

|                        |                                      |
|------------------------|--------------------------------------|
| <b>Iniziato</b>        | mercoledì, 28 gennaio 2026, 11:19    |
| <b>Stato</b>           | Completato                           |
| <b>Terminato</b>       | mercoledì, 28 gennaio 2026, 11:51    |
| <b>Tempo impiegato</b> | 32 min. 34 secondi                   |
| <b>Punteggio</b>       | 8,30/20,00                           |
| <b>Valutazione</b>     | <b>41,50</b> su un massimo di 100,00 |

**Domanda 1**

Risposta errata

Punteggio ottenuto 0,00 su 1,00

Qual è la funzione dello Shadow DOM nei Web Components?

Scegli un'alternativa:

- ☐ a. Nascondere completamente il componente dalla vista dell'utente
- ☐ b. Creare un mini-DOM protetto a cui non si applicano script, stili e eventi del documento principale
- ☒ c. Duplicare il DOM principale per migliorare le performance ✖
- ☐ d. Generare automaticamente CSS per il componente

La risposta corretta è: Creare un mini-DOM protetto a cui non si applicano script, stili e eventi del documento principale

**Domanda 2**

Risposta errata

Punteggio ottenuto 0,00 su 1,00

Si consideri il seguente frammento HTML: `<ul> Testo libero <li>Primo elemento</li> <li>Secondo elemento</li> </ul>` Quale affermazione descrive correttamente il comportamento del browser?

Scegli un'alternativa:

- ☐ a. Il browser rimuove il testo poiché non valido
- ☐ b. Il testo è figlio della lista ma non è parte semantica della lista, che è composta solo da elementi
  -
- ☒ c. Secondo il modello HTML living standard, il testo converte automaticamente il testo in un ✖
  - esplicito
- ☐ d. Il browser genera un errore di parsing bloccando la visualizzazione della lista

La risposta corretta è: Il testo è figlio della lista ma non è parte semantica della lista, che è composta solo da elementi

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Un client invia per errore la stessa richiesta PUT tre volte consecutive allo stesso URI. Secondo il principio di idempotenza REST, cosa dovrebbe accadere?

Scegli un'alternativa:

- ☐ a. Il server dovrebbe rigettare la seconda e terza richiesta con un errore 409 Conflict.
- ☐ b. Ogni richiesta dovrebbe creare una nuova versione della risorsa.
- ☐ c. Il server dovrebbe sommare gli effetti delle tre richieste.
- ☒ d. Il risultato finale sul server dovrebbe essere identico a quello di una singola richiesta PUT ✓

La risposta corretta è: Il risultato finale sul server dovrebbe essere identico a quello di una singola richiesta PUT

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Perché una callback AJAX non restituisce un valore alla funzione chiamante?

Scegli un'alternativa:

- ☐ a. Perché AJAX è sincrono
- ☐ b. Perché JavaScript non supporta return
- ☐ c. Perché il browser lo vieta
- ☒ d. Perché viene eseguita dopo la fine del flusso principale ✓

La risposta corretta è: Perché viene eseguita dopo la fine del flusso principale

**Domanda 5**

Risposta errata

Punteggio ottenuto 0,00 su 1,00

Quali sono i protocolli che Node.JS può gestire?

Scegli un'alternativa:

- ☐ a. Node.js può unicamente gestire HTTP
- ☐ b. Node.js può gestire protocolli diversi rispetto ad HTTP a patto che vengano porte well-known
- ☐ c. Node.js può implementare anche protocolli diversi rispetto ad HTTP
- ☒ d. Node.js può implementare unicamente i protocolli HTTP e HTTPS ✗

La risposta corretta è: Node.js può implementare anche protocolli diversi rispetto ad HTTP

**Domanda 6**

Completo

Punteggio ottenuto 0,20 su 1,00

Un e-commerce ti fornisce un array di oggetti prodotto (es. [{name:"Penna", price:1.5, qty:2}, ...]). Spiega, a parole, come useresti map, filter e reduce per: (1) ottenere solo i prodotti con qty > 0, (2) costruire l'elenco dei nomi da mostrare in pagina, (3) calcolare il totale (somma di price \* qty). Perché questi metodi possono rendere il codice più chiaro rispetto a un ciclo tradizionale?

