目录

[1. 概述 2](#_Toc11065)

[2. 示例场景 2](#_Toc28523)

[3. 使用步骤 3](#_Toc25799)

[4. JS语法说明 10](#_Toc22678)

## 概述

之前节目上的病人信息项或住院病人信息项元素只能固定一种文本颜色或背景效果，无法根据其显示内容设置成不同效果，因此，需要一种能根据显示内容来灵活设置其效果的解决方案。目前，已采用节目JS脚本编辑的这种方式，其使用难度一般，功能上基本满足上述需要。

## 示例场景

1. 场景一，当前显示的正在就诊数据见表一：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 病人姓名 | 状态备注 | ...... |
| 1 | 萧可玉 | 高温，急 |  |
| 2 | 闵生龙 |  |  |
| 3 | 雷婷姣 | 高温，急 |  |
| 4 | 窦茂进 | 急 |  |
| 5 | 云瑾颖 |  |  |
| 6 | 傅梅 | 高温 |  |
| 7 | 杜琳素 | 急 |  |
| 8 | 皮勇毅 | 急 |  |
| 9 | 任元全 |  |  |
| 10 | 袁国胜 |  |  |
| ...... |  |  |  |
|  |  |  |  |

表一

若当前有以下3点需求：

1. **仅**“高温”状态的病人，其姓名的字体颜色显示成红色（RGB：#FF0000），并且背景显示成蓝色（RGB：#0000FF）；
2. **包含**“急”状态的病人，其姓名的字体颜色显示成绿色（RGB：#00FF00），并且背景显示成红色（RGB：#FF0000）；
3. 其他状态的病人，其姓名的字体颜色显示成白色（RGB：#FFFFFF），并且背景显示成黑色（RGB：#000000）；
4. 场景二，当前显示的正在就诊数据见表一：

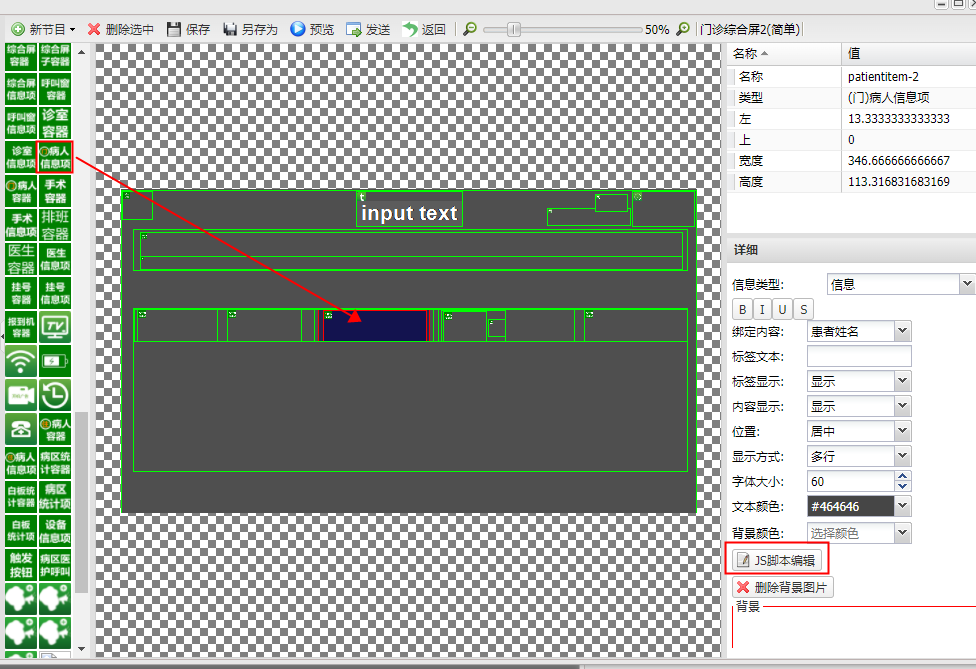
但需求改为以下3点：

1. 包含“高温”**和**“急”状态的病人，其姓名的字体颜色显示成红色（RGB：#FF0000），并且背景显示成蓝色（RGB：#0000FF）；
2. 包含“高温”**或**“急”状态的病人，其姓名的字体颜色显示成绿色（RGB：#00FF00），并且背景显示成红色（RGB：#FF0000）；
3. 其他状态的病人，其姓名的字体颜色显示成白色（RGB：#FFFFFF），并且背景显示成黑色（RGB：#000000）；

