

JS高级

基础总结深入

- 数据类型
 - 1. 分类
 - 2. 判断
 - 相关问题
 - 1. undefined与null的区别?
 - 2. 什么时候将变量赋值为null?
 - 3. 严格区别变量类型与数据类型?
- 数据、变量与内存
 - 1. 什么是数据?
 - 2. 什么是内存?
 - 3. 什么是变量?
 - 4. 内存、数据、变量三者之间的关系
 - 相关问题
 - 关于赋值与内存的问题?
 - 关于引用变量赋值问题?
 - 关于数据传递问题?
 - JS引擎如何管理内存?
- 对象
 - 1. 什么是对象?
 - 2. 为什么要用对象?
 - 3. 对象的组成
 - 4. 如何访问对象内部数据?
 - 相关问题
 - 什么时候必须使用[属性名]的方式?
- 函数
 - 1. 什么是函数?
 - 2. 为什么要用函数?
 - 3. 如何定义函数?
 - 4. 如何调用(执行)函数?
 - 5. 回调函数
 - 1. 什么函数才是回调函数?
 - 2. 常见的回调函数?
 - 6. IIFE
 - 1. 理解
 - 2. 作用
 - 7. 函数中的this

原型与原型链

- 原型(prototype)
 - 1. 函数的prototype属性
 - 2. 给原型对象添加属性(一般都是方法)
- 显式原型与隐式原型
- 原型链
 - 1. 原型链
 - 2. 构造函数/原型/实例对象的关系(图解)
 - 3. 构造函数/原型/实例对象的关系2(图解)
 - 4. 原型继承
 - 5. 原型链问题
- 探索instanceof
 - 案例1
 - 案例2
- 面试题

函数高级

- 执行上下文与执行上下文
 - 变量提升与函数提升
 - 执行上下文
 - 1. 代码分类(位置)
 - 2. 全局执行上下文
 - 3. 函数执行上下文
 - 理解
 - 执行上下文栈
 - 流程分析
 - 面试题
- 作用域与作用域
 - 作用域与作用域
 - 作用域与执行上下文
 - 作用域
 - 面试题1
 - 面试题2
- 闭包
 - 引子实例
 - 理解闭包
 - 常见的闭包
 - 闭包的作用
 - 闭包的生命周期
 - 闭包的应用: 自定义JS模块
 - 闭包的缺点及解决
 - 面试题
 - 面试题二

面向对象高级

- 对象创建模式
 - Object构造函数模式
 - 对象字面量模式
 - 工厂模式
 - 自定义构造函数模式
 - 构造函数+原型的组合模式
- 继承模式
 - 原型链继承
 - 借用构造函数继承
 - 组合继承

线程机制与事件机制

- 进程(process)
 - 程序的一次执行, 它占有一片独有的内存空间
 - 可以通过windows任务管理器查看进程
 - 是进程的一个独立执行单元
- 线程(thread)
 - 是进程运行的一个完整流程
 - 是CPU的最小调度单元
- 图解
- 进程与线程
 - 相关知识
 - 相关问题
 - JS是单线程还是多线程?
 - 浏览器运行是单线程还是多线程?
- 浏览器内核
 - 支持浏览器运行的最核心的程序
 - 不同的浏览器可能不一样
 - 内核由很多模块组成
- 定时器引发的思考
- JS是单线程执行的
- 浏览器的事件循环(轮询)模型
 - 模型原理图
 - 执行流程
 - 应用练习
 - 不足
- H5 Web Workers(多线程)
 - 介绍
 - 使用
 - 图解