



JavaScript

SWILCH-Case

```
分支结构 🗸
                     三元表达式
                     详细语法参考Java
  流程控制。
                      while ____ 详细语法参考Java
            循环结构
                    __ do-while ____ 详细语法参考Java
                     for ____ 详细语法参考Java
                    function 函数名字(){
          代码的重用
                    函数的调用 _____ 函数名();
          函数一旦重名,后面的会把前面的函数覆盖
                   在函数定义的时候,函数名字后面的小括号里的变量就是参数,目的是函数在调用的时候,用户传进来的值操作
                   此时函数定义的时候后面的小括号里的变量叫参数;写了两个变量,就有两个参数,
                   在函数调用的时候,按照提示的方式,给变量赋值--->就叫传值,把这个值就传到了变量(参数)中
          函数参数
                   形参 _____ 函数在定义的时候小括号里的变量叫形参
                   实参 ______函数在调用的时候小括号里传入的值叫实参,实参可以是变量也可以是值
                   函数可以作为参数使用,如果一个函数作为参数,那么我们说这个参数(函数)可以叫回调函数
                      在函数内部有return关键字,并且在关键字后面有内容,这个内容被返回了
                      当函数调用之后,需要这个返回值,那么就定义变量接收即可
                      如果一个函数中有return,那么这个函数就有返回值
  函数
                      如果一个函数中没有return,那么这个函数就没有返回值
                     如果一个函数中没有明确的返回值,那么调用的时候接收了,结果就是undefined
         函数的返回值
                     变量声明了,没有赋值,结果也是undefined
                     如果一个函数有参数,有参数的函数
                     如果一个函数没有参数,没有参数的函数
                     形参的个数和实参的个数可以不一致
                     return 下面的代码是不会执行的
                     函数是可以作为返回值使用的
                    数组
         arguments _
                    使用arguments对象可以获取传入的每个参数的值
                   命名函数 _______ 函数如果有名字,就是命名函数
                   匿名函数 ____
                            函数如果没有名字,就是匿名函数
          函数定义
                              把一个函数给一个变量,此时形成了函数表达式
                             var 变量=匿名函数;
                   函数表达式。
                                              如果是函数表达式,那么此时前面的变量中存储的就是一个函数,而这个变量就相当于是一个函数,就可以直接加小括号调用了
                             var f1=function (){}; _____ f1();
                                              函数表达式后面,赋值结束后,要加分号
                    声明的变量是使用var声明的,那么这个变量就是全局变量,全局变量可以在页面的任何位置使用
           全局变量。
                  除了函数以外,其他的任何位置定义的变量都是全局变量
                    全局变量,如果页面不关闭,那么就不会释放,就会占空间,消耗内存
           局部变量 _____ 在函数内部定义的变量,是局部变量,外面不能使用
           全局作用域 _____ 全局变量的使用范围
           局部作用域 _____ 局部变量的使用范围
  作用域。
          _ js没有块级作用域,只有函数除外
                       声明的变量没有var,就叫隐式全局变量
           隐式全局变量
                     全局变量是不能被删除的,隐式全局变量是可以被删除的
          定义变量使用var是不会被删除的,没有var是可以删除的
          作用域链
                       把变量的声明提前了----提前到当前所在的作用域的最上面
           提前解析代码
                      函数的声明也会被提前---提前到当前所在的作用域的最上面
  预解析
                      函数中的变量只会提前到函数的作用域中的最前面,不会出去
           预解析会分段(多对的script标签中函数重名,预解析的时候不会冲突)
                 Object Model --->文档对象模型----操作页面的元素
                 getElementById _____ 通过id访问元素
                                            返回的是一个伪数组
                 getElementsByTagName("标签的名字") <
                                          无论获取的是一个标签,还是多个标签,最终都是在数组中存储的,这行代码的返回值就是一个数组
                 getElementsByClassName() _____根据类样式的名字来获取元素
                 querySelector("选择器的名字") 根据选择器获取元素,返回来的是一个元素对象
                 querySelectorAll("选择器的名字") _____ 根据选择器获取元素,返回来的是一个伪数组,里面保存了多个的DOM对象
                                 src
                                 title
                 操作基本标签的属性
                                href
                 操作表单标签的属性 _____ name,value,type,checked,selected,disabled,readonly
                              _ 对象.style.属性=值;
                 元素的样式操作
                            対象.className=值;
元素 _____ 页面中所有的标签,元素---element,标签----元素---对象
       页面中所有的内容(标签,属性,文本(文字,换行,空格,回车)),Node
              nodeType _____ 节点的类型:1----标签,2---属性,3---文本
