JavaScript Quest 1

Total puncte 84/100

A sosit momentul primului test de cunoștințe. Prin acest quest, vei verifica informațiile pe care le-ai adunat până acum.

- Quest-ul este format din 31 întrebări de tip grilă.
- Fiecare întrebare are între 3 și 5 puncte.
- Ai nevoie de cel puţin 50 de puncte pentru a nu pierde o viaţă.
- · Ouest-ul nu este cronometrat.
- Poți trimite răspunsurile tale o singură dată. Odată trimise ele nu pot fi modificate.

Termenul limită pentru acest quest este 16 iunie 2025, ora 19:00.

Adresă de e-mail *	
antonio.soare.42005@gmail.com	

Quiz 84 din 100 puncte

Ce este event delegation?	3/3
	Ce este event delegation?

- npărțirea unui eveniment în mai multe funcții
- Adăugarea aceluiași event listener la mai multe elemente
- Adăugarea unui event listener la un element părinte pentru a gestiona evenimente de la copii
- Stergerea automată a event listeners

Feedback

Event delegation înseamnă să atașezi un event listener la un element părinte care gestionează evenimentele pentru toți descendenții.



Cum previi comportamentul implicit al unui eveniment?	3/3
event.stopPropagation()	
event.preventDefault()	✓
event.stopDefault()	
event.cancel()	
Feedback	
preventDefault() oprește comportamentul implicit al unui eveniment (ex: submit-ul unui form, click pe link).	



X Care dintre următoarele condiții va executa codul din interior?

0/3

```
if (0) {
   console.log('Primul caz')
}

if ('') {
   console.log('Al doilea caz')
}

if ('false') {
   console.log('Al treilea caz')
}

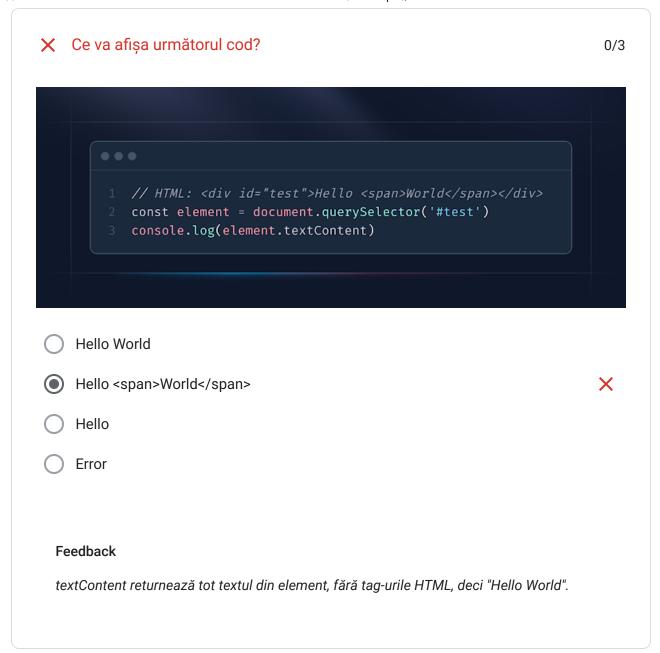
if (-1) {
   console.log('Al patrulea caz')
}
```

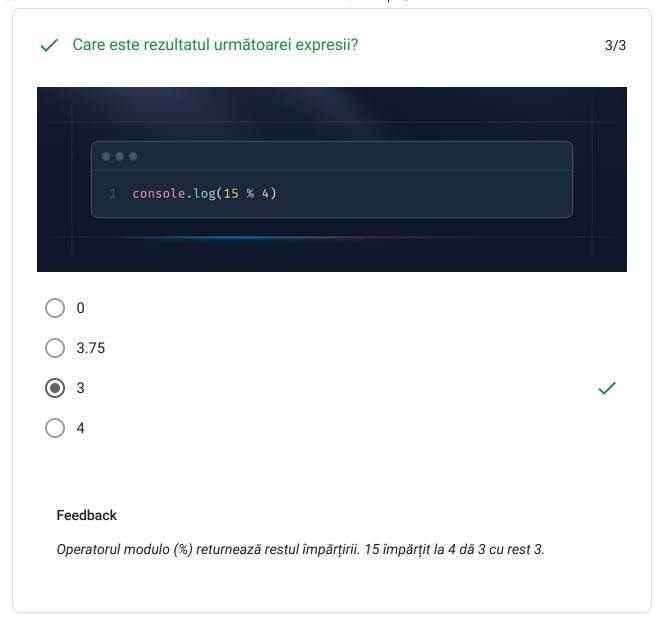
- Primul și al doilea caz
- Al treilea și al patrulea caz
- O Doar al patrulea caz
- Toate cazurile

X

Feedback

În JavaScript, valorile "falsy" sunt: false, 0, ", null, undefined, NaN. String-ul 'false' și numărul -1 sunt "truthy".





