撩一下一些必要的js工具函数

JavaScript (/lib/tag/JavaScript) 2016-04-06 16:19:52 发布

不管是什么项目，总有一些基本的功能函数默默的躺在你的工具库中，为你遮挡bug，提升性能，一起来复习下！

debounce

当监听一些scroll，resize事件时，如果我们什么都不限制，srcoll事件在滚动过程中会一直触发，极大的影响性能，这个时候，就需要节流函数debounce。debounce函数返回一个函数，在给定的时间间隔下，连续被调用将不会触发。

```
/* 事件停止被触发N秒后才会再次触发回调 * @param {Function} func - 回调执行函数 * @param {String} wait - 触发间隔 * @param {Boolean} immediate - 是否延时执行 */
function debounce(func, wait, immediate) {
    var timeout;
    return function() {
        var context = this, args = arguments;
        var later = function() {
            timeout = null;
            if (!immediate) func.apply(context, args);
        };
        var callNow = immediate && !timeout;
        clearTimeout(timeout);
        timeout = setTimeout(later, wait);
        if (callNow) func.apply(context, args);
    };
};

// Usage
var myEfficientFn = debounce(function() {
    // todo
}, 250);
window.addEventListener('resize', myEfficientFn);
```

poll

很多时候，我们需要了解某个函数的执行状态，并根据状态执行相应的处理。在没有事件通知时，需要以一定的时间间隔轮询执行状态。

阅读目录

```
/* 轮询条件函数，根据状态执行相应回调 * @param {Function} fn- 条件函数 * @param {Function} callback - 成功回调 * @param {Function} errback - 失败回调 * @param {int} timeout - 超时间隔 * @param {int} interval - 轮询间隔 */
function poll(fn, callback, errback, timeout, interval) {
    var endTime = Number(new Date()) + (timeout || 2000);
    interval = interval || 100;

    (function p() {
            // If the condition is met, we're done!
            if(fn()) {
                callback();
            }
            // If the condition isn't met but the timeout hasn't elapsed, go again
            else if (Number(new Date()) < endTime) {
                setTimeout(p, interval);
            }
            // Didn't match and too much time, reject!
            else {
                errback(new Error('timed out for ' + fn + ': ' + arguments));
            }
    })();
}

// Usage: ensure element is visible
poll(
    function() {
        return document.getElementById('lightbox').offsetWidth > 0;
    },
    function() {
        // Done, success callback
    },
    function() {
        // Error, failure callback
    }
);
```

once

实用的执行一次函数，不用多解释。虽然很简单的函数，但是防止重复加载或者初始化的习惯必须养成。

```
function once(fn, context) {
    var result;

    return function() {
        if(fn) {
            result = fn.apply(context || this, arguments);
            fn = null;
        }

        return result;
    };
}

// Usage
var canOnlyFireOnce = once(function() {
    console.log('Fired!');
});
```

getAbsoluteUrl

从字符串变量中获取绝对url路径并没有那么容易，URL构造函数必须提供必要的参数，getAbsoluteUrl函数可以单纯的从字符串输入中得到绝对路径。

```
var getAbsoluteUrl = (function() {
    var a;

    return function(url) {
        if(!a) a = document.createElement('a');
        a.href = url;

        return a.href;
    };
})();

// Usage
getAbsoluteUrl('/shirlyzhang');  // "http://imweb.io/shirlyzhang"
```

isNative

当需要重写某个函数时，必须先确定其是否是原生函数。

```
(function() {
  var toString = Object.prototype.toString;
  var fnToString = Function.prototype.toString;
  // 构造函数，数组
  var reHostCtor = /^\[object .+?Constructor\]$/;

  var reNative = RegExp('^' +
    // 强制转换（Object-toString）为字符串
    String(toString)
    // 转移特殊字符
    .replace(/[.*+?^${}()|[\]\/\\]/g, '\\$&')
    // 替换 `toString` 为 `.*?`， 保持模板通用
    // 替换 `for ...` 等字符来支持添加了额外信息的环境
    .replace(/toString|(function).*?(?=\\\()| for .+?(?=\\\])/g, '$1.*?') + '$'
  );

  function isNative(value) {
    var type = typeof value;
    return type == 'function'
      // 使用 `Function#toString` 来绕过value原生的 `toString` 方法，避免误判
      ? reNative.test(fnToString.call(value))
      // 回退到宿主对象检查。因为某些环境中，会将类型数组当做DOM方法。
      : (value && type == 'object' && reHostCtor.test(toString.call(value))) || fals
e;
  }

  module.exports = isNative;
}());

// Usage
isNative(alert); // true
isNative(myCustomFunction); // false
```

insertRule

使用css选择器来选取节点修改样式不太高效，可以选择用js新建一段css样式规则插入。

```
var sheet = (function() {
    var style = document.createElement('style');

    // WebKit 兼容
    style.appendChild(document.createTextNode(''));

    document.head.appendChild(style);
    return style.sheet;
})();

// Usage
sheet.insertRule("header { float: left; opacity: 0.8; }", 1);
```

matchesSelector

判断页面内的元素是否具有某些属性值。

```
function matchesSelector(el, selector) {
    var p = Element.prototype;
    var f = p.matches || p.webkitMatchesSelector || p.mozMatchesSelector || p.msMatche
sSelector || function(s) {
        return [].indexOf.call(document.querySelectorAll(s), this) !== -1;
    };
    return f.call(el, selector);
}

// Usage
matchesSelector(document.getElementById('myDiv'), 'div.someSelector[some-attribute=t
rue]')
```

来自： http://imweb.io/topic/5704a23a06f2400432c1397f (http://imweb.io/topic/5704a23a06f2400432c1397f)