       属性 ____ nodeName ____ 节点的名字:标签节点---大写的标签名字,属性节点---小写的属性名字,文本节点----#text
            nodeValue _____ 节点的值:标签节点---null,属性节点---属性值,文本节点---文本内容
       Obj.parentNode ____ 标签的父级节点
       Obj.parentElement ____ 标签的父级元素
       Obj.parentNode.nodeType
       Obj.parentNode.nodeName ____ 标签---大写的标签名字
       Obj.parentNode.nodeValue ____ 标签---null
       Obj.childNodes _____ 子节点
节点
      Obj.children ____ 子元素
      Obj.firstChild ____ 第一个子节点
      Obj.firstElementChild _____ 第一个子元素
      Obj.lastChild ____ 最后一个子节点
      Obj.lastElementChild ____ 最后一个子元素
       .previousSibling ____ 某个元素的前一个兄弟节点
       .previousElementSibling ____ 某个元素的前一个兄弟元素
       .nextSibling ____ 某个元素的后一个兄弟节点
      .nextElementSibling ____ 某个元素的后一个兄弟元素
             凡是获取节点的代码在谷歌和火狐得到的都是 相关的节点
       总结 ______ 凡是获取元素的代码在谷歌和火狐得到的都是 相关的元素
            ✓ 从子节点和兄弟节点开始,凡是获取节点的代码在IE8中得到的是元素,获取元素的相关代码,在IE8中得到的是undefined----元素的代码,IE中不支持
根元素 ____ html标签
       就是一件事,有事件源,触发和响应
                    1对象.addEventListener("事件类型",事件处理函数,false); _____ 谷歌和火狐支持,IE8不支持
                   2 对象.attachEvent("有on的事件类型",事件处理函数) _____ 谷歌不支持,火狐不支持,IE8支持
                       参数1:事件的类型---事件的名字,没有on
       为按钮绑定点击事件 参数2:事件处理函数---函数(命名函数,匿名函数)
                     参数3:布尔类型,目前就写false
                               参数1:事件类型---事件名字,有on
       为同一个元素绑定多个相同的事件。
                             参数2:事件处理函数---函数(命名函数,匿名函数)
                                  相同点: 都可以为元素绑定事件
                                          1.方法名不一样
       总结attachEvent()和addEventListener();
                                          2.参数个数不一样addEventListener三个参数,attachEvent两个参数
                                  不同点
                                         _ 3.addEventListener 谷歌,火狐,IE11支持,IE8不支持 attachEvent 谷歌火狐不支持,IE11不支持,IE8支持
                                          4.this不同,addEventListener 中的this是当前绑定事件的对象 attachEvent中的this是window
                                          5.addEventListener中事件的类型(事件的名字)没有on attachEvent中的事件的类型(事件的名字)有on
事件
              对象.on事件名字=事件处理函数--->绑定事件
             对象.on事件名字=null;
             对象.addEventListener("没有on的事件类型",命名函数,false);---绑定事件
             对象.removeEventListener("没有on的事件类型",函数名字,false);
              对免 attachEvent("on事件米刑" 命名函数):---绑宁事件
```

```
对象.detachEvent("on事件类型",函数名字);
                             多个元素嵌套,有层次关系,这些元素都注册了相同的事件,如果里面的元素的事件触发了,外面的元素的该事件自动的触发了
                  事件冒泡
                                          window.event.cancelBubble=true; _____IE特有的,谷歌支持,火狐不支持
                            阻止事件冒泡
                                         e.stopPropagation(); 谷歌和火狐支持,
                             1.事件捕获阶段:从外向内
                             2.事件目标阶段:最开始选择的那个
                            3.事件冒泡阶段: 从里向外
                            为元素绑定事件
                  事件阶段
                            事件触发的过程中,可能会出现事件冒泡的效果,为了阻止事件冒泡--->
DOM
                                                                                        1---->捕获阶段
                             通过e.eventPhase这个属性可以知道当前的事件是什么阶段 ____ 如果这个属性的值是: __
                                                                                        2---->目标阶段
                            一般默认都是冒泡阶段,很少用捕获阶段
                  凡是成对的标签,中间的文本内容,设置的时候,都使用innerText这个属性的方式
                  判断这个元素是不是按钮_
                                          innerText 打印标签之间的纯文本信息,会将标签过滤掉
                                         如果使用innerText主要是设置文本的,设置标签内容,是没有标签的效果的
                                          innerText可以获取标签中间的文本内容,但是标签中如果还有标签,那么最里面的标签的文本内容也能获取.---获取不到标签的,文本可以获取
                  innerText 与 innerHtml的区别。
