

**MASTER IN TECNOLOGIE INTERNET
ANNO 2011
PROGETTAZIONE WEB
LATO CLIENT**

**PROGETTO:PRENOTAZIONE ONLINE
ORA DI RIPETIZIONE**

RELATORE:

ING. STEFANO GHELARDINI

PROGETTO:PRENOTAZIONE ONLINE ORA DI RIPETIZIONE**3****A cosa serve****3****Come funziona il modulo di prenotazione****3****Il codice****6**

Codice in Javascript

6

Codice in HTML

15

CSS

18

PROGETTO:PRENOTAZIONE ONLINE ORA DI RIPETIZIONE

A cosa serve

Questo progetto, realizza la parte client di un sito che permette la prenotazione online di un'ora di lezione, di una certa materia, ad un certo orario, per un certo studente.

Per poter fare la prenotazione, lo studente (utente) deve inserire nel modulo di prenotazione: i propri dati personali, la materia, l'ora a cui fare la ripetizione.

Come funziona il modulo di prenotazione

Lo studente arriva al sito Internet e inserisce i propri dati personali

Inserisci i tuoi dati

Nome: Cognome:

E-mail: Telefono:

Fatto questo sceglie la materia di cui vuole l'ora di ripetizione

Scegli la materia

Scegli la materia di cui vuoi la ripetizione:

Italiano ▼

Va sul calendario e pone il mouse sul giorno e l'ora in cui vuole la ripetizione.

Le caselle di colore rosso rappresentano giorni (o ore) in cui non vengono effettuate le lezioni; le caselle verdi viceversa rappresentano ore in cui è possibile ottenere la ripetizione.

Seleziona giorno ed ora

Seleziona il giorno e l'ora in cui vuoi fare la ri

mer	gio	ven	sab	
22 ago	23 ago	24 ago	25 ago	
14	14	14	14	
15	15	15	15	
16	16	16	16	

La casella su cui arriva la freccia del mouse diventa gialla se quell'ora è disponibile per le ripetizioni ;

Seleziona giorno ed ora

Seleziona il giorno e l'ora in cui vuoi fare la ripetizione

mer 22 ago	gio 23 ago	ven 24 ago	sab 25 ago
14	14	14	14
15	15	15	15
16	16	16	16

quando lo studente decide di selezionare un'ora, preme il tasto sinistro del mouse.
La casella selezionata diventa rossa (l'ora non è più disponibile perché occupata dallo studente) e all'interno della casella è riportata la materia selezionata dallo studente.

Nel caso in cui lo studente si renda conto di aver sbagliato ora o giorno, può cancellare la prenotazione tramite il pulsante *cancella* posto sotto il calendario

Seleziona giorno ed ora

Seleziona il giorno e l'ora in cui vuoi fare la ripetizione

mer 22 ago	gio 23 ago	ven 24 ago	sab 25 ago
14	14	14	14
15	15	Italiano	15
16	16	16	16
17	17	17	17
18	18	18	18

Terminata la prenotazione, lo studente deve premere il pulsante *conferma*

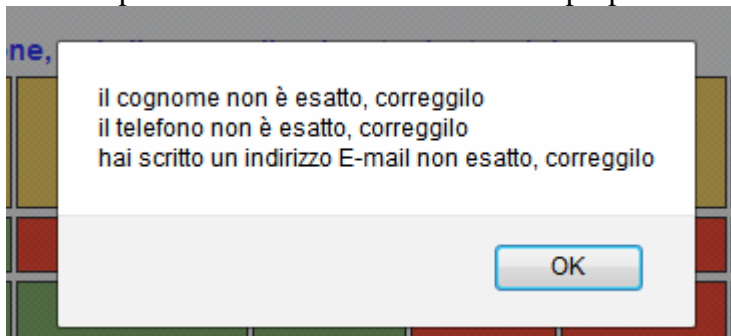
Conferma prenotazione

ciao stefano, hai prenotato un'ora di Italiano giovedì 27 settembre alle ore 14

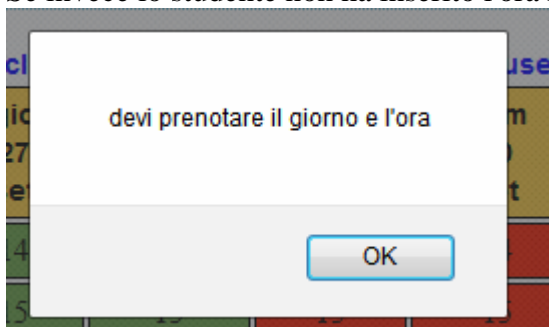
sopra al pulsante conferma, è presente un'etichetta che riassume i dati scritti dallo studente; il contenuto dell'etichetta varia in base a come viene compilato il modulo

Nel caso in cui il modulo non sia stato completato in tutte le sue parti, esiste un controllo dell'errore che evita l'invio del modulo al server.