## 使用步骤

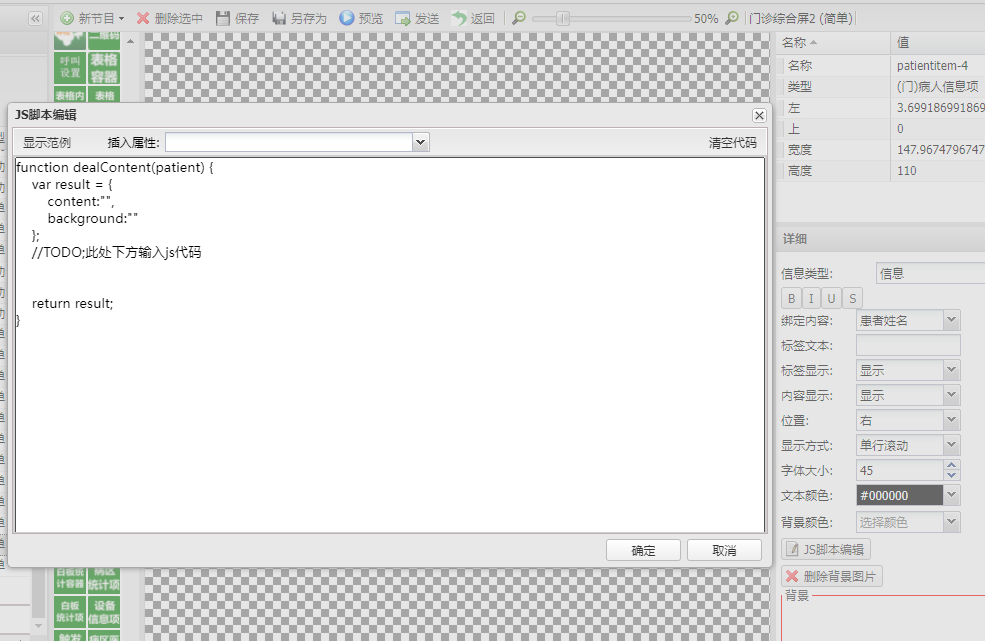
## 根据示例场景一，下面来一步步实现。

1. 病人信息项或住院病人信息项目前支持JS脚本编辑功能，图1所示，



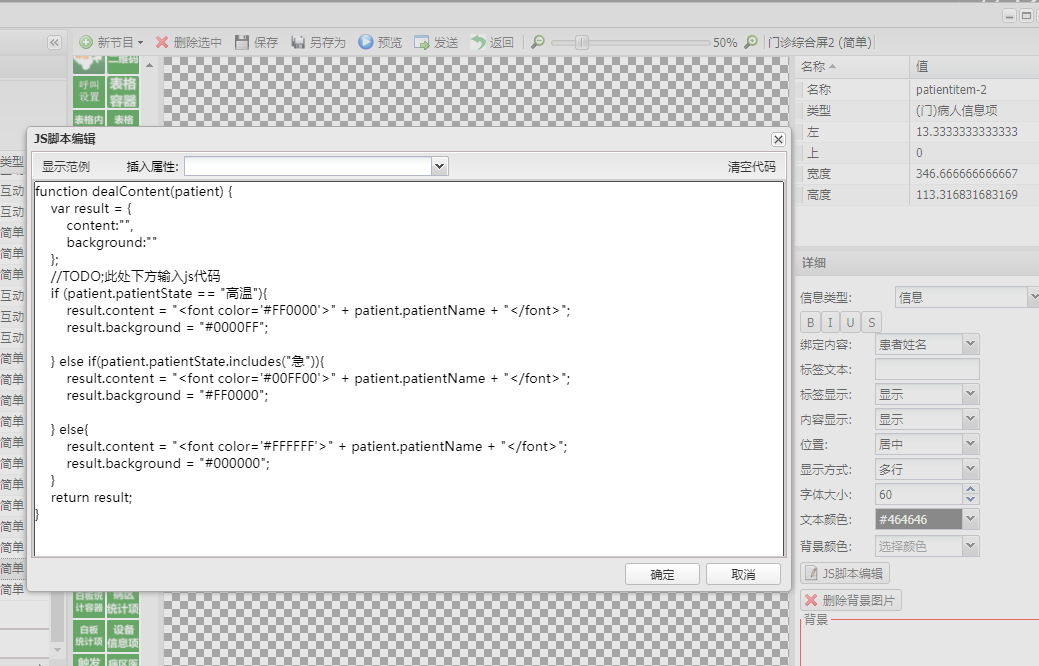
（图1）

1. 点击JS编辑按钮后，会弹出JS脚本编辑弹窗，见图2，代码编辑处默认会有模板生成，自动生成的代码不要作修改，根据提示在TODO下方输入功能代码即可。

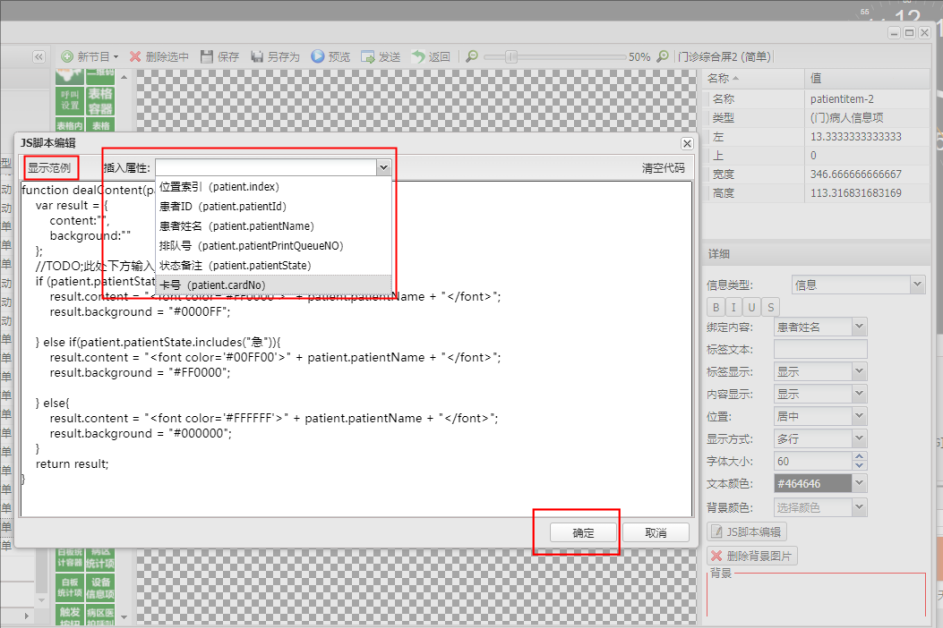


（图2）

1. 在指定位置输入JS代码后如图3所示；为方便进行编辑，您可以在编辑期间查看范例；或当用到某个属性时，可点击下拉框进行查看，并可在光标处插入属性值，见图4。



（图3）



（图4）

1. 编辑完成后，若需保存本次修改就点击“确定”按钮，取消本次修改就点击“取消”按钮或点击右上角的X按钮;
2. 若要完全放弃使用JS脚本编辑功能代码，可点击“清除代码”按钮，并点击“确定”按钮保存。
3. 为实现《示例场景一》的需求，JS功能代码可如下编辑：

function dealContent(patient) {

var result = {

content:"",

background:""

};

//TODO;此处下方输入js代码

