**一、什么是BUG**

现在人们将在电脑系统或程序中，隐藏着的一些未被发现的缺陷或问题统称为bug（漏洞）

# **二、关于BUG的5个定义是什么？**

Bug的定义可以很广泛，在软件使用过程中所出现的任何一个可疑问题，或者导致软件不能符合设计要求或满足消费者需要的问题都可以是Bug，即使这个Bug在实践中是可行的  
**（1）Bug可以真正消灭吗？**  
可以说，没有任何一个产品没有Bug，也永远不可能找出并修复所有的Bug。在修复了旧的Bug的同时，往往又会产生新的Bug  
以微软的经验，每修复三到四个Bug，一般又会产生一个新的Bug  
所以，Bug提交开发人员解决后，可能会有以下几种类型的反馈  
?  
1。Fixed：表示Bug已经被修复或更正了  
2。Duplicated：表示测试人员所找到的某个Bug已经被别人找出来了。  
3。PostPoned：表明这个Bug不是很重要，在当前阶段不用进行更正了，或者更正这个Bug风险太大，Bug本身又不会造成大的影响  
4。By Design：测试人员认为是Bug，不符合逻辑，也不符合用户的需求，但开发人员则认为是按照项目经理的设计做的  
5。Not repro：以前出现的某个Bug自动消失了，可能是处理其他Bug的时候把这个Bug一并修复掉了  
6。Won't Fix：这个Bug是一个错误，还没有重要到非要更正不可的地步，完全可以忽略不计

在解决BUG的同时还应该考虑性能问题，也就是一定要保证软件运行得很好，非常快，没有内存泄漏，不会出现越来越慢的情况

**一、移除超链接的虚线**

　　FireFox下,当你点击一个超链接时会在外围出现一个虚线轮廓. 这很轻易解决, 只需要在标签样式中加入：

outline:none.

a{

outline: none;

}

**二、chrome下p里套div造成解析错误**

<p>的合法内容为<abbr>, <audio>, <b>, <bdo>, <br>, <button>, <canvas>, <cite>, <code>, <command>, <data>, <datalist>, <dfn>, <em>, <embed>, <i>, <iframe>, <img>, <input>, <kbd>, <keygen>, <label>, <mark>, <math>, <meter>, <noscript>, <object>, <output>, <picture>, <progress>, <q>, <ruby>, <samp>, <script>, <select>, <small>, <span>, <strong>, <sub>, <sup>, <svg>, <textarea>, <time>, <var>, <video>, <wbr>，其中不包括<div>

<p>

<div></div>

</p>

在chrome里解析完之后就成了

<p></p>

<div></div>

<p></p>

**四、插入多个img标签时，换行输入代码，图片左右之间有一个空隙**

解决方案：

                让代码在同一行显示

                或者img{float：left；}

**五、img标签与文本之间或者与input标签上下不对齐**

 解决方案：

                img{vertical-align：middle；} 行不通 通过定位

**六.超链接访问过后hover样式就不出现的问题**

被点击访问过的超链接样式不在具有hover和active了,很多人应该都遇到过这个问题，解决方法是改变CSS属性的排列顺序: L-V-H-A

a:link {color: #1f3a87; text-decoration:none;}

a:visited {color: #83006f;text-decoration:none;}

a:hover {color: #bc2931; text-decoration:underline;} 改变时hover样式不生效

a:active {color: #bc2931;} 改变点击时为link的样式

**七.如何解决Chrome在应用transition时页面闪动的问题？**

方法:-webkit-transform-style:preserve-3d;或-webkit-backface-visibility:hidden;

备注：在Chrome下，使用过渡效果transition时有时会出现页面闪动

**八、 Uncaught TypeError: Super expression must either be null or a function, not undefined**

错误原因：就在super之前查找代码的错误，一般是拼写错误；大都是React.Component的C没有大写；

**九、使用React.PropTypes**

PropTypes 对象已从 React 中删除。 其过去是以 React.PropTypes 的形式被使用，但不能再使用它了。

相应的，你需要：

npm install prop-types

import PropTypes from 'prop-types'

