

# jQuery — Related Questions

A multiple-choice self-test on jQuery 3.x covering selectors, content and DOM manipulation, CSS and attributes, traversal, element data, events, and AJAX. Each question has one answer, verified against [api.jquery.com](https://api.jquery.com). Companion to the Irurueta Docs jQuery Reference.

**Note on removed methods.** Some questions offer `.bind()`, `.live()`, or `.delegate()` as options. All three were deprecated and then **removed in the jQuery 3.x line** — `.on()` replaces every one of them. They appear here only as distractors.

## jQuery basics

### 1. What is the purpose of the jQuery library?

- (a) graphic design (b) animation authoring (c) web development (d) database management

**Answer: c)** web development — specifically DOM traversal and manipulation, event handling, animation, and AJAX in the browser.

## Effects and visibility

### 2. Which jQuery method hides an element?

- (a) `.hideElement()` (b) `.visibility('hidden')` (c) `.hide()` (d) `.disappear()`

**Answer: c)** `.hide()`.

### 3. Which jQuery method displays a hidden element?

- (a) `.reveal()` (b) `.show()` (c) `.display()` (d) `.visible()`

**Answer: b)** `.show()`.

### 4. How do you animate an element to fade out?

- (a) `.fadeOut()` (b) `.animateFadeOut()` (c) `.hide('fade')` (d) `.fade()`

**Answer: a)** `.fadeOut()`.

### 5. Which method hides an element with a sliding animation?

- (a) `.slideOut()` (b) `.slideUp()` (c) `.slideHide()` (d) `.hideSlide()`

**Answer: b)** `.slideUp()`.

### 6. Which call animates an element's height over 400 milliseconds?

- (a) `$('#el').animate({height: 'toggle'}, 400)` (b) `$('#el').changeHeight('400ms')` (c) `$('#el').heightAnimate(400)` (d) `$('#el').resizeHeight({duration: 400})`

**Answer: a)** `$('#el').animate({height: 'toggle'}, 400)` — `height: 'toggle'` animates to or from zero; pass a number such as `height: 200` for a fixed target.

### 7. Which method switches an element between shown and hidden on each call?

- (a) `.toggle()` (b) `.fadeIn()` (c) `.slideUp()` (d) `.append()`

**Answer: a)** `.toggle()` — `.fadeIn()` and `.slideUp()` also change visibility but only in one direction.

### 8. Which method smoothly transitions an element between visible and hidden?

- (a) `.fadeToggle()` (b) `.slideDown()` (c) `.append()` (d) `.after()`

**Answer: a)** `.fadeToggle()` — it animates opacity and flips shown/hidden each call; `.slideDown()` only reveals.

## Content and element insertion

### 9. Which jQuery method gets or sets the HTML content of an element?

- (a) `.html()` (b) `.content()` (c) `.innerHTML()` (d) `.getText()`

**Answer: a)** `.html()` — `innerHTML` is a native DOM property, not a jQuery method.

**10. Which jQuery method gets or sets the text content of an element?**

- (a) `.text()` (b) `.textContent()` (c) `.innerText()` (d) `.getString()`

**Answer: a)** `.text()`.

**11. Which method inserts content at the end of each matched element?**

- (a) `.appendTo()` (b) `.prependTo()` (c) `.after()` (d) `.append()`

**Answer: d)** `.append()` — `.appendTo()` is the same insertion with arguments reversed; `.after()` inserts outside, as a following sibling.

**12. Which method inserts content at the beginning of each matched element?**

- (a) `.prepend()` (b) `.before()` (c) `.insertBefore()` (d) `.startWith()`

**Answer: a)** `.prepend()` — `.before()` / `.insertBefore()` insert outside, as a preceding sibling.

**13. Which method inserts elements immediately before or after an existing element?**

- (a) `.prepend()` (b) `.after()` (c) `.wrap()` (d) `.append()`

**Answer: b)** `.after()` — with its pair `.before()`; `.prepend()` / `.append()` insert inside the element.

## DOM manipulation

---

**14. Which jQuery method removes all child nodes of the matched elements?**

- (a) `.empty()` (b) `.remove()` (c) `.detach()` (d) `.unwrap()`

**Answer: a)** `.empty()` — `.remove()` deletes the matched elements themselves (with their data and handlers).

**15. Which jQuery method creates copies of existing elements?**

- (a) `.append()` (b) `.clone()` (c) `.replaceWith()` (d) `.unwrap()`

**Answer: b)** `.clone()` — pass `.clone(true)` to also copy event handlers and `.data()`.

**16. Which jQuery method removes a wrapping element from around a selected element?**

- (a) `.unwrap()` (b) `.remove()` (c) `.empty()` (d) `.detach()`

**Answer: a)** `.unwrap()` — it removes each element's parent and leaves the element in place.

## CSS, classes, attributes and dimensions

---

**17. Which jQuery method changes an element's inline CSS properties?**

- (a) `.text()` (b) `.addClass()` (c) `.css()` (d) `.attr()`

**Answer: c)** `.css()` — for styling that lives in the stylesheet, toggle a class with `.addClass()` / `.toggleClass()` instead.

