

1, 变量

var age;
命名规范
var声明提前,
typeof , instanceof
数据类型:
特殊值: null, undefined, null==undefined

parseInt(null) = NaN

算术运算符 + - * / %
12+3+"3"+2-3*2

2, 运算符和逻辑分支

++, --
var i=10;
i++ + ++i + ++i*2 + ++i

|| , &&
var b = 3 && 4;
var c = true || false

? :

if, if-else , if-elseif-else

```
var grade = "C";
switch(grade) {
  case "A":
    console.log("80~100");
    break; //如果不写break会造成case穿透
  case "B":
    console.log("70~79");
    break;
  case "C":
    console.log("60~69");
    break;
  case "D":
    console.log("<60");
    break;
  default:
    console.log("error");
}
```

3, 循环

for, while, do-while, for-in(遍历对象)
break, continue, return区别

4, 函数

普通函数, 匿名函数
arguments数组
作用域

```
var a = 1;  
function fn(a){  
    a = 2;  
}  
fn(a)  
console.log(a);
```

5, 数组

数组的声明方式
遍历数组for, for-in, 数组通过下标取值
数组的属性: length
数组的方法: push, pop, shift, unshift, reverse, sort, concat, slice, splice, join
冒泡排序
sort的排序规则: 按字符串比较规则来排序

6, 对象

对象的声明方式:
对象包含: 属性, 方法

7, 字符串

字符串的声明方式
字符串+
字符串通过下标获取字符
字符串的属性: length
字符串的方法: charAt, charCodeAt, String.fromCharCode, concat, indexOf, search, lastIndexOf, replace, substring, split, toLowerCase, toUpperCase

8, Math

Math数学函数
Math.round(3.6) //四舍五入
Math.random() //返回0-1之间的随机数
Math.max(num1, num2) //返回较大的数
Math.min(num1, num2) //返回较小的数
Math.abs(num) //绝对值
Math.ceil(19.3) //向上取整
Math.floor(11.8) //向下取整
Math.pow(x, y) //x的y次方

Math.sqrt(num) //开平方

练习: 1, 获取65~90中的随机数, 并将该数当成ASCII码, 转换成字母

```
var n = parseInt(Math.random()*26) + 65; //65~90
var ch = String.fromCharCode(n);
console.log(ch); //随机的大写字母
```

练习2: 随机颜色(#FFFFFF)

9, Date

创建日期

时间戳

日期对象的方法:

```
getDate()/setDate(): 天
getDay(): 星期几(0~6, 0表示星期天)
getMonth()/setMonth(): 月, (0~11)
getFullYear()/setFullYear(): 年
getHours()/setHours(): 时
getMinutes()/setMinutes(): 分
getSeconds()/setSeconds(): 秒
getTime(): 获取时间戳
```

```
var d = new Date(2017, 12, 23); 打印年月日
```

10, BOM

BOM的核心对象是window对象

window对象的方法: alert, confirm, prompt, open, setInterval, setTimeout, close等

window对象的属性: location, history, navigator等
location.href, location.search等

11, DOM

节点分类: 元素节点, 属性节点, 文本节点

元素节点的获取: getElementById, getElementsByTagName, getElementsByName, getElementsByClassName

节点的属性: nodeName,.nodeType, nodeValue

元素节点的属性: innerHTML, value, children, childNodes, parentNode

元素节点操作: createElement, appendChild, createTextNode, insertBefore, replaceChild, cloneNode, removeChild

属性节点操作: getAttribute, setAttribute, removeAttribute

节点间关系的属性: childNodes, firstChild, lastChild, parentNode, previousSibling, nextSibling

12, 事件

事件写法: 内联模式, 脚本模式

鼠标事件, 键盘事件, html事件

鼠标事件: onclick, ondblclick, onmousedown, onmouseup, onmousemove, onmouseover, onmouseout, onmouseenter, onmouseleave,

键盘事件: onkeydown, onkeyup, onkeypress

html事件: onselect, onchange, onfocus, onblur, onsubmit, onreset, onresize, onscroll

scrollTop:

```
var scrollTop = document.documentElement.scrollTop || document.body.scrollTop;
```

