

Started on	Thursday, 12 February 2026, 3:18 PM
State	Finished
Completed on	Thursday, 12 February 2026, 4:14 PM
Time taken	55 mins 31 secs
Marks	18.56/20.00
Grade	92.80 out of 100.00

Question 1

Correct

Mark 1.00 out of 1.00

Si consideri il seguente frammento HTML:

```
<a href="pagina.html">  
<div>Sezione cliccabile</div>  
</a>
```

Secondo il modello HTML Living Standard, questo frammento è valido?

Select one:

- ☐ a. Sì, perché <div> viene convertito in inline
- ☒ b. Sì, perché l'unico vero vincolo di <a> rimasto è il divieto di <a> annidati ✓
- ☐ c. No, essendo <a> inline non può contenere elementi block
- ☐ d. Sì, perché <a> è diventato un elemento di blocco

The correct answer is: Sì, perché l'unico vero vincolo di <a> rimasto è il divieto di <a> annidati

Question 2

Correct

Mark 1.00 out of 1.00

Perché setTimeout(fn, 0) non esegue fn immediatamente?

Select one:

- ☐ a. Perché blocca il browser
- ☐ b. Perché 0 è invalido
- ☒ c. Perché viene messo in coda dopo lo stack corrente ✓
- ☐ d. Perché serve un thread

The correct answer is: Perché viene messo in coda dopo lo stack corrente

Question 3

Correct

Mark 1.00 out of 1.00

Quale delle seguenti affermazioni sui database NoSQL, come MongoDB, è corretta?

Select one:

- ☒ a. Un database NoSQL è non relazionale, dove i dati sono organizzati in strutture con schema flessibili ✓
- ☐ b. Un database NoSQL è un database relazionale con una struttura che cambia in base alla tipologia dei dati salvati
- ☐ c. Un database NoSQL è un database non relazionale in cui i dati vengono salvati solo in formato chiave-valore.
- ☐ d. Un database NoSQL è un classico DBMS che viene interrogato con un linguaggio diverso da SQL.

The correct answer is: Un database NoSQL è non relazionale, dove i dati sono organizzati in strutture con schema flessibili

Question 4

Correct

Mark 1.00 out of 1.00

Perché nel pipelining HTTP/1.1 le risposte devono essere ordinate?

Select one:

- ☐ a. Gli header sono compressi
- ☐ b. Il server è stateless
- ☒ c. Il client non può riordinarle ✓
- ☐ d. È una scelta arbitraria

The correct answer is: Il client non può riordinarle

Question 5

Correct

Mark 1.00 out of 1.00

Cosa succede se una funzione JavaScript viene chiamata senza un parametro previsto?

Select one:

- ☐ a. Il parametro vale null
- ☐ b. Si genera un errore
- ☒ c. Il parametro vale undefined ✓
- ☐ d. La funzione non viene eseguita

The correct answer is: Il parametro vale undefined

Question 6

Complete

Mark 1.00 out of 1.00

Node.js è noto per la sua modalità di esecuzione. Cosa si intende per "Single-thread non-blocking"? Si descrivano le tecniche consigliate per gestire il main loop.

Node.js è un tool incaricato di gestire il lato server di applicazioni web. Una caratteristica peculiare di Node.js risiede nel fatto di essere single thread, per questo è necessario adottare tecniche precise per non rendere bloccanti le operazioni che deve svolgere. La soluzione è proprio gestire le chiamate tramite un main loop, attraverso cui è possibile fare programmazione parallela apparente, in pratica le funzioni sono eseguite tutte quante un po' alla volta. Questo meccanismo permette quindi di non bloccare l'applicazione, e di gestire migliaia di funzioni in contemporanea. Per la gestione asincrona delle chiamate poi esistono vari modi, tra cui l'utilizzo di callback (quando una funzione finisce la propria esecuzione comunicherà questo esternamente), l'uso di async/wait oppure di yield/next che semplificano il caso di callback hell (molte funzioni di callback che si susseguono).

Comment:

Question 7

Incorrect

Mark 0.00 out of 1.00

Si considerino queste regole CSS (nessuna usa !important): `@layer base { p { color: red; } }` e `p { color: blue; }` (fuori da layer). Quale colore avrà un `<p>`?

