

Day 09

原型

- 构造函数中的prototype指向该构造函数的原型对象
- 原型对象中的constructor指向构造函数
- 实例对象中的_proto_指向构造函数的原型对象
- 原型指向可以改变

原型链

- 实例对象和原型对象之间的关系，通过_proto_联系
- 原型指向改变，添加原型方法 —— 先进行原型改变后再添加原型方法，才能访问
- 实例属性和原型属性重名时，实例属性覆盖原型属性，在对应构造函数和原型都找不到相应属性时，会顺着原型链往上查找
- 原型链逐级往上查找，最终Object.prototype._proto_指向null

继承

- JS没有类的概念，可以通过构造函数模拟类，通过原型实现继承
- 子类构造函数的原型重设为父类构造函数的实例对象

方式

- 原型继承
 - 改变原型的指向
 - 属性非私有
- 借用构造函数
 - 在子类构造函数使用call方法调用父类构造函数
 - 可以继承属性，但不能继承方法
- 组合继承
 - 借用构造函数 + 原型继承
 - 构造函数继承属性，原型继承方法
- 拷贝继承 —— 把对象中需要共享的属性或方法，直接遍历复制到另一个对象中

函数角色

- 函数声明
- 函数表达式

函数声明有提升机制，而函数表达式没有

this

- 普通函数中的this指向window
- 对象方法中的this指向该对象
- 定时器函数中的this指向window
- 构造函数中的this指向实例对象
- 原型方法中的this指向实例对象

函数的不同调用方式

- 普通函数
- 构造函数
- 对象方法