                                         innerHTML才是真正的获取标签中间的所有内容
                                         innerHtml 打印标签之间的内容,包括标签和文本信息
                                         innerHTML主要的作用是在标签中设置新的html标签内容,是有标签效果的
                                         要设置标签内容,使用innerHTML,想要设置文本内容,innerText或者textContent,或者innerHTML,推荐用innerHTML
                                         设置标签中的文本内容,应该使用textContent属性,谷歌,火狐支持,IE8不支持
                                        设置标签中的文本内容,应该使用innerText属性,谷歌,火狐,IE8都支持
                                        如果这个属性在浏览器中不支持,那么这个属性的类型是undefined
                  innerText和textContent区别。
                                        判断这个属性的类型 是不是undefined,就知道浏览器是否支持
                                        兼容代码
                         如果想要(获取)标签及内容,使用innerHTML
                         如果想要设置标签,使用innerHTML
                         想要设置文本,用innerText,或者innerHTML,或者textContent
          经验
                  在表单标签中,如果属性和值只有一个,并且值是这个属性本身,那么在写js代码,DOM操作的时候,这个属性值,是布尔类型就可以了 _____ my$("rad1").checked = true;
                  凡是css中这个属性是多个单词的写法,在is代码中DOM操作的时候,把-干掉,后面的单词的首字母大写即可
                  在is代码中DOM操作的时候,设置元素的类样式,不用class关键字,应该使用,className
                                     _ <input type="button" value="显示效果" onclick="alert('我被点了')" />
                                    html代码中嵌入了js的代码,不方便后期的修改和维护
                            第二版
                                    __js代码很多,但是没有分离html和js该怎么做
                  编写历程
                                      开始分离html和js
                                               html标签中的id属性中存储的值是唯一的
                                               id属性就像人的身份证号码一样,不能重复,页面中的唯一的标识
                             第三版
                                     注意点
                                              从文档中找到id值为btn的这个标签(元素)
                                              document.getElementById("id属性的值");=====>返回的是一个元素对象
                                              根据id获取这个标签(元素)
                                     对于input在script上边 _____ 先有按钮,才能获取,获取之后才能注册事件
                  this关键字 _____ 如果是在当前的元素的事件中使用this,那么this就是当前的对象
                                              第一种写法 _____ <a href="http://www.baidu.com" onclick="alert('哎呀,我被点了'); return false">百度</a>
                                                          function f1() {
                                                          alert("哇塞,好漂亮哦");
                  阻止超链接的默认的跳转:return false
                                              第二种写法
                                                          return false;
                                              第三种写法 _
                       在html标签中添加的自定义属性,如果想要获取这个属性的值需要使用getAttribute("自定义属性的名字")才能获取这个属性的值
                       设置自定义属性:setAttribute("属性的名字","属性的值");
           自定义属性
                      获取自定义属性的值:getAttribute("属性的名字")
                       移除自定义属性 removeAttribute("属性的名字")
                                                   document.write("这是一个p");
                     1. document.write("标签的代码及内容");
                                                       如果是在页面加载完毕后,此时通过这种方式创建元素,那么页面上存在的所有的内容全部被干掉
                     2. 对象.innerHTML="标签及代码"; ______ my$("dv").innerHTML="窗前明月光,疑是地上霜,举头望明月,低头思故乡";
                                                     document.createElement("标签名字");对象
                                                     把元素追加到父级元素中
                     3. document.createElement("标签的名字");
                                                     点击按钮,在div中创建一个p
          元素创建
                     把新的子元素插入到第一个子元素的前面 _____ my$("dv").insertBefore(obj,my$("dv").firstElementChild);