Ad esempio se lo studente non ha inserito i propri dati otteniamo



Se invece lo studente non ha inserito l'ora in cui vuole fare la ripetizione otteniamo



Il codice

possiamo distinguere il codice in due parti:

la parte statica (che non varia quando un utente si collega al sito)

la parte dinamica che invece varia in base all' iterazione con l'utente.

La parte statica, realizzata tramite codice HTML e CSS, consiste del form in cui l'utente inserisce i propri dati ;

La parte dinamica consiste: nella creazione del calendario, nel cambiamento di colore nelle varie celle, nel cambiamento del contenuto della cella del calendario quando questa viene selezionata con il mouse;

la parte dinamica viene realizzata in javascript

Codice in Javascript

vediamo ora il codice in javascript; esso è composto da due oggetti (studente e calendario) più una variabile globale che viene utilizzata per sincronizzare i due oggetti.

La variabile globale è la seguente

```

/*****
variabile globale: serve per far comunicare dall'oggetto calendario
all'oggetto studentese è stata fatta la prenotazione di un'ora
*****/

```

```
var pass=false;
```

Per quanto riguarda l'**oggetto studente** il codice che lo gestisce è scritto di seguito

```

/*****/
//oggetto studente
/*****/
//costruttore
function Studente () {
    this.nome = null;
    this.cognome =null;
    this.telefono=null;
    this.email= null;
}

//metodi
Studente.prototype.creaStudente= function() {
    this.nome=document.getElementById('idNome').value;
    this.cognome=document.getElementById('idCognome').value;
    this.telefono=document.getElementById('idTelefono').value;
    this.email=document.getElementById('idEmail').value;
}

```

Il metodo creaStudente assegna alle variabili interne dell'oggetto studente i valori scritti dall'utente nei campi del modulo di compilazione

```
Studente.prototype.controllaDati= function() {
```

Il metodo controlla dati verifica che i dati inseriti dall'utente siano corretti, per fare questo utilizza le espressioni regolari; se i dati sono corretti (errore = false) e se la variabile pass è true (è stata selezionata un'ora di lezione) , viene effettuato il submit, altrimenti se uno o più dati inseriti non sono corretti, il metodo crea una stringa che verrà visualizzata in un prompt e dice quale dei dati è stato inserito male

```

var errore = false;
var str = "";
var schemaNome= /^[a-zA-Z]{2,}$/;
var schemaCognome= /^[a-zA-Z]{2,}$/;
var schemaTelefono= /^[0-9]{4,}$/;
var schemaEmail= /^[A-Za-z0-9]\w{2,}@[A-Za-z0-9-]{3,}\.[A-Za-z]{2,}$/;
if(!(schemaNome.test(this.nome))){
    str += "il nome non è esatto, correggilo \n";
    errore = true;
}
if(!(schemaCognome.test(this.cognome))){
    str += "il cognome non è esatto, correggilo \n";
    errore = true;
}
if(!(schemaTelefono.test(this.telefono))){
    str += "il telefono non è esatto, correggilo \n";
    errore = true;
}
if(!(schemaEmail.test(this.email))){
    str += "hai scritto un indirizzo E-mail non esatto, correggilo";
    errore = true;
}
if(errore){
    window.alert(str);
}else{
    if(pass==true){
        document.getElementById('idForm').submit();
    }
    else{
        window.alert("devi prenotare il giorno e l'ora");
    }
}
}

