1. jslint 风格一致

封装函数，让一切更简单

基本格式化

任何变量 和值都不要紧靠别的值和分隔符 而应该空一位

1. 缩进层级 同级对齐 花括号和首字符对齐不是和括号对齐

制表相当于四个字符，问题是编辑器处理不同

空格缩进2，4，8 4常用 （可以设置快捷键 tab=四个空格

1. 分号 js自动分号插入可能会有错

Return 最好不要换行 return {}可能解读错误，{}一般也不一定加分号

1. 行长度 单行不要超过80字符 其实把集合，如多参数，数组元素，对象属性拆成单行也是很有用的

4、如何换行 在符号后换行 a, 后 缩进两个tab

5、空行 不相关代码段 空行分隔 有关联代码段放一起

方法之前，局部变量定义 和第一条语句间 注释之前 逻辑片段间

6、命名驼蜂式 不用短横线 aCameraOfMine

变量名 前缀是名词

函数 前缀是动词 getThePhoto

常用前缀 is has get set can show

命名应该尽可能短，又避免理解错误 不用ijk foobar tmp之类的模糊命名，用具体的英文单词

7、常量 把变量当常量

Var MAX\_NUM=1;

8、构造函数 大驼峰

9、直接量 字符串 双引号 单引号 没有不同，就是双引号里面使用双引号要转义

换行字符串连接用+号 “hello” +”world”

数字，只有一种类型 推荐写完整的小数整数 如果需要使用小数就用10.00 不要省略 不要省略整数 .5 八进制似乎已经淘汰？(最好不用) 十六进0XA2 大写

科学计数 1e23

Null: 用来初始化 对象 和变量做比较（不一定是对象类型） 作为函数的对象型空参数，传出对象型空返回（打算返回对象时） return null; fn(null);

不用Null检测没有初始化赋值的变量 不用null检测参数if(arg4!=null)

空对象，占位符

Undefined null==undefined 但是这检测是错误的 == 和===的区别

定义了没赋值的 var a ; a===undefined 没定义的，没赋值的 typeof b typeof a 都返回undefined

尽量不要用undefined检测 禁用

5、对象直接量 不用new Object() var book={} 第一行花括号，每个属性一行缩进一个tab，最后花括号一行

Var book={

Title: ok; //空一格

}

6、数组直接量 var array = [1, 2, 3, 4, 5]; //数组每项应该分开

1. 注释

1、单行注释 // vuui 敲一个空格

（1）独占一行解释下一行 注释前空一行

（2）尾注释 包含一个缩进

（3）只有注释大段代码时才多行出现

注释是解释行的，所以需要和行缩进一致，或者在尾部

2、多行注释

可以单行或跨行

Java风格 开头结束各一行 中间星号对齐 和正文相隔一格

/\*

\* 星号后最少空一格

\*/

多行注释前有空行，和代码间无，缩进和代码一致

3、何时注释 代码很清晰时不注释 代码不明了时注释

不要注释一眼看出的功能

对整个功能块注释，对所有新定义的变量，参数批量注释

难理解的代码if(！| | ) 代码可能被误认为是错的

1. 文档注释

/\*\* @method @param @return

为所有的方法 参数 返回 对类（构造器），和传参

1. 语句和表达式

快语句都应该包含花括号，即使是单行

1、对齐 放在左边的末尾（推荐） ，或者在下一行

if(a>1){

fn(); //缩进tab

} //各占一行

2、间隔 无间隔if(){} 各一个空格 if (a>b) { （推荐） 在圆括号里面也加两个空格

3、switch () {

Case “”: //一缩进，换行 从第二个case开始，每个前后都有空行

//二缩进

Break;

Default:

另一种风格是case不缩进，和switch同级 独站一行，没有空行 语句换行，缩进一tab

4 case应该break吗？ 连续执行case理论上不会有bug 因为输入的值一般是固定的

但注意case后不break ,是会进入default的，default

如果需要连续执行，那就注释 ，如果不需要，就坚决每个打break

需要空的default吗？ 如果没有默认行为，可以省略，最好注释//无默认

4、with语句 穿透变量，用局部变量和函数 访问对象的属性和方法

With(obj){ pro++; objfun() }

1. for 循环 for for in

数组元素之间打空格吗？ 赋值前后打空格吗？

For (i=1; i<5; i++) { 每个变量前和分号间都打空格 打空格换行

Haha(); 缩进

}

Break continue return

For in 会遍历原型属性， 过滤hasOwnProperty(prop) 返回的是属性名（字符串） object[prop]

