

- 基础
 - Json协议 html协议
 - 数据类型
 - 分类
 - 判断数据类型
 - typeof
 - Object.toString.call 对象分类
 - instanceof
 - constructor
 - 传统开发四要素
 - i. 定义变量
 - ii. 获取元素
 - iii. 绑定元素
 - iv. 绑定事件
 - Arguments
 - call
 - 借用另一个对象
 - 改变this指向
- 构造函数
 - 函数声明和函数表达式区别：函数声明可提升，函数表达式不能提升
 - 构造函数和普通函数的区别：通过new关键字进行调用
 - 使用构造函数创建属性的四种方式
 - i. 参数形式
 - ii. 默认值形式
 - iii. 动态添加形式
 - iv. 混合模式

- Instanceof -- 检测该实例由哪个构造函数创建
- get set 属性取值器
- 公有属性和私有属性
- Function > function > 对象
- constructor
- 设置读写权限
- 原型
 - 原型对象不管实例化多少次，都只会生成一次
 - 原型对象本质：原型对象的属性和方法可以被所有实例共享
 - 双对象法则
 - 两个独立的对象：构造函数对象、原型对象
 - 通过__proto__联系在一起
 - __proto__属性是Function 对象的一个属性
 - prototype属性：保存地址
 - 属性屏蔽
 - 构造属性和原型属性判断
 - hasOwnProperty() -- true：实例对象；
false：原型对象
 - 原型链
 - 属性访问搜索法则
 - object.__proto__ >
Function.prototype.__proto__ >
Object.prototype.__proto__ > null
- 内存
 - 一切数据都是在内存中管理的
 - 任何变量都会占用内存
 - 实例化本质：拷贝构造函数的属性，在内存开辟一段新的区域保存这些值，定于构造函数不分配内存，实例化分配内存
 - 指针

- 栈内存&堆内存
 - 实例化在内存中保存
 - 内存分配
 - 内存回收
 - 内存泄露
- 创建对象的几种方式
 - Object
 - 内置对象
 - 构造函数
 - 原型对象
 - 拷贝
 - extend
 - 工厂
 - 第三方框架
- 三大对象
 - 内置对象
 - String
 - Date
 - Array
 - Boolean
 - RegExp
 - Number
 - BOM对象
 - Window
 - window内置全局属性和方法
 - Location
 - Document
 - History

- Navigator
 - Screen
 - Error
 - DOM对象 -- 增删查改
 - 自定义对象
- 面向对象三个作用
 - a. 封装框架
 - b. 描述数据 -- json
 - 原生绑定方法
 - i. 面向对象三大特性和继承
 - 动态创建标签绑定方式
 - formateString
 - 模板技术

1. 面向对象编程

- a. 多态
- 面向对象三大特性
 - a. 封装
 - b. 继承
 - 继承的九重境界
 1. Object中的继承 – 最简单的继承
 2. 内置对象的继承
 3. 自定义对象的继承
 4. 多种继承实现方式
 5. 第三方框架实现继承
- 面向未来变化编程和继承

- 面向接口编程和继承
- 设计模式 忘记模式 无招胜有招