6/13/25, 9:56 PM

✗ Ce problemă poate apărea în următoarea funcție?

0/3

X



- Sintaxa este incorectă
- Funcția nu returnează nimic
- Metoda toUpperCase() nu există
- Nu verifică dacă parametrul text este valid

Feedback

Funcția ar genera eroare dacă text este undefined, null sau string gol, deoarece încearcă să acceseze text[0] fără verificare.

✓ Ce se întâmplă dacă folosești await în afara unei funcții async?	3/3
Funcţionează normal	
Blochează thread-ul principal	
Generează o SyntaxError	✓
Se convertește automat la .then()	
Feedback await poate fi folosit doar în interiorul funcțiilor async sau la nivel de modul (top-level await în module ES6), altfel generează o eroare de sintaxă.	
✓ Care este cel mai bun mod de a verifica dacă o variabilă este un array?	3/3
Array.isArray(variable)	✓
variable instanceof Array	
typeof variable === 'array'	
variable.constructor === Array	
Feedback Array.isArray() este metoda recomandată pentru verificarea array-urilor, funcționând co în toate contextele.	prect



✓ Ce face Promise.all()?	4/4
Execută doar primul Promise	
Execută Promise-urile unul după altul	
Returnează primul Promise care se finalizează	
Execută Promise-urile în paralel şi aşteaptă ca toate să se finalizeze	✓
Feedback Promise.all() execută mai multe Promise-uri în paralel și așteaptă ca toate să se fina cu succes.	lizeze
Care este modul corect de a declara o variabilă în JavaScript pentru a stoca o valoare care se poate schimba?	3/3
var name = "John"	
let name = "John"	✓
const name = "John"	
variable name = "John"	
Feedback let este recomandată pentru variabile care își pot schimba valoarea, deoarece are blo scope și comportament predictibil în comparație cu var.	ock



Care dintre următoarele nume de variabile NU este valid în JavaScript?	3/3
\$price	
1stName	✓
userName	
category	
Feedback	
Numele variabilelor nu pot începe cu o cifră în JavaScript. Pot începe cu litere, underscor (_) sau simbolul dollar (\$).	e

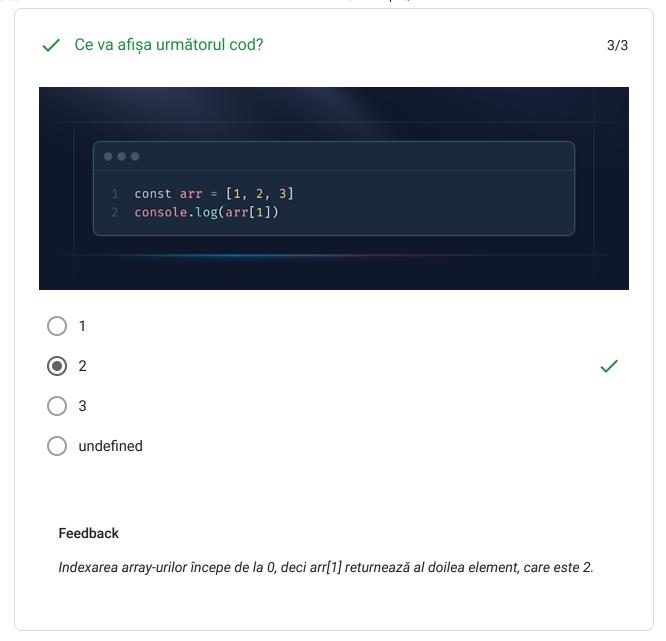
✗ Ce va afiṣa următorul cod în consolă?

