

JQuiz5 Senza la possibilità di ritorno indietro e con aggiunta di informazioni a video sul risultato del test

Nelle pagine seguenti è riportato il codice di JQuiz5.ht_, con evidenziate in rosso le modifiche o aggiunte rispetto al codice originale fornito con HotPotatoes 5.4 e in verde alcuni commenti.

Il file JQuiz5.ht_, contenuto nella stessa cartella dove avete trovato questo file, è perfettamente funzionante e basta utilizzarlo mettendolo (basta solo lui) in una apposita sottocartella della cartella source di HotPotatoes e premendo Ctrl+Alt+Shift+S per utilizzarlo al momento della compilazione del codice (vedi note nella pagina Hot Potatoes tips di <http://www.batmath.it>). Naturalmente dovete scegliere la compilazione con la versione 5 dei browser, visto che il codice si riferisce a questa versione. Nonostante alcuni miglioramenti estetici della compilazione con la versione 6 dei browser, ci piace di più la struttura del file HTML generato con la versione 5.

Attenzione: questo codice funziona solo se attivate la funzione di avanzamento domande una per una e la possibilità di invio risultati tramite mail (anche se i risultati saranno poi solo inviati a video!). In ogni caso vedete le istruzioni e l'esempio proposto sul sito <http://www.batmath.it>.

Non abbiamo qui riportato tutto il codice, ma solo le parti che hanno subito modifiche, con alcune righe sopra e sotto per poterle individuare correttamente nell'allegato file JQuiz5.ht_. I puntini di sospensione in blu identificano le zone tagliate.

```
<html>
<head>
<title>

    ...
var FocusBoxes = true;
var Score = "";
var FinalScore = "";
var CFT1 = 0; //risposte corrette al primo tentativo
var Missing = 0;//risposte non date (premuto il tasto ShowAnswers)

var PreviousPage = history.length;

[inclSendResults]
var UserName = "";
var StartTime = ""; (inizio del test)
var Startmsec = 0; (inizio del test)
(tutto il resto della sezione è stato eliminato perché inutile ai fini della presente versione)

[/inclSendResults]

[strReadingText]

[Reading]
if (is.CRAPBrowser){ReadingText=ResolveAllURLs(ReadingText);}
[/Reading]

var NavBar=";

[inclNavButtons]
```

```

NavBar+='<tr><form name="NavButtons1" action=""><td
[inclNavBarColor]bgcolor="[strNavBarColor]" [/inclNavBarColor]align="center">';
    ...
TitleCode += '<font size="-1" face="[strFontFace]" color="[strTitleColor]">' + ExerciseSubtitle +
'<br />';
TitleCode += '</font></td></tr>';

var QButtonsOpener = '<table border="0" width="85%" cellpadding="10">';
QButtonsOpener += '<tr><td align="center" valign="middle"><form name="FormAvanti">';
QButtonsOpener += '<font size="+1">';
var QButtonsCloser = '<input type="button" name="Avanti" disabled=true value="=&gt;"'
onclick="parent.ChangeQ(1)"></input>';
QButtonsCloser += '</font></form></td></tr></table>';
(sezione modificata per eliminare il tasto di ritorno indietro e disabilitare il tasto di avanzamento fino al momento opportuno)

var TableOpener = '<table border="0" width="85%">';
var QRow = ";

    ...
function StartUp(){
[inclSendResults]
//Try to get user name
    UserName = prompt('[strNamePlease]', "");
    if ((UserName == null)|| (UserName.length<2)) UserName = 'Nominativo non introdotto';
(sezione modificata per evitare errori nel caso l'utente non introduca il nome)
[/inclSendResults]

//Inizializza le variabili per il tempo del test
    var today = new Date();
    StartTime = today.toLocaleString();
    Startmsec = today.getTime();

//Write the top frame
    WriteFeedback(Instructions);

    if ((is.mac)&&(is.ns)) {
        FocusBoxes = false;
    }

    ...
function CreateStatusArrays(){
    Status.length = 0;
    for (i=0; i<I.length; i++){
        Status[i] = new Array();
        Status[i][0] = 0; // question done or not
        Status[i][1] = 0; // wrong checks
        Status[i][2] = 0; // hints asked for
        Status[i][3] = ""; // student's answer
        Status[i][4] = 0; // score for this question
}

```

```

        Status[i][5] = false; //se è stato premuto o no ShowAnswers
        Status[i][6] = ";"//per inserire la prima risposta
    }
}

function ReplaceStuff(Token, Replacement, InString){
    var i = InString.indexOf(Token);

    ...

function CheckAnswer(ShowHint, QNum){

    //if the question's already correct, exit gracefully
    if ((Status[QNum][0] == 1)&&(!Status[QNum][5])){
        WriteFeedback('Hai già risposto a questa domanda!');
        (modifiche fatte per evitare che uno prema più volte il tasto di check)
        return;
    }
    if (Status[QNum][5]) {
        WriteFeedback('Hai già chiesto le risposte a questa domanda!');
        return;
    }

    ...

//Calculate scores
    if ((Finished==true)|| (ContinuousScoring==true)){
        var Score = CalculateScore();
        Output = Output + YourScoreIs + ' ' + Score + "%. ";
    }

    if (Finished == true){
        var CFT = 0;
        for (i=0; i<Status.length; i++){
            if ((Status[i][1] == 0)&&(Status[i][2] == 0)){
                CFT++;
            }
        }
        CFT1=CFT; (risposte corrette al primo tentativo)
        Missing=0;
        for (i=0; i<Status.length; i++){
            if (Status[i][5]){
                Missing++;
            }
        }
        } (per introdurre la possibilità di stampare il numero di risposte mancanti)
        Output += CorrectFirstTime + CFT + '/' + I.length;
        WriteFeedback(Output);
    [inclSendResults]

    ...
}

if (Status[QNum][0] == 1) {BottomFrame.FormAvanti.Avanti.disabled=false;}

