite($open,$titolo."|".$autore."|".$prezzo."|".$codice."|".$categoria."|".$casa."|".$scopie."\n");
// chiudiamo il file
fclose($open);
}
?>
```

Cosa vuol dire chiudere il file?

Chiudere il file vuol dire non rende più possibile la scrittura su di esso in quel momento.

La chiusura avviene in questo modo.

Il programma comunica che non ha più bisogno del file. Tutte le strutture dati e il buffer allocato vengono rilasciate; il sistema operativo elimina il file dalla lista di quelli in uso; gli altri programmi hanno ora libero accesso al file.

Qual è l'istruzione di chiusura del file?

```
<?php
$nomefile="gatta.txt";
$testo="Tanto va la gatta al lardo che ci lascia lo zampino";
$handle = fopen($nomefile, 'w');
fwrite($handle, $testo);
fclose($handle);
?>
```

Saper spiegare riga per riga il seguente codice

```
<?php
include("setting.php");

/*****
SCRITTURA
*****/
if(isset($_POST['scrivi']))
{
    if(!is_writable($my_database_txt)){ //controlla se il file ha i permessi
        exit("il file non ha i permessi di scrittura!");
    }
    // riceviamo i dati e li filtriamo
    $bad_char = array("|", "\r\n", "\r", "\n"); //metto negli array i simboli che mi serviranno per strutturare il
record, con /r/n spostato il cursore nella riga successiva. Bisogna usare questa istruzione altrimenti i dati te li
fa vedere in un'unica riga
    $scapo = str_replace($bad_char, "", $_POST['capo']); //str_replace prende il valore reale del campo
escludendo gli spazi insignificanti, "" per lo spazio,$campo contiene il nome della variabile che ho appena
inserito in memoria
    $taglia = str_replace($bad_char, "", $_POST['taglia']);
    $prezzo = str_replace($bad_char, "", $_POST['prezzo']);
    // apriamo il file
    $open = fopen($my_database_txt, "a+"); //apri il file, con a+ apro il file e inizio a scrivere dopo
l'ultima riga
```

```
// scriviamo i dati separati dal carattere separatore
fwrite($open, $capo."|".$taglia."|".$prezzo."\r\n"); //scrivo il record
// chiudiamo il file
```

```
fclose($open);
```

```
// ritorniamo nella pagina di visualizzazione
header("location: lettura.php");
exit;
}
```

```
/******
```

```
ELIMINARE UN RIGO
```

```
*****/
```

```
else if(isset($_GET['delete']))
```

```
{
// creiamo l'array con tutti i rigi
$array_righi = file($my_database_txt);// associo il nome del record al nome del file
// eliminiamo dall'array il rigo
unset($array_righi[$_GET['delete']]);//unset vuol dire cancella,il get delete è come se sto
memorizzando il cancella
```

```
// apriamo il file resettando il contenuto
$open = fopen($my_database_txt, "w");
foreach($array_righi as $key => $value){
// ri-scriviamo tutti i rigi (rimanenti)
fwrite($open, $value);
}
```

```
fclose($open);
// ritorniamo nella pagina di visualizzazione
header("location: lettura.php");
exit;
}
```

```
/******
```

```
MODIFICARE UN RIGO
```

```
*****/
```

```
else if(isset($_POST['modifica']) AND isset($_POST['row_update']))
```

```
{
// creiamo l'array con tutti i rigi
$array_righi = file($my_database_txt);
// riceviamo i dati e li filtriamo
$bad_char = array("|", "\r\n", "\r", "\n"); //bad_char mi cancella a video i dati, ma ricorda che sul disco
fisso devono rimanere i simboli di divisione e il comando a capo
```

```
$capo = str_replace($bad_char, "", $_POST['capo']); //str_replace dice di riscrivere in un campo il
contenuto di capo
```

```
$taglia = str_replace($bad_char, "", $_POST['taglia']);
$prezzo = str_replace($bad_char, "", $_POST['prezzo']);
// ri-scriviamo il rigo (che sostituirà il precedente)
```

```
$array_righi[$_POST['row_update']] = $capo."|".$taglia."|".$prezzo."\r\n"; //row_update vuol dire
aggiornami il record
```

```
// apriamo il file resettando il contenuto
$open = fopen($my_database_txt, "w");
foreach($array_righi as $key => $value){
// ri-scriviamo tutti i rigi
fwrite($open, $value);
}
```

```
    }  
    fclose($open);  
    // ritorniamo nella pagina di visualizzazione  
    header("location: lettura.php");  
    exit;  
    }  
?>
```