

## 第十五课 分页组件

### 学习目录

- 分页组件基础结构
- 分页组件开发原理
- 分页组件开发过程

### 一 . 分页组件基础结构

```
*{  
  
    margin:0px;  
  
    padding:0px;  
  
}  
  
.wrap{  
  
    width:100%;  
  
    text-align: center;  
  
}  
  
.dataContent{  
  
    position: relative;  
  
    padding:100px 0px;  
  
    width:400px;  
  
    text-align: left;  
  
    display: inline-block;
```



学习前端，最快的进步是持续！

```
}

.pageWrap{

    width:100%;

    height:45px;

    display: inline-block;

}

.pageWrap li{

    height:45px;

    border:1px solid #ccc;

    list-style-type: none;

    display: inline-block;

    text-align: center;

    line-height: 45px;

    cursor: pointer;

    padding:0px 16px;

}

.pageWrap li:hover{

    border:1px solid #06f;

    background: #06f;

    color:#fff;

}

.pageActive{
```

```
border:1px solid #06f;

background: #06f;

color:#fff;

}

<div class="wrap">

  <ul class="pageWrap">

    <li>上一页</li>

    <li class="pageActive">1</li>

    <li>2</li>

    <li>3</li>

    <li>4</li>

    <li>5</li>

    <li>下一页</li>

  </ul>

</div>
```

## 二 . 分页组件开发原理

这个分页组件主要是要完成按页码和每页数据数量来显示数据的效果 ,实际在工作开发的时候是通过 ajax 获取数据来进行分页 , 这里我们用一个数组数据来进行模拟分页效果。

模拟分页的效果以及渲染数据到页面中的方法不算在分页组件中 ,因为这些方法都是根据不同业务逻辑来编写的 ,最好不要和分页组件写在一个构造函数中 ,分页组件关注的是如何给使用者提供数据分页效果。

1. 每次点组件的页码可以更新当前页码对应的数据。
2. 点上一页和下一页也可以获取对应的数据。
3. 点首页和尾页也可以获取的对应的数据。

### 三 . 分页组件开发过程

```
<!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>第十五课 分页组件</title>
```

```
<style>
```

```
  *{
```

```
    margin:0px;
```

```
    padding:0px;
```

```
  }
```

```
  .wrap{
```

```
    width:100%;
```

```
    text-align: center;
```

```
  }
```

```
  .dataContent{
```



学习前端，最快的进步是持续！

```
position: relative;

padding:100px 0px;

width:400px;

text-align: left;

display: inline-block;

}

.pageWrap{

width:100%;

height:45px;

display: inline-block;

}

.pageWrap li{

height:45px;

border:1px solid #ccc;

list-style-type: none;

display: inline-block;

text-align: center;

line-height: 45px;

cursor: pointer;

padding:0px 16px;

}

.pageWrap li:hover{
```