**18. How do you set the value of an HTML attribute on an element?**

- (a) `.attr()` (b) `.css()` (c) `.addClass()` (d) `.toggleClass()`

**Answer: a)** `.attr()` — for live form-control state (`checked`, `selected`, `disabled`, `current value`) use `.prop()`.

**19. What does the `.addClass()` method do?**

- (a) removes a class (b) toggles a class on and off (c) adds a class to the selected elements (d) retrieves CSS properties

**Answer: c)** adds a class to the selected elements — it accepts several space-separated names, or a function.

**20. Which jQuery method toggles a class on and off for the selected elements?**

- (a) `.toggleClass()` (b) `.removeClass()` (c) `.addClass()` (d) `.attr()`

**Answer: a)** `.toggleClass()` — a second argument forces the direction: `.toggleClass('x', isOn)`.

**21. Which jQuery tool reads or sets an element's dimensions?**

- (a) `.css()` (b) `.attr()` (c) `.addClass()` (d) `.toggleClass()`

**Answer: a)** `.css()` works, but the purpose-built methods are `.width()` / `.height()`, `.innerWidth()` / `.innerHeight()`, and `.outerWidth()` / `.outerHeight()` — they return a unitless number and their setters take a plain number.

## Traversing the DOM

---

**22. Which jQuery method selects descendants of an element that match a selector?**

- (a) `.find()` (b) `.next()` (c) `.closest()` (d) `.filter()`

**Answer: a)** `.find()`.

**23. What does the `.next()` method select?**

- (a) descendant elements (b) ancestor elements (c) sibling elements (d) elements by visibility

**Answer: c)** sibling elements — specifically the single *immediately following* sibling. Use `.nextAll()` for all following siblings, `.siblings()` for every sibling.

**24. Which jQuery method refines a set by a condition?**

- (a) `.next()` (b) `.filter()` (c) `.closest()` (d) `.prev()`

**Answer: b)** `.filter()` — it accepts a selector, an element/set, or a predicate `(index, element) => boolean`; its complement is `.not()`.

**25. Which jQuery selector matches elements that are currently hidden?**

- (a) `:visible` (b) `:hidden` (c) `.find()` (d) `.next()`

**Answer: b)** `:hidden` — note `:hidden` / `:visible` are jQuery extensions, not valid CSS, and judge rendered layout (an element with `visibility: hidden` still counts as visible).

## Element data and index

---

**26. What is the primary purpose of the jQuery `.data()` method?**

- (a) styling elements (b) attaching custom data to an element (c) creating new elements (d) parsing JSON

**Answer: b)** attaching custom data to an element — the data lives in jQuery's internal cache, not the DOM.

**27. Which values can be stored on an element with `.data()` ?**

- (a) strings and numbers only (b) objects and arrays only (c) strings, numbers, objects, or arrays (d) elements only

**Answer: c)** strings, numbers, objects, or arrays — `.data()` accepts any JavaScript value.

**28. How do you retrieve stored data from an element with `.data()` ?**

- (a) `.getData('key')` (b) `.data('key')` (c) `.retrieveData('key')` (d) `.fetch('key')`

**Answer: b)** `.data('key')` — calling `.data()` with no argument returns every stored key as an object.

**29. What does the `.index()` method determine?**

- (a) the tag name (b) an element's position among its siblings (c) the number of elements in the DOM (d) the CSS class

**Answer: b)** an element's position among its siblings — `.index(selectorOrElement)` instead reports the position within a given set.

## Event handling

---

**30. Which method handles user interactions such as clicking and hovering?**

- (a) `.bind()` (b) `.eventHandler()` (c) `.interact()` (d) `.click()`

**Answer: d)** `.click()` — in modern code, bind with `.on('click', handler)`; the no-argument `.click()` trigger form is deprecated.

**31. What is the purpose of the `.on()` method?**

- (a) disable all event handling (b) trigger an event programmatically (c) bind one or many events, optionally delegated (d) prevent propagation

**Answer: c)** bind one or many events, optionally delegated — `.on()` is the single general-purpose binder: one or more event types, an optional selector argument for delegation, and optional handler data.

