

## Esercizi di riepilogo

### Esercizio 1

(a) Scrivere un documento HTML contenente una form contenente i seguenti campi:

- cognome e nome (casella di testo editabile lunga 40 caratteri)
- sesso (selezionabile tramite due bottoni radio)
- matricola (casella di testo editabile lunga 12 caratteri)
- regione di residenza (da scegliere da un menu che riporta le 20 regioni italiane)
- email (casella di testo editabile lunga 30 caratteri)
- telefono (casella di testo editabile lunga 15 caratteri)
- anno di corso (casella di testo editabile lunga 2 caratteri)
- richieste particolari (area di testo editabile di 12 righe per 60 colonne)
- bottone di invio
- bottone di reset

(b) Aggiungere al documento HTML funzioni JavaScript che eseguono i seguenti controlli:

- (b1) verifica che il cognome e nome non sia vuoto. Questa verifica va fatta ogni volta che l'utente cambia il valore del campo cognome e nome;
- (b2) verifica che sia stata selezionata una regione. Questa verifica va fatta all'atto dell'invio della form;
- (b3) verifica che l'anno di corso sia un numero compreso tra 1 e 6 oppure sia la stringa "FC". Questa verifica va fatta ogni volta che l'utente cambia il valore del campo anno di corso;
- (b4) verifica che o l'email o il telefono siano non vuoti. Questa verifica va fatta all'atto dell'invio della form.

(c) Per ognuno dei controlli specificati al punto (b), dire se è realizzabile in HTML5 senza utilizzare codice JavaScript, e in caso positivo, spiegare come.

### Soluzione

Soluzione punti (a) e (b):

Documento HTML contenente la form e la funzione JavaScript richiesta:

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" language="javascript">
      function controllaCognomeNome() {
        if (document.registr.cognome.value=="") {
          alert("Inserire cognome");
          return false;
        }
        return true;
      }
      function controllaAnno() {
        if (document.registr.anno.value!="FC") {
          if (isNaN(document.registr.anno.value)||document.registr.anno.value=="") {
            alert("Anno di corso errato");
            return false;
          }
          else {
            var v=parseInt(document.registr.anno.value);
```

```

        if ((v<1)||v>6)) {
            alert("Anno di corso errato");
            return false;
        }
    }
}
return true;
}
function validaForm() {
    if (document.registr.regione.value=="nessuna") {
        alert("Selezionare una regione");
        return false;
    }
    if ((document.registr.email.value=="")&&(document.registr.tel.value=="")) {
        alert("Inserire o l'email o il numero di telefono");
        return false;
    }
    alert("Dati inseriti correttamente");
    return true;
}
</script>
</head>
<body>
    <form action="" method="post" name="registr" onSubmit="return validaForm();">
        cognome:
        <input type="text" name="cognome" size="40" maxlength="40" onChange="return controllaCognomeNome();">
        <br/>
        sesso:
        <input type="radio" name="sesso" value="M">M
        <input type="radio" name="sesso" value="F">F
        <br/>
        matricola:
        <input type="text" name="matricola" size="12" maxlength="12">
        <br/>
        regione:
        <select name="regione">
            <option value="nessuna" selected></option>
            <option value="valdaosta">Val d'Aosta</option>
            <option value="piemonte">Piemonte</option>
            <option value="liguria">Liguria</option>
            <option value="lombardia">Lombardia</option>
            <option value="veneto">Veneto</option>
            <option value="trentino">Trentino Alto Adige</option>
            <option value="friuli">Friuli Venezia-Giulia</option>
            <option value="emilia">Emilia-Romagna</option>
            <option value="toscana">Toscana</option>
            <option value="marche">Marche</option>
            <option value="umbria">Umbria</option>
            <option value="lazio">Lazio</option>
            <option value="abruzzo">Abruzzo</option>
            <option value="molise">Molise</option>
            <option value="campania">Campania</option>
            <option value="basilicata">Basilicata</option>
            <option value="puglia">Puglia</option>
            <option value="calabria">Calabria</option>
            <option value="sicilia">Sicilia</option>
            <option value="sardegna">Sardegna</option>
        </select>
        <br/>
        email:
        <input type="text" name="email" size="30" maxlength="30">
        <br/>
        telefono:
        <input type="text" name="tel" size="15" maxlength="15">
        <br/>

```

