

# 面向对象案例一-分页

---

分析：

案例的组成部分：

大盒子、首页、上一页、下一页、尾页 - 使用div

页码是一个div包含了5个p标签

使用者要传参数：

数据总条数、每页显示多少条

上一页、下一页使用符号使用英文字母来代替文字

这些参数都应该有默认值，调用者不传的时候，使用默认值

页码有规律：

1. 当前页处在所有页面的最中间：规律是 循环5次，初始值是当前页-2，结束页是当前页+2
2. 如果当前页小于等于3，展示1~5
3. 当前页大于等于总页数-2的时候，展示最后5个页面，初始值是总页数-4，结束页是总页数

代码：

1. 先写一个空的构造函数
2. 实例化这个构造函数得到对象
3. 分析参数
4. 在构造函数中
  - 获取元素 - 大盒子
  - 给大盒子添加了样式 - 定义了一个添加样式的方法
5. 创建小盒子 - 首页、上一页、下一页、尾页
  - 没有根据 - 找根据 - 传进来的参数 language就是根据，在另外一个方法中创建小盒子，获取不到参数，在构造函数中将参数绑定到了大对象上作为属性
  - 问题：如果调用者没有传这个参数，没有根据了，所以在构造函数中定义了一个默认值
  - 问题：如果调用者传入这个参数了，和默认值有冲突 - 将传入的参数替换掉默认值 - setDefault
  - 可以创建小盒子，不能创建放页码的那个div，在默认值中添加了一个list专门用来创建放页码的div
6. 给页码的div中创建页码
  - 根据规律创建页码
  - 发现需要判断总页数和当前页
  - 在构造函数中处理默认值结束后，添加属性总页数和当前页，总页数=向上取整(内容总条数/每页显示的条数)，当前页默认为1
  - 创建页码的过程中需要给页码的p标签设置样式以及设置当前页的样式
  - 将创建好的页码标签放到list的div标签中，发现此时没办法获取到list的div，在构造函数中添加属性list值为null，在创建小盒子的时候判断如果是list的话，给大对象的list属性赋值为创建好的div
7. 给所有小盒子以及页码的p标签添加点击事件
  - 通过事件委托来设置点击事件

- 给大盒子添加点击事件，判断事件源是哪个标签，小盒子通过类名来判断，页码标签p通过标签名来判断
  - 点击事件中主要改变当前页的值，并重新创建小盒子，重新创建页码
8. 设置禁用项 - 当当前页为1的时候，首页和上一页不能点击，当当前页为最后一页的时候，下一页和末页不能点击
- 判断当前页是否为1，是就给大盒子的第一个子元素和第二个子元素添加属性name，值为disabled
  - 给大盒子的第一个子元素和第二个子元素设置背景颜色为灰色
  - 在首页和上一页的点击事件中，多加一个判断条件，获取到事件源的name属性不等于disabled的时候才可以点击
  - 每个点击事件结束后除了要创建小盒子和页码外，还需要添加设置禁用项
  - 同理处理当前页为末页
9. 添加文本框和跳转按钮
- 创建文本框和button按钮分别追加到大盒子中
  - 在点击事件中添加button按钮的点击事件，并处理获取到的值变为数字，以及输入的数字不合理的处理
  - 在每个点击事件后还需要添加创建文本框和跳转按钮的方法
10. 设置启动器
- 将所有需要调用的方法封装到一个方法中，通过调用启动器方法简化多次调用
  - 发现点击事件放在单独的事件中没办法处理，所以将点击事件移动到构造函数中
  - 点击事件的方法中所有的\_this就可以使用this了，启动器方法就可以调用使用了
11. 使用分页的时候，发现没办法在对象中处理数据的切换，所以在构造函数中添加要执行的函数为传进来的函数，并在启动器方法的最后添加调用这个方法，需要传入当前页为参数

