

2.日期对象

math对象的属性

含义

- Date 是 ECMAScript 的 内置对象
- date 拥有构造函数 可以使用new关键字来创建对象

获得当前日期

- var d = new Date();
- 月份、星期、小时、分钟、秒、毫秒都是从0开始

获取时间的方法

- 获取年份 d.getFullYear()
- 获取月份 d.getMonth()
- 获取某日日期 d.getDate()
- 获取星期 d.getDay()
- 获取小时 d.getHours()
- 获取分钟 d.getMinutes()
- 获取秒数 d.getSeconds()
- 获取毫秒数 d.getMilliseconds()
- 获取时间戳 getTime()
- 获取GTM分钟差 getTimezoneOffset()
- 时间戳
 - 获取时间戳 d.getTime()
 - 在时间构造函数中 传入时间戳作为参数 可以将其还原成时间对象
 - 快捷方式，快速获取当前时间戳

修改日期

- 修改日期部分 setDate()
- 修改年份 setFullYear ()
- 修改月份 setMonth()
- 修改小时 setHour ()
- 修改分钟 setMinutes()
- 修改秒 setSecond()
- 设置毫秒 setMilliseconds()
- 设置date对象，以毫秒为单位 setTime ()

日期格式转换

- 转换成字符串 d.toString()
- 获得日期部分 d.toDateString()
- 获得时间部分 d.toTimeString()
- 获得本地时间格式 d.toLocaleString()
- 基于本地时间，返回日期（年月日） toLocaleDateString()
- 基于本地时间，返回（时分秒） toLocaleTlmeString ()
- 返回GMT时间格式 toGMTString ()
- 返回UTC时间格式 toUTCString()

周期性计时器(BOM)

- setInterval(); 全局函数
 - 语法 setInterval(callback,delay);
 - 参数
 - callback(function) 需要执行的函数
 - delay(number) 执行函数的间隔时间 以毫秒为单位
 - 描述
 - 周期性计时器setInterval 会自动计算delay事件
 - 每次间隔时间到 就会执行依次 callback
 - 返回值 id(number) 返回值是计时器的id
- 关闭周期性计时器 clearInterval();
 - 语法 clearInterval(id);
 - 参数 id(number) 开启计时器时的返回值
 - 描述 通过id关闭周期性计时器

常量的声明

- Math中保存了很多数学常数(常量) 不可改变的值
- // 常量的命名 全大写
- ES6中新增了一个关键字 const，用于申明一个常量

math对象的含义

- Math对象没有构造函数
- 无法创建实例
- Math对象是引用类型
- Math的属性和方法都是直接使用Math . 的操作
- math的类型是object
- object和array的数据类型都是function

引用类型和值类型

- 引用类型 在不修改地址的情况下 可以修改里面的值
- 值类型 不能修改

math对象的方法

- 1.绝对值 Math.abs(-3)
- 2.向上取整 Math.ceil(4.0000001)
- 3.自然对数 Math.log(2)
- 4.向下取整 Math.floor(-3.14)
- 5.返回最大的值 Math.max(654, 11, 5342, 453)
- 6.返回最小的值 Math.min(6345, 54, 32, 126)
- 7.计算x的y次幂 Math.pow(2, 4)
- 8.四舍五入 Math.round(3.49999999999)
- 9.开根 Math.sqrt(9)
- 10. ...操作符
 - Math.max(...arr)
 - ...的作用就是展开里面的数据，比如数组，那么就将中括号去掉，露出里面的数据
 - 1
- 11.三角函数
 - Math.sin() = 对边/斜边
 - Math.cos() = 邻边/斜边
 - Math.tan() = 对边/邻边

随机数: Math.random()

- 返回一个0-1之间的随机数(包含0 不包含1)
- 1

var d2 = new Date(1600929800248);

```
// 编写一个函数 生成一个特定范围的随机数
function random(min, max) {
  // 生成一个min-max范围内的随机数
  // return Math.round(Math.random() * (max - min) + min);
  return Math.floor(Math.random() * (max - min + 1) + min);
}
```

```
var arr = [453, 22, 654, 1, 4, 32, 6543, 543];
var obj = {
  max: Math.max(...arr),
  min: Math.min(...arr)
};
console.log(obj);
```