```

    anno di corso:
    <input type="text" name="anno" size="2" maxlength="2" onChange="return controllaAnno();">
    <br/>
    richieste particolari:
    <br/>
    <textarea name="richieste" cols="60" rows="12"></textarea>
    <br/>
    <input type="submit" value="Invia">
    <input type="reset" value="Reset">
  </form>
</body>
</html>

```

Soluzione punto (c):

I primi tre controlli del punto (b) sono realizzabili in HTML5 senza includere script. In particolare:

(b1) verifica che il cognome e nome non sia vuoto:

```
<input type="text" name="cognome" size="40" maxlength="40" required>
```

(b2) verifica che sia stata selezionata una regione;

```
<select name="regione" required>
```

(b3) verifica che l'anno di corso sia un numero compreso tra 1 e 6 oppure sia la stringa "FC";

```
<input type="text" name="anno" size="2" maxlength="2" pattern="1|2|3|4|5|6|FC">
```

Infine, il controllo (b4) (verifica che o l'email o il telefono siano non vuoti) non può essere codificato in HTML5 senza l'uso di script.

**Esercizio 2** Scrivere un documento HTML contenente una form contenente i seguenti campi:

- matricola (casella di testo editabile lunga 12 caratteri)
- cognome e nome (casella di testo editabile lunga 60 caratteri)
- data di nascita (casella di tipo date)
- sesso (menu a due opzioni, F e M)
- corso frequentato (menu a tendina contenente una lista di corsi)
- bottone di invio
- bottone di stampa del local storage degli utenti

Aggiungere funzioni JavaScript e un foglio di stile CSS tali che:

1. i dati relativi ad ogni utente devono essere memorizzati, all'atto del loro invio, nel local storage;
2. il bottone di stampa del local storage deve visualizzare, nella parte finale del documento HTML, gli utenti memorizzati nel local storage;
3. il foglio di stile CSS deve fare in modo che i dati (in particolare nome e cognome, sesso e data di nascita) relativi agli studenti maschi siano visualizzati con colore e font diverso dai dati relativi alle studentesse femmine. Non si possono usare CSS inlined (solo CSS incorporati).

## Soluzione

Documento HTML contenente la form richiesta:

```

<!DOCTYPE html>
<html>
<head>
<style>
tr.M {color:blue; font-family: "Courier New";}
tr.F {color:red; font-family: Arial;}
</style>
<script>
function inizializzaStorageUtenti(){
    if (typeof(localStorage.utenti) == "undefined") {
        localStorage.utenti="[]";
    }
}

function resetStorageUtenti(){
    localStorage.utenti="[]";
}

function stampaStorage(){
    var u = JSON.parse(localStorage.utenti);
    var l = u.length;
    var s = new String("<h3>Stato di localStorage:</h3>");
    s +="<table border=1><tr><th>cognome e nome</th><th>data di nascita</th><th>sex</th></tr>";
    for (i=0;i<l;i++)
        s += "<tr class="+u[i].sex+"><td>"+u[i].cogn+"</td><td>"+u[i].data+"</td><td>"+u[i].sex+"</td></tr>";
    s += "</table>";
    document.getElementById("vistaStorage").innerHTML = s;
    return true;
}

function inserisciUtente() {
    if (document.registrazione.cognome.value=="") {
        alert("Inserire cognome");
        return false;
    }
    if (document.registrazione.dataNascita.value=="") {
        alert("Inserire data di nascita");
        return false;
    }
    var u = JSON.parse(localStorage.utenti);
    var nextpos = u.length;
    var o = { cogn:document.registrazione.cognome.value,
        data:document.registrazione.dataNascita.value,
        sesso:document.registrazione.sesso.value };
    alert("Dati inseriti correttamente");
    u[nextpos] = o;
    localStorage.utenti = JSON.stringify(u);
    return true;
}
</script>
</head>
<body onload="inizializzaStorageUtenti()">
<form action="" method="post" name="registrazione" onSubmit="return inserisciUtente();" accept-charset="utf-8">

matricola:
<input type="text" name="matricola" size="12" maxlength="12"/>
<br/>
cognome e nome:
<input type="text" name="cognome" size="60" maxlength="60"/>
<br/>
data di nascita:
<input type="date" name="dataNascita" size="10" maxlength="10"/>
<br/>
sesso:
<select name="sesso" id="sesso">

```

