

Commento codice per realizzazione di elementi mostra nascondi con pulsanti toggle in tabella con checkbox (NON modalità random-casuale)

Codice html	Commento
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8" /> <title>tabella</title> <script type="text/javascript" src="jquery-1.11.3.js"></script> <script type="text/javascript" src="tabella_semplice_codice2.js"></script> <link rel="stylesheet" href="tabella_semplice_stili2.css" type="text/css" /> </head></pre>	<p>Il codice in rosso carica i file di...</p> <ol style="list-style-type: none">1. javascript della libreria di JQuery→ farà funzionare il sistema toggle (cioè pulsanti on/off);2. carica inoltre un altro file sempre di javascript realizzato da noi (.js);3. carica il file (.css) con gli stili realizzati da noi
<pre><body> <h1>Tabella per esercizio</h1> <h3>Studia la tabella attentamente, quindi <u>utilizza i pulsanti a sinistra</u>
 per nascondere/visualizzare il contenuto e controllare le tue conoscenze</h3>

</pre>	<p>Inizia il BODY con un titolo in dimensioni h1 e un sottotitolo in dimensioni h3</p>

Codice html	Commento
<pre> <table id="tabella" > <thead> <tr> <th></th> <th>Elementi da studiare</th> </tr> </thead> </pre>	<p>Inizia una tabella il cui nome Identificativo (id) è "tabella";</p> <p><thead> introduce, come prima riga, una intestazione di tabella... infatti di default la farà in grassetto.</p> <p>C'è una riga, che è compresa tra <tr> e </tr></p> <p>Dentro a questa riga ci sono tre colonne, che sono identificate da <th>→ non il semplice e classico <td></p> <p>perché sono colonne di intestazione</p> <p>La prima e terza colonna sono vuote, infatti c'è tag di apertura <th> e quello di chiusura </th> senza nulla in mezzo... la seconda colonna invece ha un titolo→<th>Elementi da studiare</th></p>
<pre> <tbody> <tr > <td><input type="checkbox" id="pul1"></td> <td id="riga1" class="vedo1">item 1</td> <td></td> </tr> </pre>	<p>Inizia il <tbody>, cioè il corpo della tabella.</p> <p>C'è una prima riga (compresa tra <tr> e </tr>) formata da tre colonne.</p> <p>La prima colonna ha inserito un checkbox <input type="checkbox"/>, il cui nome identificativo è "pul1"... il nome identificativo servirà per richiamare un'azione quando questo pulsante (e solo questo, non altri) viene selezionato <input checked="" type="checkbox"/>.</p> <p>La seconda riga ha come nome identificativo "riga1" (id="riga1") e una CLASS, che si chiama "vedo1"... una class è come un nome identificativo ma con la caratteristica che può eventualmente essere attribuita a più elementi, mentre un ID deve essere univoco ed essere attribuito ad un solo elemento... quindi una CLASS è più versatile... servirà per attribuire uno stile particolare alla cella, tramite i fogli di stile... in questa cella del contenuto→ "item 1" è il contenuto.</p> <p>L'ultima cella/colonna è vuota→ <td></td></p>

