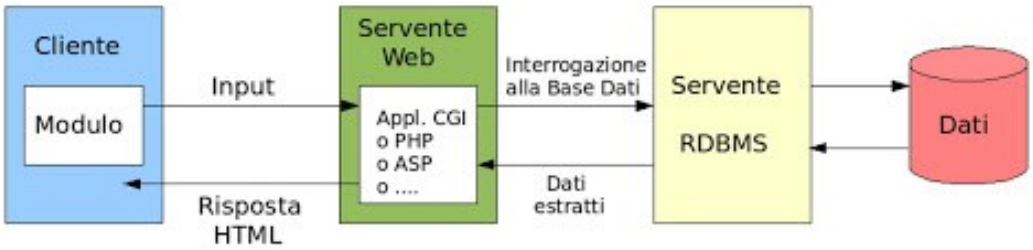
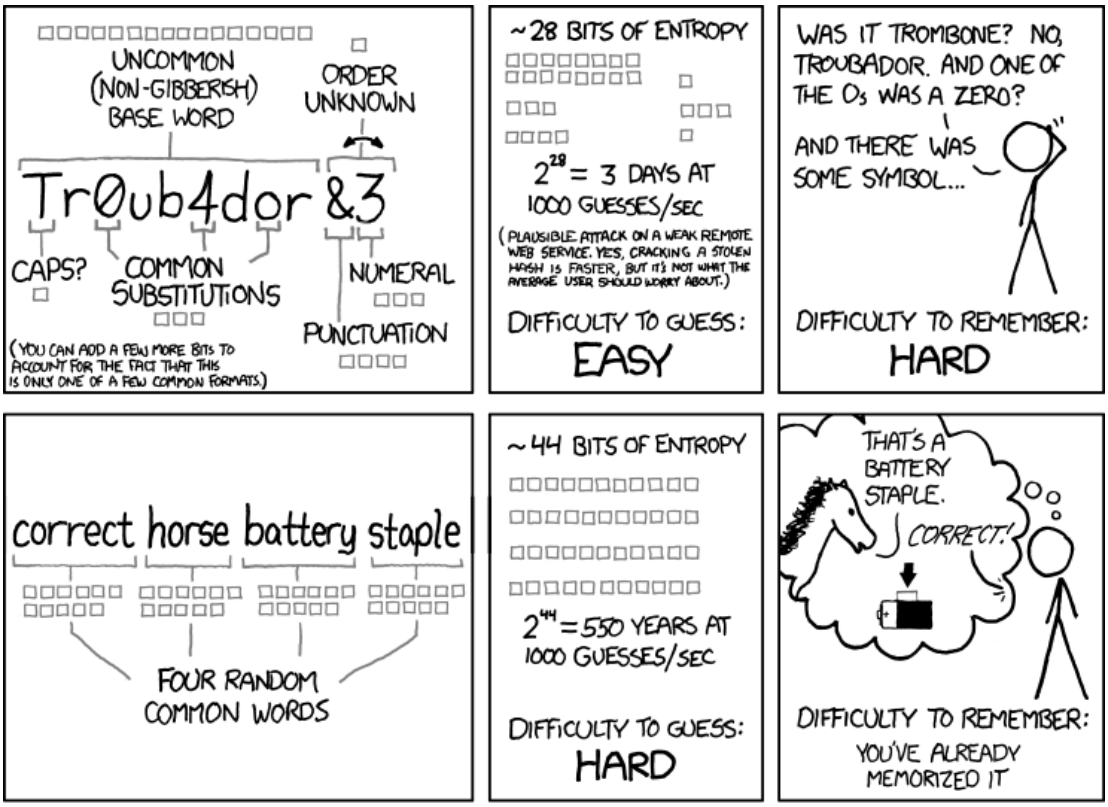
***Validare nel progetto e sviluppo di applicazioni WEB***
***ed analisi delle vulnerabilità***



