

Esercitazione 1

JS & DOM

25 novembre 2025

Introduzione

La seguente esercitazione consiste nello sviluppare un progetto di Registro Elettronico personale. In particolare, grazie a questo Registro Elettronico, è possibile salvare una lista di tutti i voti acquisiti, simulando, ovviamente in maniera semplificata, le funzioni di un reale Registro Scolastico.

Si utilizzi il codice nella cartella "Esercitazione 1 - JS&DOM" come versione iniziale su cui sviluppare il resto del progetto.

Attività 1: aggiungere i voti

Come prima cosa è necessario che il Registro Elettronico consenta all'utente di aggiungere i propri voti: modificare la versione iniziale del progetto per fare in modo che il "Form" per l'aggiunta dei voti funzioni correttamente. In particolare, è necessario:

- Aggiungere un *listener* per catturare l'evento di click sul bottone di aggiunta "add-grade-btn".
- Implementare una funzione per convalidare i dati inseriti negli input, chiamata quando l'utente clicca il bottone per aggiungere un nuovo voto. Verificare che:
 - Il campo "subject" non sia vuoto;
 - Il campo "test-info" non sia vuoto;
 - Il voto sia compreso tra 3 e 10.

Una volta superati questi controlli, la logica per l'aggiunta del voto può continuare. Se qualche controllo fallisce, stampare all'interno dell'elemento "error-box" un messaggio di errore, ovviamente senza aggiungere il nuovo voto.

Attenzione: se i controlli sono superati è necessario svuotare "error-box"

per evitare che un ipotetico errore rimanga visualizzato a schermo a seguito di una corretta aggiunta di un nuovo voto.

- Il nuovo voto viene salvato in una variabile di tipo *Object*, che racchiude tutti i voti salvati seguendo questa struttura:

```
[
  {
    "subject": "TPS",
    "gradeInfo": "Verifica su JS&DOM",
    "score": 7
  },
  {
    "subject": "TPS",
    "gradeInfo": "Verifica su NodeJS",
    "score": 8
  },
  { ... }
]
```

- Dopo l'aggiunta, stampare a console tutti i voti separatamente su varie righe (Materia: ..., Info: ..., Voto: ...).

Attività 2: stampare i voti

Dopo aver implementato la logica per aggiungere nuovi voti, è necessario stamparli tutti nel DOM, quindi:

- A seguito di ogni aggiunta, stampare i voti dentro il listato “*user-grades*”.

Attività 3: classificazione dei voti

La terza attività si occupa di UI&UX (User Interface & User Experience). L'obiettivo è:

- Implementare una funzione “*classifyScore(score)*” che prende in input una variabile *score* contenente un voto (controllare come prima cosa che contenga effettivamente un valore numerico; in caso contrario terminare l'esecuzione della funzione) e restituisce in output il nome di una classe in base al valore del voto:
 - **score < 5**
→ Restituisce la classe “*.red*”
 - **score ≥ 5 && score < 6**
→ Restituisce la classe “*.yellow*”

- **score ≥ 6 && score < 10**
→ Restituisce la classe “.green”
- **score = 10**
→ Restituisce la classe “.blue”
- Durante la stampa dei voti nel DOM, chiamare la funzione appena implementata e assegnare la classe ottenuta al rispettivo elemento di classe “grade-value”; in questo modo è possibile associare dei colori a tutti i voti salvati e mostrati.

Attività 4: media dei voti

Aggiungere un ascoltatore al bottone “average-grade” in modo che, quando viene cliccato dall’utente, il codice calcoli la media attuale di tutti i voti e la mostri all’interno del botton, al posto della scritta “MEDIA”.

Si applichino inoltre le stesse classi di colore ottenute tramite la funzione implementata nell’attività precedente.

Attività bonus: persistenza dei dati

Il progetto, nel suo stato attuale, cancella dalla memoria tutti i voti salvati se si ricarica la pagina. Per evitare questo problema, si utilizzi il localStorage tramite JavaScript per salvare tutti i voti nella memoria del browser. È inoltre necessario caricare opportunamente i valori dallo storage invece di inizializzare l’Object da 0 ogni volta.