Select one:

- ☒ a. Rosso, perché un layer dichiarato per primo ha sempre priorità sulle regole non in layer. ❌
- ☐ b. Blu, perché le regole non inserite in alcun layer hanno priorità sulle regole dentro layer (a parità di origine e importanza).
- ☐ c. Rosso, perché @layer aumenta automaticamente la specificità dei selettori contenuti.
- ☐ d. Rosso, perché le regole dentro un layer vengono trattate come se avessero !important implicito.

The correct answer is: Blu, perché le regole non inserite in alcun layer hanno priorità sulle regole dentro layer (a parità di origine e importanza).

Question 8

Correct

Mark 1.00 out of 1.00

Quale vantaggio principale offre async/await rispetto ai callback tradizionali in Node.js?

Select one:

- ☐ a. Elimina l'event loop.
- ☐ b. I callback non gestiscono errori.
- ☐ c. È sempre più veloce.
- ☒ d. Evita il callback hell rendendo il codice asincrono più leggibile. ✔

The correct answer is: Evita il callback hell rendendo il codice asincrono più leggibile.

Question 9

Correct

Mark 1.00 out of 1.00

Quale metodo produce una checkbox inizialmente NON selezionata?

Quale markup produce una checkbox inizialmente NON selezionata?

Select one:

- ☒ a. Omettere completamente l'attributo checked nell'elemento `<input type="checkbox">`. ✓
- ☐ b. Usare `checked="0"`.
- ☐ c. Usare `checked="false"`.
- ☐ d. Usare `checked="no"`.

The correct answer is: Omettere completamente l'attributo checked nell'elemento `<input type="checkbox">`.

Question 10

Correct

Mark 1.00 out of 1.00

In Turtle, come vengono indicati i blank nodes?

Select one:

- ☐ a. Con virgolette doppie `" "` oppure con il simbolo `*`
- ☐ b. Con parentesi tonde `()` oppure con il simbolo `@`
- ☐ c. Con parentesi graffe `{ }` oppure con il simbolo `#`
- ☒ d. Con parentesi quadre `[]` oppure con URI fittizi che iniziano con underscore `"_:"` ✓

The correct answer is: Con parentesi quadre `[]` oppure con URI fittizi che iniziano con underscore `"_:"`

Question 11

Complete

Mark 0.80 out of 1.00

Si descriva il concetto di falsy e truthy per i valori non booleani. Quali sono i pro e i contro?

Nel linguaggio javascript ci sono valori che sono definiti falsi o veri, così da omettere ambiguità. Alcuni esempi sono: 0, null, undefined, false, "", definiti sempre falsi, altri "0", "null", "false", true, 1 che sono sempre definiti veri. L'utilità è lampante, definirli rigidamente non crea dubbio sulla valutazione di essi. In aggiunta questi meccanismi permettono una grande versatilità nel codice, ad esempio per controllare che una stringa sia vuota non serve lanciare una funzione da librerie esterne, o implementate internamente, che controlli la lunghezza della stringa e successivamente fare un check se la lunghezza sia maggiore di 0, basta fare:

if string --> allora la stringa non sarà vuota

else --> allora la stringa risulta vuota

Comment:

Question 12

Correct

Mark 1.00 out of 1.00

Quale delle seguenti affermazioni descrive correttamente il concetto di "stateless" nel contesto delle API REST?

Select one:

- ☐ a. Le risposte del server non possono includere header di caching perché potrebbero creare uno stato implicito.
- ☐ b. Le API REST devono sempre usare cookie per mantenere lo stato tra client e server.
- ☒ c. Il server non mantiene informazioni sullo stato della sessione client tra richieste successive; ogni richiesta deve contenere tutte le informazioni necessarie. ✓
- ☐ d. Il client non può mantenere uno stato locale e deve richiedere lo stato al server ad ogni interazione.

The correct answer is: Il server non mantiene informazioni sullo stato della sessione client tra richieste successive; ogni richiesta deve contenere tutte le informazioni necessarie.

Question 13

Correct

Mark 1.00 out of 1.00

Quale delle seguenti affermazioni descrive correttamente il rapporto tra React, JSX e Babel?

Select one:

- ☐ a. Babel e JSX sono dipendenze obbligatorie per eseguire React
- ☐ b. Babel è il framework di React
- ☐ c. JSX è un linguaggio eseguito direttamente dal browser usando Babel per identificare moduli React
- ☒ d. JSX è una sintassi estesa usata da React e viene trasformata in JavaScript standard tramite Babel ✓

The correct answer is: JSX è una sintassi estesa usata da React e viene trasformata in JavaScript standard tramite Babel

Question 14

Correct

Mark 1.00 out of 1.00

Per far sì che text-overflow: ellipsis mostri "..." su una singola riga di testo, quali altre condizioni sono tipicamente necessarie?