0/3

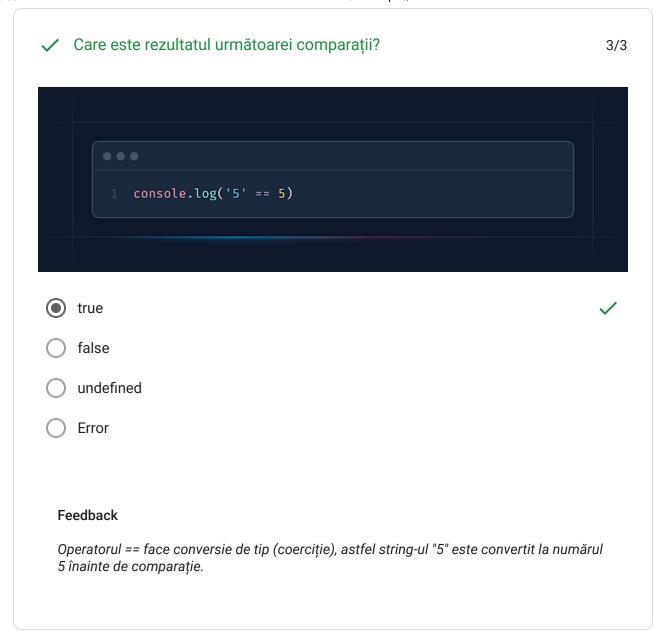
- Ambele vor afışa "Text vizibil Text ascuns Text invizibil"
- textContent: "Text vizibil", innerText: "Text vizibil Text ascuns Text invizibil"
- textContent: "Text vizibil Text ascuns Text invizibil", innerText: "Text vizibil"
- Ambele vor afışa doar "Text vizibil"

Feedback

textContent returnează tot textul din element, inclusiv din elementele cu display: none și visibility: hidden, în timp ce innerText respectă stilizarea CSS și exclude textul din elementele ascunse sau invizibile.



Care este comportamentul unui closure în JavaScript?	3/3
O funcție care se apelează pe sine	
O funcție care nu returnează nimic	
O funcție care se execută imediat	
O funcție care are acces la variabilele din scope-ul extern	✓
Feedback	
Un closure permite unei funcții interioare să acceseze variabilele din funcția exterioară chiar și după ce funcția exterioară s-a terminat.	i,



✓ Ce returnează o funcție async?	4/4
Un Promise	✓
O Un callback	
O Un obiect async	
Valoarea returnată direct	
Feedback	
Funcțiile async returnează întotdeauna un Promise, chiar dacă returnezi o valoare obișnuită în funcție.	

✓ Ce rezultat va produce următoarea prelucrare de array? 3/3 1 const numbers = [1, 2, 3, 4] const result = numbers.filter((n) => n > 2).map((n) => n * 2) console.log(result) [1, 2, 3, 4] [2, 4, 6, 8] [4, 6, 8] **(**6, 8] Feedback filter() păstrează doar elementele > 2 (rezultând [3, 4]), apoi map() le înmulțește cu 2 (rezultând [6, 8]).

Cum gestionezi erorile într-o funcție async?	4/4
catch()	
try/catch	✓
throw/catch	
o error handler	
Feedback În funcțiile async, erorile se gestionează folosind blocurile try/catch, similar cu code sincron.	ul
Ce se întâmplă când încercăm să accesăm o variabilă declarată cu let înainte de declararea ei?	3/3
Funcționează normal	
Generează ReferenceError	✓
Returnează null	
Returnează undefined	
Feedback Variabilele declarate cu let și const nu pot fi accesate înainte de declarare, generânderoare "Cannot access before initialization".	d o



✓ Ce se întâmplă cu această funcție când este apelată?

3/3



- Returnează 42
- Returnează undefined
- Returnează obiectul (value: 42)
- Generează o eroare de sintaxă

Feedback

Din cauza Automatic Semicolon Insertion (ASI), JavaScript inserează un semicolon după return, astfel funcția returnează undefined în loc de obiect.

✓ Ce va afişa următorul cod? 3/3 function test() { let x = 1if (true) { console.log(x) console.log(x) test() 1, 1 2, 1 2, 2 Error Feedback Variabilele declarate cu let au block scope. Variabila x din if este diferită de cea din funcție.

✓ Care este diferența principală între appendChild() și append()?	3/3
appendChild() poate adăuga doar noduri, append() poate adăuga și stringuri	✓
appendChild() este mai rapid, append() este mai lent	
appendChild() returnează elementul adăugat, append() returnează undefined	
Nu există diferență, sunt sinonime	
Feedback	
appendChild() acceptă doar noduri DOM, în timp ce append() poate accepta atât noduri cât și stringuri de text și poate adăuga mai multe elemente simultan.	

