

## JavaScript - DOM Manipulation Quiz (Medium)

**Q1. What is the primary difference between the collection returned by `document.querySelectorAll('.item')` and `document.getElementsByClassName('item')`?**

- A) `querySelectorAll` returns a static `NodeList`, while `getElementsByClassName` returns a live `HTMLCollection`.
- B) `querySelectorAll` returns a live `NodeList`, while `getElementsByClassName` returns a static `HTMLCollection`.
- C) `querySelectorAll` can only select one element, while `getElementsByClassName` can select multiple.
- D) `querySelectorAll` returns an Array, while `getElementsByClassName` returns a `NodeList`.

**Q2. When adding a large number of elements to the DOM in a loop, what is the most performant approach to minimize browser reflows and repaints?**

- A) Appending each new element directly to the document body inside the loop.
- B) Creating a `DocumentFragment`, appending elements to it, and then appending the fragment to the DOM once.
- C) Concatenating a large HTML string and setting the parent's `innerHTML` property after the loop.
- D) Using `document.write()` inside the loop for each new element.

**Q3. In an event listener attached to a parent element (`''`), if a user clicks on a child button (`''`), what will `event.target` and `event.currentTarget` refer to?**

- A) Both will refer to the parent `''`.
- B) `event.target` will be the child `''`, and `event.currentTarget` will be the parent `''`.
- C) `event.target` will be the parent `''`, and `event.currentTarget` will be the child `''`.
- D) Both will refer to the child `''`.

**Q4. Consider the HTML `''`. If a user types 'new' into the field, what will `inputEl.value` and `inputEl.getAttribute('value')` return respectively?**

- A) 'new', 'new'
- B) 'initial', 'initial'
- C) 'new', 'initial'
- D) 'initial', 'new'

**Q5. Which property should you use to get a collection of only the direct child *\*elements\** of a node, excluding text nodes and comments?**

- A) `.childNodes`
- B) `.children`
- C) `.querySelectorAll('*')`
- D) `.innerElements`

**Q6. What does the method `element.classList.toggle('active')` do?**

- A) It always adds the 'active' class to the element.
- B) It always removes the 'active' class from the element.
- C) It returns `true` if the 'active' class is present, and `false` otherwise.
- D) It adds the 'active' class if it's absent and removes it if it's present.

**Q7. What is a key advantage of using `element.append()` over `element.appendChild()`?**

- A) `append()` has better support in older browsers.
- B) `append()` can accept multiple nodes and DOMStrings, while `appendChild()` only accepts a single Node.
- C) `append()` returns the node that was added, while `appendChild()` returns nothing.
- D) `append()` is less performant but easier to read.

**Q8. In an event listener for a form submission, what is the purpose of calling `event.preventDefault()`?**

- A) To stop the event from bubbling up to parent elements.
- B) To remove the event listener after it has executed once.
- C) To prevent the browser's default form submission behavior (e.g., reloading the page).
- D) To validate the form fields before submission.

**Q9. How do you correctly access the value '123' from the element `user` using the `dataset` property?**

- A) `user.dataset['data-user-id']`
- B) `user.dataset.dataUserId`
- C) `user.dataset.userId`
- D) `user.getAttribute('dataset.userId')`

**Q10. What is the difference between `node.cloneNode()` and `node.cloneNode(true)`?**

- A) `cloneNode()` creates a deep copy, while `cloneNode(true)` creates a shallow copy.
- B) `cloneNode()` creates a shallow copy (the node only), while `cloneNode(true)` creates a deep copy (the node and its descendants).
- C) `cloneNode()` also copies event listeners, while `cloneNode(true)` does not.
- D) `cloneNode(true)` is an invalid method call and will throw an error.

**Q11. To insert `newNode` right before `referenceNode`, what is the correct syntax?**

- A) `referenceNode.insertBefore(newNode)`
- B) `newNode.parentNode.insertBefore(referenceNode)`
- C) `referenceNode.parentNode.insertBefore(newNode, referenceNode)`
- D) `document.insertBefore(newNode, referenceNode)`

**Q12. To find an element with class `.item` that is a descendant of an element with ID `#container`, which is the most efficient query?**

- A) `document.querySelectorAll('#container .item')[0]`
- B) `document.getElementById('container').querySelector('.item')`

- C) `document.getElementsByClassName('item')[0]`
- D) `document.querySelector('.item')`

**Q13. You have a click listener on a button and another on its parent `div`. To prevent the `div`'s listener from firing when the button is clicked, what should you call in the button's event handler?**

- A) `event.preventDefault()`
- B) `event.stopImmediatePropagation()`
- C) `event.stopPropagation()`
- D) `return false;`

**Q14. Which of the following correctly replaces an existing `oldNode` with a `newNode` in the DOM?**

- A) `oldNode.replaceWith(newNode)`
- B) `newNode.replace(oldNode)`
- C) `oldNode.parentNode.insertAfter(newNode, oldNode)`
- D) `document.body.replace(oldNode, newNode)`

**Q15. What is the value of `element.style.backgroundColor` if the element's background was set to 'blue' in an external CSS file?**

- A) 'blue'
- B) 'rgb(0, 0, 255)'
- C) `undefined`
- D) An empty string

## Answer Key:

- Q1: A
- Q2: B
- Q3: B
- Q4: C
- Q5: B
- Q6: D
- Q7: B
- Q8: C
- Q9: C
- Q10: B
- Q11: C
- Q12: B
- Q13: C
- Q14: A
- Q15: D