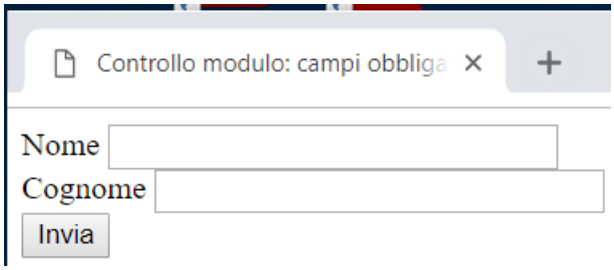

***Validare un modulo :***

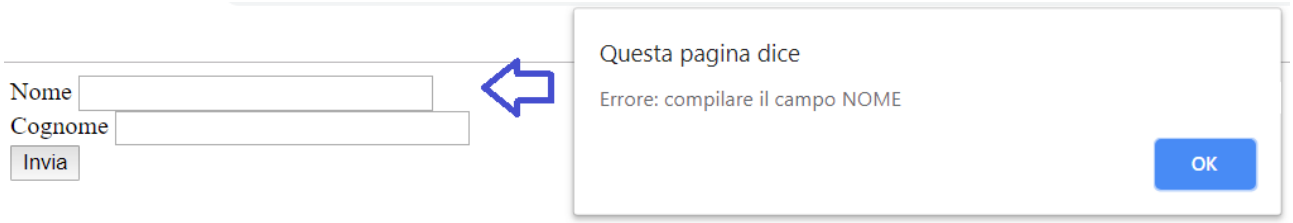

***creare***
***e validare***
***password "forti":***



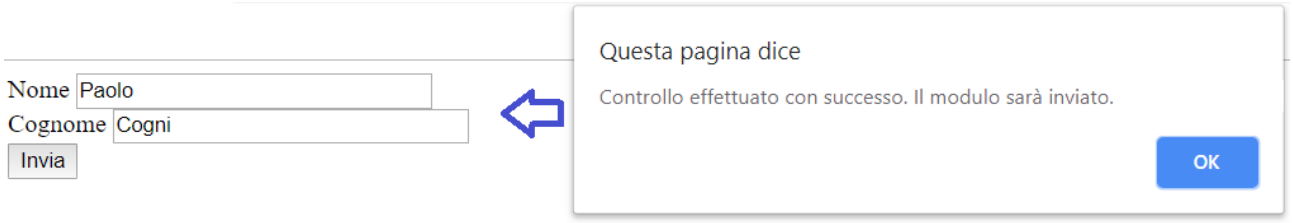
## *Controlli sui form*

***Semplice controllo su campi obbligatori con JavaScript***




Controllo in sequenza …. simile alert per COGNOME




***Codice***[1]

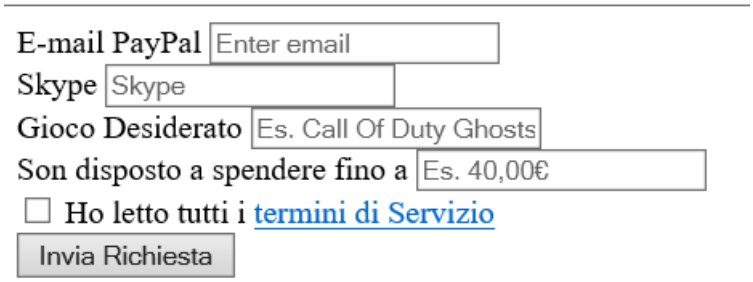
```
<html>
<head>
<title>Controllo modulo: campi obbligatori</title>
<script language="JavaScript">
function controllo(){
with(document.modulo) {        /* serve ad effettuare il codice nel form "modulo" della pagina (document) */
if(nome.value=="") {
        alert("Errore: compilare il campo NOME");
        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>
</head>
```

1 Da w3schools analoghi esempi "*JavaScript Form Validation*" (con funzioni definite dall'utente)

```
<body>
<! -- da http://www.comefaccio.net/tutorial/Javascript/139-Rendere-obbligatori-i-campi-di-un-form.html  -->
        <form name="modulo" id="modulo" onSubmit="return controllo();" method="post" action="">
                Nome <input name="nome" type="text" id="nome" size="30" />
                <br />
                Cognome <input name="cognome" type="text" id="cognome" size="30" />
                <br />
                <input type="submit" name="Submit" value="Invia" />
         </form>
 <body>
 </html>
```

***Prova codice:*** http://www.new345.altervista.org/PHP/email/controllo.htm (in JavaScript)

***Semplice form con invio email***

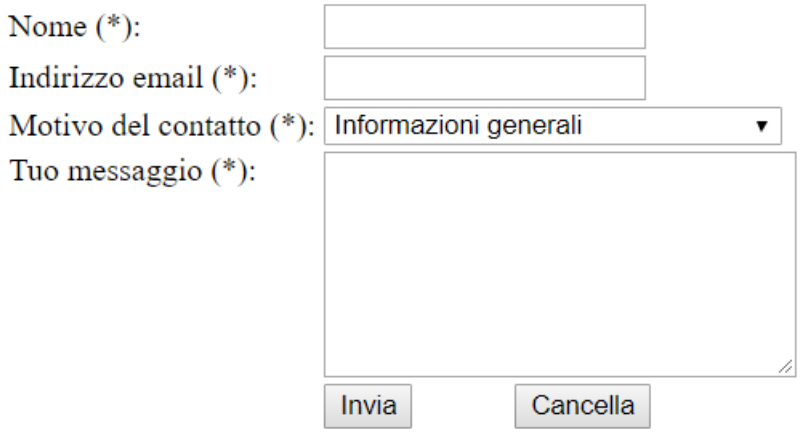



***Prova codice:*** http://www.new345.altervista.org/PHP/email/invio.htm

senza controllo smarcatura ...per lettura termini servizio solo pagina da creare

***contatti***

Modulo per contatti: ~ (*) campi obbligatori




tratto da il quaderno di mike

***NB:*** *pagina archiviata* ***"Controllo di un form e conservazione dei dati inseriti"***

***https://web.archive.org/web/20150523151854/****http://www.ilquadernodimike.altervista.org/files_php/php_preservare_i_dati.php?page=php*

