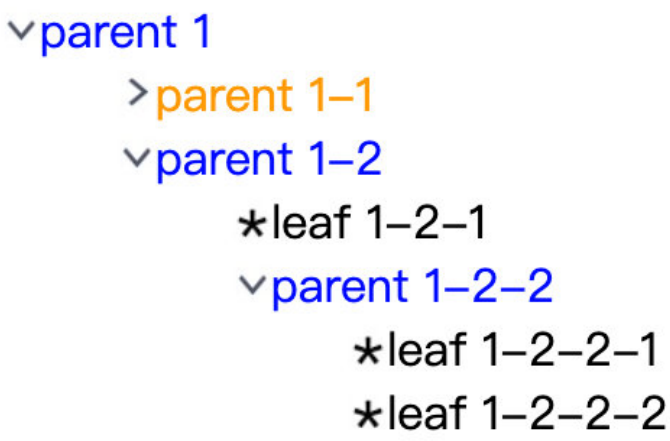
第三次作业实验报告

1.实验介绍：

使用js,css和html文件开发一个树形控件，要求能够完成正常的结构组织，在点击某个节点时，可以展开或者收起其子节点，并且出发回调函数，在展开或收起时采取动画过渡。

实现效果如下：



2.具体实现：

具体实现思路如下：

首先需要解析代表输入树的多维数组。定义了两个类`Tree`和`Node`,`Tree`类代表构建的树，`Node`类代表树上每个节点。

`Node`类具有id,title,children,expand等属性，其中children和expand具有默认值。同时具有`generate(parent)`方法，调用后用于生成父节点为parent的子树并加入DOM，代码如下：

```
this.generate = function(parent) {
    let ele = document.createElement('li');
    if (this.children.length == 0) {
        ele.innerHTML = '<span class="leaf"+ onclick="toggle(this)" id=' +
this.id + '>' + '<img class="leafimg" src=' + imgsrc2 + '>' + this.title +
'</span>';

    } else {
        if (this.expand) {
            ele.innerHTML = '<span class="open"+ onclick="toggle(this)" id=' +
this.id + '>' + '<img class="openimg" src=' + imgsrc + '>' + this.title +
'</span>';
```

```
        } else {
            ele.innerHTML = '<span class="close"+ onclick="toggle(this)" id='
+ this.id + '>' + '<img class="closeimg" src=' + imgsrc + '>' + this.title +
'</span>';
        }
    }

    if (this.children.length > 0) {
        let nextpar = document.createElement('ul');
        ele.appendChild(nextpar);
        for (let item of this.children) {
            item.generate(nextpar);
        }
    }
    parent.appendChild(ele);
}
```

可以看到，在该方法中，我们对某个节点的两种情况进行了讨论，当该节点的children为空数组时，即该节点为叶子节点，则设置节点class为leaf，设置id等属性并绑定toggle函数，直接加入DOM树；如果该节点children不为空，则根据expand属性设置class为open或close，设定其余属性后创建ul标签作为其子节点的父辈容器，并递归调用`generate`函数。

`Tree`类具有container，tree属性，并且根据输入初始化clicked函数，并定义了初始化函数`init()`。tree属性即为根据输入生成的树的根节点。`init`函数用于生成树组件并加入DOM，其中数组建最外层为`<div>`标签，其`id`设置为Tree的`container`属性，随后调用根节点tree的`generate`方法生成完整的树并加入DOM。最后根据`<span>`的className决定初始是否展开。`init()`函数实现如下：

```
this.init = function() {
    let list = document.getElementsByTagName("body")[0];
    let warp = document.createElement('div');
    warp.id = this.container;
    warp.innerHTML = '<ul id="wrap"></ul>';
    list.appendChild(warp);
    this.tree.generate(document.getElementById("wrap"));

    let spanList = warp.getElementsByTagName("span");
    for (let span of spanList) {
        let content = span.nextElementSibling;
        if (content) {
            if (span.className == "open") {
                content.style.transform = "scaleY(" + 1 + ")";
                content.style.height = "auto";

            } else {
                content.style.transform = "scaleY(" + 0 + ")";
                content.style.height = 0;
            }
        }
```

```
    }
}
```

`Toggle(span)`函数在某个`<span>`被点击时被调用，该函数会调用回调函数`clicked(id)`函数，其中id即为被点击的`<span>`的id，然后根据目前该`<span>`的className决定要执行展开或收起，具体操作为修改类名以配合CSS，并在此处加入了动画操作。

```
function toggle(span) {
    let id = span.id;
    tree.clicked(id);
    let par = span.nextElementSibling;
    let node = document.getElementById(id);
    if (par) {
        if (span.className == 'open') {
            span.className = 'close';
            let arrow = span.getElementsByTagName("img")[0];
            arrow.className = 'closeimg';

            par.style.transformOrigin = "top";
            par.style.height = "0px";
            par.style.transform = "scaleY(" + 0 + ")";
            par.style.transition = "height .2s,transform .2s";
        } else {
            span.className = 'open';
            let arrow = span.getElementsByTagName("img")[0];
            arrow.className = 'openimg';

            par.style.transformOrigin = "top";
            par.style.height = "50px";
            par.style.transform = "scaleY(" + 1 + ")";
            par.style.transition = "height .2s,transform .2s";
        }
    }

}
```

关于动画：

本实验中，动画主要分为两部分：1.标签箭头的旋转 2.子节点的收起与展开动画

对于标签箭头的旋转，我采用了以下方法实现：首先制作了两个图片的DataURL内嵌于代码中，分别是 > 和 * 的图像，其中 * 用于叶子节点。然后使用`transform:rotate(90deg)`用于旋转90°作为展开后的节点箭头，使用`transition:transform .2s`用于设置持续时间以形成动画效果。

对于收起与展开动画，我是用了`scaleY`与`height`属性完成缩放，其中当展开时，设定为`scaleY(1)`，并设定`height=0`；当收起时，设定为`scaleY(0)`，并设置`height = auto`。这样就完成了缩放，并通过`transition`将上面的动作加入延迟时间用以形成动画过渡效果。

3.难点与问题：

在本次实验中，我先后遇到了以下几个难点与问题：

1.不了解如何使用css和HTML组件进行交互：

解决方法：上网查阅资料并了解了DOM的相关知识。

2.对于点击后的动画实现：

解决方法：最开始我直接使用>字符进行箭头的标注，但发现这样不是很方便旋转，因此我选择了插入图片用于做旋转操作。并且经查阅资料了解到transition属性来达到过渡动画的效果。

在子节点的展开与收起动作中，我先后尝试了以下几个方案：

- 使用`display:none`和`display:block`切换完成展开与收起动作，这个方法可以完成动作本身，但是不容易加入动画效果。
- 使用`visibility`属性设置是否可见与max-height设置最大高度实现缩放，这个方法适合加入动画，但是我发现，通过对一个节点设置`visibility: hidden`后，其自身会隐藏，但其子节点并不会，如果对其自身和所有子节点全部设置属性使其可以隐藏，则再次展开时恢复子节点原先状态较为困难，对此可以考虑使用两个属性四种状态同时控制：是否隐藏与是否展开。但是这样的方法过于繁琐。
- 最后我查阅资料，发现了`scaleY`和设置`height`的结合，可以完成上述两个任务的结合，能够正确缩放并很好地实现动画效果。