代码：

```
// 1. 写一个空的构造函数
function Page(classname,obj={}){
    // 第一步获取元素
    this.box = document.querySelector("."+classname);
    // 2. 给大盒子设置样式
    // 专门定义一个方法来设置样式
    this.setStyle(this.box,{
        width:"800px",
        height:'40px',
        border:"1px solid #000",
        display:"flex",
        "justify-content":"center",
        alignItems:"center"
    });
    // 将参数绑定到对象上成为属性
    this.obj = obj; // 如果调用者不传参数，这个属性就是空值，没有默认值
    // 自定义默认值 - 默认值和参数结构一致
    this.default = {
        language:{
            first:"首页",
            prev:"上一页",
            list:"list",
            next:"下一页",
            last:"尾页"
        },
```

```

        pageData:{
            total:1000,
            pageSize:12
        }
    };
    // 使用传进来的参数替换掉默认值
    this.setDefault();
    // 设置默认执行的函数
    this.show = this.obj.show || function(){};
    // 计算总页数
    this.totalPage = Math.ceil(this.default.pageData.total/this.default.pageData.pageSize);
    // 设置一个属性，用来存放页码的div
    this.list = null;
    // 设置一个当前页的属性
    this.currentPage = 1;
    var _this = this;
    this.box.onclick=function(){
        _this.qiehuanPage();
    }
    // 3.创建小盒子
    // this.createSmallBox();
    // 4.创建页码
    // this.createPage();
    // // 6.限制首页和上一页在第一页的是后不能点击
    // this.setDisabled();
    // // 5.点击标签让页码切换

    // this.qiehuanPage();
    // // 添加文本框和按钮
    // this.jump();
    this.init();
}
// 设置启动器
Page.prototype.init=function(){
    this.createSmallBox();
    this.createPage();
    this.setDisabled();
    this.jump();
    this.show(this.currentPage);
}
// 添加文本框和按钮
Page.prototype.jump=function(){
    var input = document.createElement("input");
    input.type = 'number';
    this.setStyle(input,{
        width:"50px",
        height:"19px",
        border:"none",
        outline:"none",
        border:"1px solid #000",
        marginRight:"5px",
        paddingLeft:"2px",
    });
}

```

```

    this.box.appendChild(input);
    var button = document.createElement("button");
    button.innerText = 'GO';
    button.style.height='23px';
    this.box.appendChild(button);
}
// 设置禁用项
Page.prototype.setDisabled=function(){
    if(this.currentPage == 1){
        this.box.children[0].setAttribute("name","disabled");
        this.box.children[1].setAttribute("name","disabled");
        this.box.children[0].style.background = '#ccc';
        this.box.children[1].style.background = '#ccc';
    }
    if(this.currentPage==this.totalPage){
        this.box.children[3].setAttribute("name","disabled");
        this.box.children[4].setAttribute("name","disabled");
        this.box.children[3].style.background = '#ccc';
        this.box.children[4].style.background = '#ccc';
    }
}
// 点击标签让页码切换
Page.prototype.qiehuanPage=function(e){
    // 使用事件委托, 给大盒子绑定事件, 判断是点击的哪一个事件源
    var e = e || window.event;
    var target = e.target || e.srcElement;
    if(target.className == 'first' && target.getAttribute("name")!="disabled"){
        // console.log(target.getAttribute("name"));
        this.currentPage = 1;
        // this.createSmallBox();
        // this.createPage();
        // this.setDisabled();
        // this.jump();
        this.init();
    }else if(target.className == 'prev' && target.getAttribute("name")!="disabled"){
        this.currentPage--;
        this.init();
    }else if(target.className == 'next' && target.getAttribute("name")!="disabled"){
        this.currentPage++;
        this.init();
    }else if(target.className == 'last' && target.getAttribute("name")!="disabled"){
        this.currentPage = this.totalPage;
        this.init();
    }else if(target.nodeName == "P" && target.innerText-0!=this.currentPage){
        this.currentPage = target.innerText-0;
        this.init();
    }else if(target.nodeName == "BUTTON"){
        if(target.previousElementSibling.value-0<=1){
            this.currentPage = 1;
        }
        if(target.previousElementSibling.value-0>=this.totalPage){
            this.currentPage = this.totalPage;
        }
    }
}