```

Infine per quanto riguarda l'oggetto **calendario** abbiamo

```

/*****
//oggetto calendario
*****/

function Calendario() {
    /*crea un array di nome tab che conterra' il nome del colore di ogni cella
    del calendario*/
    this.tab = new Array();
    /*crea un array di nome parola che conterra' la stringa scritta in ogni
    cella del calendario*/
    this.parola = new Array();
    /*crea 2 matrici formate da 6 elementi
    for (i = 0; i < 6; i++) {
        this.tab[i] = new Array();
        this.parola[i] = new Array();
    }

    /* inizializzo a verde tutte le componenti della matrice tab, e parola con
    le ore delle lezioni */
    for(i=0;i<6;i++){
        for(j=0;j<14;j++){
            this.tab[i][j]="verde";
            var oraIn = 14-1+i;
            var oraOut = 14+i;
            this.parola[i][j]= oraIn ;
        }
    }

    /*creo gli array che contengono giorno (gio) mese (me)
    numero del giorno (ndg)*/
    this.gio = new Array();
    this.me = new Array();
    this.ndg = new Array();

    /*altre variabili non di tipo array*/
    this.calendario;
    this.MESE = ["gen", "feb", "mar", "apr", "mag", "giu", "lug", "ago", "set",
    "ott", "nov", "dic"];
    this.MESE_COMPL = ["gennaio", "febbraio", "marzo", "aprile", "maggio",
    "giugno", "luglio", "agosto", "settembre", "ottobre", "novembre", "dicembre"];
    this.GIORNO = ["dom", "lun", "mar", "mer", "gio", "ven", "sab"];
    this.GIORNO_COMPL = ["domenica", "lunedì", "martedì", "mercoledì", "giovedì",
    "venerdì", "sabato"];
    /*mese m, numero del giorno d, giorno g, della data di oggi (adesso)
    this.adesso = new Date();
    this.m = null;
    this.d = null;
    this.g = null;

```

Vengono creati 6 oggetti array, ogni array contiene una riga della tabella che rappresenterà il calendario. Poiché il calendario è formato da 14 giorni, ogni oggetto array, quando verrà istanziato, sarà formato da 14 componenti. tab[][] memorizza il colore di ogni cella da cui è formato il calendario mentre parola[][] serve a memorizzare la parola scritta all'interno di ogni cella del calendario

La variabile "adesso" è un oggetto che contiene la data odierna


```
//inizializzazione data di oggi
this.m = this.adesso.getMonth(); // 0 = gennaio, 1 = febbraio, ...
this.d = this.adesso.getDate(); // 1 = primo del mese,...
this.g = this.adesso.getDay();
```

inizializzo le variabili "m", "d", "g" con i valori:
del mese, del numero del giorno e del nome del
giorno della data odierna

```
//mese mn,numero del giorno dn,giorno gn, della data di domani(nuova data)
this.nuovaData;
this.mn = null;
this.dn = null;
this.gn = null;
```

```
//altre variabili
this.idCella=null;
this.calendarioHTML;
this.nomeMateria;
this.giorno;
this.numGiorno;
this.mese1;
this.ore;
```

```
// variabile semaforo dell'oggetto Calendario
this.passa=false;
```

La variabile passa diventa true dopo che ho selezionato un'ora di
ripetizione

```
//coordinate dell'elemento selezionato
this.coord_i;
this.coord_j;
}
```

Il metodo "creaCalendario" crea dinamicamente il calendario
cioè genera il codice HTML necessario al Browser per
visualizzare la tabella che rappresenta il calendario

```
Calendario.prototype.creaCalendario= function() {
var content = '<table name="tabella" id="idTabella" >\n';
for(i=0;i<6;i++) {
// inizializza la 1° riga del calendario con le date
if(i==0){
```

i==0 indica la prima riga della tabella; a questa riga viene
assegnato il colore "bianco"e viene scritta, all'interno di ogni cella
da cui è formata la riga, la data di ogni giorno che compare nel
calendario

```
content += '<tr>';
```

```

for(j=0;j<14;j++){
    this.tab [i][j]="bianco";
    /*crea una nuova data che vale 1 oppure
    2 oppure 3 ecc giorni in più della data attuale (adesso)*/
    this.nuovaData = new Date();
    var nuovoGiorno = (j + this.d + 1);
    this.nuovaData.setDate(nuovoGiorno);
    this.gn = this.nuovaData.getDay();
    this.dn = this.nuovaData.getDate();
    this.mn = this.nuovaData.getMonth();
    var data = this.GIORNO[this.gn] + '<br>' + this.dn + '<br>' +
    this.MESE[this.mn];
    this.parola[i][j] = data;
}

```

Questa parte di codice serve a creare il calendario a partire dalla data odierna. Infatti a partire dalla data di oggi , viene creata una nuova data, incrementata di un numero di giorni pari al valore della variabile J del ciclo for, in modo da avere la data di oggi, domani, domani l'altro, e così via, fino ad arrivare al quattordicesimo giorno dopo quello attuale.