***Prova codice:*** http://www.new345.altervista.org/PHP/email/contatti.php

in XAMPP warning pur inseriti campi

```php
<?php
foreach($_GET as $item){  // oppure foreach($_POST as $item)  se metodo POST
   if($item == ''){                          // oppure if(empty($item))
      echo "Errore, devi compilare tutti i campi!";
   exit;
   }
}
/* oppure per controllo
  if(isset($_GET['skype']))
    {
    if(empty($_GET['skype']))
        die('Il campo skype è vuoto!');
    }
    else die('il campo skype non esiste!');
*/

$mail = "infcol10@gmail.com";                          /* mail valida */
$oggetto = "GoldKey";
$testo = "PayPal: ".$_GET['paypal']."
Skype: ".$_GET['skype']."
Gioco: ".$_GET['game']."
Budget: ".$_GET['money'];

mail($mail,$oggetto,$testo);  /* arriverà una email dal server da dove la mandiamo e quindi
                                 anche col nome del server come mittente (per esempio spedendo
                                 una E-Mail dal server Altervista, vedremo, come mittente, Apache
                                mail($destinatario, $oggetto, $testo, $mittente);
                                   con
                                      $destinatario ="nomeutente@dominio.it";
                                      $oggetto ="prova di email";
                                      $testo="questa è una prova";
                                      $mittente='From: $nomemittente<>\n';

                                ad esempio 'From: "mittente: " <pippo.pluto@gmail.com> \r\n'
                                                  per vedere come mittente l'indirizzo
                                                  dell'utente che ci invia l'E-Mail.

                                Se vogliamo che, una eventuale risposta, venga inviata a un indirizzo
                                diverso dal mittente aggiungere

                                          $mittente .= "Reply-to: topolino@gmail.com \r\n";

                                Se abbiamo necessità di inviare e-mail con immagini
                                o file multimediali, dobbiamo inviare un e-mail in formato HTML.
                                Per fare ciò dobbiamo aggiungere al quarto parametro degli header
                                specifici:
                                          $mittente = "MIME-Version: 1.0 \r\n";
                                         $mittente .= "Content-type: text/html; charset=utf-8 \r\n";
                                         $mittente .= 'From: "mittente: "<pippo.pluto@gmail.com> \r\n';
                                         $mittente .= "Reply-to: info@gmail.com \r\n";

                                 ed inserire tag nel testo (si veda altro esempio) */
echo "Risponderemo alla tua richiesta entro 24h";
?>
```

***Segmento tratto da pagina[2] archiviata:***

```
<?php
                                                    /* modulo invio */
function inviaform() {
// viene preparato il messaggio da inserire nel corpo della lettera
$messaggio="Questa email ti è stata inviata dal tuo sito: NomeSito.
- Utente: $_POST[nome],
- email: $_POST[indirizzo],
- Motivo del contatto: $_POST[motivo].\n
- Messaggio: $_POST[testo]";
// viene preparato il mittente
$header = "From: ". $Name . " <" . $_POST[indirizzo] . ">\r\n";
// e finalmente viene spedita la E-Mail
mail("nomeutente@dominio.it", "Invio email da: $_POST[nome]", $messaggio , $header);
                                                                    /* da sostituire con URL */
// tutto quel che segue è il messaggio che assicura l'utente che la E-Mail è stata inviata
echo "<div class=\"titpag\">Invio e-mail</div>\n";
echo "<div id=\"main\">
....attendi.... <br /><br />
...IL TUO MESSAGGIO VERRA' INVIATO....</div>";
echo "<div class=\"titpag\">GRAZIE PER AVER INVIATO IL MESSAGGIO</div>\n
<p class=\"p\">Egregio sig. " . $_POST[nome] . ",
la ringraziamo per averci scritto, </p>\n
<p class=\"p\">Le risponderemo al pi&ugrave; presto.</p>\n
<p class=\"p\">Webmaster</p>\n";
}
?>
```

