

TECNOLOGIE WEB
Esame del 29/01/2026
Tempo a disposizione: 2 ore

REGOLE.

Per l'esame potete accedere solamente a:

- localhost
- <https://achecker.csr.unibo.it/achecker/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).**
- **I file su cui lavorerete dovranno chiamarsi come il vostro nome utente (se la vostra mail unibo è *marcoantonio.deluigi23@studio.unibo.it* allora il vostro esercizio javascript si chiamerà *marcoantonio.deluigi23.js*)**
- **Per consegnare dovete creare un unico file zip con i file all'interno della cartella della consegna.**

ESERCIZIO N. 1 (7 punti)

HTML

Scrivere il codice HTML5 accessibile e semanticamente corretto per realizzare la tabella che segue, con caption “Conference Program”:

TIME/Conference Room	RENO	LAUGHLIN	VIRGINIA CITY	CARSON CITY
08:15 - 9:30	Keynote Speech			
09:30 - 10:00	Networking & Coffee Break			
10:00 - 12:00	Track 1	Track 9	Track 2	Work-in-Progress 2
12:00 - 13:00	Lunch			
13:00 - 15:30	Track 6	Track 5	Work-in-Progress 3	Work-in-Progress 4
15:30 - 16:00	Networking & Coffee Break			
16:00 - 18:00	Panel 1		Panel 2	

Gli aspetti presentazionali relativi al bordo della tabella non sono necessari, ma potrebbero esservi utili per verificare meglio lo span di alcune celle, se necessario. Per questo motivo potete usare eventualmente CSS inline.

Il documento deve essere HTML5 valido e deve essere accessibile secondo le WCAG 2.1 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%.
- La tabella occupa il 90% della larghezza della pagina, ed ha un padding laterale sinistro pari al 5%.
- La tabella ha un bordo di tipo double, di colore Dark Green, larghezza 5px.
- La tabella ha una ombreggiatura orientata a destra e in basso, con un offset di 10px, una sfocatura di 5px, di colore Green.
- Le celle di intestazione hanno background Dark Green, il testo è di colore White, il bordo è nero, di tipo solid, larghezza 1 px.
- Le celle di contenuto hanno background White, il testo è di colore Dark Green, il bordo è Green, di tipo solid, larghezza 1 px.
- Le celle di contenuto con i seguenti valori: "Keynote Speech", "Networking & Coffee Break", "Lunch", si differenziano dalle altre celle di contenuto per quanto riguarda il background che deve essere di colore Green, con opacity pari a "0.4".

L'uso delle media-query non è richiesto.

ESERCIZIO N. 3 (5 punti)

Domanda

Descrivere brevemente i passaggi e gli scambi di comunicazione che avvengono tra browser e server Web dal momento in cui viene digitato o cliccato un indirizzo Web al momento in cui la corrispondente pagina Web viene renderizzata dal browser.

NB: Consegnare la risposta in un file txt

ESERCIZIO N. 4 (7 punti)

Javascript

Considerando il file **esercizio_javascript.html**, sviluppare il codice JavaScript necessario a implementare le seguenti specifiche:

1. Caricare il contenuto del file **studenti.json** al caricamento della pagina.
2. Creare dinamicamente una tabella HTML contenente l'elenco completo degli studenti, visualizzando per ciascuno tutte le informazioni presenti nel file JSON. La tabella va aggiunta come un ulteriore figlio del tag **main**.
3. Popolare dinamicamente il menu a tendina con id **corsoSelect** con l'elenco dei corsi di laurea presenti nel file JSON, evitando duplicazioni.
4. Al cambiamento del valore selezionato nel menu a tendina, la tabella deve essere aggiornata mostrando esclusivamente gli studenti iscritti al corso selezionato.
5. Se l'opzione “Tutti i corsi” è selezionata, la tabella deve mostrare nuovamente l'elenco completo degli studenti.

La manipolazione della tabella e del menu deve avvenire tramite accesso e modifica del DOM, senza ricaricare la pagina.

L'unica modifica al file html consentita è il link al file javascript che dovrà chiamarsi come riportato nelle istruzioni all'inizio del compito.

ESERCIZIO N. 5 (7 punti)

PHP

Scrivere il codice PHP che consenta di registrare una misurazione per un sensore oppure di recuperare dati aggregati (in formato JSON).

Nel caso la richiesta fosse di tipo **POST**:

- Controllare che i parametri `sensor_id`, `value`, `timestamp` siano settati e non nulli.
- Verificare che `value` sia numerico.
- Controllare che il sensore indicato da `sensor_id` esista nella tabella `sensors`.
- Inserire la misurazione nella tabella `measurements`.
- Restituire in JSON la misurazione appena salvata (almeno `sensor_id`, `value`, `timestamp`). Se il sensore non esiste o i dati non sono validi, restituire un JSON con chiave `error`.

Nel caso la richiesta fosse di tipo **GET**:

- Se il campo `sensor_id` è presente: calcolare e restituire `min`, `max` e `media` dei valori registrati per quel sensore.
- Se non esistono dati per la richiesta, restituire un JSON di errore.

La pagina PHP deve SEMPRE restituire un JSON valido.

Dovete supporre che il db esista (nome database: `iot_db`; nome tabelle: `sensors` e `measurements`; username: `root`, pw:) e che la tabella sia strutturata secondo le istruzioni che trovate nel file “README_DB.txt”