                     my$("dv").replaceChild();
                     移除父级元素中第一个子级元素 _____ my$("dv").removeChild(my$("dv").firstElementChild);
                     点击按钮删除div中所有的子级元素,判断父级元素中有没有第一个子元素 __
          Browser Object Model 浏览器对象模型,操作浏览器
                    window.alert("您好啊");//以后不用,测试的时候使用
                    window.prompt("请输入帐号");
                    window.confirm("您确定退出吗");
                               页面加载的时候,按钮还没出现,页面没有加载完毕
                    加载事件
                                             只要页面加载完毕,这个事件就会触发
                               window.onload _
                                             页面中所有的内容,标签,属性,文本,包括外部引入js文件
                    扩展事件。
                              window.onbeforeunload _____ 页面关闭之前触发的
                    扩展的事件_
                                window.onunload _____ 页面关闭后才触发的事件
                              window.location.hash ____ 地址栏上#及后面的内容
                              window.location.host ____ 主机名及端口号
                              window.location.hostname ____ 主机名
                              window.location.pathname ____ 文件的路径---相对路径
                              window.location.port ____ 端口号
                    location
                              window.location.protocol __
          window
                              window.location.search _
                                                 搜索的内容
                                                 location.href="http://www.jd.com"; _____ 属性
BOM
                                                 location.assign("http://www.jd.com"); _____ 方法
                              设置跳转的页面的地址
                                                 location.reload(); ____ 重新加载--刷新
                                                 location.replace("http://www.jd.com"); _____ 没有历史记录
                             window.history.forward(); ____ 前进
                    history
                            window.history.back(); ____ 后退
                               window.navigator.userAgent _____判断用户浏览器的类型
                    navigator
                              window.navigator.platform _____ 判断浏览器所在的系统平台类型
                                                     参数1:函数
                                                     参数2:时间---毫秒---1000毫秒--1秒
                             循环定时,
                                       setInterval()
                                                     返回值就是定时器的id值
                                                     结束 ____ clearInterval()
                    定时器
                                                      参数1:函数
                                                      参数2:时间---1000毫秒---1秒
                             一次性的定时
                                          setTimeout
                                                      返回值:该定时器的id
                                                      结束 ____ clearTimeout(timeld);
                     offsetWidth:获取元素的宽
                     offsetHeight:获取元素的高
          offset系列。
                     offsetLeft:获取元素距离左边位置的值
                     offsetTop:获取元素距离上面位置的值
ECMAScript
```

是一门解释性语言
是一门基于对象的语言
是一门基于对象的语言

1.在一对script的标签中有错误的js代码,那么该错误的代码后面的js代码不会执行

2.如果第一对的script标签中有错误不会影响后面的script标签中的js代码执行

3.script的标签中可以写什么内容 type="text/javascript"是标准写法或者写language="JavaScript"都可以

但是,目前在我们的html页面中,type和language都可以省略,原因:html是遵循h5的标准

4.有可能会出现这种情况:script标签中可能同时出现type和language的写法

5.script标签在页面中可以出现多对

6.script标签一般是放在body的标签的最后的,有的时候会在head标签中,目前讲课的时候都在body标签的后面(body中的最后) 7.如果script标签是引入外部js文件的作用,那么这对标签中不要写任何的js代码,如果要写,重新写一对script标签,里面写代码