Questi elementi rendono più leggibile il codice rispetto ad un ciclo in quanto sono istruzioni di alto livello, e quindi con solo delle chiamate a funzione posso trovare ciò che mi è stato richiesto. Userei tali funzioni per andare a trovare i dati che mi interessano e li salverei all'interno di variabili locali, per poi andare a calcolare ciò che mi interessa.

Commento:

**Domanda 7**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Perché lo stato applicativo in HTTP deve stare nel client?

Scegli un'alternativa:

- ☒ a. Il server non mantiene contesto ✓
- ☐ b. I cookie sono obbligatori
- ☐ c. Il server è più veloce
- ☐ d. È imposto da REST

La risposta corretta è: Il server non mantiene contesto

**Domanda 8**

Completo

Punteggio ottenuto 0,40 su 1,00

Si descriva la funzionalità di un form HTML. A cosa servono, cosa sono le ACTION e quali elementi li costituiscono

I form HTML, delineati dall'apposito tag <form>, sono dei componenti specifici per andare a realizzare tutte quelle parti di pagina per fare login, ad esempio. Il form ha anche la possibilità di definire quale sarà l'azione HTTP da compiere in fase di submit.

Commento:

**Domanda 9**

Risposta errata

Punteggio ottenuto 0,00 su 1,00

Un byte 10xxxxx trovato come primo byte utile in UTF-8 indica:

Scegli un'alternativa:

- ☐ a. Un flusso malformato
- ☐ b. Un ideogramma
- ☐ c. Un carattere ASCII
- ☒ d. Un carattere Latin-1 ✖

La risposta corretta è: Un flusso malformato

**Domanda 10**

Risposta errata

Punteggio ottenuto 0,00 su 1,00

Un elemento ha position: sticky ma non rimane "appiccicato" durante lo scroll. Qual è una causa corretta secondo lo standard?

Scegli un'alternativa:

- ☐ a. Sticky funziona solo se l'elemento è dentro un contenitore con display: flex.
- ☒ b. Sticky funziona solo se si imposta esplicitamente display: block sull'elemento sticky. ✖
- ☐ c. Serve sempre z-index esplicito, altrimenti sticky non si attiva.
- ☐ d. Manca un valore non-auto per almeno un inset (es. top) sull'asse in cui deve diventare sticky.

La risposta corretta è: Manca un valore non-auto per almeno un inset (es. top) sull'asse in cui deve diventare sticky.

**Domanda 11**

Risposta errata

Punteggio ottenuto 0,00 su 1,00

Con object-fit: cover su un <img>, quale comportamento è corretto?

Scegli un'alternativa:

- ☐ a. L'immagine resta sempre interamente visibile (letterboxing) mantenendo l'aspect ratio.
- ☐ b. L'immagine riempie completamente il contenitore mantenendo l'aspect ratio, e può essere ritagliata (clipped) se necessario.
- ☐ c. object-fit: cover riduce automaticamente la dimensione (in KB) dell'immagine scaricata per adattarla al contenitore.
- ☒ d. L'immagine viene stirata per riempire il contenitore senza mantenere l'aspect ratio. ✖

La risposta corretta è: L'immagine riempie completamente il contenitore mantenendo l'aspect ratio, e può essere ritagliata (clipped) se necessario.

### Domanda 12

Risposta errata

Punteggio ottenuto 0,00 su 1,00

Qual è il valore di `res`? `const res = [1,2,3].forEach(x => x * 2);`

Scegli un'alternativa:

- ☐ a. `undefined`, perché `forEach` non restituisce un nuovo array (serve per side-effect).
- ☐ b. Un "iteratore" che produce i valori raddoppiati quando viene consumato.
- ☒ c. `[2,4,6]`, perché la callback raddoppia gli elementi. ❌
- ☐ d. `[1,2,3]`, perché `forEach` restituisce l'array originale.

La risposta corretta è: `undefined`, perché `forEach` non restituisce un nuovo array (serve per side-effect).

### Domanda 13

Completo

Punteggio ottenuto 0,70 su 1,00

Si descrivano i metodi di Update: `updateOne`, `updateMany` e `replaceOne` in MongoDB. A cosa servono e quali sono le loro differenze? Per ciascuno si mostri un esempio pratico di implementazione.