事件对象:

```
var evt = e || event;
```

事件对象的属性:

button

which

clientX/clientY: 可视区域的x和y值

pageX/pageY: 页面内容区域的x和y值

screenX/screenY: 显示屏幕的x和y值

offsetX/offsetY: 鼠标点击元素的位置距离该元素左边界和上边界的距离

keyCode, charCode, ctrlKey, shiftKey, altKey

元素节点的属性:

offsetLeft/offsetTop: left值(不带单位), top值(不带单位)

offsetWidth/offsetHeight: width值(不带单位), height值(不带单位)

事件的传播方式: 冒泡, 捕捉

阻止冒泡:

```
evt.cancelBubble = true; //取消冒泡, IE  
evt.stopPropagation(); //停止传播, 非IE
```

阻止右键菜单

```
document.oncontextmenu = function(){  
    return false;  
}
```

阻止超链接

拖拽: onmousedown, onmousemove, onmouseup

事件监听器: addEventListener("click", fn, false); removeEventListener("click",
fn);

```
attachEvent("onclick", fn); detachEvent("onclick", fn);
```

获取css样式

```
function getStyleAttr(obj, attr){  
    if (window.getComputedStyle){
```

```

        return getComputedStyle(obj, null)[attr] ;
    }
    return obj.currentStyle[attr] ;
}

```

13, Cookie

cookie: 浏览器本地存储的一种技术, 存储文本(字符串)

格式: name=value;[expires=过期时间];[path=访问路径];[domain=域名];[secure]

document.cookie = "username=zhangsan";

var d = new Date(); //当前时间

d.setDate(d.getDate()+7); //7天后的日期

document.cookie = "username6=tianqi;expires=" + d + ";path="/;

14, 正则表达式

创建正则: var reg = new RegExp("box");

var reg = /box/;

g: 全局, i: 忽略大小写, m: 换行模式

正则表达式的方法: test, exec

字符串使用正则的方法:

str.match(): 匹配成功则返回数组, 否则返回null

str.replace(): 替换

str.search(): 查找子串第一次出现的下标, 如果没有找到则返回-1

str.split(): 拆分

//正则符号

. : 表示任意单个字符(除了换行)

* : 表示前面的字符可以出现任意多次(0次或者多次),

[] : 表示范围

[a-z]: 小写字母

[a-zA-Z]: 大写或小写字母

[a-zA-Z0-9_]: 字母, 数字或下划线

{ } : 表示出现的字数

{3} : 让前面的字符出现3次

{3,} : 让前面的字符出现3次或以上

{3,6} : 让前面的字符出现3~6次

^ : 非字符

+ : 前面的字符可以出现1次或多次

\w: 数字, 字母下划线, 等价于[a-zA-Z0-9_]

\d: 数字, 等价于[0-9]

^ \$/: 完全匹配

\s : 空格

| : 或

() : 分组,捕获

用户名为数字字母下划线, 第一个不能为数字, 长度在6~20: `/^[a-zA-Z_]\w{5,19}$/`
手机号(13或14或15或18开头的11位数字): `/^((13[0-9])|(14[57])|(15([0-3]|[5-9]))|(18[05-9]))\d{8}$/`
身份证: `/^\d{17}(\d|X)$/`
电子邮箱: `/^(\w)+(\.\w+)*@(\w)+((\.\w+)+)$/`
邮政编码(共6位数字, 第一位不能为0): `/^[1-9]\d{5}$/`

15, ES5, ES6

严格模式: `use strict`
`let`, `const`, 分别用于创建变量和常量, 可以形成块级作用域
字符串, 数组, 函数新增方法
箭头函数 `=>`
对象属性和方法的简写, 对象新增的方法
`Set`, `Map`, 解构赋值

16, 运动

熟练使用定时器 `setInterval`, `setTimeout` 的用法
匀速运动
缓冲运动
抛物线运动
弹性运动
透明度淡入淡出
链式运动
轮播图效果
放大镜, 瀑布流

17, 面向对象