```

<option value="F">F</option>
<option value="M">M</option>
</select>
<br/>
corso frequentato:
<select name="corso" id="corso">
<option value="Elettronica">Elettronica</option>
<option value="LTW">LTW</option>
<option value="PFP">PFP</option>
<option value="SO">SO</option>
</select>
<br/>
<input type="submit" value="Invia"/>
<input type="button" value="Stampa Storage" onclick="stampaStorage();" />
<input type="button" value="Reset Storage" onclick="resetStorageUtenti();" />
</form>
<hr/>
<div id="vistaStorage"></div>
<hr/>
</body>
</html>

```

Per comodità è stato aggiunto un bottone “Reset Storage” per la cancellazione di tutti gli utenti dal local storage.

**Esercizio 3** Dato il seguente documento HTML:

```

<html>
  <body>
    inizio documento
    <hr/>
    <div>
      prima zona di visualizzazione dei documenti
    </div>
    <hr/>
    <div>
      seconda zona di visualizzazione dei documenti
    </div>
    <hr/>
    fine documento
  </body>
</html>

```

Modificare il documento HTML in modo da poter selezionare, tramite un menu con 8 opzioni, il caricamento asincrono di 8 diversi documenti HTML (nomi dei documenti: doc1.htm, doc2.htm, doc3.htm, doc4.htm, doc5.htm, doc6.htm, doc7.htm, doc8.htm): i primi quattro documenti vanno visualizzati all’interno del primo elemento div, mentre gli altri quattro documenti vanno visualizzati all’interno del secondo elemento div.

**Soluzione**

```

<!DOCTYPE html>
<html>
  <body>
    inizio documento
    <hr/>
    <select name="menuDoc">
      <option value="doc1">doc1</option>
      <option value="doc2">doc2</option>
      <option value="doc3">doc3</option>
      <option value="doc4">doc4</option>
      <option value="doc5">doc5</option>
      <option value="doc6">doc6</option>
      <option value="doc7">doc7</option>
      <option value="doc8">doc8</option>
    </select>
  </body>
</html>

```

```

<hr/>
<div id="zona1">
    prima zona di visualizzazione dei documenti
</div>
<hr/>
<div id="zona2">
    seconda zona di visualizzazione dei documenti
</div>
<hr/>
fine documento<br/>
<script>
    document.getElementsByTagName("select")[0].onchange=caricaDocumento;

    function caricaDocumento(e) {
        var httpRequest = new XMLHttpRequest();
        httpRequest.prevTarget = e.target;
        httpRequest.onreadystatechange = gestisciResponse;
        httpRequest.open("GET", e.target.value + ".htm", true);
        httpRequest.send();
    }
    function gestisciResponse(e) {
        if (e.target.readyState == 4 && e.target.status == 200) {
            var c=e.target.prevTarget.value.charAt(3);
            if (c<="4") c="1"; else c="2";
            document.getElementById("zona"+c).innerHTML = e.target.responseText;
        }
    }
</script>
</body>
</html>

```

**Esercizio 4** Scrivere un documento HTML che risolve il precedente esercizio 5 utilizzando JQuery.

### Soluzione

