

La crittografia lato client è migliore di quella lato server da ogni punto di vista. Infatti con la crittografia lato client il processo è effettuato sul tuo dispositivo. Sul server giunge solo la versione criptata, quella in chiaro non lascerà mai il tuo dispositivo.

La validazione lato client fornisce immediatamente il feedback, l'utente non ha bisogno di attendere l'invio del form e il ritorno di conferma/errore.

l'attributo placeholder: alla prima lettera digitata vedremo scomparire il testo segnaposto a vantaggio del nostro testo.

```
<!doctype html>

<html>
<head>
  <title>Disegnare i forms con HTML5</title>
<style>
  body {
    margin: 0; padding: 0;
    font-size: 12px;
    font-family: "Verdana", sans-serif;
    color: #444;
  }
  section {
    width: 400px;
    height: 230px;
    margin: 0 auto;
    border-left: 1px solid #ccc;
    border-right: 1px solid #ccc;
    margin-top: 10px;
    padding: 15px;
    padding-top: 5px;
  }
  fieldset {
    clear: left;
    border: 0px;
    border-bottom: 1px dotted #dcdcdc;
    padding: 10px;
    height: 30px;
  }
  label {
    font-weight: bold;
    width: 100px;
    float: left;
  }
  input, textarea {
    border: 1px solid #dcdcdc;
    padding: 4px;
    width: 250px;
```

```

    color: #444;
    float: left;
}

button {
    width: 15em;
    padding: .5em;
    color: #ffffff;
    text-shadow: 1px 1px 1px #000;
    border: solid thin #882d13;
    -webkit-border-radius: .7em;
    -moz-border-radius: .7em;
    border-radius: .7em;
    -webkit-box-shadow: 2px 2px 3px #999;
    box-shadow: 2px 2px 2px #bbb;
    background-color: #ce401c;
    background-image: -webkit-gradient(linear, left top, left bottom,
    from(#e9ede8), to(#ce401c),color-stop(0.4, #8c1b0b));
}

.firstSection {
    border-top: 1px solid #ccc;
    height: 200px;
}

.lastSection {
    border-bottom: 1px solid #ccc;
    height: 185px;
}

.longLine {
    height: 70px;
}

.note {
    font-size: 11px;
    font-style: italic;
}

.noBottomBorder {
    border: 0;
}
</style>
</head>

<body>
<form>
<section class="firstSection">
<h2>Dati anagrafici</h2>
<p class="note">L'asterisco (*) indica i campi obbligatori</p>
<fieldset>
    <label>Nome *</label><input name="nome" id="nome" type="text" required autofocus="autofocus">

```

```

</fieldset>
<fieldset>
    <label>Numero Figli</label><input name="n_figli" id="n_figli" type="number" min="0" max="10">
</fieldset>
</section>
<section>
    <h2>Dati commerciali</h2>
    <fieldset>
        <label>Telefono *</label><input name="telefono" id="telefono" type="tel" required>
    </fieldset>
    <fieldset>
        <label>P. Iva</label><input name="piva" id="piva" type="text">
    </fieldset>
    <fieldset>
        <label>Email *</label><input name="email" id="email" type="email" placeholder="nome@dominio.it"
required>
    </fieldset>
    <fieldset>
        <label>Sito Personale</label><input name="sito" id="sito" type="url" placeholder="www.sito.com">
    </fieldset>
</section>
<section class="lastSection">
    <h2>Informazioni extra</h2>
    <fieldset class="longLine">
        <label>Commenti</label><textarea name="commenti" id="commenti" placeholder="www.sito.com"
maxlength="100" wrap="soft"></textarea>
    </fieldset>
    <fieldset class="noBottomBorder">
        <button name="commenti" id="commenti">Invia</button>
    </fieldset>
</section>
</form>
</body>
</html>

```

Espressioni regolari javascript

[a-zA-Z]{6}[0-9]{2}[a-zA-Z][0-9]{2}[a-zA-Z][0-9]{3}[a-zA-Z]

Essa indica una sequenza di sei caratteri alfabetici, seguita da due caratteri numerici, una lettera, due numeri, una lettera, tre numeri ed infine una lettera. Vi ricorda qualcosa? E' lo schema di un codice fiscale.

Le espressioni regolari sono schemi (o pattern) che descrivono sinteticamente un insieme di stringhe utilizzando specifici caratteri speciali detti metacaratteri.

I metacaratteri più usati sono rappresentati dall'asterisco (*), che indica la presenza di zero o più occorrenze del carattere precedente, {n} indica la presenza di n occorrenze del carattere o dell'espressione che lo precede.

. Il punto indica un qualsiasi carattere

\w indica un carattere alfanumerico

\d indica una cifra numerica

/pattern /i indica che la ricerca non tiene conto della differenza di caratteri maiuscoli/minuscoli.

[^...] per escludere uno qualsiasi dei caratteri in parentesi

| (or logico) per esprimere un'alternativa tra due espressioni

**** (cosiddetto "escape")

{N} {Nmin,} {Nmin,Nmax} per specificare la numerosità (esatta, minima, minima e massima)

? indica zero o al più una occorrenza

+ indica una o più occorrenze di un'espressione

\$ corrisponde alla fine della stringa

^ corrisponde all'inizio della stringa

regexp.test(stringa) restituisce true se all'interno della stringa compare l'espressione, false altrimenti

/\w+@\w+\.\w{2,4}/i indirizzo di posta elettronica

/\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/ indirizzo IP

/[a-f0-9]+/i valori esadecimale

(https?:\V)?([\da-z\.-]+)\.([a-z\.]{2,6})([\V\w\.-]*)*\V? un URL

Possibile formato di una password: alfanumerica, minimo 8, massimo 16 caratteri, almeno una maiuscola

una minuscola ed una cifra, almeno una maiuscola, una minuscola ed una cifra, funzione che restituisca

true se la pwd è corretta, false altrimenti

function is_pwd_OK(p)

{

return

(/ ^[a-zA-Z0-9]{8,16}\$/.test(p) &&

/[A-Z]/.test(p) &&

/[a-z]/.test(p) &&

/[0-9]/.test(p)

)

}