

1. 字符串方法

`String.fromCharCode(index)` 获取任意 Unicode 码对应的字符

`substring(start,end)` 截取字符串, `start` 开始的下标, `end` 结尾的下标, 不包含 `end` 本身; 如果两个参数是负数, 自动转成 0.

2. Date 对象

用于对日期和时间进行存储和计算

(1) 创建 Date 对象

`new Date('2018/11/11 10:20:30')`

`new Date(2018,10,11,10,20,30)` //第 2 个数月份的范围 0~11

`new Date()` 存储当前的系统时间

`new Date(1000*60*60*24)` 存储的是距离计算机元年的毫秒数对应的日期时间

(2) 获取 Date 对象的信息

`getFullYear/getMonth/getDate/getHours/getMinutes`

`/getSeconds/getMilliseconds/getDay(星期 0~6)`

`/getTime(距离计算机元年毫秒数)`

(3) 转为本地字符串

`toLocaleString()` // 年-月-日 时:分:秒

`toLocaleDateString()` //年-月-日

`toLocaleTimeString()` //时:分:秒

(4) 设置 Date 对象的信息

`setFullYear/setMonth/setDate/setHours/setMinutes`

`/setSeconds/setMilliseconds/`

`setTime` 设置距离计算机元年毫秒数, 一旦使用后, 年月日时分秒都会受到影响

练习: 创建对象保存员工的**入职时间**'2018-11-16', 3年后到期计算**到期时间**, 合同到期前1个月续签合同, 假如**续签时间**是周末, 提前到周五。提前一周通知人事准备续签(**提醒时间**)

3. Number 对象

`new Number(值)` 创建对象

将一数据转为 Number 对象, 本质还是数字。

`Number.MAX_VALUE` 获取计算机能存储的最大值

`Number.MIN_VALUE` 获取计算机能存储的最小值

`toFixed(n)` 保留小数点后 n 位

`toString()` 将数字转为字符串类型

数字+" //隐式转为字符串类型

4. Boolean 对象

`new Boolean(值)` 创建布尔对象, 本质上将数据转为布尔型

`Boolean(值)` 转为布尔型

`!!值` 隐式将数据转为布尔型

`toString()` 将布尔型数据转为字符串

null 的作用: 用于释放(销毁)一个引用类型的数据。

5. 错误处理

`SyntaxError` 语法错误, 错误的使用了中文, 缺少括号...程序不会执行。

`ReferenceError` 引用错误, 使用未声明的变量

`TypeError` 类型错误, 错误的使用了数据的类型, 错误的使用括号

`RangeError` 范围错误, 参数的使用超出了范围

`new Array(-1)`

引用错误、类型错误、范围错误出现后, 会影响后续代码的执行。

异常处理

```
try{
    尝试执行的代码, 可能出现错误
}catch(err){
    err: 捕获的错误信息
    处理错误的方案
}
```

6. ES6 新特征

ECMAScript6=> ES2017 ES2018...

(1) 块级作用域——局部作用域

使用 `let` 关键字声明的变量, 只能在块级作用域下访问, 不能被外部访问。

块级作用域: { }, for、while、do-while 都是块级作用域

(2) 箭头函数

是回调函数的另一种写法, 和匿名函数不一致

```
sort( (a,b)=>{return a-b;} )
```

如果箭头函数的函数体中只有一行代码, 并且是以 `return` 形式, 可以简化为

```
sort((a,b)=>a-b)
```

练习: 双色球: 随机取 1~33 之间的 6 个数字不能重复, 并且是从小到大排序, 取 1~16 之间的 1 个数字 (最后), 组成一个新数组。