1.输入语法：prompt(‘请输入你的语言’)

2.输出： alert() document.write() console.log()

3.undefined 表示没有赋值 null 表示赋值了，但是内容为空

4.typeof 关键字检测数据类型

5.强制转换Number(数据) parseInt(数据)： 只保留整数 ；parseFloat(数据)： 可以保留小数

6.转换为字符型: String(数据) 变量.toString(进制)

7.continue：结束本次循环，继续下次循环 ; break：跳出所在的循环

8.数组.push() 方法将一个或多个元素添加到数组的末尾，并返回该数组的新长度 (重点)

arr.unshift(新增的内容) 方法将一个或多个元素添加到数组的开头，并返回该数组的新长度

数组. pop() 方法从数组中删除最后一个元素，并返回该元素的值

数组. shift() 方法从数组中删除第一个元素，并返回该元素的值

数组. splice(start, deleteCount) 方法 删除指定元素 start 起始位置: 指定修改的开始位置（从0计数） deleteCount: 表示要移除的数组元素的个数 可选的。 如果省略则默认从指定的起始位置删除 到最后

9.遍历对象中, for k in obj,获得对象属性是那个，获得值是那个？ 获得对象属性是 k 获得对象值是 obj[k]

10. Math

random：生成0-1之间的随机数（包含0不包括1）

ceil：向上取整

floor：向下取整

max：找最大数

min：找最小数

pow：幂运算

abs：绝对值

11.根据id获取一个元素document.getElementById(‘aa’)

根据标签获取一类元素 获取页面所有div document.getElementByTagName(‘div’)

根据类名获取元素获取页面所有类名为aa的 document.getElementByClassName(‘aa’)

12.

document.write() 方法 只能追加到body中

¬

元素.innerText 属性 只识别内容，不能解析标签

¬ 元素.innerHTML 属性 能够解析标签从

13.

className 赋值会覆盖以前的类名 元素.className=’active’

元素.classList.remove(‘类名’)删除一个类

元素.classList.add(‘类名’)追加一个类

元素.classList.toggle(‘类名’)切换一个类

classList 是追加和删除不影响以前类名

14. let 变量名=setInterval(函数，间隔时间)

clearInterval(变量名)

15．监听事件类型：click 鼠标点击 mouseenter 鼠标经过 mouseleave 鼠标离开

focus 获得焦点 blur 失去焦点

Keydown 键盘按下触发 Keyup 键盘抬起触发

input 用户输入事件

16. 父节点查找: parentNode 属性 子元素.parentNode

¬ 返回最近一级的父节点 找不到返回为null

17. 子节点查找：childNodes 获得所有子节点、包括文本节点（空格、换行）、注释节点等 children重点 仅获得所有元素节点 返回的还是一个伪数组

父元素.children

18. 兄弟关系查找： 1. 下一个兄弟节点 ¬ nextElementSibling 属性

2. 上一个兄弟节点 ¬ previousElementSibling 属性

19.创建一个新的元素节点doucment.createElement(‘标签名’)

插入到这个父元素的最后 父元素.appendChild(要插入的元素)

插入到子元素的前面 父元素.insertBefore(要插入的元素，在哪个元素前面)

20 元素.cloneNode(布尔值)：cloneNode会克隆出一个跟原标签一样的元素，括号内传入布尔值 ¬ 若为true，则代表克隆时会包含后代节点一起克隆

¬ 若为false，则代表克隆时不包含后代节点 ¬ 默认为false

21. 父元素.removeChild(要删除的元素) 删除元素要通过父元素删除

22.

getFullYear() 获得年份 获取四位年份

getMonth() 获得月份 取值为 0 ~ 11

getDate() 获取月份中的每一天 不同月份取值也不相同

getDay() 获取星期 取值为 0 ~ 6

getHours() 获取小时 取值为 0 ~ 23

getMinutes() 获取分钟 取值为 0 ~ 59

getSeconds() 获取秒 取值为 0 ~ 59

1. 实例化时间对象 ¬ new Date()

2. 时间对象方法 ¬ 时间对象里面的方法转换实际所用

3. 时间戳 ¬ date.getTime() ¬ +new Date() ¬ Date.now() ¬ 重点记住 +new Date() 因为可以返回当前时间戳或者指定的时间戳