同类热门经验

1. Node.js 初体验 (/lib/view/open1326870121968.html)
2. JavaScript开发规范要求 (/lib/view/open1352263831610.html)
3. 使用拖拉操作来自定义网页界面布局并保存结果 (/lib/view/open1325064347889.html)
4. Nodejs入门学习，nodejs web开发入门，npm、express、socket配置安装、nodejs聊天室开发 (/lib/view/open1329050007640.html)
5. 利用HTML5同时上传多个文件 - resumable.js (/lib/view/open1327591300671.html)
6. nide：一个不错的Node.js开发工具IDE (/lib/view/open1325834128750.html)

相关文档 — 更多 (http://www.open-open.com/doc)

- Nodejs中文文档.pdf (http://www.open-open.com/doc/view/74ffe3fa3eb24e23a61dc36ab95e93f0)
- 基于第一个PhoneGap(cordova)的应用详解.docx (http://www.open-open.com/doc/view/aff67a42dfcc42a2afaa48651a659a97)
- JavaScript 标准参考教程 - Node.js 概述.pdf (http://www.open-open.com/doc/view/09480785c08f43fc8ea0618ccf2847b9)
- JavaScript 加强.ppt (http://www.open-open.com/doc/view/69691eea5564475490bdc690893f5a32)

相关经验 — 更多 (http://www.open-open.com/lib)

- 以开发者的视角整理编排的前端开发所使用语言的主流学习资源 (/lib/view/open1456802317250.html)
- Cordova开发前的准备工作 (/lib/view/open1434699808067.html)
- 抛弃jQuery，深入原生的JavaScript (/lib/view/open1398340435562.html)
- JavaScript初学者应知的24条最佳实

相关讨论 — 更多 (http://www.open-open.com/solution)

- 什么是Node.js？ (http://www.open-open.com/solution/view/1318473088937)
- 用NodeJS打造你的静态文件服务器 (http://www.open-open.com/solution/view/1321344823593)
- 那些年，追过的开源软件和技术 (http://www.open-open.com/solution/view/1425959150201)

- JavaScript 入门学习笔记总结2.doc (http://www.open-open.com/doc/view/2d879c4b038d463f9dbc7dea75f93e5d)
- 精通Ext JS 第二版.pdf (http://www.open-open.com/doc/view/e73b4a47802d41a3b2ab072479384844)
- Node.js入门经典.pdf (http://www.open-open.com/doc/view/864dac02a053402e85183f4a498a6bb6)
- node.js入门之buffer类的使用详解.docx (http://www.open-open.com/doc/view/deb470bdbae64c12a1406b120b8c0d24)
- 基于Gulp的前端自动化.pdf (http://www.open-open.com/doc/view/c531a6f797254d0888eeffa77e3ac364)
- 开发Backbone.js应用.pdf (http://www.open-open.com/doc/view/677fb06d95f440ac99ab2e98d8407e8c)
- 韩顺平html+css+js之JS笔记.doc (http://www.open-open.com/doc/view/10937f5ec47e4fd181db3d066053a5a7)
- Backbone.js 中文 API.pdf (http://www.open-open.com/doc/view/b2914a99a50c425681af5b799408885c)
- 常用的javascript验证正则表达式.pdf (http://www.open-open.com/doc/view/a21ade8bbbb44b5e82687aeaad88f35a)
- Learning Node.pdf (http://www.open-open.com/doc/view/b9595c4711f94452aef70783ce813844)
- 利用impress.js构建令人印象深刻的幻灯片.pdf (http://www.open-open.com/doc/view/45bf01c0be0240aab608391a55dac01c)
- JS RegExp 正则表达式.pdf (http://www.open-open.com/doc/view/e37b8e209ac74fe1882e2422389b77c3)
- web开发调试工具.pdf (http://www.open-open.com/doc/view/4ca2c0e3b12945f9b2d15f80a6e2bebf)
- 开发Backbone.js应用程序.pdf (http://www.open-open.com/doc/view/5d578a190fee40cabaef026280fa7bf6)
- PhoneGap中文api.pdf (http://www.open-open.com/doc/view/ecb54caa96ab4b5c8678e8267948d8c0)
- JS入门教程.pdf (http://www.open-open.com/doc/view/c14f187d11d24d55811bfe832a29587b)

- 践（译） (/lib/view/open1378729348928.html)
- 100+ 超全的 web 开发工具和资源 (/lib/view/open1464676693194.html)
- Vue.js 源码学习笔记 (/lib/view/open1439344021458.html)
- 听飞狐聊JavaScript设计模式系列14 (/lib/view/open1452240695120.html)
- ES6：下一版本的JavaScript的新特性 (/lib/view/open1453297166620.html)
- Node.js内存泄漏分析 (/lib/view/open1456659632687.html)
- 【译】寻找头绪：编写可维护的JavaScript (/lib/view/open1464080808193.html)
- 【译】寻找头绪：编写可维护的JavaScript (/lib/view/open1464081646045.html)
- 给 JavaScript 初心者的 ES2015 实战 (/lib/view/open1447222864319.html)
- jQuery的性能优化，你知道几条 (/lib/view/open1388202452360.html)
- 解开面条代码: 怎样书写可维护JavaScript (/lib/view/open1463732040577.html)

- Web开发从学些JavaScript开始 (http://www.open-open.com/solution/view/1417659580167)
- Web开发人员最喜爱的10款流行JavaScript库 (http://www.open-open.com/solution/view/1379903308476)
- Javascript闭包学习 (http://www.open-open.com/solution/view/1321491048452)
- 76个JavaScript教程资源免费下载 (http://www.open-open.com/solution/view/1372818526987)