**32. Which method stops an event from propagating to parent elements?**

(a) `.halt()` (b) `.stopEvent()` (c) `event.stopPropagation()` (d) `event.preventDefault()`

**Answer: c)** `event.stopPropagation()` — called on the event object inside the handler. `event.preventDefault()` is different: it cancels the default action, not propagation. Returning `false` from a jQuery handler does both.

**33. What does `event.preventDefault()` do?**

(a) it stops the browser's default action for the event (following a link, submitting a form, ticking a checkbox) (b) it stops the event bubbling to ancestors (c) it triggers the event automatically (d) it improves event delegation

**Answer: a)** it stops the browser's default action for the event — it does *not* stop other handlers running or the event bubbling (that is `event.stopPropagation()` / `event.stopImmediatePropagation()`).

**34. Which method binds events to elements matching a selector, including ones added later?**

(a) `.register()` (b) `.bind()` (c) `.attach()` (d) `.on()`

**Answer: d)** `.on()` — `$(parent).on('click', 'li', handler)` (event delegation); `.bind()` / `.live()` / `.delegate()` never supported this current form or were removed.

**35. Which method handles keyboard events such as a key press?**

(a) `.click()` (b) `.hover()` (c) `.keypress()` (d) `.submit()`

**Answer: c)** `.keypress()` — but both the `keypress` event and this shorthand are deprecated; use `.on('keydown', ...)` / `.on('keyup', ...)` and read `event.key`.

## AJAX

---

**36. What does AJAX stand for?**

(a) Advanced JavaScript and XML (b) Asynchronous JavaScript and XML (c) Automated JavaScript and XHTML (d) Active JavaScript and JSON

**Answer: b)** Asynchronous JavaScript and XML — the "XML" is historical; responses today are usually JSON.

**37. In `$.ajax()`, what do options such as `type`, `url`, `data`, and `dataType` configure?**

(a) the server's response time (b) the browser version (c) the AJAX request itself (d) the user's IP address

**Answer: c)** the AJAX request itself — `type` is a legacy alias for `method`.

**38. Which shorthand method makes a GET request for JSON data?**

(a) `$.send()` (b) `$.fetch()` (c) `$.getJSON()` (d) `$.request()`

**Answer: c)** `$.getJSON()` — it is `$.get()` with `dataType: 'json'`; the native global `fetch()` is unrelated.

**39. What kind of request does `$.post()` make?**

(a) GET (b) POST (c) PUT (d) DELETE

**Answer: b)** POST.

**40. What is the purpose of the `success` callback in an AJAX request?**

(a) handle errors (b) run after the request completes, success or not (c) send data to the server (d) handle the successful response from the server

**Answer: d)** handle the successful response from the server — `success` / `error` / `complete` are the legacy style; prefer the jqXHR promise (`.done()` / `.fail()` / `.then()`), Promises/A+ compliant since jQuery 3.

**41. What is the main advantage of shorthand methods such as `$.get()` and `$.post()`?**

(a) they handle any request type (b) they are more secure (c) they make the code shorter and more readable (d) they give better server performance

**Answer: c)** they make the code shorter and more readable — they are thin wrappers over `$.ajax()` with a fixed `method` / `dataType`; anything more (headers, `timeout`, `beforeSend`) still needs `$.ajax()`.

**42. Which jQuery method is the core of its AJAX functionality?**

(a) `$.get()` (b) `$.post()` (c) `$.getJSON()` (d) `$.ajax()`

**Answer: d)** `$.ajax()` — `$.get()`, `$.post()`, and `$.getJSON()` are all shorthands that call it internally.

## Building an interactive list

---

These refer to the worked example in the jQuery Reference (*Worked Example: An Interactive List*).

**43. In the rule `.red span { color: red; text-decoration: line-through; }`, what is the `red` class for?**

(a) set the background colour (b) mark a crossed-out ("done") item with a line-through (c) change the font size (d) underline the text

**Answer: b)** mark a crossed-out ("done") item with a line-through — the rule also sets the text colour red, but "line-through" is the part being tested.

**44. What does the click handler on a delete button do?**

(a) adds a new list item (b) removes the input field (c) removes the parent `<li>` from the list (d) toggles the `red` class

**Answer: c)** removes the parent `<li>` from the list — the delegated handler calls `.closest('li').remove()`.

**45. What triggers creation of a new list item?**

(a) clicking the "Add to List" button (b) hovering over a list item (c) typing in the input field (d) reloading the page

**Answer: a)** clicking the "Add to List" button — binding `submit` on the surrounding form is more robust because it also fires on the Enter key.