task24(作业)

问答

1.ajax 是什么？有什么作用？

Ajax是Asynchronous JavaScript and XML的缩写，这一技术能够向服务器请求额外的数据而无需卸载整个页面，会带来良好的用户体验。传统的HTTP请求流程大概是这样的，

- 浏览器向服务器发送请求
- 服务器根据浏览器传来数据生成response
- 服务器把response返回给浏览器
- 浏览器刷新整个页面显示最新数据
- 这个过程是同步的，顺序执行

AJAX 在浏览器与 Web 服务器之间使用异步数据传输（HTTP 请求）从服务器获取数据

这里的异步是指脱离当前浏览器页面的请求、加载等单独执行，这意味着可以在不重新加载整个网页的情况下，通过JavaScript发送请求、接受服务器传来的数据，然后操作DOM将新数据对网页的某部分进行更新，使用Ajax最直观的感受是向服务器获取新数据不需要刷新页面等待了。

https://zhuanlan.zhihu.com/p/22564745?refer=study-fe

2.前后端开发联调需要注意哪些事情？后端接口完成前如何 mock 数据？(npm install -g server-mock)

前后端联调是一种**真实业务数据**和 **本地mock数据**之间来回切换以达到前后端分离架构下的不同开发速度时数据交换的一种方式方法
需要注意的事情：

- 约定前后端联调的时间。
- 约定双方需要传输的数据和接口，在接口文档中确定好参数- 的名称、格式等。
  约定请求和响应的格式和内容。

mock数据就是html发送一个ajax的请求。这个请求到哪里去，然后后端如何去响应这个请求。后端去获取数据，并且定义接口；前端编写页面，并且和后端进行交互。

mock数据的方法有：

一.使用XAMPP等工具，编写PHP文件来进行测试。

二.使用server-mock或mock.js（http://mockjs.com/ ）搭建模拟服务器，进行模拟测试（优点是不需要熟练掌握后台PHP语言，采用熟悉的js语法）；

步骤：

1.安装node.js，呼出cmd命令
2.选取一个文件夹，使用npm install -g server -mock进行全局安装
3.输入mock start可以启动一个web 服务器，他的根目录就是你选取的文件夹，启动完成之后，web服务器就可以展示了
4.浏览器输入localhost:8080就是你选取的文件夹
5.使用mock init会自动的在文件夹下生成3个文件
6.当html使用url对接口进行请求，会被router.js里相同的接口接受比如:

```
app.get("/loadMore",function(req,res){
 //接受名为loadMore的php的参数
 res.send({status:0,//向html发出正确的回参
           msg:"hello 饥人谷"//回参中的值
          })
 })
```

3.点击按钮，使用ajax获取数据，如何在数据到来之前防止重复点击?

方法一：

```
使用setTimeout，使在数据到来之前按钮不被触发

var clicktag = 0;
            $('.btn').click(function () {
                if (clicktag === 0) {
                    clicktag = 1;
                    $(this).addClass("otherThings");
                    console.log('3秒才能触发一次');
                    setTimeout(function () { clicktag = 0; }, 3000);
                }
            });
```

http://js.jirengu.com/levo/1/edit?js,console,output

方法二：
可设置标记变量flag，初始时设置flag为true.在用户点击提交按钮后，判断flag是否为true，如果是则发送ajax请求，并把flag设置为false。 等服务端给出响应后再把flag设置为true;

```
 var ready = true;
    $('.add-more').on('click', function(){
        ...
        if(!ready){
            return;
        }
        ready = false;
```

```
        $.ajax({
            ...
            complete: function(){
                ready = true;
            }
        });
    });
```

代码

1.封装一个 ajax 函数，能通过如下方式调用

```
function ajax(opts){
    // todo ...
}
document.querySelector('#btn').addEventListener('click', function(){
    ajax({
        url: 'getData.php',   //接口地址
        type: 'get',               // 类型， post 或者 get,
        data: {
            username: 'xiaoming',
            password: 'abcd1234'
        },
        success: function(ret){
            console.log(ret);       // {status: 0}
        },
        error: function(){
           console.log('出错了')
        }
    })
});
```

HTML代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>mock</title>
</head>
<body>
<input type="text" name="username" id="username" placeholder="请输入用户名">
<button id="btn">获取信息</button>
<dl id="ct">