```

<!DOCTYPE html>
<html>
<body>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js">
    </script>
    inizio documento
    <hr/>
    <select name="menuDoc">
        <option value="doc1">doc1</option>
        <option value="doc2">doc2</option>
        <option value="doc3">doc3</option>
        <option value="doc4">doc4</option>
        <option value="doc5">doc5</option>
        <option value="doc6">doc6</option>
        <option value="doc7">doc7</option>
        <option value="doc8">doc8</option>
    </select>
    <hr/>
    <div id="zona1">
        prima zona di visualizzazione dei documenti
    </div>
    <hr/>
    <div id="zona2">
        seconda zona di visualizzazione dei documenti
    </div>
    <hr/>
    fine documento<br/>
    <script>
        $(document).ready(function(){

```

```

    $("#select").change(function(){
    var c=this.value.charAt(3);
    if (c<="4") c="1"; else c="2";
    $("#zona"+c).load(this.value+".htm",
    function(responseTxt, statusTxt, xhr){
        if(statusTxt == "error") alert("Errore" + xhr.status + ":" + xhr.statusText);
    });
    });
    });
</script>
</body>
</html>

```

**Esercizio 5** Rappresentare in XML i seguenti dati in formato JSON:

```

[
  {
    "autori": ["Anna Bianchi","Mario Rossi","Gianni Verdi"],
    "titolo": "Titolo 1",
    "data": { "anno": 2019, "mese": 10, "giorno": 1 },
    "tipoPub": "libro"
  },
  {
    "autori": ["Paola Gialli","Gianni Verdi"],
    "titolo": "Titolo 2",
    "data": { "anno": 2020, "mese": 2, "giorno": 15 },
    "tipoPub": "articolo"
  },
  {
    "autori": ["Mario Rossi","Maria Neri"],
    "titolo": "Titolo 3",
    "data": { "anno": 2018, "mese": 1, "giorno": 31 },
    "tipoPub": "libro"
  }
]

```

## Soluzione

```

<?xml version="1.0"?>
<bibliografia>
  <voce>
    <autori>
      <autore>Anna Bianchi</autore>
      <autore>Mario Rossi</autore>
      <autore>Gianni Verdi</autore>
    </autori>
    <titolo>
      Titolo 1
    </titolo>
    <data>
      <anno>2019</anno>
      <mese>10</mese>
      <giorno>1</giorno>
    </data>
    <tipoPub>
      Libro
    </tipoPub>
  </voce>
  <voce>
    <autori>
      <autore>Paola Gialli</autore>
      <autore>Gianni Verdi</autore>
    </autori>

```

```

<titolo>
  Titolo 2
</titolo>
<data>
  <anno>2020</anno>
  <mese>2</mese>
  <giorno>15</giorno>
</data>
<tipoPub>
  Articolo
</tipoPub>
</voce>
<voce>
  <autori>
    <autore>Maria Neri</autore>
    <autore>Mario Rossi</autore>
  </autori>
  <titolo>
    Titolo 3
  </titolo>
  <data>
    <anno>2018</anno>
    <mese>1</mese>
    <giorno>31</giorno>
  </data>
  <tipoPub>
    Libro
  </tipoPub>
</voce>
</bibliografia>

```

**Esercizio 6** Scrivere un foglio di stile XSLT che trasforma in HTML i documenti XML del tipo dell'esempio precedente permettendo una opportuna visualizzazione sul browser dei dati contenuti.

### Soluzione

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html"/>

  <xsl:template match="/*">
    <html>
      <body>
        Bibliografia:
        <ol>
          <xsl:apply-templates select="voce"/>
        </ol>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="voce">
    <li>
      <xsl:value-of select="tipoPub"/>:
      <ul>
        <xsl:apply-templates select="*" />
      </ul>
    </li>
  </xsl:template>

  <xsl:template match="autori">
    <li>Autori:
      <ol>
        <xsl:for-each select="autore">

```



```

        <li><xsl:value-of select="."/></li>
    </xsl:for-each>
</ol>
</li>
</xsl:template>

<xsl:template match="titolo">
    <li>Titolo:
        <xsl:value-of select="."/>
    </li>
</xsl:template>

<xsl:template match="data">
    <li>Data:
        <xsl:value-of select="giorno"/><xsl:value-of select="mese"/><xsl:value-of select="anno"/>
    </li>
</xsl:template>

<xsl:template match="text()"/>

</xsl:stylesheet>

```