

一、什么是JavaScript

- 1、HTML是网页的结构，CSS是网页的外观，而JavaScript是页面的行为。
- 2、HTML页面是静态的（只供浏览），平常我们所见到的各种网页特效就是使用JavaScript实现的。

二、JavaScript的特点

1. 动态改变页面内容

HTML页面是静态的，一旦编写，内容是无法改变的。JavaScript可以弥补这个不足，可以将内容动态地显示在网页中。

2. 动态改变网页的外观

JavaScript通过修改网页元素的CSS样式，达到动态地改变网页的外观。

3. 验证表单数据

我们常见的在各大网站中的注册中的验证功能，就是JavaScript实现的。

4. 响应事件

JavaScript是基于事件的语言。

三、JavaScript编辑工具

常用的JavaScript编辑工具有：

- (1) 记事本；
- (2) WebStorm；√
- (3) Visual Studio；
- (4) sublime Text

四、html文档中引用JavaScript脚本的三种方式

1. 行内使用
2. 页面内嵌
3. 引用外部js文件

五、了解两个对象

1. 窗口对象window

一个浏览器窗口就是一个window对象，window对象主要用来控制打开窗口或关闭窗口、控制窗口的大小和位置、窗口历史等等。一句话，window对象就是用来操作“浏览器窗口”的一个对象

2. 文档对象document

document对象是window对象中的子对象，操作的都是HTML文档

四、内容

1.DOM操作

1.1 什么是DOM

- DOM, 全称“Document Object Model (文档对象模型)”
- 在前端开发时, 我们往往需要在页面某个地方添加一个元素或者删除元素, 这种添加元素、删除元素的操作就是通过操作DOM来实现的。
- 可以把整个html文档理解为dom树, 每一个元素节点都看成一个“dom对象”。由于元素节点也是一个对象, 因此他们有自身的属性和方法

1.2 获取元素

- document.getElementById(): 根据id获取元素

```
<p id="pid1">这是第一段</p>
<p id="pid2">这是第二段</p>

<script>
    window.onload = function () {
        //根据id属性获取元素
        var p1 = document.getElementById("pid1");
        console.log(p1);
        //根据id属性获取元素中的文本内容
        var p2 = document.getElementById("pid2").innerText;
        console.log(p2);
    }
</script>
```

- document.getElementsByName(): 根据name属性获取元素

```
姓名: <input type="text" name="uname"> <br>
密码: <input type="password" name="pwd"> <br>
<input type="button" value="登录" onclick="login()">
<script>
    function login() {
        //根据name属性值获取元素, 获取到的是元素数组
        var name = document.getElementsByName("uname")[0].value;
        var pwd = document.getElementsByName("pwd")[0].value;
        alert("姓名是: " + name + ",密码是: " + pwd)
    }
</script>
```

- document.getElementsByTagName(): 根据标签名称获取元素

```
<button onclick="getInput()">获取所有的input</button>
<script>
    function getInput() {
        //根据标签名称获取元素, 获取到的是NodeList对象
        var inputs = document.getElementsByTagName("input");
        for(var i = 0; i < inputs.length; i++) {
            alert(inputs[i].value);
        }
    }
</script>
```

- document.getElementsByClassName(): 根据类名获取元素

```
<p id="pid1" class="pclass"><span>这是第一段</span></p>
<p id="pid2" class="pclass">这是第二段</p>
<script>
    function getPClass() {
        //根据类名获取元素
        var pclasses = document.getElementsByClassName("pclass");
        for(var i = 0; i < pclasses.length; i++) {
            alert(pclasses[i].innerHTML)
        }
    }
</script>
```

- document.querySelector(): 根据CSS选择器的规则, 返回第一个匹配到的元素
- document.querySelectorAll(): //根据CSS选择器的规则, 返回所有匹配到的元素

```
<div id="div1">
    <p id="p1" class="class1">
        我是第一个P</p>
    <p id="p2" class="class2">
        我是第二个P</p>
</div>
<script>
    window.onload = function () {
        var node = document.querySelector("#div1 > p");
        alert(node.innerHTML);

        var node1 = document.querySelector(".class2");
        alert(node1.innerHTML);

        var nodelist = document.querySelectorAll("p");
        alert(nodelist[0].innerHTML + " - " + nodelist[1].innerHTML);
    }
</script>
```

1.3 创建和增加节点

- createElement () : 创建元素节点
- createTextNode(文本内容) : 创建文本节点
- appendChild () : 末尾追加方式插入节点

```

<button onclick="addNode()">在末尾增加节点</button>
<script>
    function addNode() {
        var h2 = document.createElement("h2");
        h2.innerHTML = "二级标题";
        document.getElementById("div1").appendChild(h2);
        var img = document.createElement("img");
        img.setAttribute("src", "animal.png");
        document.getElementById("div1").appendChild(img)
    }
</script>