23. clientX/clientY ➢ 获取光标相对于浏览器可见窗口左上角的位置 ➢

offsetX/offsetY ➢ 获取光标相对于当前DOM元素左上角的位置

24

事件对象.stopPropagation()阻断事件流动传播

25

➢ mouseover 和 mouseout 会有冒泡效果

➢ mouseenter 和 mouseleave 没有冒泡效果(推荐)

26

方法.prenentDefaule()链接不跳转

27

1. 事件委托是委托给了谁？
2. ➢ 父元素
3. 2. 如何找到真正触发的元素？ ➢ e.target 事件对象.target

28

1. scrollWidth和scrollHeight是得到元素什么的宽高？

¬ 内容 ¬ 不包含滚动条

1. 被卷去的头部或者左侧用那个属性？是否可以读取和修改？

¬ scrollTop / scrollLeft ¬ 可以读取，也可以修改（赋值）

1. 检测页面滚动的头部距离（被卷去的头部）用那个属性？

¬ document.documentElement.scrollTop

29

1. offsetWidth和offsetHeight是得到元素什么的宽高？

¬ 内容 + padding + border

1. offsetTop和offsetLeft 得到位置以谁为准？

¬ 带有定位的父级 ¬ 如果都没有则以 文档左上角为准

30.

获取宽高： ¬ 获取元素的可见部分宽高（不包含边框，滚动条等）¬ clientWidth和clientHeight

λ 获取位置： ¬ 获取左边框和上边框宽度 ¬ clientLeft和clientTop 注意是只读属性

31.

window.addEventListener('resize', function () {

          console.log(document.doucmentElement.clientWidth)

        })检测屏幕宽度

32.

setTimeout 仅仅只执行一次，所以可以理解为就是把一段代码延迟执行, 平时省略window

33

¬ location.href 属性获取完整的 URL 地址，对其赋值时用于地址的跳转

¬ search 属性获取地址中携带的参数，符号 ？后面部分

¬ hash 属性获取地址中的啥希值，符号 # 后面部分

¬ reload 方法用来刷新当前页面，传入参数 true 时表示强制刷新

34

localStorage

localStorage.setItem(key, value) 存储数据：

localStorage.getItem(key) 获取数据：

localStorage.removeItem(key) 删除数据

存储复杂数据类型存储 本地只能存储字符串,无法存储复杂数据类型.需要将复杂数据类型转换成JSON字符串,在存储到本地

JSON.stringify(复杂数据类型)

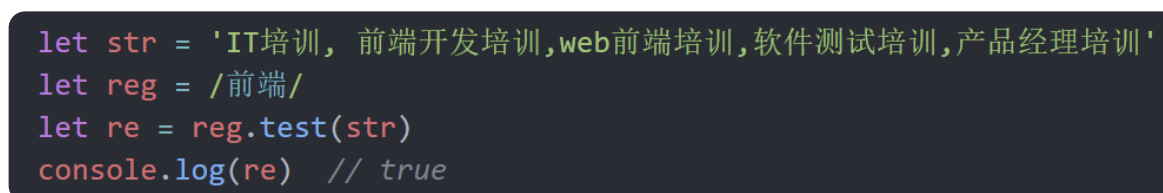
¬ 将复杂数据转换成JSON字符串 存储 本地存储中

JSON.parse(JSON字符串)