学习前端，最快的进步是持续！

```
        border:1px solid #06f;

        background: #06f;

        color:#fff;

    }

    .pageActive{

        border:1px solid #06f;

        background: #06f;

        color:#fff;

    }

</style>

</head>

<body>

    <div class="wrap">

        <div class="dataContent">数据内容区域</div>

        <ul class="pageWrap">

            <!-- <li>上一页</li>

            <li class="pageActive">1</li>

            <li>2</li>

            <li>3</li>

            <li>4</li>

            <li>5</li>
```

```
<li>下一页</li> -->
```

```
</ul>
```

```
</div>
```

```
<script>
```

```
var data = {
```

```
  total:25,
```

```
  dataArr:[
```

```
    {id:1,title:'标题 1',content:'内容 1'},
```

```
    {id:2,title:'标题 2',content:'内容 2'},
```

```
    {id:3,title:'标题 3',content:'内容 3'},
```

```
    {id:4,title:'标题 4',content:'内容 4'},
```

```
    {id:5,title:'标题 5',content:'内容 5'},
```

```
    {id:6,title:'标题 6',content:'内容 6'},
```

```
    {id:7,title:'标题 7',content:'内容 7'},
```

```
    {id:8,title:'标题 8',content:'内容 8'},
```

```
    {id:9,title:'标题 9',content:'内容 9'},
```

```
    {id:10,title:'标题 10',content:'内容 10'},
```

```
    {id:11,title:'标题 11',content:'内容 11'},
```

```
    {id:12,title:'标题 12',content:'内容 12'},
```

```
    {id:13,title:'标题 13',content:'内容 13'},
```

```
    {id:14,title:'标题 14',content:'内容 14'},
```

```
    {id:15,title:'标题 15',content:'内容 15'},
```

```
{id:16,title:'标题 16',content:'内容 16'},  
  
{id:17,title:'标题 17',content:'内容 17'},  
  
{id:18,title:'标题 18',content:'内容 18'},  
  
{id:19,title:'标题 19',content:'内容 19'},  
  
{id:20,title:'标题 20',content:'内容 20'},  
  
{id:21,title:'标题 21',content:'内容 21'},  
  
{id:22,title:'标题 22',content:'内容 22'},  
  
{id:23,title:'标题 23',content:'内容 23'},  
  
{id:24,title:'标题 24',content:'内容 24'},  
  
{id:25,title:'标题 25',content:'内容 25'}  
  
]  
  
};
```

```
function copyObj(obj1,obj2){  
  
    for(var i in obj1){  
  
        obj2[i] = obj1[i]  
  
    }  
  
}
```

```
function Page(){  
  
    this.settings = {  
  
        id:'pageWrap',
```



```
        data:[],

        total:1,

        pageSize:3,

        pageActive:1,

        prevText:"",

        nextText:"",

        firstText:"",

        lastText:"

    };

    this.pageWrap = null;

    this.activePageNum = 1;

    this.allPageNum = 1;

}

Page.prototype.init = function(option){

    copyObj(option,this.settings);

    this.getWrap();

    this.createPageWrap();

    this.eventCenterFn();

}

Page.prototype.getWrap = function(){
```

```
this.pageWrap = document.getElementsByClassName(this.settings.id)[0];

console.log(this.pageWrap);
}

Page.prototype.createPageWrap = function(){
    var str = "";

    if(this.settings.data && this.settings.data.length != 0){
        var data = this.settings.data;

        this.allPageNum = Math.ceil(this.settings.total /
this.settings.pageSize); //总页数

        for(var i=0; i<this.allPageNum; i++){
            if(this.settings.pageActive == (i + 1)){
                str += '<li class="pageActive pageItem" data-value="'+(i
+ 1)+'">' + (i + 1) + '</li>';

                continue;
            }

            str += '<li class="pageItem" data-value="'+(i + 1)+'">' + (i + 1)
+ '</li>';
        }

        this.pageWrap.innerHTML = '<li data-value="first"> 首页 </li><li
data-value="prev"> 上一页 </li>' + str + '<li data-value="next"> 下一页 </li><li
```

```
data-value="last">尾页</li>;
```

```
}
```

```
}
```

```
Page.prototype.eventCenterFn = function(){
```

```
    var that = this;
```

```
    this.pageWrap.addEventListener('click',function(e){
```

```
        var e = e || window.event;
```

```
        var target = e.target;
```

```
        if(target.tagName == 'LI' && target.className == 'pageItem'){
```

```
            that.selectPageFn(target);
```

```
        }else{
```

```
            that.selectPrevNext(target);
```

```
        }
```

```
    },false);
```

```
}
```

```
Page.prototype.selectPageFn = function(target){
```

```
    this.selectActiveFn(target);
```

```
    this.selectPageData(target);
```

```
}
```

```
Page.prototype.selectActiveFn = function(target){  
  
    var pageItem = document.getElementsByClassName('pageItem');  
  
    for(var i=0;i<pageItem.length;i++){  
  
        pageItem[i].classList.remove('pageActive');  
  
    }  
  
    target.classList.add('pageActive');  
  
}
```

```
Page.prototype.selectPageData = function(target){  
  
    var activeNum = Number(target.getAttribute('data-value'));  
  
    var selectDataEvent = new CustomEvent('selectDataEvent',{  
  
        detail:{  
  
            activeNum:activeNum,  
  
            pageSize:this.settings.pageSize,  
  
        }  
  
    });  
  
    this.pageWrap.dispatchEvent(selectDataEvent);  
  
    this.activePageNum = activeNum;  
  
}
```

```
Page.prototype.selectPrevNext = function(target){  
  
    var activeValue = target.getAttribute('data-value');
```

```
var activeNum = 1;

var pageItem = document.getElementsByClassName('pageItem');

if(activeValue == 'prev' && this.activePageNum > 1){

    activeNum = this.activePageNum - 1;

    this.selectPageFn(pageItem[activeNum - 1]);

}

if(activeValue == 'next' && this.activePageNum < this.allPageNum){

    activeNum = this.activePageNum + 1;

    this.selectPageFn(pageItem[activeNum - 1]);

}

if(activeValue == 'first'){

    this.selectPageFn(pageItem[0]);

}

if(activeValue == 'last'){

    this.selectPageFn(pageItem[this.allPageNum - 1]);

}

}

var p1 = new Page();

var option = {

    id:'pageWrap',

    data:data.dataArr,
```

```
        total:data.total
    }

    p1.init(option);

    p1.pageWrap.addEventListener('selectDataEvent',function(e){

        var num = e.detail.activeNum;

        var pageSize = e.detail.pageSize;

        var pageData = getMockData(num,pageSize);

        createData(pageData);

    });

    var                pageData                =

getMockData(p1.settings.pageActive,p1.settings.pageSize);

    createData(pageData);


function getMockData(pageCount,pageSize,arr){//模拟分页

    var pageCount = pageCount || 1;

    var pageSize = pageSize || 10;

    var arr = data.dataArr || 10;

    var index = (pageCount - 1) * pageSize;

    var newArr = (index + pageSize >= arr.length) ? arr.slice(index,

arr.length) : arr.slice(index, index + pageSize);

    return newArr;

}
```

```
function createData(data){//新建内容区域

    var str = "";

    var dataContent =

document.getElementsByClassName('dataContent')[0];

    for(var i=0;i<data.length;i++){

        str += '<div>' + data[i].id + data[i].title + data[i].content + '</div>';

    }

    dataContent.innerHTML = str;

}

</script>

</body>

</html>
```

**谢谢观看！**

我是星星课堂老师：周小周