***Nb: in altri segmenti da sostituire il tag font con uso CSS e l'uso obsoleto delle funzioni mysql con "mysqli"***

## PHP Form Validation Example

* required field

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male Other *

Submit

***prova** codice[3] da w3schools*

```
function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
```

***e per validare indirizzo email:***
***prova codice[4]***
***nell'uso di funzione filter_var***

---

2 Pagina archiviata https://web.archive.org/web/20150523151854/http://www.ilquadernodimike.altervista.org/files_php/php_preservare_i_dati.php?page=php

3 Da PHP 5 mysqli_real_escape_string() per aggiungere sequenze di escape.

4 Con uso funzione filter-var

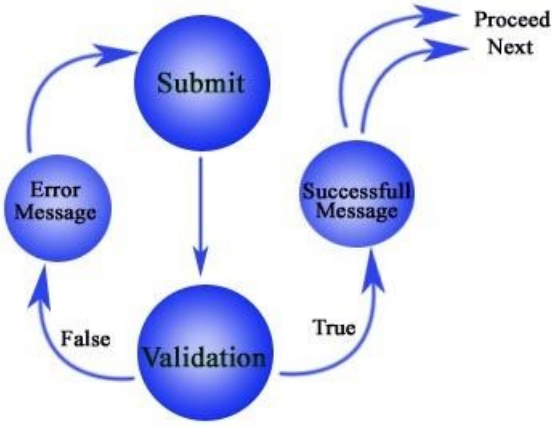
***Altre risorse online:***

*usando* ***bootstrap:*** *https://www.w3schools.com/bootstrap4/bootstrap_forms.asp*

*tag* ***HTML5****:* https://www.w3schools.com/html/html_form_input_types.asp

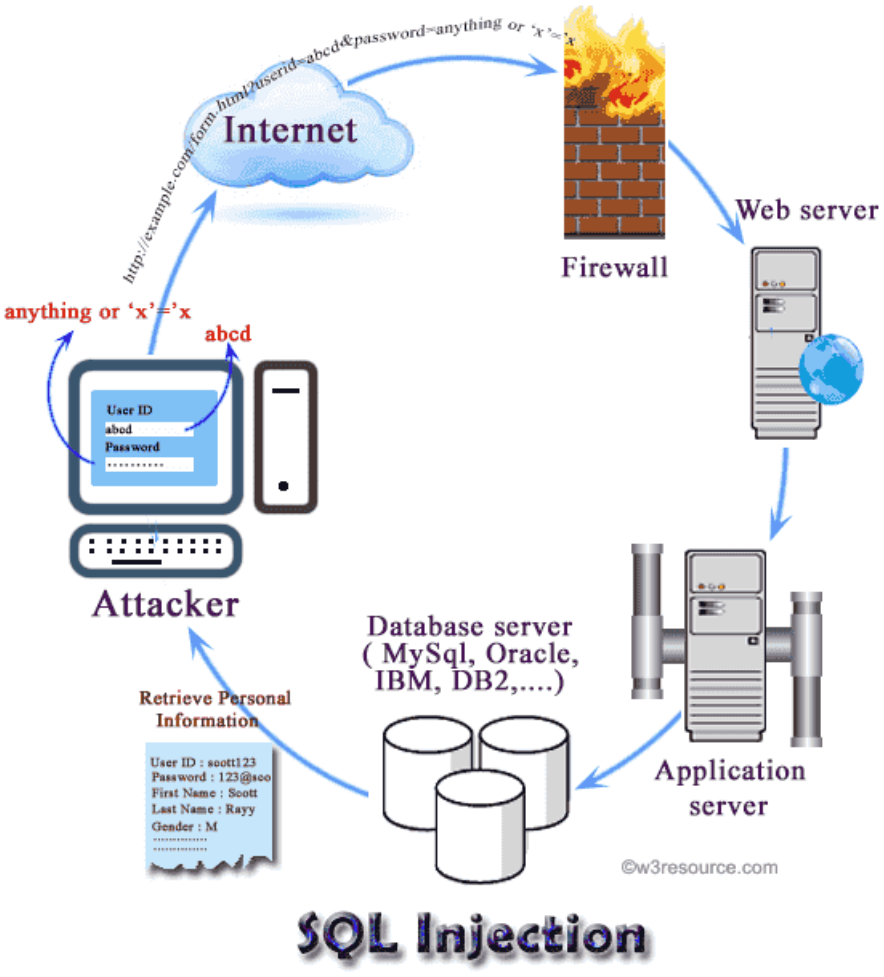
***PHP Tutorials for beginners: gestire forms***

