第一个实验

实验要求

1. 构造一个JSON对象，属性 学生姓名，学生年龄，学生所在学院；

2.学生所在学院 学院名称，学院坐落地址

3. 页面加载以后，创建两个JSON 学生对象，并且把第一个学生的姓名赋值给页面上一个id=“s1”的 input 元素。

实验说明

这次实验主要是学会怎么解析json数据 ，从后天传入的一定是字符串，所以需要我们进行转换，然后进行一个个数据的读取，通过getElementById()函数获取到dom对象，然后等待用户点击按钮，最终将解析好的数据填入就行。

以下是html代码

<form id="formCss" action="javascript:" >

    <div>

        <h3>学生姓名：</h3> <input id="s1" type="text" > <br>

        <h3>学生年龄：</h3> <input id="age" type="text"> <br>

        <h3>学生所在学院：</h3> <input id="collegeName" type="text"> <br>

        <h3>学院地址：</h3> <input id="collegeAddress" type="text"> <br>

    </div>

    <input type="button" id="valueButton" value="文本框赋值" >

</form>

以下是javascript代码

<script type="text/javascript">

    window.onload = function (){

        let rawData = '{"name":"张三", "age" : 18, "college": {"collegeName": "计算机科学与技术学院","collegeAddress":"j13教学楼" }}';

        let oinfomation = document.getElementById("infomation");

        oinfomation.innerHTML = "后端发送过来的数据" + rawData;

        let  student1=JSON.parse(rawData);  //处理字符数据 使用两个对象来存储

        let student2 = JSON.parse(rawData);

        let name = student1.name;   //将名字 年龄 学院名称和地址用变量保存起来

        let age = student1.age;

        let collegeName = student1.college.collegeName;

        let collegeAddress = student1.college.collegeAddress;

        let oButton = document.getElementById("valueButton");  //获取dom对象 以便接下来的操作

        let nameText = document.getElementById('s1');   //这里将会填写学生的姓名一栏

        let ageText = document.getElementById("age");

        let collegeNameText = document.getElementById("collegeName");

        let collegeAddressText = document.getElementById("collegeAddress");

        oButton.onclick  = function () { //点击按钮的时候触发此函数 将后台给的json数据解析后放入到文本框之中

            nameText.value = name;    //将

            ageText.value = age;

            collegeNameText.value = collegeName;

            collegeAddressText.value = collegeAddress;

        };

    }

</script>

第二个实验

实验要求

1. 构造一个JSON对象，属性 学生姓名，学生年龄，学生所在学院；

2.学生所在学院 学院名称，学院坐落地址

3. 页面加载以后，创建两个JSON 学生对象，并且把第一个学生的姓名赋值给页面上一个id=“s1”的 input 元素。

实验说明

这次实验主要是主要是学习正则表达式的使用，因为要验证不少东西，需要写的代码也比较多，而且验证结束后需要想办法告知用户那些东西通过了验证，那些东西未通过，好让用户按照要求进行修改

以下是html代码

<body>

    <div id="infomation">

    </div>

    <form id="formCss" action="javascript:" >

        <div>

            <h3>email：</h3> <input id="email" type="email" > <br>

            <h3>域名：</h3> <input id="domain" type="text"> <br>

            <h3>手机号码：</h3> <input id="telNum" type="tel"> <br>

            <h3>身份证号：</h3> <input id="id" type="text"> <br>

            <h3>密码：</h3> <input id="password" type="password"> <br>

            <h3>地址：</h3> <input id="ipAddress" type="text"> <br>

        </div>

        <input type="button" id="valueButton" value="验证信息" onclick="">

    </form>

    <div id="inputTextArea">

        <textarea id="textarea1" cols="60" rows="10"></textarea>

    </div>

</body>

    <div id="example">

    <h3>测试样例</h3> <br>

    123@qq.com   验证成功<br>

    http://baidu.com   验证成功<br>

    13154769321  验证成功<br>

    370873200503201458  验证成功<br>

    qwe123  验证失败 密码至少是8位<br>

    115.145.21.1  验证成功<br>

    </div>

以下是javascript代码，

<script type="text/javascript">

    window.onload = function (){

        let oButton = document.getElementById("valueButton"); //以下代码是用来获取dom对象 以便后期操作

        let oEmail = document.getElementById("email");

        let oDomain = document.getElementById("domain");

        let oTelNum = document.getElementById("telNum");

        let oId = document.getElementById("id");

        let oPassword = document.getElementById("password");

        let oIpAddress = document.getElementById("ipAddress");

        let oInfomation = document.getElementById("infomation");

        let oArray = new Array(oEmail,oDomain,oTelNum,oId,oPassword,oIpAddress);  //将这些dom对象用数组进行存储

        let information = "邮箱：     可以以若干字母和数字开头 ," +

            "但是必须中间包含 `@` 和 `.` 符号, \n" +

            "最后的一级域名长度在2-4位之间 \n";

        information += "域名：   需要以http://或者https://开头，支持域名中有中划线的情况 \n";

        information += "电话号码 ：   需要为11位且必须以1开头,第二个数字不能是2,其他数字随意 \n";

        information += "身份证号：  需要是15位或者是18位 且符合一定的格式 考虑了以x为结尾的情况 \n";

        information += "密码：  需要至少8位，可以是数字大小写字母 \n";

        information += "ip地址：    可以是ipv4地址 也可以是ipv6地址 \n";