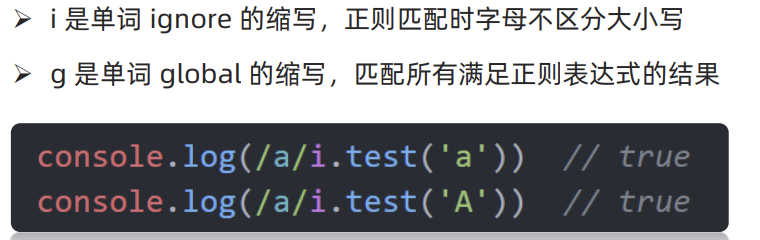
¬ 将JSON字符串转换成对象 取出 时候使用

35

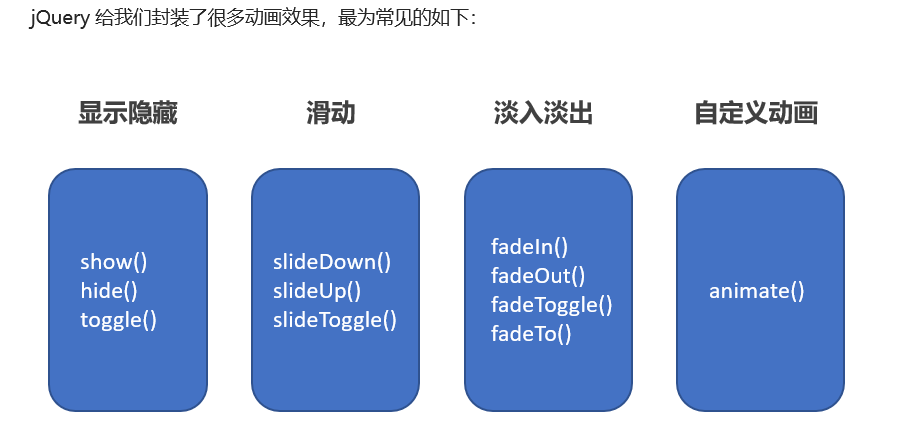
1.正则表达式使用分为几步？ ¬ 定义正则表达式 ¬ 检测查找是否匹配

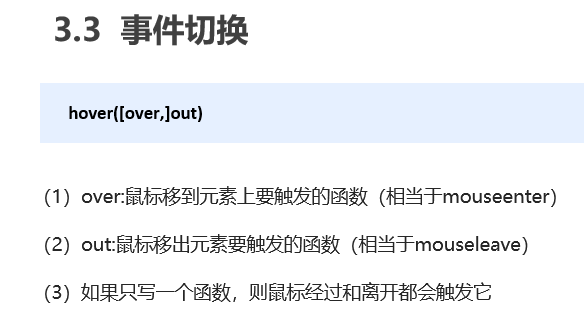


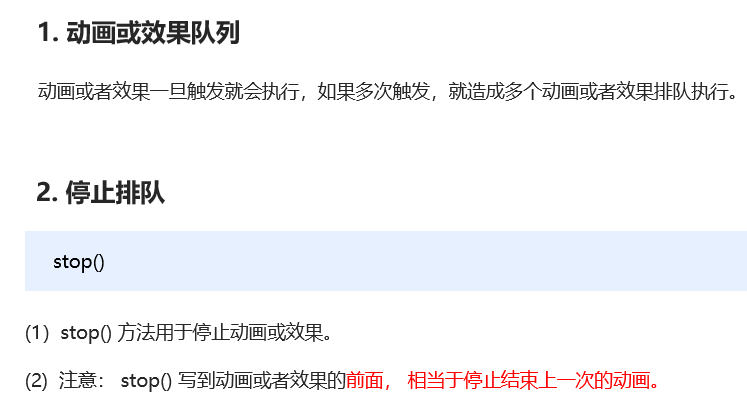


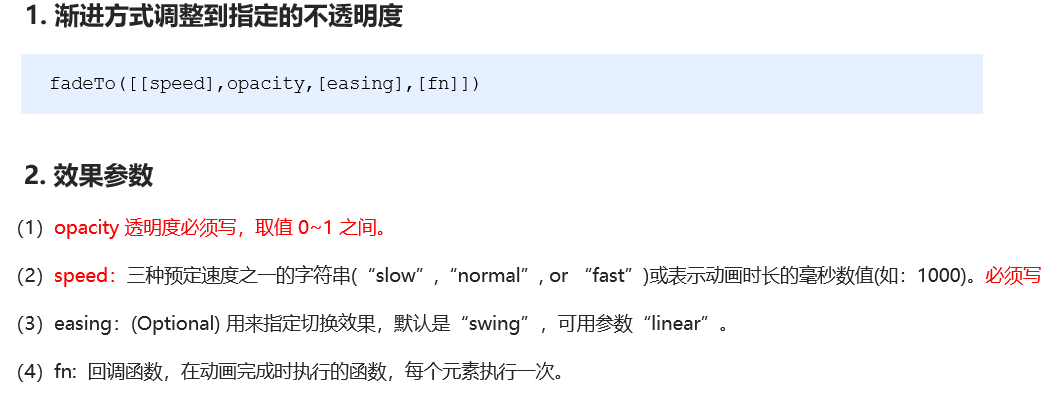


36









37