***validare form***




***per evitare vulnerabilità***

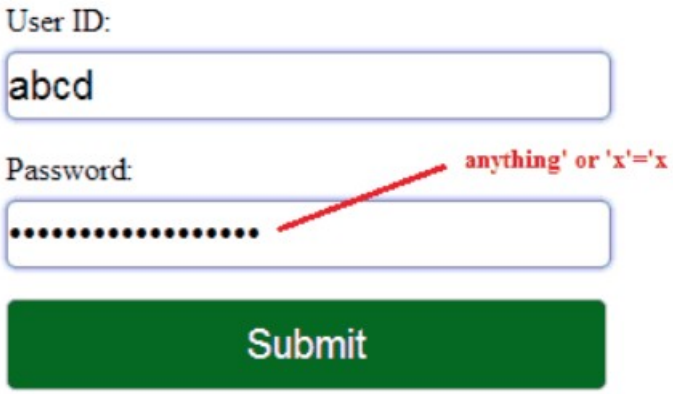
***tipico attacco:*** SQL injection




***Per una sitografia più completa:***

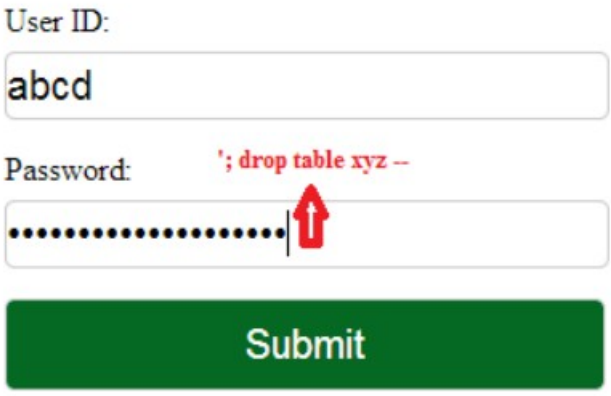
***"vulnerabilità***
***nelle applicazioni web"***

*Accedere*




*e scoprire dati*
*che si l'amministratore pensava nascosti:*


*Cancellare …*




```
select * from user_details
where userid = 'abcd'
and password = ''; drop table xyz
```

*… come*
*illustrato*
*nel fumetto*

## Who is Bobby Tables?

*pagina bobby-tables.com per illustrare SQL-Injection*
*(fumetto tratto da https://xkcd.com/327/)*

From the webcomic *xkcd*


## *Creare e validare password "forti"*

### *Esempi di codice e RegEx*


***Esempio completo con CSS e JS da*** w3schools *senza richiedere nella password* ***caratteri speciali usando pattern*** *="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"*

***Esempi che richiedono anche caratteri speciali:***

*// Validate password strength lato server*
*$uppercase = preg_match('@[A-Z]@', $psw);*
*$lowercase = preg_match('@[a-z]@', $psw);*
*$number = preg_match('@[0-9]@', $psw);*

*$stringa = '/paternCaratteriSpeciali/'; // fornendo elenco*
*// dei caratteri speciali che si possono inserire*

*$specialChars = preg_match('stringa', $psw);*


*in PHP delimitatori del pattern / oppure @*
*[^\w] significa che si richiede almeno un carattere*
*non alfanumerico né underscore*
*volendo considerare come carattere speciale anche l'underscore si fornisce pattern che lo comprende*
*ad esempio volendo i seguenti caratteri speciali !@#$%^&*()\-_=+{};:,<.>*

*$stringa = '/[!@#$%^&*()\-_=+{};:,<.>]/';*

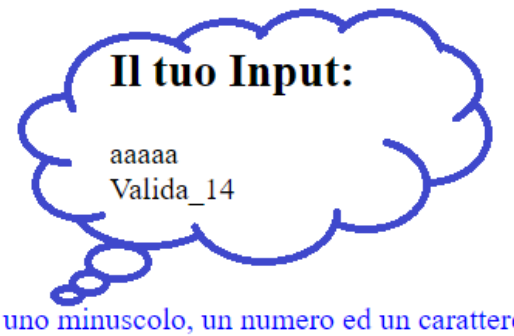
**Validare Password**

Username

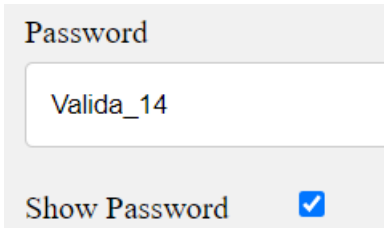
Password Submit

*Password sicura*

Password corretta:
almeno 8 cifre, almeno un carattere maiuscolo, uno minuscolo, un numero ed un carattere speciale




NB*: le app non ammettono i seguenti caratteri speciali: . , [ ( ) ] ' & ; ? \ : ; "*
*sono* ***accettati*** *! @ # $ % ' - / = ^ \ _ ` { } ~ +*

