

TECNOLOGIE WEB
Esame del 17/02/2023 – Compito A
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 valido, ben formato, accessibile e semanticamente corretto per realizzare un documento che contenga una intestazione di primo livello “Prenotazione Volo” e un form diviso in due parti:

- La prima parte, dal titolo “Dati Anagrafici”, deve consentire i seguenti input:
 - Nome, Cognome, Data di nascita, Nazionalità.
- La seconda parte, dal titolo “Dati Volo”, deve consentire i seguenti input:
 - Aeroporto di partenza, aeroporto di destinazione, numero di passeggeri (massimo consentito: 5), Data di partenza
 - Un insieme di controlli per consentire la scelta tra viaggio andata e ritorno e viaggio di sola andata.
 - Un insieme di controlli per consentire la scelta tra: economy, premium economy, business, prima classe.
 - Uno o più controlli per consentire l'indicazione di date flessibili.
- Bottoni di Submit e Cancella della form

Il documento deve essere HTML5 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. La dimensione deve essere del 100%.
- Il form occupa il 90% della larghezza della pagina, ed ha un padding laterale sinistro pari al 5%.
- Il form ha un bordo di tipo dashed, di colore Dark orange, larghezza 5px.
- I fieldset hanno testo di colore dark orange, un bordo solid, di colore Dark orange, larghezza 2 px.
- I fieldset hanno una ombreggiatura orientata a sinistra e in alto, con un offset di 5px, una sfocatura di 10px, di colore Orange.
- I controlli di testo hanno uno sfondo di colore dark orange, con colore del testo white.
- Le label hanno colore del testo dark orange.
- I buttoni hanno colore di sfondo dark orange e colore del testo white, in grassetto. Sull'hover, colore di sfondo e di foreground si scambiano.

L'uso delle media-query non è richiesto.

ESERCIZIO N. 3 (5 punti)

Descrivere in che cosa consiste il concetto di “cascading” dei fogli di stile.
Scrivere la soluzione su un file .txt

ESERCIZIO N. 4 (7 punti)

Javascript

Dato il file html esercizio_javascript.html in allegato, modificare il file ‘soluzione.js’ utilizzando JavaScript in modo tale che:

- Al click sul bottone Leggi Personaggio si dovrà:
 - Leggere il contenuto dell’input e controllare che sia un numero intero positivo.
 - Fare una richiesta GET al file personaggio.json
 - In caso di successo, visualizzare i dati del personaggio sotto forma di elenco non ordinato nel main.
- Al click sul bottone Leggi Personaggi si dovrà:
 - Fare una richiesta POST al file personaggi.json
 - In caso di successo, visualizzare i dati dei personaggi sotto forma di tabella, considerando che:
 - il nome deve essere una cella di intestazione
 - la tabella deve essere accessibile.

NB: NON SONO AMMESSE MODIFICHE AL FILE HTML

ESERCIZIO N. 5 (7 punti)

PHP

Scrivere il codice PHP valido (ovvero che esegua correttamente su server web Apache) che consenta di inserire un nuovo personaggio di Harry Potter o di ritornare l'insieme dei personaggi salvati su database (in formato JSON), in base al tipo di richiesta.

In questa pagina occorrerà quindi:

Controllare anzitutto se la richiesta sia di tipo POST o GET.

- Nel caso la richiesta fosse di tipo POST:
 - controllare che le variabili inviate (name, species, gender, house, dateOfBirth, yearOfBirth, ancestry, eyeColour, hairColour) siano settate e non nulle
 - inserire il nuovo personaggio all'interno del database
 - restituire il personaggio appena creato
 - in caso di errore, restituire un messaggio di errore.
- Nel caso la richiesta fosse di tipo GET:
 - controllare che sia passato il campo id:
 - Se non presente, selezionare tutti i personaggi e restituirli in formato JSON (esempio in data.json).
 - Altrimenti se presente, restituire il personaggio con id passato come input, se NON presente restituire un messaggio di errore.
- Dovete supporre che il db esista (nome database: esami; nome tabella: harrypotter; username: root, pw:) e che la tabella "harrypotter" sia strutturata e riempita secondo le istruzioni che trovate nel file "README_DB.txt".
- La pagina PHP deve SEMPRE restituire un JSON valido