Codice html	Commento
<pre> <tr> <td><input type="checkbox" id="pul2"></td> <td class="vedo2" id="riga2">item 2 item 2 item2 item2 item 2 item 2 item 2 item2 </td> <td></td> </tr> </pre>	<p>Il codice della prima riga si ripete, quasi identico perché facciamo la seconda riga della tabella → sarebbe la terza se contiamo anche la riga di intestazione.</p> <p>Notare le uniche differenze:</p> <p>l'identificativo della checkbox adesso si chiama id="pul2"... dobbiamo cambiare nome ovviamente perché così potremo riferirci a questo pulsante e solo a questo... il suo nome deve essere diverso dalla precedente checkbox;</p> <p>l'identificativo della seconda riga è id="riga2" e anche la CLASS si chiama "vedo2" invece che "vedo1"...</p> <p>INSOMMA IN OGNI RIGA CAMBIEREMO GLI IDENTIFICATIVI e le CLASS</p>
<pre> <tr> <td><input id="pul3" type="checkbox"></td> <td id="riga3" class="vedo3">item 3 item3 item 3 </td> <td></td> </tr> </pre>	<p>Il codice si ripete, con le dovute modifiche dei nomi degli identificativi, per il numero di righe desiderato... se ci fermassimo qui avremmo TRE righe, ma se continuiamo con il codice possiamo avere il numero di righe che desideriamo...</p> <p>notare che nella cella centrale, oltre ad essere inserito del testo (item 3 item 3 item 3), è inserita anche una immagine → <code></code></p> <p>alt="castello" sta per → testo alternativo, cioè, se per qualche motivo l'immagine non fosse visibile o ci fosse uno screenreader per non vedenti, al posto dell'immagine viene visualizzata/letta la scritta "castello"; l'allineamento è in mezzo/centrato (align="middle")</p>

Codice html	Commento
<pre><tr> <td><input id="pul7" type="checkbox"></td> <td id="riga7" class="vedo7">item 7</td> <td></td> </tr> </tbody> </table> </body> </html></pre>	<p>Chiusura della tabella, dopo aver inserito il numero di righe desiderato →</p> <pre></tbody> </table></pre> <p>Chiusura del codice html</p> <pre></body> </html></pre>

Codice del file di javascript creato da noi e che in parte richiama elementi dalla libreria di JQuery

Codice javascript → file .js	Commento
<pre>\$(document).ready(function(){ /*-----*/</pre>	<p>Si inizia il codice di javascript sempre con questa riga di codice che fondamentalemente impartisce l'ordine di implementare le istruzioni sottostanti solo quando il documento è stato completamente caricato → DOCUMENTO READY, cioè "pronto"</p> <p>Notare la parentesi graffa aperta... dovrà essere chiusa in fondo al documento... tutte le istruzioni staranno tra questa prima parentesi graffa di apertura e l'ultima proprio in fondo... insomma si tratta di una SCATOLA CONTENITORE di tutto il codice</p> <p>/*-----*/ è un separatore che ho inserito nel codice solo per separare le diverse sezioni e poter leggere meglio il codice → NON influisce sul corretto funzionamento del codice perché è un COMMENTO...</p> <p>I Commenti in javascript si mettono tra i simboli /* commento commento commento e */</p>
	<p>Il file di javascript si scrive con il BLOCCO NOTE, come si fa con i file degli stili, ma quando si salva si mette, dopo il nome, l'estensione .js (punto js)... nel nostro caso il file si chiama tabella_semplice_codice2.js</p>

