**Optimize Selectors**

Selector optimization is less important than it used to be, as more browser implement document.querySelectorAll() and the burden of selection shifts from jQuery to the browser. However, there are still some tips to keep in midn.

**ID-Based Selectors**

Beginning your selector with an ID is always best.

// fast

$('#container div.robotarm');

// super-fast

$('#container').find('div.robotarm');

The $.fn.find approach is faster because the first selection is handled without going through the Sizzle selector engine — ID-only selections are handled using document.getElementById(), which is extremely fast because it is native to the browser.

**Specificity**

Be specific on the right-hand side of your selector, and less specific on the left.

// unoptimized

$('div.data .gonzalez');

// optimized

$('.data td.gonzalez');

Use tag.class if possible on your right-most selector, and just tag or just .class on the left.

Avoid excessive specificity.

$('.data table.attendees td.gonzalez');

// better: drop the middle if possible

$('.data td.gonzalez');

A "flatter" DOM also helps improve selector performance, as the selector engine has fewer layers to traverse when looking for an element.

**Avoid the Universal Selector**

Selections that specify or imply that a match could be found anywhere can be very slow.

$('.buttons > \*'); // extremely expensive

$('.buttons').children(); // much better

$('.gender :radio'); // implied universal selection

$('.gender \*:radio'); // same thing, explicit now

$('.gender input:radio'); // much better