</dl>
```

```
<script type="text/javascript">

    function dealWith(userInfo){ //输出HTML的函数
        var str = '<dt>性别：</dt>';
        str += '<dd>'+userInfo.sex+'</dd>';
        str += '<dt>年龄：</dt>';
        str += '<dd>'+userInfo.age+'</dd>';
        document.querySelector('#ct').innerHTML = str;  //输出HTML
    }
    function ajax(opts){  //封装的ajax函数
        var xmlhttp = new XMLHttpRequest();
        var dataStr = '';
        for( var key in opts.data){
            dataStr += key + '=' + opts.data[key]+'&'
        }
        dataStr = dataStr.substr(0,dataStr.length-1); //给后台的内容

        if(opts.type.toLowerCase()==='post'){ //post格式
            xmlhttp.open(opts.type,opts.url,true);
            xmlhttp.setRequestHeader("Content-type", "application/x-www-form-ur
lencoded");
            xmlhttp.send(dataStr);
        }

        if(opts.type.toLowerCase() === 'get'){ //get格式
            xmlhttp.open(opts.type,opts.url+'?'+dataStr,true);
            xmlhttp.send();
        }

        xmlhttp.onreadystatechange = function(){  //反馈
            if(xmlhttp.readyState ==4 && xmlhttp.status == 200){
                var json = JSON.parse(xmlhttp.responseText);
                opts.success(json);
            }
            if(xmlhttp.readyState == 4 && xmlhttp.status == 404){
                opts.error();
            }
        };
    }

    document.querySelector('#btn').addEventListener('click',function(){
        ajax({
            url:'ajax_simple.php',
            type:'post',
            data:{
                username:document.querySelector('#username').value,
                password:document.querySelector('#password')
            },
```

```
            success:function(jsonData){
                dealWith(jsonData);
            },
      error:function(){
        console.log('error')
      }
        });
    });
</script>
</body>
</html>
```

PHP文件代码：

```
    <?php
    $username = $_POST['username'];  //创建索引数组
    //$username = $_GET['username'];
    if($username === 'kevin'){
        $ret = array('sex'=>'男','age'=>23);
    }elseif ($username === 'david') {
        $ret = array('sex'=>'男', 'age'=>30);
    }else{
        $ret = array('sex'=>'女','age'=>18);
    }
    echo json_encode($ret);
    ?>
```

2.1实现如下加载更多的功能(代码)

第1行

第2行

第3行

第4行

第5行

第6行

第7行

第8行

第9行

第10行

第11行

第12行

再加载5个

2.2不用PHP而用JS文件实现

3.实现注册表单验证功能

要点是首先要封装几个用的到的function：

```
function hasClass(el,cls){
    var reg = new RegExp( '\\b'+cls+'\\b' );
    return reg.test(el.className);
}

function singleAddClass(el,cls){
    if( !hasClass(el,cls) ){
        el.className += ' '+cls;
    }
}

function addClass(el,cls){
    if( el.length && el.length>0){
        for(var i=0;i<el.length;i++){
            singleAddClass(el[i],cls);
        }
    }else{
            singleAddClass(el,cls);
        }
}

function singleRemoveClass(el,cls){
    if( hasClass( el,cls ) ){
        el.className = el.className.replace(cls,'').replace(/\s{2,}/g,' ');
    }
}

function removeClass(el,cls){
    if(el.length && el.length>0){
        for(var i=0;i<el.length;i++){
            singleRemoveClass(el[i],cls);
        }
    }else{
            singleRemoveClass(el,cls);
        }
}
```

针对用户名和密码设置相应的正则式去验证：

```
function isValidUsernm(str){
    return /^\w{3,10}$/.test(str);
}

function isValidPsw(str){
    if(str.length>15||str.length<6){
        return false;
```

```
    }
    if(/\W/.test(str)){
        return false;
    }
    if(/(^[A-Z]+$)|(^[a-z]+$)|(^[0-9]+$)|(^_+$)/g.test(str)){
        return false;
    }
    return true;
}
```

验证的思路：

1、用户名要合法（用正则式筛选）
2、用户名未被占用（用一个PHP文件验证）
3、第一次密码要合法（用正则式筛选）
4、第二次密码要合法（用正则式筛选）
5、第二次密码要与第一次一致

代码