Select one:

- ☐ a. Serve overflow: visible così il testo esce e viene sostituito automaticamente.
- ☐ b. Basta impostare text-overflow: ellipsis; il browser tronca comunque senza altre regole.
- ☒ c. L'elemento deve avere overflow limitato (es. overflow: hidden), white-space: nowrap e una larghezza (o max-width) che causi overflow. ✓
- ☐ d. È necessario impostare anche un line-height fisso.

The correct answer is: L'elemento deve avere overflow limitato (es. overflow: hidden), white-space: nowrap e una larghezza (o max-width) che causi overflow.

Question 15

Complete

Mark 0.86 out of 1.00

Si descrivano il DOMElement e i suoi metodi utilizzabili per navigare il DOM.

Il DOM è l'interfaccia che permette di interagire con la pagina HTML. Tra le molte categorie di DOM esiste il DOMElement che permette la comunicazione con gli elementi della pagina. Inizialmente l'accesso agli elementi era iterativo tramite l'uso di "children", ma ad oggi è stata snellita questa pratica attraverso l'uso di molti metodi, tra cui:

getElementById(id elemento)

getElementsByTagName(nome elemento se esistente)

getElementsByTagName(tag elemento)

getElementsByClass(classe elemento)

Comment:

incompleto

Question 16

Correct

Mark 1.00 out of 1.00

Perché FEFF può essere usato come BOM ma FFFE no?

Select one:

- ☐ a. È casuale
- ☐ b. FEFF è ASCII
- ☐ c. UTF-8 lo richiede
- ☒ d. FFFE è un codice proibito ✓

The correct answer is: FFFE è un codice proibito

Question 17

Correct

Mark 1.00 out of 1.00

Quale affermazione descrive correttamente lo scope delle variabili del JavaScript moderno?

Select one:

- ☐ a. var, let e const hanno lo stesso scope
- ☐ b. const non ha scope
- ☒ c. let e const hanno scope di blocco, var dipende dal contesto ✓
- ☐ d. Tutte sono globali

The correct answer is: let e const hanno scope di blocco, var dipende dal contesto

Question 18

Correct

Mark 1.00 out of 1.00

Un parser UTF-8 incontra un byte iniziale 11110xxx. Cosa può dedurre?

Select one:

- ☐ a. È Latin-1
- ☒ b. Il carattere occupa quattro byte ✓
- ☐ c. È ideografico BMP
- ☐ d. È un byte di continuazione

The correct answer is: Il carattere occupa quattro byte

Question 19

Correct

Mark 1.00 out of 1.00

Qual è la differenza tra il carattere escaped %2F e il carattere / quando utilizzato nella componente path di un URI?

Select one:

- ☐ a. Il carattere / viene normalizzato automaticamente dal browser
- ☒ b. %2F viene interpretato come carattere letterale, mentre / ha significato gerarchico ✓
- ☐ c. Nessuna, i caratteri escape vengono sempre decodificati solo lato client
- ☐ d. Il carattere %2F, a differenza del carattere /, è sempre vietato negli URI HTTP

The correct answer is: %2F viene interpretato come carattere letterale, mentre / ha significato gerarchico

Question 20

Complete

Mark 0.90 out of 1.00

Si descriva che cosa si intende per RDF, a cosa serve e quali sono i vantaggi del suo utilizzo. Si presenti inoltre un esempio di tripla RDF.

L'RDF è un concetto fondamentale nel Semantic Web e si riferisce al modo di raggruppare sintagmi relativi a livello semantico. Il raggruppamento avviene tramite triple composte da Soggetto, Predicato e Oggetto. Ad esempio, "Matteo mangia la pasta", può essere rappresentato da una tripla RDF composta da "Matteo" come soggetto, collegato a "la pasta" (l'oggetto) tramite il verbo "mangia". È proprio questo **collegamento** in triple, che rappresenta in modo chiaro e flessibile, ma perfetto per una gestione di dati a grafo, la base del grande risultato finale che si ottiene a seguito di raggruppamenti per Tesauri, per e tra Ontologie, che porta ad un enorme grafo di collegamenti semantici tra (molti) dati.

Comment:

esempio in linguaggio naturale

Jump to...