Codice javascript → file .js	Commento
<pre data-bbox="163 201 797 568"> /*-----*/ \$(function(){ \$('#pul1').click(function(){ \$(".vedo1").toggleClass("nonvedo1"); }); }); /*-----*/ </pre>	<p data-bbox="819 201 2029 320">C'è una funzione di JQuery → <code>\$(function(){</code> Il simbolo \$ richiama proprio JQuery... questa funzione viene chiusa con ultimo <code>});</code>;</p> <p data-bbox="819 344 1962 464">In mezzo ci sta un'altra funzione che viene attivata quando si fa click sulla checkbox che si chiama pul1... se vi ricordate, il nome ID si richiama mettendo davanti il simbolo #</p> <p data-bbox="819 488 1984 743"><u>Questa funzione svolge la seguente azione: modifica la CLASS che si chiama "vedo1" e la sostituisce con la CLASS che si chiama "nonvedo1"</u> → sono nomi che ho inventato io e che ho "settato" nel file degli stili css (lo analizziamo dopo)... cmq anticipiamo... alla classe "vedo1" ho settato un attributo di opacity =1 (che è il massimo, quindi massima visibilità... 100%).</p> <p data-bbox="819 751 2007 831">Alla classe "nonvedo1" ho settato un attributo di opacity=0 (trasparenza totale, quindi invisibile).</p> <p data-bbox="819 839 1984 1007">L'attributo di opacity, a differenza di quello che usavamo precedentemente, mantiene la cella al suo posto, ma non si vede!!! Quindi evitiamo quel problema della cella che, scomparendo, "apre un buco" e quindi ogni elemento si sposta di un posto a sinistra.</p> <p data-bbox="819 1031 2029 1190">ToggleClass svolge la funzione, implementata da JQuery, di verificare in che stato si trova il pulsante... se è attiva la classe "vedo1" la sostituisce con "nonvedo1" e viceversa... cioè il controllo lo fa JQuery con codice già scritto nel file di JQuery.</p> <p data-bbox="819 1214 2018 1430">Ultima cosa da notare → <code>\$(".vedo1").toggleClass("nonvedo1");</code> Il punto, che ho messo in rosso, viene utilizzato al posto del simbolo # <u>quando ci si riferisce al nome di una classe...</u> in ("nonvedo1") il punto NON c'è perché il codice toggle sostituisce semplicemente il nome della classe... il punto c'è già nel codice, quindi non si deve mettere.</p>

Codice javascript → file .js	Commento
<pre> /*-----*/ \$(function(){ \$('#pul2').click(function(){ \$(".vedo2").toggleClass("nonvedo2"); }); }); /*-----*/ </pre>	<p>Il codice si ripete identico per il numero di righe che abbiamo messo nella tabella... 7 righe, 7 pulsanti checkbox, 7 porzioni di codice su toggleClass... notare che unica differenza è data dai nomi del pulsante “pul2” e delle classi “vedo2” e “nonvedo2”... mi raccomando... intendo dire che si apre una volta sola <code>\$(document).ready(function){</code></p> <p>poi al suo interno si ripete la porzione di codice <code>\$(function){</code></p> <pre> \$('#pul2').click(function(){ \$(".vedo2").toggleClass("nonvedo2"); }); }); </pre> <p>per quante volte è necessario... poi si chiude con un'ultima <code>});</code> (vedi sotto)</p>
<pre> }); </pre>	<p>ricordarsi di chiudere con una <code>});</code>;</p> <p>si tratta della graffa e rotonda che chiude la prima riga di codice... se vi ricordate...</p> <pre> \$(document).ready(function){ </pre>

Codice del file .css che gestisce gli stili del documento

Codice css → file .css	Commento
<pre>.vedo1{ opacity: 1; } .nonvedo1{ opacity: 0; } .vedo2{ opacity: 1; } .nonvedo2{ opacity: 0; } .vedo3{ opacity: 1; } .nonvedo3{ opacity: 0; }</pre>	<p>Il codice va ripetuto con le modifiche dei nomi per il numero di righe che abbiamo messo in tabella... 7 righe, 7 pulsanti checkbox... allora 7 volte ripetuto il codice... qui mi fermo a 3</p> <p>Il punto iniziale richiama il nome di una classe (quindi NON dimenticarlo), come il # richiama il nome di un id.</p> <p>Per il resto il codice l'ho già spiegato precedentemente... opacity=1 → visibilità totale... opacity=0 → visibilità nulla... ovviamente in altri casi, NON in questo, se volessimo ottenere una semi trasparenza potremmo settare la opacity ad un valore intermedio → .6 (punto 6) sta per 60%... .2 (punto 2) sta per 20% e così via.</p>
	<p>Il file .css si scrive con il BLOCCO NOTE; quando si salva si mette, dopo il nome, l'estensione .CSS (punto css)... nel nostro caso il file si chiama tabella_semplice_stili2.css</p>

