* 新增的关键字与语法
  + const
    - 定义常量或者不会重复定义的变量
    - 定义了数据后不能够再定义新值，否则会throw TypeError/SyntaxError
    - 数组可以push或者pull数据，不会出现错误，但不可以再定义为其他类型
  + let
    - 只改变当前范围内变量的值，当变量超出范围后，返回原值，如下：

let x = 1;

if (x === 1) {

let x = 2;

console.log(x);

// expected output: 2

}

console.log(x);

// expected output: 1

* + template string(模块化字符串)
    - 使用字符串变量替代原有各字符相加的情况
      * console.log(“This is your ${word}”)
        + ${word} 就可以自动引用名为word变量
  + arrow function(箭头函数)
    - 原本的函数定义的更为简单，有点像c#中的匿名函数
      * var x = (a,b) => { return a+b; };
        + a,b为参数，=>后的函数实体
        + 如果后面的实体只有一行代码甚至可以把大括号都去掉
    - 使用了这种方法确实可以简化代码量，但是会无视大括号的使用范围
  + default(函数参数默认值)
    - 在没有定义参数数值的情况下，容易产生数值错误，此时可以为没有定义数值的参数进行默认赋值
      * function add(a=10,b=20){return a + b;} //为a,b赋默认值10,20
  + for…of….(新的foreach遍历方法)
    - 原本数组遍历需要使用arr.forEach(function(item,index,arr){ …… }) 才可以遍历数组内的内容，比较繁琐
    - for…of…语法
      * for(let item of items){ ….. }
  + Object destructure(对象解构)
    - 变量对象化复制数值
      * var {firstName,lastName} = instructor;
        + 如果instructor中含有firstName,lastName则var 变量也会相应的赋值，并可供其他同一级/下级范围使用
      * var {firstName:first,lastName:last} = instructor;
        + 功能同上,但除了赋值外还更改了变量的名称为first，last
  + Array destructure(数组解构)
    - 和对象解构基本类似
  + Class(面向对象的类申明)
    - 原本使用function模拟生成类的功能，现在可以用class关键字替代为更好用的方法，可以不再使用prototype来实现内部方法的功能
      * class Person{

constructor(firstName,lastName){

this.firstName = firstName;

this.lastName = lastName;

}

show(){ //内置方法，原本需要通过prototype的方式才可以添加

console.log(“Hello ${this.firstName} ${this.lastName}”);

}

static isPerson(obj){ //内置静态方法需要添加static关键字

return obj.constructor === Person;

}

}

* + - 继承
      * 使用extends 关键字继承原有类型
    - 使用超类的方法
      * 使用super关键字
      * 使用多参数构建器 super(…arguments)
    - hash map
      * 建立map
        + var items = new Map;
      * 添加对象
        + items.set(“key”,”value”)
      * 删除对象
        + items.del(“key”)
      * 获得数值
        + items.get(“key”)
      * 获得迭代
        + items.keys()
    - set
      * set会自动将数组中重复的数据剔除
    - promise
      * 见Aysnc javascript
    - promise.all
      * promise队列，按次序完成，中间有reject直接跳转至catch部分
    - yield
      * 生成器，同c#,java,python
    - padstart
      * 自动补全字符长度(从开头)
    - padend
      * 自动补全字符长度(从尾部)
    - async await 异步
      * 同c#