2020年6月26日星期五

* 函数
  + 可以在声明前调用
  + 如果在浏览器中，写在全局的函数：var fun = function(){return 1};，其实也是方法哦，它是属于 window 对象的
  + this
    - 表示当前方法属于谁
    - 服务器中的全局函数：指向全局对象[object\_global]
    - 浏览器中的全局函数：指向window对象
  + call方法
    - call() 的第一个参数就是改变的 this 指向，
    - 如果为 null 则函数的 this 不变，
    - 注意，如果在严格模式下 （函数体或全局的开头有这句话：'use strict'），this 会变成 null。
    - 如果函数本身有参数，则从 call 的第二个参数开始写起。
* 在网页中使用JavaScript
  + 直接写在HTML文件中
    - <script type="text/javascript"> 这里写 JavaScript </script>
    - 可以放在任意位置，但是前面的script先执行
      * 页面初始化的js必须放在head里
  + 分开
    - <script src="script.js"></script>
  + 事件绑定
    - 元素.事件 = 函数名
* document.write()
  + 让内容输出到页面中去。
  + document.write()可用于直接向 HTML 写内容。
  + 使用 document.write() 输出到页面，浏览器是可以识别 html 内容的，比如各种标签及特殊字符等。
  + 在 document.write() 方法下如果想输出连续的空格的话，在字符串中多打几个空格符是不行 的，可以使用输出 html 标签&amp;nbsp;（在 html 中代表空格哦~）来解决。
* 警告框
  + 突然弹出一个小窗口
  + 不点击"确定"或者关闭按钮，就不能对网页做任何操作
  + 使用语法：alert（字符串或变量）
* Element.setAttribute(attribute\_name, new\_attribute\_value)
  + 若不存在，则新增
* 打开新窗口
  + 使用语法：window.open(<URL>, <窗口名称>, <参数字符串>)
* window对象
  + 指当前的浏览器窗口
* 计时器有四种方法：
  + setTimeout() ：在指定的毫秒数后调用函数或计算表达式。
  + clearTimeout() ：取消由 setTimeout() 方法设置的 timeout。
  + setInterval() ：按照指定的周期（以毫秒计）来调用函数或计算表达式。
  + clearInterval() ：取消由 setInterval() 设置的 timeout。下面我们会详细介绍这四种方法哦~