```

```

        if(target.previousElementSibling.value-0==this.currentPage ||
target.previousElementSibling.value==''){
            return;
        }
        this.currentPage = target.previousElementSibling.value-0;
        this.init();
    }
}
// 创建页码
Page.prototype.createPage=function(){
    // 如果总页数小于5 - 获取总页数
    if(this.totalPage<=5){
        for(var i=1;i<=this.totalPage;i++){
            var p = document.createElement("p");
            p.innerText = i;
            this.setStyle(p,{
                border:"1px solid #000",
                margin:"0 5px",
                padding:"0 5px",
            });
            // 要给当前页设置特殊的样式 - 找当前页 - 去构造函数中定义一个当前页的属性
            if(i == this.currentPage){
                this.setStyle(p,{
                    background:"red",
                    color:"#fff"
                });
            }
            // 创建好的页码p标签放到list的div中 - 问题：找不到list的div
            this.list.appendChild(p);
        }
    }else{ // 总页数大于5
        // 当前页小于等于3 - 1~5页
        if(this.currentPage<=3){
            for(var i=1;i<=5;i++){
                var p = document.createElement("p");
                p.innerText = i;
                this.setStyle(p,{
                    border:"1px solid #000",
                    margin:"0 5px",
                    padding:"0 5px",
                });
                if(i == this.currentPage){
                    this.setStyle(p,{
                        background:"red",
                        color:"#fff"
                    });
                }
                this.list.appendChild(p);
            }
        }else if(this.currentPage>=this.totalPage-2){ // 当前页大于等于总页数-2
            for(var i=this.totalPage-4;i<=this.totalPage;i++){
                var p = document.createElement("p");
                p.innerText = i;
            }
        }
    }
}

```

```

        this.setStyle(p,{
            border:"1px solid #000",
            margin:"0 5px",
            padding:"0 5px",
        });
        if(i == this.currentPage){
            this.setStyle(p,{
                background:"red",
                color:"#fff"
            });
        }
        this.list.appendChild(p);
    }
}
}else{ // 当前页在页码的最中间
    for(var i=this.currentPage-2;i<=this.currentPage+2;i++){
        var p = document.createElement("p");
        p.innerText = i;
        this.setStyle(p,{
            border:"1px solid #000",
            margin:"0 5px",
            padding:"0 5px",
        });
        if(i == this.currentPage){
            this.setStyle(p,{
                background:"red",
                color:"#fff"
            });
        }
        this.list.appendChild(p);
    }
}
}
}

// 设置默认值的方法
Page.prototype.setDefault=function(){
    for(var attr in this.obj.language){
        this.default.language[attr] = this.obj.language[attr];
    }
    for(var attr in this.obj.pageData){
        this.default.pageData[attr] = this.obj.pageData[attr];
    }
}

// 创建小盒子方法
Page.prototype.createSmallBox=function(){
    this.box.innerHTML = '';
    // 中间的页码盒子和上一页首页等盒子长的不一样，修饰的时候也要区别对待，不好在这里做了
    // 参数中有language，正好对应其他4个盒子，想通过遍历这个参数，来创建小盒子
    for(var attr in this.default.language){ // 获取不到传进来的参数，将参数绑定到对象的属性上，在这里就能使用this来使用了
        // 可以创建小盒子了
        var div = document.createElement("div");
        div.className = attr;
        if(attr != 'list'){

```

```
div.innerText = this.default.language[attr];
this.setStyle(div,{
    border:"1px solid #000",
    margin:"0 5px",
    padding:"0 5px",
});
}else{
    this.list = div;
    this.setStyle(this.list,{
        display:"flex",
        "justify-content":"center",
        alignItems:"center"
    });
}
this.box.appendChild(div);
}
}
// 设置样式的方法
Page.prototype.setStyle=function(ele,styleObj){
    for(var attr in styleObj){
        ele.style[attr] = styleObj[attr];
    }
}
```