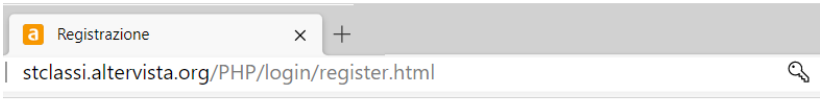



### *Utili esempi - JavaScript*

- *per consentire all'utente di visualizzare la password:*
  *tratto da* ***Toggle Password Visibility***
- *per* ***testare campi vuoti*** *da* w3schools
- **Input Type** ***HTML5*** *e* **pattern** *per validare da* **HTML5 Pattern** ***numeri di telefono, nome*** (ad esempio *username per Twitter*) ***etc…***

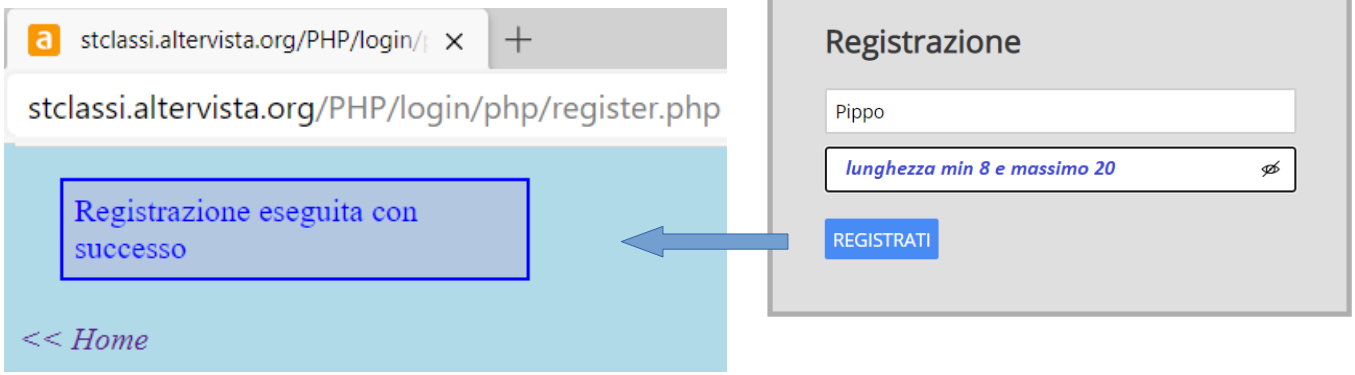
## Caso d'uso di sessioni - *web site in costruzione*