Le date così ottenuti sono scritte nella variabile `parola[i][j]` e contengono il giorno (scritto in lettere) la data (numero) e il mese (scritto in lettere) di quella data.

```

content += '<th class="'+ this.tab[i][j] + '"><div class=\"divTitolo\">'
          + this.parola[i][j] + '</div></th>' + '\n';
this.gio[j] = this.gn;
this.me[j] = this.mn;
this.ndg[j] = this.dn;
}
content += '</tr>';

```

Qui vengono create le restanti celle della tabella cioè quelle che non fanno parte dell'intestazione

La variabile `content` contiene la stringa, scritta in HTML, che permette la visualizzazione della tabella sul browser.

In particolare ad ogni cella che forma l'intestazione della tabella, viene assegnata una class per il colore di sfondo (colore assegnato tramite il valore della variabile `tab[i][j]`), dopodiché all'interno della cella viene creato un div, che serve a dare al titolo un carattere evidenziato rispetto al resto.

Fatto questo, all'interno della cella e del div, viene scritta la parola da visualizzare nella cella tramite la variabile `parola[i][j]`

```

}else{
content += '<tr>';
        //crea le altre righe della tabella
for(j=0;j<14;j++){
    idCella = i + "," + j;
    /* pongo ROSSE le domeniche e i sabati*/
    this.nuovaData = new Date();
    var nuovoGiorno = j + this.d + 1;
    this.nuovaData.setDate(nuovoGiorno);
    this.gn = this.nuovaData.getDay();
    if (this.GIORNO[this.gn]=="dom" || this.GIORNO[this.gn]=="sab") {
        this.tab[i][j]="rosso"; //rosso
    }
    /*singola cella della tabella: la classe dellacella mi è data da tab
    il contenuto della cella da parola */
    content += '<th class="'+ this.tab[i][j] + '" id="'+ idCella +
              '" onClick="c.prenota(this)">' + this.parola[i][j] + '</th>' + '\n';
        }
    content += '</tr>';
}
}
content += '</table>\n';
this.calendarioHTML = content;
}

```

Nelle celle relative ai giorni sabato e domenica, viene posto il colore di sfondo rosso. Per fare questo viene variato il valore di `tab[i][j]` da verde a rosso

Viene completata la variabile `content` con il resto delle celle da cui è formata la tabella calendario.

Rispetto alle celle che formano l'intestazione, viene tolto il div mentre viene aggiunto l'attributo `id` calcolato in una riga precedente del codice, che sarà utile in seguito per modificare la singola cellula.

Inoltre su queste celle, e solo su queste celle, è presente il metodo JavaScript

`c.prenota(this)` dove: `c` rappresenta un oggetto di tipo calendario, `prenota()` è il metodo considerato, `this` rappresenta l'oggetto corrente (cella del calendario) su cui è posto il cursore del mouse e il cui valore è passato al metodo `prenota`.

Questo metodo si attiva quando faccio click col pulsante sinistro del mouse su quella cella della tabella.

Il metodo `visualizza` permette di far vedere il calendario sul browser

```
Calendario.prototype.visualizza = function () {
    document.getElementById('idCalendario').innerHTML=this.calendarioHTML;
}
```

Il metodo `prenota` serve a effettuare la prenotazione cioè, a far cambiar colore e a scrivere all'interno della cella selezionata il nome della materia scelta dall'utente.
Per ottenere questo risultato, viene sostituito il calendario visualizzato nel browser con una nuova versione in cui la cella selezionata cambia il proprio contenuto

```
Calendario.prototype.prenota = function (elemento) {
    /*devo ricavare l'ID dell'elemento della tabella con cui faccio click con il mouse */
    for(i=0;i<6;i++){
        for(j=0;j<14;j++){
```

