一. JAVA 语言代码规范

（1）为包，类，方法，变量取一个好名字，使代码易于理解

1.能清晰的表达意图：使用完整的描述性单词，避免使用单个字母，未形成惯例的缩写来命名。例如：int elapsedTimeInDays 要比 int d 好得多；

2.避免造成误导：有误导的名字比表达不清的名字更有危害性，比如 String accountList 其实并不是一个 List，a=1;是数字 1 还是字母 l;TTLCONFUSION 于 TTLCONFUSION 名称太相似；

3.避免不必要的编解码：代码被人阅读的次数会很多，因此要注意可读性，避免不必要的人脑编解码，比如在 JAVA 中不建议采用匈牙利命名法，Java 是强类型的语言，且 IDE 已经很先进，在编译时就能发现类型错误，因此匈牙利命名法无用武之地，况且它可能导致错误信息，比如 PhoneNumber strPhone 当类型变更后，名称未变更导致提供错误信息；

4.能区分出意思：比如以下名称，product,productInfo,productDate 表达的意思差不多，让人从名称并不能区分出它们各自代表的东西有什么不同，因此建议不要在变量/类名后加 info,data,object 等一般意义的词；

5.不用或少用缩写：小于 15 个字母的一般不用缩写。超过 15 个字母的，可以采用去掉原因字母的方法或者行业内约定俗成的缩写，且缩写保持驼峰格式，不要编造不符合习惯的缩写。比如 serviceDataPoint,可以缩写为 svcDataPnt,不可缩写成 SDP.如果要用缩写，仅第一个字母大写其余小写如:getHTTPRequest 改为 getHttpRequest;

（2）禁止使用魔鬼数字

直接使用数字造成代码难以理解，也难以维护。应采用有意义的静态变量或枚举来代替。

例外情况：有些情况下，如循环或比较时采用数字 0，-1,1,这些情况可采用数字；

（3）常量命名，由全大写单词组成，单词间用下划线分割，且用 static final 修饰；

示例：

错误：static int MAXUSERNUM = 200;　　static String s = "Launcher"

推荐：static final int MAX_USER_NUM = 200; static final String APPLICATION_NAME = "Launcher"

（4）变量，属性命名，使用名词，并采用首字母小写的驼峰命名法；

驼峰命名是指第一个单词字母使用小写，剩余单词首字母大写其余字母小写的大小写混合发。含有几个意义的属性，名称尽量包含复数；

示例：

不好：String customername;　　List<String> u = new ArrayList<String>;

推荐：String customerName;　　List<String> users = new ArrayList<String>;

(5)方法的命名，用动词和动宾结构，并采用首字母小写的驼峰命名法；

格式如下

get+非布尔属性名();

is+布尔属性名();

set+属性名();

has+名词/形容词

动词()

动词+宾语()

其中动词()主要用在动作作用在对象自身上，如 document.print();

(6)类和接口的命名，采用首字母大写的驼峰命名法；

类的命名不应用动词，而应用名词，比如 Customer,WikiPage,Account,避免采用类似 Manager,Processor,Data,Info 这样模糊的词；

示例：

不好：public class info{}

推荐：public class OrderInformation()

(7)包的命名，由一个或若干个单词组成，所有的字母均为小写；

包名采用域后缀倒置的加上自定义的包名，采用小写字母，都应该从 com.taobao 开头(除一些特殊原因)，再加上产品名称和模块名称，部门内部应该规划好包名的范围，防止产生冲突；

示例：com.taobao.mobilecontrol.views;

二. JavaScript 代码风格

（1）基本格式

1.缩进

建议每级 4 个空格，可以给编辑器设置 tab = 4 个空格，自动转换

2.分号

不要省略分号，防止 ASI（自动插入分号）错误

3.行宽

每行代码不超过 80 个字符，过长应该用操作符手动断行

4.断行

操作符在上一行末尾，且下一行缩进 2 级，如果是赋值语句，还应该和等号后面部分对齐

5.空行

函数声明与函数声明、变量声明与函数声明、函数内部的逻辑块之间都应该有空行隔开

6.命名

变量名/函数名：Camel（驼峰）规则，首词首字母小写，后续词首字母大写，其余部分小写

常量名：C 语言式，全大写，下划线分词

构造函数：Pascal 规则，所有词首字母大写，其余部分小写

7.字面量

字符串：双引号包裹，断行用[+]操作符，*不要*用\转义字符

数值：*不要*省略小数点前后的部分，*不要*用八进制形式

Null：只把 null 当作 Object 的占位符，*不要*用来检测形参，也*不要*用来检测未初始化的变量

Undefined：应该把所有对象都初始化为 null，以区分未定义和未初始化

对象字面量/数组字面量：*不要*用构造函数方式声明对象和数组

(2)注释

1.单行注释

行尾：用 1 级缩进隔开代码，而且//后面要有一个空格

独占一行：用来注释下面，要与被注释的代码保持相同的缩进

行首：用来注释多行代码

2.多行注释

用来包裹大段注释，推荐 Eclipse 风格，例如

注意：

a.多行注释上方留一个空行

b.*星号后面留一个空格

c.多行注释至少三行（因为第一行和最后一行后面不加注释）

3.在哪里添注释

a.不能自解释的代码

b.故意的，但看起来像是有错的地方

c.针对浏览器的 hack

4.文档注释

应该给各个函数添注释，包括功能描述、参数、返回值、抛出的错误等等，例如推荐的 Eclipse 风格：

(3)语句和表达式

1.花括号对齐方式

建议行尾风格，不推荐次行风格

2.块语句空格

if 后的圆括号部分前后各有一个空格，例如：

3.switch 语句

缩进：case 与 switch 对齐，break 缩进 1 级

case 贯穿：用空行或注释//falls through 表明 case 贯穿是故意的

default：保留 default 或者用注释//no default 表明没有 default

4.with 语句

不用

5.for 循环

所有变量都应该在函数体顶部声明，包括 for 循环初始化部分用到的变量，避免 hosting（提升）引发 bug（可能会屏蔽全局变量）

6.for-in 循环

*不要*用来遍历数组，用的时候记得加上 hasOwnProperty 过滤，如果故意遍历原型属性，应该用注释说明

(4)函数、操作符

1.变量声明

函数体 = 变量声明 + 函数声明 + 逻辑语句。用空行隔开各个部分

2.函数声明

先声明再使用，*千万不要*把函数声明放在 if 分支里，因为浏览器理解不同，而且 ES 没给标准

3.函数调用

圆括号前后都*不加*空格，避免和块语句混淆

4.匿名函数立即执行

把立即执行的匿名函数用圆括号包裹，避免与匿名函数声明混淆

5.严格模式

*不要*在全局作用域开严格模式，只在函数内部开，给多个函数开可以用匿名函数立即执行限定严格模式的作用域

6.判断等于

只用===和!==

7.eval

不用 eval()和 new Function()，用匿名函数优化 setTimeout()和 setInterval()

8.基本包装类型

*不要*用 new Boolean(), new String(), new Number()