```

- insertBefore () : 在指定节点前插入新节点

```

<button onclick="insertNode()">在指定位置增加节点</button>
<script>
    function insertNode() {
        var img = document.createElement("img");
        img.setAttribute("src", "animal.png");
        img.setAttribute("title", "羊驼");
        var div1 = document.getElementById("div1");
        document.body.insertBefore(img, div1);
    }
</script>

```

- cloneNode () : 克隆节点，参数默认为false，只复制当前节点，若参数为 true，将递归复制当前节点的所有子孙节点

```

<button onclick="copyNode()">复制节点</button>
<ul id="myList1"><li>Coffee</li><li>Tea</li></ul>
<ul id="myList2"><li>Water</li><li>Milk</li></ul>
<script>
    function copyNode(){
        var itm = document.getElementById("myList2").lastChild;
        var cln = itm.cloneNode(true);
        document.getElementById("myList1").appendChild(cln);
    }
</script>

```

1.4 删除和替换节点

- removeChild () : 删除节点
- replaceChild() : 替换节点

```

<button onclick="delNode()">删除节点</button>
<button onclick="repNode()">替换节点</button>

<script>
    function delNode() {
        var dNode = document.getElementById("animal");
        document.body.removeChild(dNode);
    }

```

```

    }
    function repNode(){
        var oldImg = document.getElementById("animal");
        var newImg = document.createElement("img");//创建节点
        newImg.setAttribute("src", "a.png");
        document.body.replaceChild(newImg,oldImg);
    }
</script>

```

2.属性操作

1.1 获取属性: getAttribute()

1.2 设置属性: setAttribute()

1.3 移除属性: removeAttribute()

```

<br>
<br>
<input type="button" value="获取src属性" onclick="getSrc()">
<input type="button" value="设置title属性" onclick="setTitle()">
<input type="button" value="删除title属性" onclick="delTitle()">
<script>
    function getSrc(){
        //获取属性
        var imgSrc = document.getElementById("img1").getAttribute("src");
        alert(imgSrc)
    }
    function setTitle(){
        //设置属性
        document.getElementById("img1").setAttribute("title","动物")
    }
    function delTitle(){
        //设置属性
        document.getElementById("img1").removeAttribute("title")
    }
</script>

```

3.内容操作

3.1 表单值操作

```

姓名: <input type="text" name="uname"> <br>
密码: <input type="password" name="pwd"> <br>
性别: <input type="radio" name="sex" value="男">男 <input type="radio" name="sex"
value="女">女 <br>
爱好: <input type="checkbox" name="hobby" value="读书">读书
<input type="checkbox" name="hobby" value="音乐">音乐
<input type="checkbox" name="hobby" value="电影">电影 <br>
<input type="button" value="获取信息" onclick="getInfo()">
<input type="button" value="设置信息" onclick="setInfo()">
<script>

```

```

function getInfo() {
    console.log(document.getElementsByName("uname")[0].value);
    console.log(document.getElementsByName("pwd")[0].value);
    var sex = document.getElementsByName("sex");
    for(var i = 0; i < sex.length; i++) {
        if(sex[i].checked) {
            console.log(sex[i].value)
        }
    }
    var hobbies = document.getElementsByName("hobby");
    for(var i = 0; i < hobbies.length; i++) {
        if(hobbies[i].checked) {
            console.log(hobbies[i].value)
        }
    }
}

function setInfo() {
    document.getElementsByName("uname")[0].value = "刘备";
    document.getElementsByName("pwd")[0].value = "123456"
    var sex = document.getElementsByName("sex");
    for(var i = 0; i < sex.length; i++) {
        if(sex[i].value == "男") {
            sex[i].checked = true;
        }
    }
    var hobbies = document.getElementsByName("hobby");
    for(var i = 0; i < hobbies.length; i++) {
        if(hobbies[i].value == "音乐") {
            hobbies[i].checked = true
        }
    }
}
}
</script>

```

3.2 普通元素文本操作

```

<p id="pid1"></p>
<p id="pid2"></p>
<input type="button" value="获取信息" onclick="getInfo()">
<input type="button" value="设置信息" onclick="setInfo()">
<script>
    function setInfo() {
        document.getElementById("pid1").innerText = "这是第一段内容";
        document.getElementById("pid2").innerHTML = "<span style='color: red'>这是第一段内容</span>";
    }
    function getInfo() {
        console.log(document.getElementById("pid1").innerText)
        console.log(document.getElementById("pid2").innerHTML)
    }
}
</script>