Devo determinare il valore delle coordinate I e J della cella della tabella calendario che è stato passato al metodo `prenota`.
Per fare questo, faccio una scansione di tutti gli indici delle celle che formano il calendario e creo una stringa di valore: i, j diversa per ogni cella del calendario, quindi confronto l'id della cella passata al metodo `prenota` con la stringa ottenuta sopra.
Quando i valori coincidono, ho trovato le coordinate I e J della cella passata al metodo `prenota` coordinate che vengono memorizzati nelle variabili `coord_i` `coord_j`

```
var stringa= i + "," + j;
if(elemento.id==stringa){
    /*memorizza i valori i e j dell'elemento selezionato*/
    this.coord_i=i;
    this.coord_j=j;

    /*seleziono l'elemento della tabella solo se è verde*/
    if(c.tab[i][j] == "verde"){
```

Se il colore della cella è verde effetto una serie di operazioni:
- memorizzo giorno, mese e numero del giorno della cella selezionata
- memorizzo in `parola[i][j]` il nome della materia prenotata dall'utente (dove I e J rappresentano le coordinate della cella selezionata dall'utente

```
/*faccio diventare l'elemento della tabella rosso*/
c.tab[i][j]="rosso";

/* giorno, mese, num del giorno, dell'ora selezionata */
this.giorno = this.GIORNO_COMPL[this.gio[j]];
this.numGiorno = this.ndg[j];
this.mese1 = this.MESE_COMPL[this.me[j]];
this.ore = this.parola[i][j];
```

```

/* memorizza in parola [i][j] il nome della materia scelta */
var indiceMateria=document.getElementById("idMateria").selectedIndex;
this.nomeMateria=
    document.getElementById("idMateria").options[indiceMateria].value;
c.parola [i][j]=this.nomeMateria;

```

```

/*creo la nuova tabella con l'elemento selezionato*/
var content1 = '<table name="tabella" id="idTabella" >\n';
for(i=0;i<6;i++) {

```

A questo punto , i dati di interesse sono memorizzati negli array di variabili `tab[i][j]` e `parola[i][j]`; per dare la sensazione di una modifica della tabella che rappresenta il calendario, viene in realtà visualizzato un nuovo calendario aggiornato con la casella selezionata

```

content1 += '<tr>';
if(i==0){
    content1 += '<tr>';
    for(j=0;j<14;j++){
        content1 += '<th class="' + this.tab[i][j] + '>'
                    + this.parola[i][j] + '\n';
    }
}
else{
    content1 += '<tr>';
    //crea le altre righe della tabella
    for(j=0;j<14;j++){
        //singola cella della tabella
        content1 += '<th class="' + this.tab[i][j] + '>'
                    + this.parola[i][j] + '\n';
    }
}
}
}

```

```

/*Visualizza il nuovo calendario*/
content1 += '</table>\n';
document.getElementById('idCalendario').innerHTML=content1;

```

La parte successiva del codice serve per visualizzare, sopra l'etichetta del bottone conferma, i dati relativi all'ora di prenotazione

```

/*Crea il messaggio data di prenotazione */
var messaggio='ciao ' + document.getElementById('idNome').value
    + ', hai prenotato un\'ora di ' + this.nomeMateria + ' '
    + this.giorno + ' ' + this.numGiorno + ' ' + this.mese1
    + ' alle ore ' + this.ore;
var str = '<label for="idSubmit">'+ messaggio + ' </label>';

```

```

/*Scrivi il messaggio data di prenotazione sull'etichetta posta sopra
il bottone conferma*/
document.getElementById('idBottoni').innerHTML=str;

```

La variabile `passa` viene posta a `true` quando si scrive per la prima volta il messaggio nell'etichetta sopra il pulsante conferma

```

    this.passa=true;
    /*ponendo passa= true si abilita il metodo corregginome che permette di
    correggere il nome se è scritto male*/
  }
}
}
pass=true;
}