I metodi update servono per andare a modificare dei valori all'interno del database. `updateOne` serve per aggiornare un singolo elemento passando il filtro all'elemento e il nuovo valore che vogliamo impostare, `updateMany` serve per fare la stessa cosa di `updateOne` ma lo fa su più elementi (ad esempio, `updateOne` posso farlo per modificare il saldo del conto corrente di un singolo utente a fronte di un bonifico mentre `updateMany` posso usarlo per scalare il numero di prodotti in un magazzino a seguito di un acquisto multiplo). `replaceOne` invece serve per andare a sostituire un elemento specifico con un altro elemento specifico nuovo senza crearlo da 0.

Commento:

### Domanda 14

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In UTF-16, quando viene usata una coppia di surrogate?

Scegli un'alternativa:

- ☐ a. Per ASCII
- ☐ b. Per lettere accentate
- ☐ c. Per ideogrammi BMP
- ☒ d. Per caratteri fuori dal BMP ✔️

La risposta corretta è: Per caratteri fuori dal BMP

### Domanda 15

Risposta errata

Punteggio ottenuto 0,00 su 1,00

Cosa restituisce una query SPARQL quando viene eseguita su un dataset RDF?

Scegli un'alternativa:

- ☐ a. Un documento JSON-LD completo
- ☒ b. Un nuovo grafo RDF contenente solo triple ✖
- ☐ c. Tabelle di dati con le variabili selezionate, non triple RDF
- ☐ d. Un file XML con tutti i metadati

La risposta corretta è: Tabelle di dati con le variabili selezionate, non triple RDF

### Domanda 16

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è la differenza tra window.location e window.history in un'applicazione JavaScript client-side?

Scegli un'alternativa:

- ☐ a. History può caricare nuove risorse HTTP nel browser, mentre location no
- ☐ b. Sia location che history sono alias dello stesso oggetto, dovuti alla standardizzazione di EcmaScript.
- ☐ c. Location si occupa di leggere l'URL mentre history lo modifica
- ☒ d. Location modifica l'URL corrente eseguendo il redirect, mentre history gestisce l'array degli URL visitati durante la navigazione. ✔

La risposta corretta è: Location modifica l'URL corrente eseguendo il redirect, mentre history gestisce l'array degli URL visitati durante la navigazione.

### Domanda 17

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Considera: `console.log("A"); setTimeout(()=>console.log("B"),0); console.log("C");` Quale ordine di output è corretto?

Scegli un'alternativa:

- ☒ a. A, C, B ✔
- ☐ b. C, A, B
- ☐ c. B, A, C
- ☐ d. A, B, C

La risposta corretta è: A, C, B

**Domanda 18**

Risposta errata

Punteggio ottenuto 0,00 su 1,00

Nel contesto del Semantic Web, ogni tripla RDF ha quale caratteristica in termini di ragionamento?

Scegli un'alternativa:

- ☒ a. È automaticamente collegata a tutte le altre triple dello stesso dataset ✖
- ☐ b. Rappresenta un'unità atomica di ragionamento senza connessioni implicite con altre triple
- ☐ c. Eredita sempre le proprietà del soggetto
- ☐ d. Deve essere validata da un'ontologia OWL per essere valida

La risposta corretta è: Rappresenta un'unità atomica di ragionamento senza connessioni implicite con altre triple

**Domanda 19**

Risposta errata

Punteggio ottenuto 0,00 su 1,00

In un URI come `https://example.com/pagina?lang=it#sezione2`, quale componente viene gestita solo dal client e NON viene inviata al server durante il recupero della risorsa?

Scegli un'alternativa:

- ☐ a. Il path `/pagina`, perché il server identifica la risorsa solo tramite host e porta.
- ☒ b. Lo schema `https`, perché è solo un'indicazione per il browser e non influisce sulla richiesta. ✖
- ☐ c. La query `?lang=it`, perché serve solo a "filtrare" localmente la pagina nel browser.
- ☐ d. Il fragment `#sezione2`, perché viene separato prima della dereferenziazione ed è usato dal user agent.

La risposta corretta è: Il fragment `#sezione2`, perché viene separato prima della dereferenziazione ed è usato dal user agent.

**Domanda 20**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Perché `Array.from(NodeList)` è spesso necessario?

Scegli un'alternativa:

- ☐ a. Per rendere la lista immutabile
- ☐ b. Per clonare il DOM
- ☒ c. Per poter usare i metodi degli array ✔
- ☐ d. Per migliorare il parsing

La risposta corretta è: Per poter usare i metodi degli array

Vai a...