da **https://guidaphp.it/tutorial/form-registrazione-login-php-mysql**

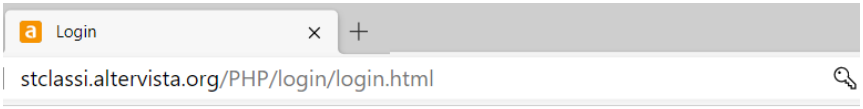

*form di registrazone*

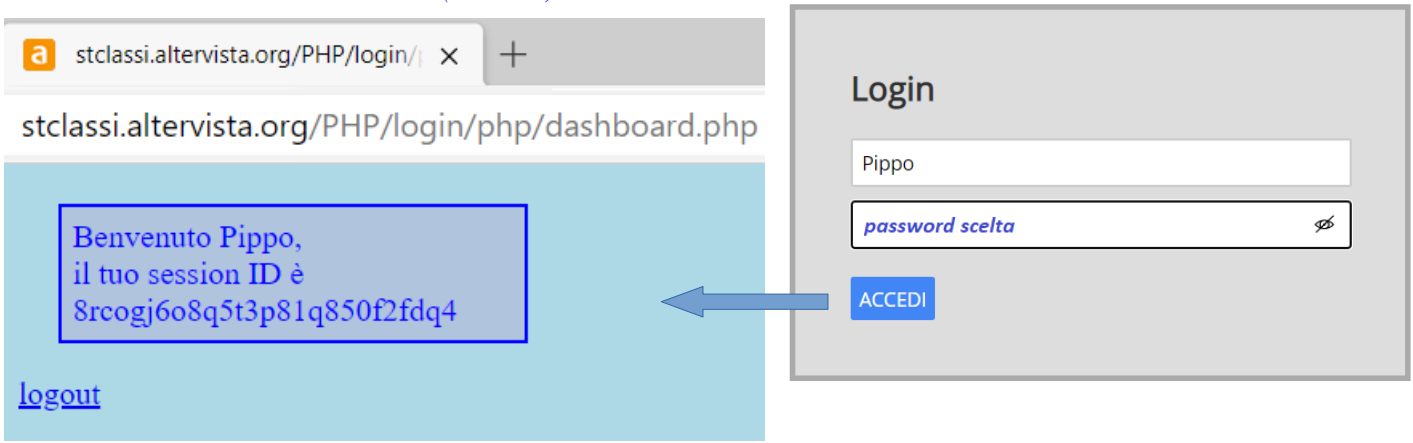
*e pagina server side (estratto)*




*form per accesso all'area riservata*

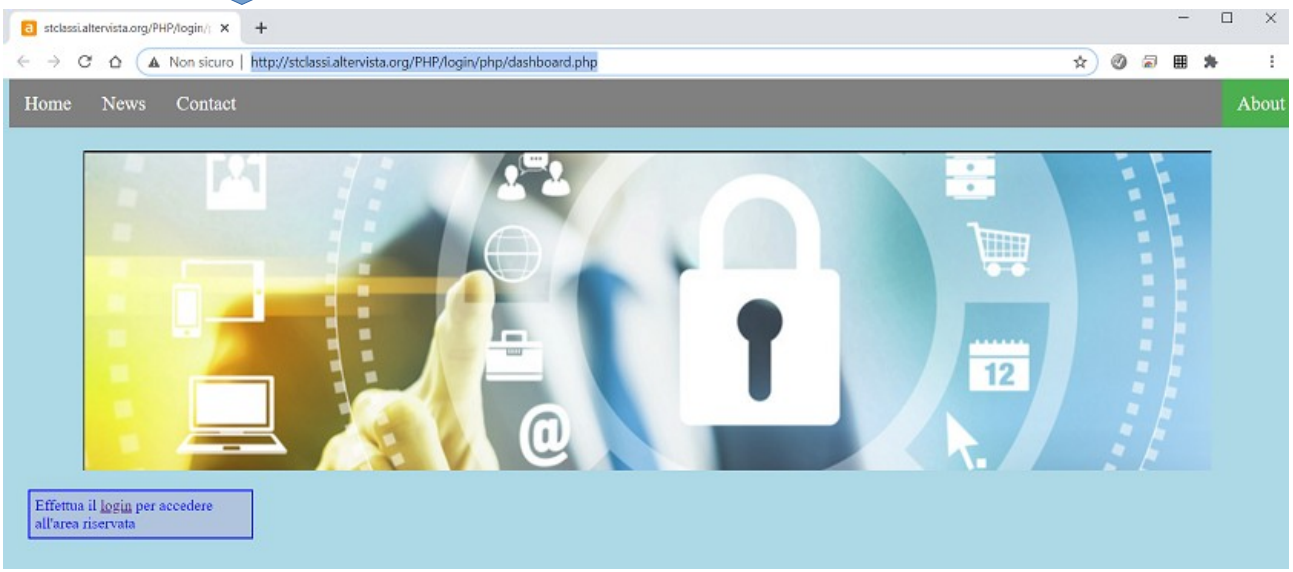
*e validazione lato server (estratto)*



## *Segmenti di codice*

```php
<?php                    // login utente
if (isset($_SESSION['session_id'])) {
   $session_user = htmlspecialchars($_SESSION['session_user'], ENT_QUOTES, 'UTF-8');
   $session_id = htmlspecialchars($_SESSION['session_id']);   // prevenire attacchi di tipo XSS convertendo
                                                              // eventuali tag HTML presenti nella stringa da form
   printf("Benvenuto %s,<br>il tuo session ID è %s", $session_user, $session_id);
  << altro codice  per accedere all'area riservata>>
   printf("%s", '<a href="logout.php">logout</a>');
} else {
     printf("Effettua il %s per accedere all'area riservata </p>", '<a href="../login.html">login</a>');
}
?>
```




```php
<?php          // validazione account di registrazione
if (isset($_POST['register'])) {
  //$username = $_POST['username'] ?? '';   // da PHP 7 operatore ternario di confronto null-coalescing
  $username = isset($_POST['username'])  ? $_POST['username'] : '';  // equivalente per  versioni  precedenti
  $password  = isset($_POST['password']) ? $_POST['password'] : '';
  $isUsernameValid = filter_var( $username,
                                 FILTER_VALIDATE_REGEXP, [ "options" => [  "regexp" => "/^[a-z\d_]{3,20}$/i"  ] ] );
   $pwdLenght = mb_strlen($password);
   if (empty($username) || empty($password)) {
      $msg = 'Compila tutti i campi %s';
   } elseif (false === $isUsernameValid) {
      $msg = 'Lo username non è valido. <br> Sono ammessi solamente caratteri
              alfanumerici e l\'underscore. <br>Lunghezza minina 3 caratteri.
             <br>Lunghezza massima 20 caratteri';
   } elseif ($pwdLenght < 8 || $pwdLenght > 20) {
      $msg = 'Lunghezza minima password 8 caratteri.
            <br>Lunghezza massima 20 caratteri';
   } else {
      $password_hash = password_hash($password, PASSWORD_BCRYPT); // un hash non è altro che una stringa
                       // di lunghezza fissa ottenuta mediante l'applicazione di una funzione di hash alla stringa di partenza
?>
```