```

```

}

function ShowAnswers(QNum){
//Se hai già chiesto altre volte le risposte, manda un messaggio
if(Status[QNum][5]) {
    WriteFeedback('Hai già chiesto le risposte un\'altra volta!');
    return;
}
//Se hai già risposto esci, con grazia;
if (Status[QNum][0] == 1) {
    WriteFeedback('Hai già risposto correttamente a questa domanda');
    return;
}
//Cambia il valore di Status[QNum][5]
Status[QNum][5] = true;

//If the question isn't done, mark it as done. Score will be 0;
Status[QNum][0] = 1;
Status[QNum][1] = -1;
Status[QNum][2] = -1;

//Get the correct answers and put them into a string
var Output = TheseCorrect + '<br />';

Output += I[QNum][1][0][0] + '<br />';

[inclAlsoCorrect]
for (var i=1; i<I[QNum][1].length; i++){
    Output += I[QNum][1][i][0] + '<br />';
}
[/inclAlsoCorrect]

//Check whether the quiz is finished
var Finished = true;
for (var i=0; i<Status.length; i++){
    if (Status[i][0] == 0){Finished = false;}
}

//Se non è finito manda un messaggio
if (!Finished) {
    Output += 'Non potrai più rispondere a questa domanda!<br />';
}

//Calculate scores
if ((Finished==true)||((ContinuousScoring==true))){
    var Score = CalculateScore();
    Output = Output + YourScoreIs + ' ' + Score + '%. ';
}

if (Finished == true){
    var CFT = 0;
    for (i=0; i<Status.length; i++){
}

```

```

                if ((Status[i][1] == 0)&&(Status[i][2] == 0)){
                    CFT++;
                }
            }
            CFT1 = CFT;
            Missing=0;
            for (i=0; i<Status.length; i++){
                if (Status[i][5]){
                    Missing++;
                }
            }
            Output += CorrectFirstTime + CFT + '/' + I.length;
        
```

[inclSendResults]

```

            setTimeout('SendResults(' + Score + ')', 50);

```

[/inclSendResults]

```

        }
        WriteFeedback(Output);
        if (Status[QNum][0] == 1) {BottomFrame.FormAvanti.Avanti.disabled=false;}
    }

```

(funzione quasi completamente modificata)

[Reading]

```

function ShowReading(){

    ...

```

[/inclTimer]

[inclSendResults]

```

function SendResults(Score){

    var today = new Date();
    var TestTime = Math.round((today.getTime()-Startmsec)/60000);

    var ResultForm = '<html><body>';
    ResultForm += '<p><font color="#0000ff"><b>Questo è il risultato del test che hai appena eseguito.</b></font>';
    ResultForm += '<p>Titolo del Test: [strExerciseTitle].</p>';
    ResultForm += '<p><b>Il tuo nome: '+UserName+'.</b></p>';
    ResultForm += '<p>Inizio del test: '+StartTime+'</p>';
    ResultForm += '<p>Durata del test: '+TestTime+' minuti.</p>';
    ResultForm += '<p>Punteggio parziale ottenuto: '+Score+'%.</p>';
    ResultForm += '<p>Il punteggio parziale tiene conto delle risposte date e delle eventuali correzioni apportate dopo la prima risposta, assegnando un punteggio via via decrescente all'aumentare dei tentativi.<p>';
    ResultForm += '<p>Risposte corrette al primo tentativo: '+ CFT1 + '/' + I.length +'.'</p>;
    ResultForm += '<p>Risposte non fornite: '+ Missing + '/' + I.length +'.'</p>;
    ResultForm += '<p>Risposte fornite al primo tentativo: </p><p>';
    for (i=0; i<I.length;i++){
        if (Status[i][6]==") Status[i][6]=*****";
        if ((Status[i][1] == 0)&&(Status[i][2] == 0)){
            ResultForm += 'Domanda n°: '+i+' Prevista: '+I[i][1][0]+'' - Fornita: '+Status[i][6]+'<br />';
        }
    }
}

```

```

        }
        if ((Status[i][1] != 0) || (Status[i][2] != 0)) {
            if (Status[i][5]) {
                ResultForm += '<font color="#0000ff">*Domanda n°: '+i+
Prevista: '+I[i][1][0][0]+ - Fornita: '+Status[i][6]+'.</font><font color="#ff0000"> Hai chiesto la
soluzione!</font><br />';
                }
                if (!Status[i][5]) {
                    ResultForm += '<font color="#0000ff">*Domanda n°: '+i+
Prevista: '+I[i][1][0][0]+ - Fornita: '+Status[i][6]+'.</font><br />';
                }
            }
        }
        ResultForm += '<p align="center"><a href="#"'
        onclick="window.close()">chiudi</a></p>';
        ResultForm += '</body></html>';

        newwin=window.open("",'risposta','toolbar=no,status=no,scrollbars=no,resizable=no,width=7
50,height=550');
        newwin.document.open();
        newwin.document.write(ResultForm);
        newwin.document.close();
    }
    (funzione quasi completamente riscritta)

[/inclSendResults]

function Blank(){
    return '<html><body>&copy;Half-Baked Software. Loading...</body></html>';
}

...
</html>

```