1. 变量、函数、运算符

1、声明 变量声明会**自动提前**到函数的顶部

For(var )声明 由于没有let就没有块作用域 这个var会定义在外面，而且被提升到函数顶

所以可以把同一作用域所有的变量声明在顶部

单var语句 var a = 10, //赋值号前后都留空格 分隔变量都留空格

B=1. 每个变量单独一行，赋值符对齐 变量缩进对齐，没有赋值的在最后写

2、函数声明 调用可在声明前 建议先声明再写函数 function do(item) {} //和花括号空一格 代码缩进

不应该用 if () { 函数声明}

3。、调用 没有间隔 a();

4、匿名函数 var a = function() {}

立即执行 function() {} () 不容易看出 必须用圆括号包围

Var a = (function() {} () ); 看看括号的分隔 包多重括号时

* 1. 严格模式 可以用于局部，不建议全局严格

“use strict

7、相等和强制类型转换 == ！= 类型不同是就会类型转换

5==”5” 字符串会转数字，而不是转ascii编码 1==true 0==false

如果是对象，另一个不是 valueOf() 如果没有valueOf() 就toString()

Null==undefined

不要用== 用=== !==

8、eval( 字符串 ) “语句” “count + 5”

9、包装类型 让基础类型有对象行为 name = “” name.toUpperCase()

基础类型 当成对象使用的时候，本身就会创建一个临时实例，然后销毁

name.prop=1 发现无法加上，因为不是对象

不建议直接使用包装类型 String Number 系统会自动调用

“”.toNumber() 是不行的？

1. UI层松耦合

耦合 一个组件修改了，另一个也要改

无法无耦合 总要传递信息的

文本 样式 行为

2、css中的js 表达式（已经不存在）

3、通过js来修改style element.style.color=”red”; .cssText=”color:red;left:10px;”

如何是好实现？不修改style属性 而是增加减少className += “ ok”; 类名需要空格分隔

可以接受的，重定位操作 修改元素坐标

4、js和html分离 不写 onclick

写 getElementById(); addEvenrLisener()

1. js中的html

innerHTML 嵌入多行代码 动态dom dom操作

用php模板技术代替？ 不太好

Xml获取？

客户端模板 {{}}

1. 避免全局变量

执行环境 “全局对象”-window对象 所有全局声明的函数和变量都是全局对象的属性

1. 命名冲突 和其他开发的代码 内置API
2. 难以管理依赖 依赖全局的函数
3. 修改全局 处处报错， 如果不依赖而是作为参数传入，传参是解除依赖的好办法
4. 不好测试，

2、 创建全局 无意间省略 var 甚至无意间覆盖了系统属性 禁止未声明赋值

3、使用单全局变量 命名空间 yui jquery 其他功能代码都挂在对象下

4、命名空间 在全局变量下继续分类dom mail 对象.对象.属性

5、模块 amd

6、0全局变量 立即执行函数

七、事件处理

Event 回调 一般只会用到部分属性 target

（1）拆分逻辑代码 handler(event){ disolver(event); }

(2) 拆分时其实不建议分发event对象 应该把event的属性提取出来传入 discover(x,y) {} 这样代码和event就没有耦合

3、handler 对event进行处理 然后转发其中信息给逻辑函数 阻止默认，阻止冒泡

1. 避免空比较

If (item != =null) 如果不为空？ 这个意思就不对，任何基础类型，对象都和null不等

1. 基本类型比较 typeof 返回字符串 “string” number boolean undefined
2. Null不用于检测语句 除非你真的希望是null 空对象，这样必须提前设为null，如果没设，那是undefined

Typeof null object

3、对象检测 所有对象typeof返回 object

obj Instanceof 构造 可以检测原型链，所以不是最好的方法

1. 检测函数 typeof function
2. 检测数组 帧问题frame Object.prototype.toString.call(value) [object Array]

isArray()

5、检测对象属性是否存在 if ( obj[a] ) 只要为假就出错 0 “” false null undefined 不合适，可能定为其中之一 不要检测null

In 符 if (a in b )

**数组的 a [] 需要空格？**

1. 配置数据从代码中分离
2. 硬编码的值 称为配置数据 “写死”

Alert(“ok”) var s = “ok”;

字符串 Url className 不建议用硬编码，应该用可变的变量作用中间

常见配置数据：可变值 ，字符串，url,设置，重复出现的值(如全局常量，变量)