```

La variabile `pass` viene posta a `true` quando è stata fatta la prenotazione

Il metodo `correggiNome` serve a cambiare il nome dell'utente nell'etichetta sopra il bottone conferma nel caso in cui l'utente cambi il proprio nome sul modulo di compilazione. Questo metodo si attiva solo se è stata fatta la prenotazione (`pass==true`) e se si è scritto il messaggio nell'etichetta sopra il pulsante conferma `this.passa==true`

```

Calendario.prototype.correggiNome = function (elemento) {

  /* il metodo è attivo solo se passa è true e pass è true,
  passa è true quando è abilitato il metodo cambia nome che permette di
  correggere il nome se è scritto male
  pass è true dopo che ho selezionato l'ora di ripetizione*/
  if((this.passa==true)&&(pass==true)){
    /*correggi il nome sopra il bottone conferma prenotazione */
    var messaggio='ciao ' + document.getElementById('idNome').value
      + ', hai prenotato un\'ora di ' + this.nomeMateria + ' '
      + this.giorno + ' ' + this.numGiorno + ' ' + this.mese1
      + ' alle ore ' + this.ore;
    var str = '<label for="idSubmit">'+ messaggio + ' </label>'
    document.getElementById('idBottoni').innerHTML=str;
  }
}

```

Il metodo `cancella` serve a cancellare la prenotazione sul calendario. Effettua le seguenti operazioni:

- cambia il colore della cella selezionata dall'utente da Rosso a verde,
- sostituisce il nome della materia con l'orario,
- visualizza l'oggetto calendario modificato
- cancella la parte del messaggio dell'etichetta sopra il bottone "conferma" che riguarda la data di prenotazione e lo sostituisce con la frase "devi scegliere il giorno e l'ora"
- pone la variabile `pass` a `false` (la prenotazione è stata cancellata)

```
Calendario.prototype.cancella = function () {
    this.tab[this.coord_i][this.coord_j]="verde";
    this.parola[this.coord_i][this.coord_j]=14-1+this.coord_i;
    c.visualizza();
    var messaggio='ciao ' + document.getElementById('idNome').value
        + ' devi scegliere il giorno e l\'ora';
    var str = '<label for="idSubmit">'+ messaggio + ' </label>';
    document.getElementById('idBottoni').innerHTML=str;
    pass=false;
}
```

Codice in HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<!-- miapagina.html -->
<html>

    <head>
        <title> Prenotazioni </title>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <link rel="stylesheet" type="text/css" href="stile.css" >
        <script type="text/javascript" src="./mioscript.js">
        </script>
        <script type="text/javascript">
            // gli oggetti gli creo nell'head per evitare side effect
            //(effetti collaterali)
            var s = new Studente();
            var c = new Calendario();
            c.creaCalendario();
        </script>
    </head>
```

Vengono creati un oggetto studente e un oggetto calendario. Il metodo `creaCalendario()` crea la stringa in codice HTML che rappresenta il calendario, ma non lo visualizza; la visualizzazione del calendario sarà effettuata dal metodo `visualizza()` posizionato nel punto del codice HTML dove effettivamente vogliamo far apparire il calendario

```

<body id="idCorpo">
  <h1> PRENOTAZIONE ORA DI RIPETIZIONE </h1>
  <form action="" id="idForm">

```

Viene creato il fieldset che contiene gli elementi per scrivere i dati dell'utente;
 il metodo `c.correggiNome(this)` serve a modificare il nome che compare nell'etichetta sopra il pulsante conferma se l'utente decide di correggere tale nome.
 Sono usati tag `<div>` e `` per creare l'effetto di una tabella senza usare il tag `<table>`

```

<fieldset class="delimita" >
  <legend>Inserisci i tuoi dati </legend>
  <div class="centra">
    <span >
      <label for="idNome"> Nome:</label><br>
      <input type="text" name="nome" id="idNome" size="25"
        onChange ="c.correggiNome(this)">
    </span>
    <span>
      <label for="idCognome"> Cognome:</label><br>
      <input type="text" name="cognome" id="idCognome" size="25" >
    </span>
    <span>
      <label for="idEmail"> E-mail:</label><br>
      <input type="text" name="email" id="idEmail" size="25">
    </span>
    <span>
      <label for="idTelefono"> Telefono:</label><br>
      <input type="text" name="telefono" id="idTelefono" size="25" >
    </span>
  </div>
</fieldset>

<fieldset class="delimita">
  <legend>Scegli la materia</legend>
  <div class="centra3">
    <label for="idMateria"> Scegli la materia di cui vuoi
      la ripetizione:</label><br>
    <select name="materia" id="idMateria" >
      <option id="idItaliano" value="Italiano">Italiano</option>
      <option id="idMatematica"
        value="Matematica">Matematica</option>
      <option id="idInglese" value="Inglese">
        Inglese</option>
      <option id="idInformatica"
        value="Informatica" >Informatica</option>
    </select>
  </div>
</fieldset>

```