        oInfomation.innerText = information;

        let aa= "";

        let boolArray = new Array(false,false,false,false,false,false);  //该数组用于判断某一项测试是否通过了测试

        let nameArray = new Array("邮箱" , "域名" ,"电话号码" ,"身份证号" ,"密码" ,"IP地址");

        let emailPatten =  /^[A-Za-z\d]+([-\_.][A-Za-z\d]+)\*@([A-Za-z\d]+[-.])+[A-Za-z\d]{2,4}$/;  //以下都是正则表达式

        let domainPatten = /^((http:\/\/)|(https:\/\/))?([a-zA-Z0-9]([a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])?\.)+[a-zA-Z]{2,6}/;

        let telNumPatten = /^1[3456789]\d{9}$/;

        let id18Patten = /^([1-6][1-9]|50)\d{4}(18|19|20)\d{2}((0[1-9])|10|11|12)(([0-2][1-9])|10|20|30|31)\d{3}[0-9Xx]$/;

        let id15Patten =  /^([1-6][1-9]|50)\d{4}\d{2}((0[1-9])|10|11|12)(([0-2][1-9])|10|20|30|31)\d{3}$/;

        let passwordPatten = /^[a-zA-Z\d\_]{8,}$/;

        let ipAddressPatten = /^((2[0-4]\d|25[0-5]|[01]?\d\d?)\.){3}(2[0-4]\d|25[0-5]|[01]?\d\d?)$/;

        let onsubmit =function (){   //当点击测试按钮的时候进行测试

            //测试6个文本框输入内容是否符合要求 符合要求的话 boolArray数组的对应位为true

            if(emailPatten.test(oEmail.value))

                boolArray[0] = true;

            if(domainPatten.test(oDomain.value))

                boolArray[1] = true;

            if(telNumPatten.test(oTelNum.value))

                boolArray[2] = true;

            if(id15Patten.test(oId.value) || id18Patten.test(oId.value))

                boolArray[3] = true;

            if(passwordPatten.test(oPassword.value))

                boolArray[4] = true;

            if(ipAddressPatten.test(oIpAddress.value))

                boolArray[5] = true;

            for(let i =0; i<boolArray.length ;i++){

                if(boolArray[i]==true)

                  aa +=  oArray[i].value +"   " + nameArray[i] + "测验通过 \n";

                else

                    aa += oArray[i].value + "   " + nameArray[i] + "测验失败\n";

            }

            textarea1.value = aa;    //更新文本框中的内容 显示测试结果给用户

            aa = "";

            for(let j = 0;j<boolArray.length;j++){

                boolArray[j] = false;

            }

        };

        oButton.onclick = onsubmit;

    }

</script>

第三个实验

第三个实验主要是来预习servlet,需要手动配置tomcat和web.xml 等信息,其实并不复杂 ,但是有很多小问题需要解决,比如tomcat的字符集是utf-8 ,而windows的字符集是GBK,这样tomcat启动的时候就会出现乱码,需要我们修改tomcat的配置文件.

因为我是用maven创建的

以下是文件的目录结构

│ pom.xml

├─src

│ ├─main

│ │ ├─java

│ │ │ └─tech

│ │ │ └─haonan

│ │ │ FirstServlet.java //路径为 localhost:8080/hello

│ │ │ JspServlet.java //路径为 localhost:8080/index

│ │ │

│ │ └─resources

│ └─webapp

│ │ index.jsp // JspServlet重定向到这个文件

│ └─WEB-INF

│ web.xml // 用来配置 拦截网络访问

以下是pom.xml文件中引入内容

   <dependency>  <!--引入servlet的包 以便使用里边的类-->

      <groupId>javax.servlet</groupId>

      <artifactId>javax.servlet-api</artifactId>

      <version>4.0.1</version>

      <scope>provided</scope>

    </dependency>

FirstServlet.java 类中的内容

public class FirstServlet extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        //下面三行代码是设置传输的格式 否则中文会出现乱码

        resp.setCharacterEncoding("UTF-8");

        req.setCharacterEncoding("UTF-8");

        resp.setContentType("text/html;charset=UTF-8");

        resp.getWriter().println("<h1>hello world</h1>");

        resp.getWriter().println("servlet 演示");

        resp.getWriter().flush();

        resp.getWriter().close();

    }

}

JspServlet.java 类中的内容

public class JspServlet extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        req.getRequestDispatcher("/index.jsp").forward(req, resp);

    }

}

Web.xml文件中的内容

<!DOCTYPE web-app PUBLIC

 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

 "http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

  <display-name>hellodemo</display-name>

  <servlet>

    <servlet-name>abc</servlet-name>

    <servlet-class>tech.haonan.FirstServlet</servlet-class>

  </servlet>

  <servlet-mapping>

    <servlet-name>abc</servlet-name> <!--拦截 localhost:8080/hello 访问去 FirstServlet 这个类-->

    <url-pattern>/hello</url-pattern>

  </servlet-mapping>

  <servlet>

    <servlet-name>index</servlet-name>

    <servlet-class>tech.haonan.JspServlet</servlet-class>

  </servlet>

  <servlet-mapping>

    <servlet-name>index</servlet-name><!--拦截 localhost:8080/index 访问去 JspServlet 这个类-->

    <url-pattern>/index</url-pattern>

  </servlet-mapping>

</web-app>