TECNOLOGIE WEB

Esame del 13/06/2022 – Compito in Presenza  
Tempo a disposizione: 2 ore

**REGOLE.**

* Per l’esame potete accedere solamente a:

localhost

<http://achecker.csr.unibo.it/checker/index.php> (per validare l’accessibilità)

<https://validator.w3.org/> (per validare il documento HTML)

<https://www.w3schools.com/>

* L’accesso ad altri siti è disabilitato.
* Avete anche disponibile una versione di w3schools locale nel disco remoto dove avete trovato il testo del compito.
* Come editor di testo, è possibile scegliere tra Visual Studio Code e Notepad++

**IMPORTANTE**:

**lavorate all’interno della cartella della consegna, e consegnate TUTTI i file delle soluzioni (non solo il doc del compito).**

|  |
| --- |
| ESERCIZIO N. 1 **(7 punti)** |

Scrivere il codice HTML5 accessibile e semanticamente corretto per realizzare un documento che contenga una form (con metodo GET) per l’acquisto di libri di testo, strutturata in gruppi di campi come segue:

* + Un gruppo di campi per la selezione dei titoli: “Sistemi operativi. Concetti ed esempi, di Abraham Silberschatz et al.”, “Linguaggi di programmazione. Principi e paradigmi, di Maurizio Gabbrielli e Simone Martini”, “I moderni sistemi operativi, di Andrew Tanenbaum”, “Fondamenti di reti di calcolatori, di Andrew Tanenbaum”;
  + Un gruppo di campi per il tipo di formato: “copertina flessibile”, “copertina rigida”, “formato elettronico”;
  + Un gruppo di campi per la consegna: nome, cognome, via, numero civico, città, provincia, CAP, orario flessibile (con radio button);
  + Un campo corrispondente al pagamento alla consegna;
  + Un bottone di Submit e un bottone di Annulla.

Il documento deve essere HTML5 valido e deve essere accessibile secondo le WCAG2.0 a livello A (la validazione con tool automatici dell’accessibilità non è di per sé sufficiente).

|  |
| --- |
| ESERCIZIO N. 2 **(6 punti)** |

**CSS**

Dato il file html relativo all’esercizio N. 1, realizzare il file .css (esterno), tenendo in considerazione quanto segue:

* Tutti i font devono avere lo stesso font-family, che deve essere Arial.
* I testi dei legend devono avere colore di sfondo Orange e colore del testo Dark Red. I bordi sono arrotondati e il testo è allineato al centro. I box devono avere una ombreggiatura di colore gray con spostamento in alto e a destra pari a 5px e una sfocatura pari a 10px.
* La form deve avere le seguenti caratteristiche:
  + Colore di sfondo Yellow.
  + Bordo color Orange, spessore 2px, di tipo solid.
  + I bordi devono essere arrotondati. L’arrotondamento deve essere espresso in pixel e deve essere pari all’ultima cifra del vostro numero di matricola. Se questa cifra è 0 allora l’arrotondamento deve essere pari a 10 pixel.
  + La form deve avere una ombreggiatura di colore gray con spostamento in alto e a destra pari a 10px e una sfocatura pari all’ultima cifra del numero di matricola. Se questa cifra è 0 allora la sfocatura deve essere pari a 10 pixel.
* Il testo delle label deve essere di colore nero, dimensione pari al 120%.
* I bottoni devono avere sfondo Orange, con testo di colore Black e dimensione 150%. I bottoni devono avere i bordi arrotondati (l’arrotondamento deve essere espresso in pixel e deve essere pari all’ultima cifra del numero di matricola. Se questa cifra è 0 allora l’arrotondamento deve essere pari a 10 pixel).
* I fieldset hanno il bordo color Dark Red.

|  |
| --- |
| ESERCIZIO N. 3 **(5 punti)** |

Descrivere brevemente le principali novità introdotte da CSS3.

|  |
| --- |
| ESERCIZIO N. 4 **(7 punti)** |

**JQuery**

L’esercizio è da svolgere insieme a quello di Php

Dato il file html ‘esercizio\_javascript.html', creare il file ‘soluzione.js’ JavaScript o jQuery in modo tale che:

* Al caricamento della pagina:
  + Dovranno essere nascosti il form, i tag span e il bottone Valuta Soluzione.
* Al click sul bottone “Nuova partita” si dovrà:
  + Fare una richiesta alla pagina index.php. Alla risposta del server, creare una tabella 9\*9, le cui celle compilata secondo quanto restituito dal server.
  + Visualizzare il form
  + Mostrare il form, rimuovendo l’eventuale contenuto dei tag di input.
* Al click sul bottone “Aggiungi” si dovrà:
  + Leggere gli input riga, colonna e valore.
  + Controllare che riga, colonna e valore siano compresi tra 1 e 9 ed in caso contrario visualizzare un messaggio di errore.
  + Aggiungere il valore nella tabella.
* Al click sul bottone “Valuta Soluzione” si dovrà:
  + Leggere i numeri della tabella contenuto della tabella.
  + Inviarli al server
  + In base alla risposta, visualizzare uno dei due span.
* **NB: NON SONO AMMESSE MODIFICHE AL FILE HTML**

|  |
| --- |
| ESERCIZIO N. 5 **(7 punti)** |

**PHP**

L’esercizio è da svolgere insieme a quello di jQuery

Scrivere il codice PHP valido (ovvero che esegua correttamente su server web Apache) che consenta di:

* Creare una nuova partita, aggiungendo la relativa riga nel database e restituendo lo stato iniziale come json. Lo stato inziale deve essere una stringa di 81 caratteri composta da 8 numeri tra 1 e 9 mentre i restanti elementi devono essere 0. La soluzione inziale deve essere valida (lo stesso numero non può essere presente nella stessa riga, colonna o quadrato). L’id della partita va salvato in un cookie.
* Controllare che la soluzione inviata dall’utente in post sia:
  + Sia relativo alla partita corrente.
  + Valida: in ogni riga, colonna e quadrato (3\*3) siano presenti i numeri da 1 a 9.
  + Rispetti lo stato iniziale della soluzione.
  + Restituire poi il risultato sotto forma di json.
* **Dovete supporre che il db esista (nome database: esami, username: root, pw: ) e che sia strutturato secondo le istruzioni che trovate nel file "README\_DB.txt".**