```

<fieldset class="delimita">
  <legend>Seleziona giorno ed ora</legend>
  <div class="centra2">
    <label for="idCalendario"> Seleziona il giorno e l'ora
      in cui vuoi fare la ripetizione, poi clicca con il
      pulsante destro del mouse
    </label>
  </div>
  <div id="idCalendario">
    <script type="text/javascript">
      c.visualizza();
    </script>
  </div>
  <div >
    <input type="button" name="cancella" value="cancella"
      onclick="c.cancella();" >
  </div>
</fieldset>

```

visualizza il calendario
realizzato con javascript

Quando viene premuto il bottone conferma, si attiva il
metodo `cancella()`, che cancella la prenotazione
effettuata dall'utente sul calendario

```

<fieldset class="delimita">
  <legend>Conferma prenotazione</legend>
  <div id="idBottoni"></div>
  <div >
    <input type="button" id="idConferma" name="conferma"
      value="Conferma" onclick="s.creaStudente();
      s.controllaDati();" >
  </div>
</fieldset>
</form>

```

Quando faccio clicca sul pulsante "conferma" viene
richiamato il metodo `creaStudente()` dell'oggetto `s`
che associa alle variabili interne dell'oggetto
studente i contenuti presenti negli input del form
dopodiché viene richiamato il metodo
`controllaDati()` (stesso oggetto) che effettua il
submit cioè l'invio dei dati della forma al server, se i
dati inseriti dall'utente sono occorre

```

<div class="w3c">
  <span>
    <a href="http://validator.w3.org/check?uri=referer">
      
    </a>

    <a href="http://jigsaw.w3.org/css-validator/check/referer">
      
    </a>
  </span>
</div>

</body>
</html>

```

CSS

```
body {
  background-color: white;
  color: black;
}
```

```
div.centra {
  margin-left: 9%;
  width: 95%;
}
```

div.centra rappresenta il selettore: **div** è l'elemento HTML su cui viene applicato il CSS, **centra** rappresenta una classe **margin-left** è la proprietà **9%** è il valori che assume la proprietà

```
div.central {
  margin:auto;
  width: 20em;
}
div.centra2 {
  margin-left: 9%;
  width: 46em;
}
div.centra3 {
  margin-left: 9%;
  width: 20em;
}
```

```
th.verde:hover {
  background-color: #FFFF00;
  color: #191970;
}
```

hover:pseudo classe dinamica;quando passo con il mouse sopra questa cella, cambia il colore della cella che diventa gialla

```
h1 {
  background-color: #FFFFFF;
  color: #00008B;
  font: bold 1.4em Arial,Helvetica,sans-serif;
  text-align: center;
}
```

```
span {
  display: inline;
  float: left;
  padding: 6px;
  width: 16em;
}
```

Questo CSS serve per rendere mobile gli input text dove sono scritti i dati dell'utente.
In particolare la proprietà float permette di posizionare un oggetto a sinistra del contenitore, lasciando che tutto il resto del contenuto si posizioni alla destra.

```
input.txt {
  border: 1px inset #00008B;
  color: blue;
  width: 200px;
}
```

```

label {
  color: blue;
  font: bold 0.9em Arial,Helvetica,sans-serif;
  width: 40%;
}
fieldset.delimita {
  border: 1px dotted #61B5CF;
  margin-top: 0.6em;
  padding: 0.4em;
}
legend {
  background-color: #FFFFFF;
  color: #00008B;
  font: bold 1.1em Arial,Helvetica,sans-serif;
}
th {
  background-color: #E2E2E2;
  border: 1px solid black;
  color: #000000;
  /*font-weight: normal;*/
  margin: 0;
  padding: 0.2em 0.4em;
}

```

```

.bianco {
  background-color: #FFCD40;
  color: black;
}
.rosso {
  background-color: #FF1E00;
  color: black;
}
.verde {
  background-color: #68A447;
  color: black;
}

```

Colori delle varie celle che formano il calendario

```

table {
  margin: auto;
  width: 85%;
}

```

Dimensione della tabella che rappresenta il calendario
La proprietà `margin: auto` consente di centrare la tabella rispetto al contenitore.

```

.w3c {
  margin: auto;
  width: 10%;
}
.divTitolo{
  font: bold 1em Helvetica,sans-serif;
  min-width:2em;
}

```