When you know that a certain condition will not change throughout the life of the program, it makes sense to test the condition only once. Browser sniffing (or feature detection) is a typical example.

[复制代码](javascript:void(0);)

// BEFORE

var utils = {

addListener : function(el, type, fn) {

if ( typeof window.addEventListener === 'function') {

el.addEventListener(type, fn, false);

} else if ( typeof document.attachEvent === 'function') {// IE

el.attachEvent('on' + type, fn);

} else {// older browsers

el['on' + type] = fn;

}

},

removeListener : function(el, type, fn) {

// pretty much the same...

}

};

// AFTER

// the interface

var utils = {

addListener : null,

removeListener : null

};

// the implementation

if ( typeof window.addEventListener === 'function') {

utils.addListener = function(el, type, fn) {

el.addEventListener(type, fn, false);

};

utils.removeListener = function(el, type, fn) {

el.removeEventListener(type, fn, false);

};

} else if ( typeof document.attachEvent === 'function') {// IE

utils.addListener = function(el, type, fn) {

el.attachEvent('on' + type, fn);

};

utils.removeListener = function(el, type, fn) {

el.detachEvent('on' + type, fn);

};

} else {// older browsers

utils.addListener = function(el, type, fn) {

el['on' + type] = fn;

};

utils.removeListener = function(el, type, fn) {

el['on' + type] = null;

};

}

[复制代码](javascript:void(0);)

**References*:***

*JavaScript Patterns -*by Stoyan Stefanov (O`Reilly)