然后就可以使用它啦，如： PropTypes.string 。

如果你错误地使用了 React.PropTypes，会得到这样的错误提示：

TypeError: Cannot read property 'string' of undefined

PropTypes 提供一系列验证器，可用于确保组件接收到的数据类型是有效的。在本例中, 我们使用了 PropTypes.string。当传入的 prop 值类型不正确时，JavaScript 控制台将会显示警告。出于性能方面的考虑，propTypes 仅在开发模式下进行检查。

**十、3像素BUG**

条件 浮动块元素与未浮动元素处于同一行。解决：设置非浮动元素浮动。

**十一、a>b**

### 10>”11” false

“10”>11 false

“10”>”11” false

1. ”10” false

“9”>10 false

“9”>”10” true

可以看到之前都还是很正常的嘛，但是到了'9'与'10'的对比就出现问题啦。  
这是因为在js中两个字符串对比的是他们的ASCII码。所以就会出现问题  
解决方法:  
写的时候注意一点就好啦

## **input放在a标签里面单击不能获取input的光标（IE环境下）**

双击才可以获得焦点，目前有的解决方案：

不要给a标签添加href属性；

不要在外面套上a标签。

1. **标签属性设置为inline-block出现隐藏的长度为4px的间隙**

首先说下：这不是bug，这是符合规范的应该有的表现，但是有时这种间距对我们布局或处理兼容性产生影响，因此需要去除掉它

<ul>

<li>开心</li>

<li>愉悦</li>

<li>郁闷</li>

</ul>

解决1：

ul{

font-size:0;

}

li{

font-size:14px;

}

**结尾：在开发过程中如何尽量避免减少BUG的出现**

1、正式开发前，做好准备工作。一个开发任务分配下来，不要直接上手就写。可以先在自己的小本子上画画写写，勾勒出大概的项目设计，业务难点。有没有重点需要攻克的技术，提前跟组长或技术经理报备好。这样再着手写代码，就不容易出现卡壳，影响心情，导致 Bug 的产生。

2、提前做好项目的任务规划，合理安排开发时间，减少加班的次数也能一定程度上少写一点 Bug 。毕竟有哪个程序员愿意天天加班呢？

3、开发过程中，要及时测试。不要等到全部的功能写完在一起测试，这样很容易出大问题。最好的方式就是做完一个小功能，就自己先跑起来，点点看看。及时测试，及时发现问题。早发现，早解决。

4、要重视 IDE 的报错或者警告，不要觉得能跑起来，就没问题。可能当前开发的部分还无关痛痒，但是多积累几个警告以后，可能会引发更大的问题。所以每当遇到 IDE 报错的时候，自己一定要点开看看，不要就随他去了。

5、整个功能写完以后，先自己多自测。全面的考虑使用场景，只要能想到的特殊环境都要测试一下看看，不要想着有些环境很少有人用，就不测试。往往这些场景下的漏洞，就可能引发崩溃。

6、自己平时有事没事就可以进行 Code Review。但是看自己的代码容易出现当局者迷，所以最好找个要好的同事，互相 Review 代码。省的在提交之后，被组长 Review 不通过，打回重写。。。

7、常见的 Bug 或者已经犯了几次的问题一定要记录下来，空下来的时候，经常多翻翻。好记性不如烂笔头

8、可以向资历老，技术牛的同事请教下经验，一般程序员还是乐于助人的多，不会藏着掖着。开发经验多了，避免不了遇到的Bug也会多一些。每个老师傅都会有些心得，所以需要学会向同事请教。

虽说自己现在比几年前仔细多了，但今年还是“创造”了不少 Bug ... 深知自己还差得很远，需要提高和注意的地方还有很多。

自己写的 Bug ，流着泪也要改完。