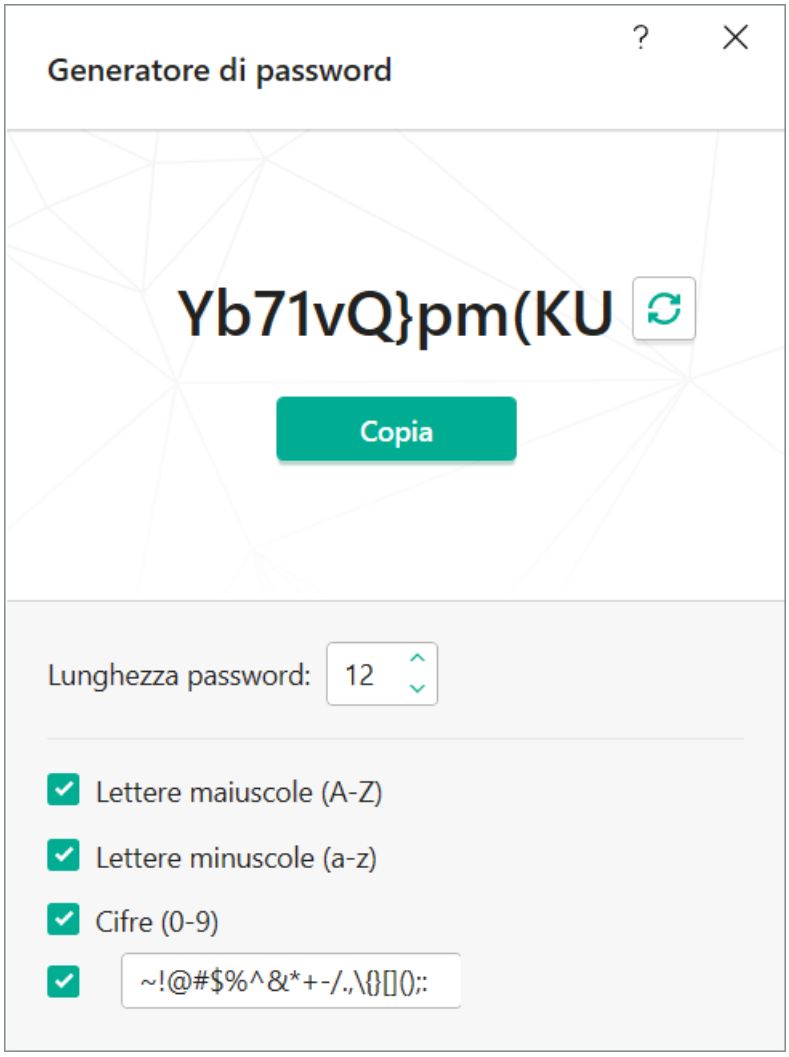
Lo username non è valido.
Sono ammessi solamente caratteri alfanumerici e l'underscore.
Lunghezza minina 3 caratteri.
Lunghezza massima 20 caratteri

Lunghezza minima password 8 caratteri.
Lunghezza massima 20 caratteri

## *Robustezza della password* *(wikipedia)*

*La robustezza della password è una misura di efficienza contro le varie tipologie di attacchi che una password può subire. La robustezza indica di **quanti tentativi** ha bisogno un aggressore, che non ha accesso diretto alla password, per indovinarla e violarla, introducendosi così illegalmente in account e sistemi informatici.*

*La **forza** della password si ricava da una funzione tra **lunghezza**, **complessità** e **imprevedibilità** della stringa di caratteri usati.*




*Quattro mosse per creare una password forte e facile da ricordare*

**Primo:**"Pensare ad una frase, al testo di una canzone, alle citazioni di un film, ad una filastrocca o a qualcosa di simile, che sia facile da ricordare".

**Secondo:**"Prendere la prima lettera delle prime tre o prime cinque parole".

**Terzo:**"Aggiungere, tra una lettera e l'altra, un carattere speciale (ad esempio: #, @, / e simili)". Adesso, è possibile basare tutte le proprie password **uniche** su questa singola stringa di caratteri.

**Quarto**: Prendete la stringa base e "annotate la prima parola che si associa al sito o alla piattaforma" cui accedere.
"Se si sta creando una password per l'account di Facebook – suggerisce Kaspersky - si potrebbe associare al social media il colore blu, presente nel logo. Basterebbe, quindi, aggiungere la parola 'blu'".
Ma si tratta sempre di una parola basata su un'associazione di idee soggettiva (ed è proprio questa la sua forza).

*Suggerimenti da David Jacoby, ricercatore di Kaspersky Lab*