**if (patient.patientState == "高温"){**

**result.content = "<font color='#FF0000'>" + patient.patientName + "</font>";**

**result.background = "#0000FF";**

**} else if(patient.patientState.includes("急")){**

**result.content = "<font color='#00FF00'>" + patient.patientName + "</font>";**

**result.background = "#FF0000";**

**} else{**

**result.content = "<font color='#FFFFFF'>" + patient.patientName + "</font>";**

**result.background = "#000000";**

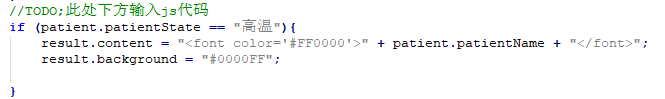
**}**

return result;

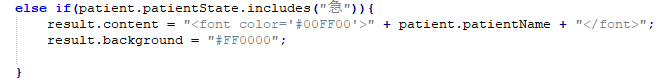
}

代码解读：

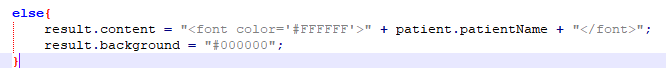
1. if(...)是条件，而{.......}括起来的是设置效果区；
2. 对于第一个if括起来的语句，若病人状态等于“高温”，则设置文本颜色为红色（result.content=....#FF0000.....），背景为蓝色（result.background=”#0000FF”）；