```

4. 样式操作

4.1 行内样式

4.2 行外样式

```
<style>
    #div1 {
        color: blue
    }
</style>
<body>
    <div id="div1" style="font-size: 36px">这是一个div</div>
    <button onclick="getStyle1()">获取div的行内样式</button>
    <button onclick="getStyle2()">获取div的行外样式</button>
<script>
    window.onload = function () {
        var div1 = document.getElementById("div1");
        div1.style.width = "200px";
        div1.style.height = "200px";
        div1.style.backgroundColor = "yellow"
    }
    function getStyle1() {
        var div1 = document.getElementById("div1");
        console.log(div1.style.backgroundColor)//yellow
        console.log(div1.style.color) //空白
        console.log(div1.style.fontSize)//36px
    }
    function getStyle2() {
        var div1 = document.getElementById("div1");
        //document.defaultView:document关联的window对象
        //大部分情况下, 可以直接通过 window 对象调用getComputedStyle()
        //getComputedStyle()有两个参数: 元素和伪类, 仅支持读取样式
        // var computedStyle = document.defaultView.getComputedStyle(div1,null);//ie9以
上
        var computedStyle = div1.currentStyle;//ie9以下
        console.log(computedStyle.backgroundColor)//rgb(255, 255, 0) || yellow
        console.log(computedStyle.color) //rgb(0, 0, 255) || blue
        console.log(computedStyle.fontSize)//36px

    }
</script>
```

5. 类属性操作

```
<style>
    .div1 {
        font-size: 36px;
        color: blue
    }
    .div2 {
        width: 200px;
        height: 200px;
```

```

        border: 1px solid red;
    }
    .div3 {
        background-color: yellow;
    }
</style>
<div id="div1" class="div1 div2">
    这是一个div
</div>
<button onclick="getClass()">获取类</button>
<button onclick="setClass()">添加类</button>
<button onclick="delClass()">移除类</button>
<button onclick="containsClass()">是否包含某个类</button>
<button onclick="toggleClass()">切换类</button>
<script>
    var div1 = document.getElementById("div1");
    function getClass() {
        console.log(div1.className)
        console.log(div1.classList)
    }
    function setClass() {
        // div1.className = "div3";
        div1.classList.add("div3")
    }
    function delClass() {
        // div1.className = ""; //全部移除
        div1.classList.remove("div2")
    }
    function containsClass() {
        console.log(div1.classList.contains("div2"))
    }
    function toggleClass() {
        div1.classList.toggle("div2")
    }
</script>

```

6.事件操作

6.1 鼠标相关

- onclick: 单击事件
- ondblclick: 双击
- onmousedown: 按下鼠标
- onmouseup: 释放鼠标
- onmouseover: 移上
- onmouseout: 移开

6.2 键盘相关

- onkeydown: 按下键盘
- onkeyup: 释放键盘
- onkeypress: 按下并释放


```

姓名: <input type="text" name="uname" onkeydown="setStyle()" >
<script>
    function setStyle() {
        if(event.keyCode == 13) { //enter键
            var uname = document.getElementsByName("uname")
            //获得焦点背景色为粉色
            uname[0].style.backgroundColor = "pink"
        }
    }
</script>

```

6.3 body相关

- onload: 页面加载
- onresize: 改变大小
- onscroll: 滚动条滚动

6.4 form相关

- onsubmit: 提交表单
- onfocus: 获得焦点
- onblur: 失去焦点
- onchange: 常用于下拉列表的选项发生改变触发的时间。按钮没有这个事件

```

姓名: <input type="text" name="uname" onblur="checkName()" onfocus="setStyle()">
<span id="sid" style="color: red"></span>
<script>
    function checkName() {
        var uname = document.getElementsByName("uname");
        if(uname[0].value == "") {
            //失去焦点背景色为粉色
            uname[0].style.backgroundColor = "yellow"
            document.getElementById("sid").innerText = "姓名不能为空"
        } else {
            document.getElementById("sid").innerText = ""
        }
    }
    function setStyle() {
        var uname = document.getElementsByName("uname")
        //获得焦点背景色为粉色
        uname[0].style.backgroundColor = "pink"
    }
</script>

```

```

<select name="fruit" id="fruit" onchange="getFruit()">
    <option value="apple">苹果</option>
    <option value="banana">香蕉</option>
    <option value="orange">橘子</option>
    <option value="strawberry">草莓</option>
</select>
<span id="show"></span>

```

```
<script>
function getFruit(){
    var mySelect = document.getElementById("fruit");
    //返回下拉列表中的选项的数目
    for(var i = 0; i < mySelect.length; i++) {
        if (mySelect[i].selected) {
            alert(mySelect[i].value)
        }
    }
    //selectedIndex 设置或返回下拉列表中被选项目的索引号
    var index = mySelect.selectedIndex;
    //options 返回包含下拉列表中的所有选项的一个数组
    var fruitName = mySelect.options[index].text;
    // var fruit = mySelect.options[index].value;
    var fruit = mySelect.value;
    var show = document.getElementById("show");
    show.innerHTML = "选择的水果是" + fruit + "-" + fruitName
        + "<br/>索引号为: " + index
        + "<br/>下拉列表选项数目为: " + mySelect.length;
}
```