2017-12-2

现在开始JavaScript任务2

做完这个任务，就可以掌握利用JavaScript进行页面跳转和传参；使用JavaScript对input等页面元素进行复杂的操作；进一步理解JavaScript数组；了解JavaScript正则表达式

首先，我们来看看需要什么样的技能

* JavaScript页面跳转
* JavaScript获取input标签value
* JavaScript数组
* JavaScript数组乱序
* JavaScript键盘事件
* JavaScript正则表达式

刚刚已经下载了需要的软件，现在看看它的页面是怎样的

不对，页面之间是怎样跳转的呢？

之前使用的是<a>标签，href设置需要跳转到的页面就好

好的，页面跳转功能好像可以这样搞。

获取input标签的value

试着写一写。完成。

JavaScript数组，在之前的学习中已经学了一点。

JavaScript数组乱序？

先休息一下……

11:30:51

继续

那就抄袭这个算法咯。先试试是怎样的。

可以用。

接下来是键盘事件

onkeydown 事件，会在用户按下一个键盘按键的时发生。感觉和onclick差不多

onkeydown="SomeJavaScriptCode"

|  |  |
| --- | --- |
| **数字值** | **实际键值** |
| 48到57 | 0到9 |
| 65到90 | a到z（A到Z） |
| 112到135 | F1到F24 |
| 8 | BackSpace（退格） |
| 9 | Tab |
| 13 | Enter（回车） |
| 20 | Caps\_Lock（大写锁定） |
| 32 | Space（空格键） |
| 37 | Left（左箭头） |
| 38 | Up（上箭头） |
| 39 | Right（右箭头） |
| 40 | Down（下箭头） |

？？

算是可以用得上了？

正则表达式

继续

试试使用正则表达式。跟着例子试试替换

var aft\_org=org.replace(/something/gi,"any???");

例如这样的语句，/被替换内容/后面的g代表把所有复合条件的匹配，i表示忽略大小写(一次匹配一个)，m表示多行匹配，可以加起来一起使用。

[]直接放在//里面即可。

|  |  |
| --- | --- |
| [abc] | 查找方括号之间的任何字符。 |
| [0-9] | 查找任何从 0 至 9 的数字。 |
| (x|y) | 查找任何以 | 分隔的选项。 |
| \d | 查找数字。 |
| \s | 查找空白字符。 |
| \b | 匹配单词边界。 |
| \uxxxx | 查找以十六进制数 xxxx 规定的 Unicode 字符。 |
| n+ | 匹配任何包含至少一个 n 的字符串。 |
| n\* | 匹配任何包含零个或多个 n 的字符串。 |
| n? | 匹配任何包含零个或一个 n 的字符串。 |

看来知识点已经差不多了。开始。

似乎在样式上面要花点时间……

先这样吧。

19时1分44秒

继续

刚刚稍微做了一下页面。大概是这样的效果，还有用了任务1的技能点，获取class的元素(input)的value，所有的input都是一样的class，可以通过document.getElementsByClassName('num-input')[i].value来取其值。

现在还需要做的就是应用onkeydown事件。事实上，我已经忘记了……

这个例子有点奇怪……使用字母上面一排的数字是无法输入的，但是使用小键盘的数字键就可以，应该是他们的keyCode不一样吧。

改造一下正则表达式，那就可以只能输入数字了。

……

……

……

好的。改造完成，应用到页面上面，也是成功的。

接下来就是应用游戏逻辑，人数上面有限制的。

|  |  |
| --- | --- |
| 总人数 | 杀手数 |
| 4-5 | 1 |
| 6-8 | 2 |
| 9-11 | 3 |
| 12-14 | 4 |

按照这个的话，写一些选择语句就好。

好像可以了。

先这样咯。