

第十三课 组件化开发过程

学习目录

- 搭建基础结构
- 编写面向过程代码
- 改写成面向对象组件化代码

一 . 搭建基础结构

每当我们开发一些 js 功能的时候，第一步要做的是先要熟悉整个功能业务逻辑，思考整体的基础结构，这个结构不仅包含 html\css，也包括相应的 js 逻辑结构，所以先要根据功能需求在页面中搭建一套基础结构，我们要封装的组件是一个模态框组件，有标题栏，有中间内容区域，有底部操作按钮。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

  <title>第十三课 组件化开发过程</title>

  <style>

    *{

      margin:0px;
```

```
        left:0px;

    }

    .model{

        width:350px;

        height:200px;

        border:1px solid #ccc;

        border-radius:10px;

        margin:100px auto;

    }

    .model .header{

        width:100%;

        height:35px;

        line-height: 35px;

        border-bottom:1px solid #ccc;

    }

    .model .header .title{

        display: inline-block;

        width:90%;

        text-indent: 15px;

    }

    .model .header .close{

        display: inline-block;
```



学习前端，最快的进步是持续！

```
width:10%;

float: right;

text-align: center;
}

.model .content{

padding:10px;

height:130px;

overflow-y: auto;

box-sizing: border-box;
}

.model .footer{

height:35px;

border-top:1px solid #ccc;
}

.model .footer span{

float:right;

margin:5px 10px 0px 5px;

box-shadow: none;

padding:1px 8px;

color: #fff;

font-size: 14px;

cursor: pointer;
```

```
}

.model .footer .confirmBtn{

    background: orange;

}

.model .footer .cancelBtn{

    background: green;

}

</style>

</head>

<body>

    <div id="wrap">

        <div class="model">

            <div class="header">

                <div class="title">星星课堂 web 前端培训</div>

                <div class="close">x</div>

            </div>

            <div class="content">星星课堂真好</div>

            <div class="footer">

                <span class="confirmBtn">确定</span>

                <span class="cancelBtn">取消</span>

            </div>

        </div>

    </div>

</body>

</html>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

二．编写面向过程代码

编写好基本结构之后，我们可以按照面向过程的方式来动态的新建模态框，在我们编写好的面向过程的模态框中，我们可以已进入页面就新建一个模态框，模态框的标题和内容可以通过参数来指定，并且我们可以通过点按钮显示模态框，也可以点关闭隐藏模态框，点确定按钮和取消按钮也可以用对应事件函数来编写我们的业务逻辑。

但是这里有个最大的问题，如果我想再来一个一样的模态框，我的事件操作就要重新写一遍，假如我想中间内容显示成列表内容，我也要重新构建一套模态框，因为我如果在先用的模态框的构建函数中改变就会影响到其他普通内容的模态框，因此这样就增加了代码量，所以我们需要又到组件化原理来重新构建我们的模态框。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>第十三课 组件化开发过程</title>
```



学习前端，最快的进步是持续！

```
<style>
```

```
*{
```

```
margin:0px;
```

```
left:0px;
```

```
}
```

```
.model{
```

```
width:350px;
```

```
height:200px;
```

```
border:1px solid #ccc;
```

```
border-radius:10px;
```

```
margin:100px auto;
```

```
}
```

```
.model .header{
```

```
width:100%;
```

```
height:35px;
```

```
line-height: 35px;
```

```
border-bottom:1px solid #ccc;
```

```
}
```

```
.model .header .title{
```

```
display: inline-block;
```

```
width:90%;
```

```
text-indent: 15px;
```

```
}

.model .header .close{

    display: inline-block;

    width:10%;

    float: right;

    text-align: center;

    cursor: pointer;

}

.model .content{

    padding:10px;

    height:130px;

    overflow-y: auto;

    box-sizing: border-box;

}

.model .footer{

    height:35px;

    border-top:1px solid #ccc;

}

.model .footer span{

    float:right;

    margin:5px 10px 0px 5px;

    box-shadow: none;
```

```
padding:1px 8px;

color: #fff;

font-size: 14px;

cursor: pointer;

}

.model .footer .confirmBtn{

    background: orange;

}

.model .footer .cancelBtn{

    background: green;

}

</style>

</head>

<body>

    <div id="wrap">

        <!-- <div class="model">

            <div class="header">

                <div class="title">星星课堂 web 前端培训</div>

                <div class="close">x</div>

            </div>

            <div class="content">星星课堂真好</div>
```



```
<div class="footer">

    <span class="confirmBtn">确定</span>

    <span class="cancelBtn">取消</span>

</div>

</div> -->

</div>

<button id="openModelBtn">打开模态框</button>

<script>

    var openModelBtn = document.getElementById('openModelBtn');

    openModelBtn.onclick = function(){

        show();

    }

    function create(title,content){

        var wrap = document.getElementById('wrap');

        var model = document.createElement('div');

        model.setAttribute('class','model');

        var str = '';

        str += '<div class="header"> <div class="title">' + title + '</div>'

        str += '<div class="close">x</div> </div>'

        str += '<div class="content">' + content + '</div>'
    }
}
```

```
str += '<div class="footer"><span class="confirmBtn"> 确 定  
  
</span><span class="cancelBtn">取消</span></div>';  
  
model.innerHTML = str;  
  
wrap.appendChild(model);  
  
return model;  
  
}  
  
function show(){  
  
    model.style.display = 'block';  
  
}  
  
var model = create('标题 1','内容 1');  
  
var confirmBtn = document.getElementsByClassName('confirmBtn')[0];  
  
document.getElementsByClassName('confirmBtn')[0];  
  
var cancelBtn = document.getElementsByClassName('cancelBtn')[0];  
  
confirmBtn.onclick = function(){  
  
    console.log('确定');  
  