2、配置抽离封装 封装在数组，结构，对象中 config = {}

3、配置数据的形式 ，json？ Jsonp ：json作为调用函数的参数，让服务端返回js

九、自定义错误

1、throw new Error(“error”) 传入的参数是message 表达错误信息

Message可以是任何类型，包括数字，字符，变量，对象，等

New Error(“发生错误，行号” + line + ) 错误信息是由变量和字符拼接出来的

发生错误的行号，函数名，失败原因

2、何时抛出 if检查后抛出 很多类库都可以检查和发出错误

3、错误处理 拦截错误，避免程序崩溃， 进行一定的处理后恢复try catch()

4、错误类型 eval range synax type uri

5、创建自己的错误类型 继承 重写message

十一、避免用错对象

不动原生对象，Object Array，即使想添加功能也应该封装一个新对象 dom window bom

类库对象

不覆盖已存在的对象/库的方法，不新增，不删除。。。所有的处理在自己的对象实例中修改

3、更好的途径

扩展这些对象 继承

原型继承 Object.create() 可以有个参数对象，将参数中的属性方法加入到新对象，会覆盖

基于类型的继承 newobj.prototype=new Old()

构造器继承 B(name){ A.call(this,name) } B.protorype=new A()

4．门面模式（装饰器，包装器）

给原对象添加方法 ，但是是通过包装对象使用的

原对象可以全局穿透到包装对象中，也可以作为参数传入

有些类似代理？ New Wrapper(obj) wrapper.add()

如果不是添加方法，而是改变方法，让对象和其他对象有**统一接口**，就称为适配器

1. polyfill 把高版本es用低版本js实现 用在低兼容的浏览器
2. 阻止对象被修改

禁止扩展Object.preventExtension() isExtensible

密封 .seal() isSealed()

冻结 .freeze isFrozen() 锁定的对象无法解锁，除非修改原代码

十二、浏览器嗅探

1. 用户代理检测 user-agent 用户代理字符串 navigator.userAgent

不好用，要随着浏览器更新 有库

2、特性检测 对浏览器特性进行检测 是否可以使用某个js特性，是否实现了某个方法？

检测标准js方法 检测浏览器特定方法（有可能需要浏览器前缀）

不存在方法时设法自己提供一个

3、特性应该分别检测 不能认为有了某种特性，其他的就一定存在 比如有getid就有getName 特性推断

4、避免浏览器推断 存在某种方法就认为是某种浏览器 无法用部分来定义整体

十三、 自动化 自动化工具 生产版和发布版，压缩代码，合并

1. 文件目录结构

一个文件一个对象？ 相关的文件相同目录

类库代码独立，分出lib文件夹 测试代码

Build src test

十四、Ant

所有的自动化系统都需要配置 ,build.xml 如何编译 ，如何管理各种依赖， 合并css，合并html ，js？ 校验jslint

文件精简和压缩 去掉注释，去掉无用空格，换行 避免出错需要用解析型压缩工具

gzip 响应头content\_encoding:gzip

运行时压缩：服务端进行，发送压缩的html css js

构建时压缩：

十四、 文档生成工具 jsdoc

从代码生成文档

/\*\*

\*函数解释

\*参数解释

\*返回值解释

\*/

十五、自动化测试

跨浏览器

基于浏览器

附录风格指南

A4运算符 间距 前后一个空格 包括 = &&

A5括号间距 （a + b） 靠边

括号内有括号时

调用和数组是 a() a[]

A6 对象直接量 缩近 属性 a: b;

A7 对象方法属性 应该生成文档

注释生命 TODO下一步 HACK捷径 XXX有问题需修复 FIXME REVIEW改动需评审

A8 变量声明 单 var 换行 变量和赋值号缩进

A9 函数声明 func() 匿名函数 function() { 和括号无空格

A10 命名 下划线只用于私有变量 方法 （不被外部调用，只在对象内部被调用）

名词 动词

构造函数 类名 非动词

常量 大写，下划线分隔词

A 7.8 赋值 右侧比较 a = (i < count);

A 10 三元表达式 只用于赋值 a= (a > b)a : b; 不要用于代替if函数调用

A11 语句 分号结数

返回语句 return a;

复合语句 {}

If 不允许省略花括号

If () {

} else if ( ) {

}

花括号的尾应该和 **起始的层级**对齐 注意是和相应层级同缩进 如果是最外层应该左对齐

For语句： for ( )不允许声明变量，除非let if () {}

For in for of怎么弄？ 变量应该声明在哪个作用域？ 千万不要不声明直接for (i) 这样会渗透到全局

Swith () { case :一个缩进 case后的语句再缩进一下

空行： 两行 类和接口之间

单行 方法之间 局部变量和方法头之间 注释之前 语句逻辑块之间

空格： a [] 跟括号时 （函数除外） 参数列表 运算符号 一元不用空格 for表达式应该隔开 括号内分隔 [a, b, c, ]