1. 对于第二个else if括起来的语句，若病人状态包含“急”，则设置文本颜色为绿色（result.content=....#00FF00.....），背景为红色（result.background=”#FF0000”）；



1. 对于第三个else括起来的语句，若不符合以上条件（②和③）的其他病人，则设置文本颜色为白色（result.content=....#FFFFFF.....），背景为黑色（result.background=”#000000”）；



1. ”==”表示等于，“**.includes**”表示包含，比如病人状态为“高温，急”，那么该病人包含“高温”的状态，也包含“急”的状态，等于“高温，急”的状态，不等于“急，高温”的状态，以及不等于“高温”的状态；
2. 《示例场景一》的最终实现效果，见图5，图6：

（图5）



（图6）

1. 目前开始处理《示例场景二》的需求，制作节目过程与(1) ~ (6)步骤一致;
2. 为实现示例场景二的需求，JS功能代码可如下编辑：

function dealContent(patient) {

var result = {

content:"",

background:""

};

//TODO;此处下方输入js代码

if (patient.patientState.includes("高温") && patient.patientState.includes("急")){

result.content = "<font color='#FF0000'>" + patient.patientName + "</font>";

result.background = "#0000FF";

} else if(patient.patientState.includes("高温") || patient.patientState.includes("急")){

result.content = "<font color='#00FF00'>" + patient.patientName + "</font>";

result.background = "#FF0000";

} else{

result.content = "<font color='#FFFFFF'>" + patient.patientName + "</font>";

result.background = "#000000";

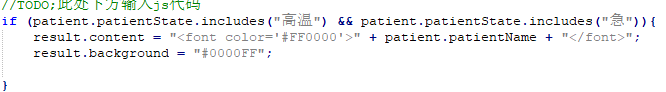
}

return result;

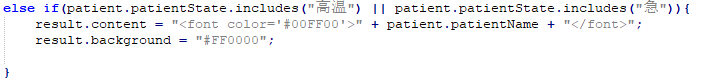
}

代码解读：

1. if(...)是条件，{.......}括起来的是设置效果区；
2. 对于第一个if括起来的语句，表示当病人包含“高温”**和**“急”的状态时，就设置文本颜色为红色（result.content=....#FF0000.....），背景为蓝色（result.background=”#0000FF”）；



1. 对于第二个else if括起来的语句，表示当病人包含“高温”**或**“急”的状态时，就设置文本颜色为绿色（result.content=....#00FF00.....），背景为红色（result.background=”#FF0000”）；



1. 对于第三个else括起来的语句，表示当病人不符合以上条件（②和③）时（即其他病人），就设置文本颜色为白色（result.content=....#FFFFFF.....），背景为黑色（result.background=”#000000”）；



1. “**.includes**”表示包含，比如病人状态为“高温，急”，那么该病人包含“高温”的状态，也包含“急”的状态；“&&”表示并且，“||”表示或者.
2. 《示例场景二》的最终实现效果，见图7，图8：



图7



图8

## JS语法说明

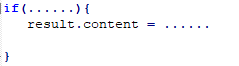
1. if(...)或else if(...)或else都是条件语句，而{.......}括起来的是设置显示效果区；只有条件符合之后才会执行{.......}里的设置效果，任何一个条件符合之后，后续的条件都会被忽略掉；优先从第一个条件开始进行判断；
2. “==”表示等于，“.includes”表示包含；
3. “&&”表示并且，“||”表示或者；
4. “//TODO;此处下方输入js代码” 和 “return result; ”语句之间可填写的JS代码一般形式有以下几种。

(提示：可在result.content语句下方加设背景result.background = ......;)

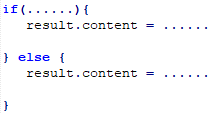
形式一：



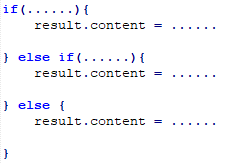
形式二：



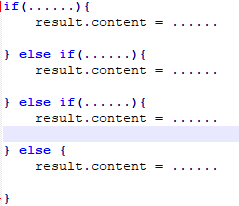
形式三：



形式四：



形式五：



若result.content或result.background属性没有设置任何值，会默认使用原节目元素上配置的显示效果。result.background属性可以赋颜色值（RGB）、相对路径、或者图片访问链接。

**注意：相对路径写法示例**

result.content = “/opt/resources/picture/1.jpg”;

其中“/opt/resources/”是固定的，后面的“picture/1.jpg”路径具体看实际图片放置的位置，如下图所示。

