现在来简要地分享一下代码的行文规范。

虽然编译器不在乎空白字符，哪怕全部连起来也仍然是“正确的”代码。然而，一份整齐的代码会让人赏心悦目，阅读与加工变得简单方便，生活也变得清爽了起来，何乐而不为呢？

我们以Html代码为例，简要谈谈漂亮的代码是怎么写出来的。

样例代码：cook.html。一份短小精悍的佳作。



观察这份代码，可以找到以下简洁的规律呢。

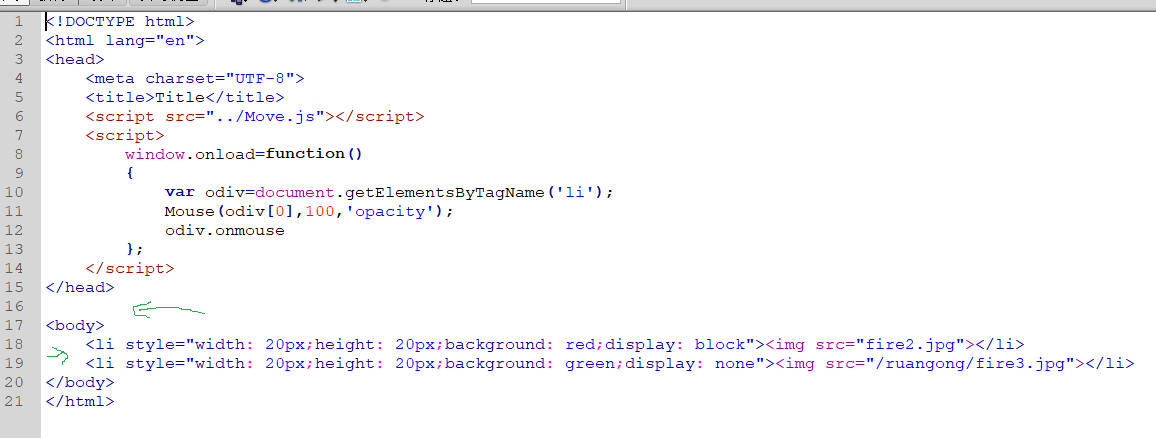
（为了阅览方便，规定一个Tab是4个空格。）

首先，除非是单行，换行缩进几乎都是成对存在的。这样带来一个方便：



标箭头的部分成对存在。有左括号就有右括号，有斜杠终止标签就有开头的标签。

然后观察这部分：



在head和body之间插入一个空行，表示分段。

如有必要，可以在每个不相干的大结构之间插入空行，表示分段。但是，不要在结构内部插入空行，这容易引起混乱。

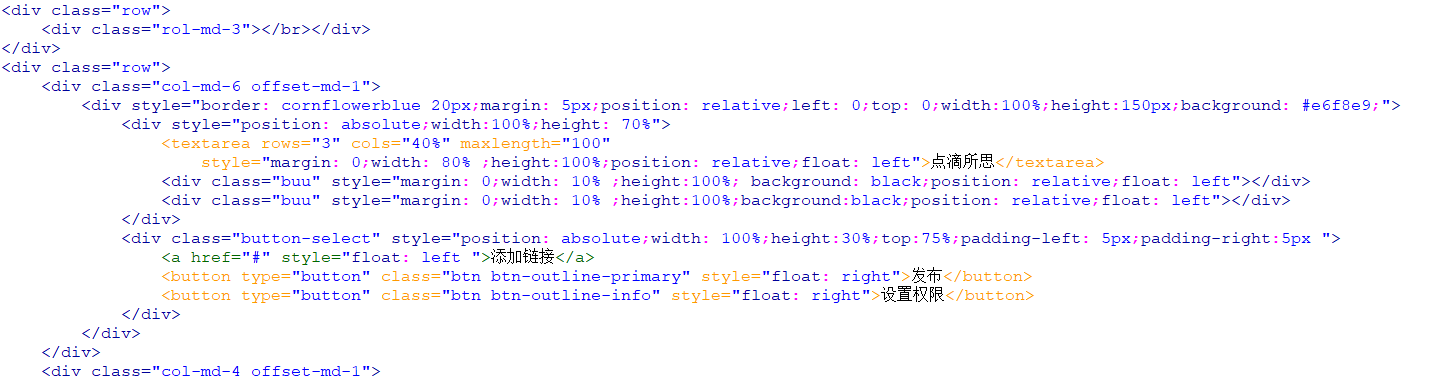
在每个层级结构之间，应当写入一个缩进。

在这个例子中，body中的两行可以展开写（写成6行，两层缩进），也可以合并写（如图）。

如果有多层级结构，只有最里边的层级可以合并写，外面的层级必须展开。

如果一个层级过长，导致换行，建议不要合并写，而采用展开写。

关于“一行过长”的问题解释，可以看下面这个例子：



这是index.html的一个局部。看其中textarea那一行。

如果写成一行，将导致换行至最左。

解决方案是，将中间部分最长的属性“style”前面换一行，并新加一个缩进（这里的缩进表示承接），使得可以容纳下这一行代码。

再看一个例子：



这是user.html的一个局部。一共只有style一个超长的属性，采用上述策略已无法保证不发生超出一行的现象。

这时可以将其中的每一个分量并列，在下方多写一个缩进并列出：

第一行的末尾是左双引号。

后面每一行，均为分量开头，分号结尾。直至最后一行结束。

这样的代码，最大限度地保障了“层次明确，上下对齐”的目标，不仅看着美观，给debug也带来了巨大的方便。