Codice css → file .css	Commento
<pre>#tabella{ width: 1000px; position: absolute; top: 200px; z-index:1; }</pre>	<p>I settaggi degli altri elementi del file sono molto semplici... non meritano particolari commenti... qui per la tabella abbiamo stabilito la larghezza (width) la posizione assoluta, quindi da specificare (dall'alto 200px → top: 200px;) per default, se non si specifica nulla, viene posizionata completamente a sx.</p> <p>z-index è la posizione nello spazio, cioè il livello... più alto è il numero, più in alto viene posizionato l'elemento... per capirci... un elemento con z-index=0 sta sotto a un elemento con z-index=1... se i due elementi si trovano nella stessa identica posizione, quindi sovrapposti, si vedrà solo z-index=1 perché z-index=0 sta sotto.</p>
<pre>td, th { background-color: white; font-family: arial, verdana, sans serif; border: solid 1px; /*border-collapse: collapse;*/ border-color: blue; border-spacing: 5px; text-align: left; <!-- spacing inserisce spazio tra il bordo di celle adiacenti/--> }</pre>	<p>Settaggi per le celle... colori dello sfondo (white), tipo di carattere, dimensioni, colore e tipo di bordo, famiglia del font</p> <p>Questa riga di codice <code>/*border-collapse: collapse;*/</code> è “commentata”, cioè in pratica resa “non funzionante” → provate a cancellare <code>/*</code> e <code>*/</code> per vedere cosa accade</p>

Se volessimo ottenere un effetto ulteriore, cioè quello di cliccare sulla checkbox per nascondere gli item da studiare, ma nello stesso tempo volessimo che comparisse, al posto del testo nascosto, una domanda guida... dovremmo utilizzare un piccolo trucchetto: mettere una tabella sopra l'altra... più in alto (sopra) la tabella con gli item da nascondere... sotto la tabella con le domande guida; poiché, utilizzando il pulsante checkbox, rendiamo invisibile la cella della tabella, si potrà vedere la cella della tabella che sta sotto con la domanda guida... esempio:

tabella (sta sopra)

<input type="checkbox"/>	Cristoforo Colombo	
<input type="checkbox"/>	Item 2	
<input type="checkbox"/>	Item 3	
<input type="checkbox"/>	Item 4	
<input type="checkbox"/>	Item 5	
<input type="checkbox"/>	Item 6	

tabella1 (sta sotto)

<input checked="" type="checkbox"/>	Navigatore genovese	
<input type="checkbox"/>		
<input checked="" type="checkbox"/>	Domanda guida su item 3	
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		

Selezionando le checkbox della prima e terza riga, queste diventano trasparenti (opacity=0) per cui si intravede la tabella sottostante solo per le righe 1 e 3.

Per ottenere questo risultato dovremo fare i seguenti ulteriori passaggi, rispetto al codice precedente:

1. nel codice html inserire il codice per avere una seconda tabella; il suo id potrebbe essere id="tabella1"; in questa tabella inserire le domande guida
2. nel codice degli stili css inserire...

```
#tabella1{  
width: 1000px;  
position: absolute;  
top: 200px;  
z-index:0;  
}
```

Notare che "tabella1" ha stesse dimensioni e stessa posizione di "tabella" però

tabella avrà **z-index:1**

tabella1 avrà **z-index:0**

per cui tabella sta sopra e nasconde tabella1

Inoltre dobbiamo dare a tabella un ulteriore settaggio → **background-color: transparent;**

codice CSS completo di tabella e tabella1 nella prossima pagina

Codice css → file .css	Commento
<pre>#tabella{ background-color: transparent; width: 1000px; position: absolute; top: 200px; z-index:1; }</pre>	<p>Il background-color deve essere settato a “transparent” perché mi sono accorto che, anche se rendiamo invisibili le singole celle (opacity=0), si continua a vedere il bianco dello sfondo della tabella per cui non si riesce a intravedere la tabella sottostante...</p> <p>con background-color: transparent invece si riesce a vedere la tabella che sta sotto</p>
<pre>#tabella1{ width: 1000px; position: absolute; top: 200px; z-index:0; }</pre>	