}  
  
cancelBtn.onclick = function(){  
  
    console.log('取消');  
  
}
```

```
var close = document.getElementsByClassName('close')[0];

close.onclick = function(){

    model.style.display = 'none';

}

</script>

</body>

</html>
```

三．搭建基础结构

在写好面向过程的模态框之后，我们可以进一步来改写成面向对象组件化的模态框，来解决之前面向过程模态框带来的问题。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

    <title>第十三课 组件化开发过程</title>

    <style>
```

```
*{  
  
    margin:0px;  
  
    left:0px;  
  
}  
  
.model{  
  
    width:350px;  
  
    height:200px;  
  
    border:1px solid #ccc;  
  
    border-radius:10px;  
  
    margin:100px auto;  
  
    display: none;  
  
}  
  
.model .header{  
  
    width:100%;  
  
    height:35px;  
  
    line-height: 35px;  
  
    border-bottom:1px solid #ccc;  
  
}  
  
.model .header .title{  
  
    display: inline-block;  
  
    width:90%;  
  
    text-indent: 15px;
```

```
}

.model .header .close{

    display: inline-block;

    width:10%;

    float: right;

    text-align: center;

    cursor: pointer;

}

.model .content{

    padding:10px;

    height:130px;

    overflow-y: auto;

    box-sizing: border-box;

}

.model .footer{

    height:35px;

    border-top:1px solid #ccc;

}

.model .footer span{

    float:right;

    margin:5px 10px 0px 5px;

    box-shadow: none;
```

```
padding:1px 8px;

color: #fff;

font-size: 14px;

cursor: pointer;

}

.model .footer .confirmBtn{

    background: orange;

}

.model .footer .cancelBtn{

    background: green;

}

</style>

</head>

<body>

    <div id="wrap">

        <!-- <div class="model">

            <div class="header">

                <div class="title">星星课堂 web 前端培训</div>

                <div class="close">x</div>

            </div>

            <div class="content">星星课堂真好</div>
```

```
<div class="footer">

    <span class="confirmBtn">确定</span>

    <span class="cancelBtn">取消</span>

</div>

</div> -->

</div>

<button id="openModelBtn1">打开模态框 1</button>

<button id="openModelBtn2">打开模态框 2</button>

<script>

    function Model(){

        this.settingsOption = {

            title:'标题',

            content:'内容'

        };

        this.modelDom = null;

        this.tools = {

            copyObj:function(obj1,obj2){

                for(var i in obj1){

                    obj2[i] = obj1[i];

                }

            }

        }

    }

}
```

```
}  
  
}
```

```
Model.prototype.init = function(option){  
  
    this.tools.copyObj(option,this.settingsOption);  
  
    this.modelDom = this.create();  
  
    this.event();  
  
}
```

```
Model.prototype.create = function(){  
  
    var wrap = document.getElementById('wrap');  
  
    var model = document.createElement('div');  
  
    model.setAttribute('class','model');  
  
    var str = '';  
  
    str += '<div class="header"><div  
class="title">'+this.settingsOption.title+'</div>  
  
    str += '<div class="close">x</div></div>  
  
    str += '<div  
class="content">'+this.settingsOption.content+'</div>  
  
    str += '<div class="footer"><span class="confirmBtn"> 确定  
</span><span class="cancelBtn">取消</span></div>';  
  
    model.innerHTML = str;
```



```
wrap.appendChild(model);

return model;

}
```

```
Model.prototype.event = function(){

    var that = this;

    var confirmBtn = document.getElementById('confirmBtn');

    this.modelDom.getElementsByClassName('confirmBtn')[0];

    var confirmEvent = new CustomEvent('confirmEvent',{

        detail:{

            params : '确定参数'

        }

    })

    confirmBtn.onclick = function(){

        that.modelDom.dispatchEvent(confirmEvent);

    }

    var cancelBtn = document.getElementById('cancelBtn');

    this.modelDom.getElementsByClassName('cancelBtn')[0];

    var cancelEvent = new CustomEvent('cancelEvent',{

        detail:{
```

```
        params : '取消参数'

    }

    })

    cancelBtn.onclick = function(){

        that.modelDom.dispatchEvent(cancelEvent);

    }

    var close = this.modelDom.getElementsByClassName('close')[0];

    close.onclick = function(){

        that.close();

    }

}

Model.prototype.show = function(){

    this.modelDom.style.display = 'block';

}

Model.prototype.close = function(){

    this.modelDom.style.display = 'none';

}
```

```
var option1 = {  
  
    title:'标题 111'  
  
};  
  
var m1 = new Model();  
  
m1.init(option1);  
  
var openModelBtn1 = document.getElementById('openModelBtn1');  
  
openModelBtn1.onclick = function(){  
  
    m1.show();  
  
}  
  
m1.modelDom.addEventListener('confirmEvent',function(e){  
  
    console.log(e + 1);  
  
});  
  
m1.modelDom.addEventListener('cancelEvent',function(e){  
  
    console.log(e + 1);  
  
});  
  
var option2 = {  
  
    title:'标题 666'  
  
};  
  
var m2 = new Model();  
  
m2.init(option2);  
  
var openModelBtn2 = document.getElementById('openModelBtn2');
```



学习前端，最快的进步是持续！

```
openModelBtn2.onclick = function(){  
  
    m2.show();  
  
}  
  
m2.modelDom.addEventListener('confirmEvent',function(e){  
  
    console.log(e + 2);  
  
});  
  
m2.modelDom.addEventListener('cancelEvent',function(e){  
  
    console.log(e + 2);  
  
});  
  
</script>  
  
</body>  
  
</html>
```

谢谢观看！

我是星星课堂老师：周小周