**基于jq的纯js版json配置工具开发**

——————2019.7.30韵方

首先来看一下成品吧，没有写css略微有点小丑啊不要介意哈：

****

然后直接上代码哈，代码其实一共可以分为以下的这么几步完成。1.编写html组件（包括目前看到的所有按钮和text框）。2.编写+-号的逻辑，包括自动生成编号和文本框以及他们的id。3.编写生成json按钮的代码。此处分为3小步。（1）将原始字符串拆分成两个队列，一个队列存放文本内容，另一个队列存放配置字符数字（2）将要配置的字符逐行存入二维数组中，二维数组每行为一个配置，每列为配置的一行数据（3）合并字符串并输出在对应的位置上。花里胡哨的就不多说了，直接上代码。

首先是html模块的代码，此处不过多解释了，就是一些很基础的组件和属性。



然后是加减配置字符div模块的代码，这里要注意配置textarea的id，因为下面会用到哦。有加模块和减模块两个模块。



然后就是对原始字符串进行解析，使当他读到<n>时会自动去匹配配置好的字符串，然后进行一一的对应，替换成我们最终要的字符。这里我使用了两个队列的数据结构，一个队列存放无<n>的字符串，另外一个队列存放n的值，以便为我们去下面的配置好的字符那里去匹配。

这里要提一下js入队列出队列的基本语法。

.unshift()//入队列

.pop()//出队列



再然后就是将需要配置的字符存起来，以便与去匹配需要匹配的字符了。这里我用了二维数组的形式进行储存。

****

最后就是合并单元格的操作了，将队列拿来交叉一一出队列，其中一个数字的队列去匹配二维数组中的数据，逐行逐行重复执行，从而得到最后的结果。



注意了，在此时我碰到了一个问题，就是每次逐行重复执行的时候，需要对队列进行一次进出的操作，当然这可以用环状队列来实现，但我最初是用普通队列复制一个新的队列来实现，这时我发现只要改变新复制的队列，原先的队列也会发生改变。

经过查找相关的资料，我知道了原来js的队列使用的是地址引用，而我们一般采用=的形式赋值，只会对其地址进行复制，所以改了新的，原来的也就变了，所以，我们需要采取强复制的方式进行赋值并且进行出入的操作。

以下为强赋值的代码，中间经过了一层普通的值的引用，所以就不会只复制地址了。



最后，介于es6新语法的规范，上述代码中所有的var均可用const代替，谢谢。

知识点：js弱复制与强复制的产生原因，区别和解决办法

Js队列的基本操作