

# CSS selectors For QA/testers

## 1. Basic Selectors

Selector	What it selects	QA Use Case
*	All elements	Rarely used, can check global styles
element	All elements of that type (div, p, button)	Verify default styles for elements
.class	All elements with a specific class	Check component styling
#id	Element with specific ID	Target unique elements for validation
[attribute]	Elements with an attribute	Check inputs, links, data attributes
[attribute=value]	Elements with exact attribute value	Validate specific input fields or buttons

## 2. Combinators

Selector	What it selects	QA Use Case
parent child	All descendants of parent	Nested elements styling
parent > child	Direct children only	Validate hierarchical layout
prev + next	Adjacent sibling	Check spacing, hover interactions
prev ~ siblings	All following siblings	Verify grouped elements styling

## 3. Pseudo-classes (QA-Focused)

Selector	What it selects	QA Use Case
:hover	Element on hover	Test hover states
:focus	Element when focused	Inputs, accessibility testing
:active	Element during click	Buttons, links
:disabled	Disabled form elements	Verify form behavior
:checked	Checked radio/checkbox	Form validation
:first-child	First child of parent	Layout verification
:last-child	Last child	Styling consistency
:nth-child(n)	nth child of parent	Verify dynamic lists
:not(selector)	Elements not matching selector	Conditional checks

## 4. Pseudo-elements (QA-Focused)

Selector	What it selects	QA Use Case
::before	Content inserted before element	Decorative elements/icons
::after	Content inserted after element	Check visual cues or counters
::placeholder	Placeholder text in inputs	Verify styling of input hints

## 5. Attribute Selectors

Selector	Example	QA Use Case
[attr^="value"]	Starts with	Input fields starting with a prefix
[attr\$="value"]	Ends with	Class or ID naming consistency
[attr*="value"]	Contains	Buttons/links containing text/data

## 6. Grouping & Universal

Selector	What it selects	QA Use Case
selector1, selector2	Multiple selectors	Check multiple elements in one test
*	All elements	Check default/global style

### QA-Focused Tips

1. **Inspect via DevTools:** Use selectors in Chrome/Firefox console for quick checks:

```
document.querySelectorAll('.btn-primary') // returns all matching buttons
```

- 2. **Test dynamic states:** Hover, focus, active, checked — often missed but crucial.
- 3. **Check nested elements:** Especially when layouts break in mobile view.
- 4. **Target pseudo-elements:** Sometimes errors appear only in ::before or ::after.
- 5. **Attribute selectors:** Useful for testing dynamically generated IDs or data-\* attributes.

# CSS Quick Reference & QA Checklist

## 1. Layout & Positioning

Item	What to Check	Tools / Notes
Box Model	Verify margins, padding, borders, content sizing	DevTools → Elements → Computed
Display Property	Check <code>block</code> , <code>inline</code> , <code>flex</code> , <code>grid</code> , <code>none</code>	Look for hidden elements ( <code>display:none</code> )
Positioning	<code>static</code> , <code>relative</code> , <code>absolute</code> , <code>fixed</code> , <code>sticky</code>	Ensure elements don't overlap incorrectly
Z-index	Layering of modals, popups, dropdowns	Inspect stacking context
Flex/Grid	Alignment, spacing, wrapping	Test responsiveness & visual layout

## 2. Responsive Design

Item	What to Check	Tools / Notes
Media Queries	Layout adapts correctly at breakpoints	DevTools → Responsive mode
Relative Units	<code>%</code> , <code>em</code> , <code>rem</code> scale correctly	Test resizing text or containers
Viewport Units	<code>vw</code> , <code>vh</code> , <code>vmin</code> , <code>vmax</code> scaling	Especially for full-screen sections
Flex/Grid Responsiveness	Items wrap, shrink, or grow correctly	Check on mobile/tablet/desktop

## 3. Typography & Text

Item	What to Check	Tools / Notes
Fonts	Correct <code>font-family</code>	Ensure fallback fonts are acceptable
Font Sizes & Line Height	Readability across devices	Check scaling with <code>em</code> , <code>rem</code>
Text Alignment	<code>text-align</code> , vertical alignment	Especially for buttons, labels
Text Overflow	<code>overflow</code> , <code>text-overflow: ellipsis</code>	Check long text does not break layout

## 4. Colors & Visuals

Item	What to Check	Tools / Notes
Color	Correct <code>color</code> , <code>background-color</code> , <code>border-color</code>	Verify branding consistency
Opacity & Transparency	<code>opacity</code> , <code>rgba()</code>	Check visibility of overlapping elements
Borders & Shadows	<code>border</code> , <code>border-radius</code> , <code>box-shadow</code>	Ensure consistent UI components

Item	What to Check	Tools / Notes
Backgrounds	Images, gradients, size, repeat	Check for scaling / tiling issues

## 5. Interactivity & States

Item	What to Check	Tools / Notes
Hover/Focus States	<code>:hover</code> , <code>:focus</code> , <code>:active</code>	Buttons, links, input fields
Transitions	<code>transition</code>	Smoothness of animation
Animations	<code>animation</code> , <code>keyframes</code>	Triggered properly and not janky
Disabled / Hidden States	<code>:disabled</code> , <code>display:none</code>	Ensure UI behaves as expected

## 6. Advanced / Optional

Item	What to Check	Tools / Notes
Pseudo-elements	<code>:before</code> , <code>:after</code>	Verify decorative content or icons
CSS Variables	<code>--var-name</code>	Theme consistency (light/dark mode)
Vendor Prefixes	<code>-webkit-</code> , <code>-moz-</code>	Cross-browser compatibility
Framework Classes	Bootstrap, Tailwind, Material UI	Ensure correct usage / overrides

## 7. Accessibility Checks

Item	What to Check	Tools / Notes
Color Contrast	Meet WCAG standards	Tools: Lighthouse, Axe
Focus Visibility	<code>:focus</code> state visible	Tab through UI components
Hidden Content	<code>display:none</code> vs <code>visibility:hidden</code>	Ensure screen readers handle properly

## 8. Practical QA Workflow

1. Open **DevTools** → Inspect Element
2. Check **Box Model & Positioning**
3. Switch to **Responsive Mode** → Resize screen
4. Test **hover/focus states** & interactivity
5. Verify **colors, fonts, backgrounds, borders**
6. Test on **different browsers** → Chrome, Firefox, Safari, Edge
7. Optional: **Check animations, transitions, pseudo-elements**