

对象练习

- 什么是面向对象 什么是面向过程
- 分别介绍对象的三大特征
- 对象也是一个变量，但对象可以包含多个值（多个变量），每个值以 对呈现。
- 定义一个对象 包含下列属性 name age sex addr(居住地) job(职业)
包含下列方法
eat, sleep, playgame, watchtv, read
函数体内自定义内容 不可为空
- 给上面的题里的对象添加 height(身高) weight(体重) 属性并赋值
- 给上面题里的对象添加 下列方法
meituan(点外卖 参数需要传手机号和居住地址 想吃的东西 返回下单成功)
cook(做饭 需要传入时间 判断当前是哪一顿饭)
travel(需要传入目的地 返回我去哪里玩啦)
listenmusic(传入你喜欢的歌手 控制台打印最喜欢的曲子)
work (传入上班时间和下班时间 判断是否为996)
- 访问上面对象定义的以及添加后的所有属性（两种方式访问 先指明可以通过什么方式访问）
格式为 如name字段: 姓名：+name的属性值
通过调用不同的方法来制定行程（按照正常顺序调用上面对象的每一个方法）
- 探索对象创建的其他方式 每种都举例尝试（不得少于三个方法 五个属性）
- 探索遍历函数的所有属性
- 五大主流浏览器的名字为键 内核为值 创建一个对象
- 下面代码的执行结果是：

```
function person(firstname,lastname,age,eyecolor)
{
    this.firstname=firstname;
    this.lastname=lastname;
    this.age=age;
    this.eyecolor=eyecolor;
    return [this.firstname,this.lastname,this.age,this.eyecolor,this]
}

var myFather=new person("John","Doe",50,"blue");
var myMother=person("Sally","Rally",48,"green");
console.log(myFather)
console.log(myMother)
```

- 下面代码的执行结果是什么 为什么

```
const user = {
  name: "John"
};
user.name = "Pete";
alert(user.name);
```

- 如何访问下面对象的bird属性

对象为：

```
var user={
  fire:"fox",
  likes:{
    type:"animal",
    bird:"fly"
  }
}
```

给user对象添加一个属性名是name 值是user

给user对象添加一个方法 触发后可以删除user对象的likes里面的type

创建一个对象 属性值为用户输入的自定义字符（最少三个属性）

- 创建一个对象 包含如下内容

- code 值为200
- msg 值为"请求成功"
- data 值为数组[]
- data中包含如下内容
 - shopname 商品名
 - pic 商品图片
 - desc 商品简介
- price 商品价格
 - kucun 库存
 - xiaoliang 销量
 - caozuo 操作

注 写的时候把值赋上

- 把上面那个题目用构造函数的重新写一份，并生成4个内容不同的对象
- 创建一个对象 包含如下内容
 - store 商店名称
 - addr 商店地址
 - sale: 方法 输出xxx商店正在大甩卖
- 遍历上面的两个对象 并输出每个的键和值
- 从外面给商店的对象增添一个shoplist的属性 该属性是一个数组 包含英雄名称跟英雄价格 货币单位以金币计算（5个王者荣耀就ok）
- 把第一题中的data中的对象里面的caozuo 属性删除
- 分析

```
var F = function () {}
Object.prototype.a = function () {}
Function.prototype.b = function () {}

var f = new F()
// 请问f有方法a 方法b吗
```

- 写出p1,p2,person,obj,Function,Object等的原型链
function Person(){}

```
let p1 = new Person()  
let p2 = new Person()  
let obj = {}
```

- 通过原型链实现祖孙三层继承关系（每个对象包含三个属性，两个方法）
- 访问上个题中原型中的对象
- 用原型链继承封装dom操作
 - 创建元素
 - 修改属性
 - 修改内容
 - 添加事件