✓ Ce se întâmplă când încerci să modifici o proprietate pe un tip primitiv? 4/4



- Afișează "world"
- Generează o eroare
- Afişează undefined
- Convertește string-ul în obiect

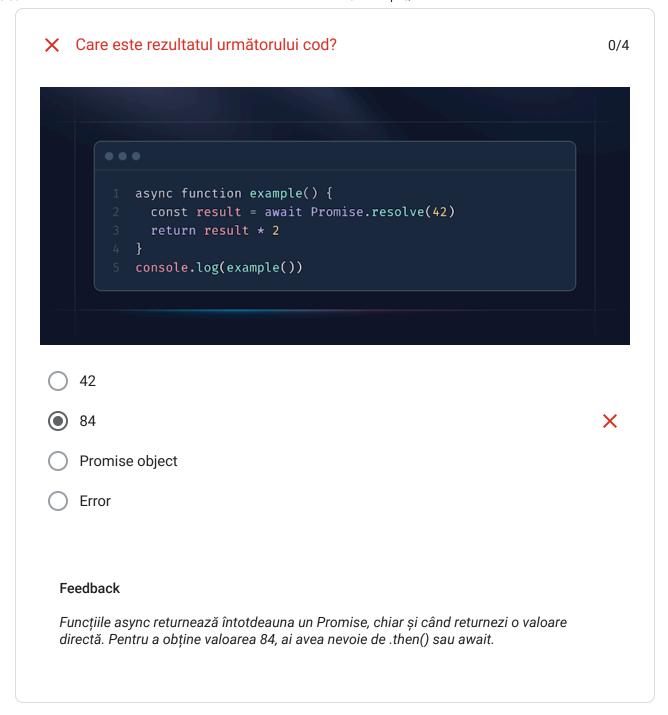
Feedback

Tipurile primitive sunt immutable în JavaScript. Când încerci să adaugi o proprietate, aceasta nu se păstrează și returnează undefined când o accesezi.

✓ Ce face metoda .finally() pe un Promise?	3/3
Nu există o astfel de metodă	
Se execută doar dacă Promise-ul eșuează	
Se execută doar dacă Promise-ul are succes	
Se execută indiferent de rezultatul Promise-ului	✓
Feedback Metoda .finally() se execută întotdeauna, indiferent dacă Promise-ul este fulfilled sau rejected.	
Ce se întâmplă când apelezi o funcție cu mai puține argumente decât parametrii declarați?	3/3
Generează eroare	
Funcția nu se execută	
Parametrii lipsă sunt null	
Parametrii lipsă sunt undefined	✓
Feedback Parametrii pentru care nu sunt furnizate argumente primesc valoarea undefined.	



Ce metodă folosești pentru a crea un element HTML nou?	3/3
document.createElement('div')	✓
document.newElement('div')	
document.addElement('div')	
new Element('div')	
Feedback document.createElement() creează un nou element HTML cu tag-ul specificat.	



✓ Ce se întâmplă când o funcție async aruncă o eroare?	3/3
C Eroarea este ignorată	
Eroarea oprește execuția aplicației	
Eroarea este transformată în warning	
Promise-ul returnat este rejected cu acea eroare	✓
Feedback Când o funcție async aruncă o eroare (throw), Promise-ul pe care îl returnează automat funcția devine rejected cu acea eroare ca motiv.	
✓ Ce se întâmplă când un Promise este rejected și nu este prins?	4/4
 ✓ Ce se întâmplă când un Promise este rejected și nu este prins? ☐ Se ignoră eroarea 	4/4
	4/4
Se ignoră eroarea	4/4
Se ignoră eroarea Se reîncearcă automat	4/4
 Se ignoră eroarea Se reîncearcă automat Se generează o UnhandledPromiseRejection 	4/4



✓ Ce este hoisting în JavaScript?	4/4
Comprimarea codului JavaScript	
Optimizarea performanței JavaScript	
Executarea codului în ordine inversă	
 Mutarea declaraţiilor de variabile şi funcţii în partea de sus a scope-ului 	✓
Feedback Hoisting este comportamentul JavaScript de a "muta" declarațiile de variabile și funcții î partea de sus a scope-ului lor.	ìn
✓ Care este diferența principală între push() și unshift() pentru array-uri?	3/3
Nu există diferență	
push() adaugă elemente, unshift() le elimină	
push() adaugă la sfârșitul array-ului, unshift() la început	✓
push() adaugă la început, unshift() la sfârșitul array-ului	
Feedback push() adaugă elemente la sfârșitul array-ului, iar unshift() adaugă elemente la începutu array-ului.	I

Acest formular a fost creat în domeniul Digital Nation. - <u>Contactează proprietarul formularului</u>

Acest formular pare suspect? <u>Raportează</u>

Formulare Google

