# 什么是原型和原型链

每个对象都有一个原型属性，在浏览器中显示为\_\_proto\_\_，它指向了构造函数的prototype属性，它是一个原型对象

这样同类对象就共用了同一个原型对象，实现了对象间的属性共享

在访问对象属性时，如果实例本身没有该属性，则会从原型上查找，如果原型上仍没有该属性，则从原型对象的原型上继续向上查找，直到为最终为null停止查找。这样的实例与原型之间的层级关系称为原型链

# 实现继承的几种方法

**原型链继承**

将父实例赋值给子类的prototype，这样子实例的原型指向了父实例，继承了父实例的属性

缺点：父实例的私有实例属性被继承为了原型属性

**构造函数继承**

在子类的构造函数中调用父类的构造函数，并把子类的this传给父类，可以继承父类实例属性

缺点：无法继承原型属性，所以这种方式使用较少

**组合继承**

结合了原型链继承和构造函数的继承。

缺点：调用了两次构造函数

**寄生组合继承**

在组合继承的基础上，不新建父类实例，而通过Object.create将原型指向父类的prototype

**多重继承**

通过寄生组合继承多个父类

# new 过程发生了什么

[代码示例](https://link.juejin.cn?target=https://codesandbox.io/s/yuan-xing-lian-i172po?file=/src/%E6%A8%A1%E6%8B%9Fnew.js" \o "https://codesandbox.io/s/yuan-xing-lian-i172po?file=/src/%E6%A8%A1%E6%8B%9Fnew.js" \t "/private/var/folders/p8/gzh3syc172q36kk89443cqrc0000gn/T/com.kingsoft.wpsoffice.mac/wps-dingmiao/x/_blank)

创建一个对象，该对象的原型指向构造函数的prototype属性

将该对象赋值给构造函数中的this，并执行构造函数

如果构造函数返回一个对象，则返回该对象，没有则返回创建的对象

# 什么是闭包

闭包就是能读取其他函数内部变量的函数

内部函数存在对外部函数作用域变量的引用就会形成闭包

闭包可以实现对函数内部属性和方法的私有化

经典场景：函数返回一个函数，在外部仍可以访问内部变量，防抖节流

# js事件循环机制

js是单线程，所有代码放到执行栈中执行

当遇到异步函数时，将异步回调放入一个任务队列中

执行完执行栈中的同步任务，会去任务队列里按顺序执行寄存的函数

如此往复，称为事件循环

异步任务又分为

宏任务（macro-task）:setTimeout、setInterval...

微任务（micro-task）:Promise...

同步任务->微任务->宏任务

# this指向

普通函数指向函数的调用者，箭头函数指向函数所在的作用域，注意理解作用域,只有函数的{}构成作用域,对象的{}以及 if(){}都不构成作用域;

# call、bind、apply的作用区别

三者都能改变函数this指向

apply、call调用函数，传参形式不同，apply接受一个数组参数，call接受多个参数，以队列形式传入

bind不调用函数，它返回新函数，传入的参数将作为新函数的值传入

# Promise和 手写promise

Promise是处理异步任务的一种方式

一个Promise必然处于pending(待定)、fulfilled（已成功）、rejected（已失败）三种状态

通过resolve和reject切换promise状态，通过then方法通知promise执行结果

通过定时器和任务队列的方式实现异步任务

# async/await

async/await可以让异步操作像同步一样执行，可以让多个异步任务以队列的方式依次执行

await后面接Promise对象

await只能在async函数中使用，不然会报错

async函数返回的是一个Promise对象

async/await通过generator函数实现

# ES6语法

es6泛指es5.1以后的js下一代标准语法，自2015年每年推出一个版本，包括ES2015-ES2020等

解构赋值、箭头函数、模板字符串、展开运算符（扩展对象，只能做到当对象属性是 基本数据类型 才是 深拷贝，如果是 引用数据类型，那就是浅拷贝）。。。

# setTimeout执行机制

# webpack原理、构建的过程

[「吐血整理」再来一打Webpack面试题](https://juejin.cn/post/6844904094281236487" \o "https://juejin.cn/post/6844904094281236487" \t "/private/var/folders/p8/gzh3syc172q36kk89443cqrc0000gn/T/com.kingsoft.wpsoffice.mac/wps-dingmiao/x/_blank)

webpack是一个整合打包的工具

功能：模块打包，将不同模块的代码整合一起，并保证引用正确。解析特殊语法，es6、vue、less、jsx等，转译成浏览器能识别的语言。功能扩展，通过插件等方式拓展功能，例如按需加载、代码压缩等，提高自动化和工程化程度

构建过程： 读取配置、确认入口文件、编译模块、完成输出

# webpack plugin和loader区别

# 是否有 手写webpack插件

# sourcemap是什么

# 热更新HMR原理

# 精度问题产生的原因和解决办法

# 两个超大数字相加，不能使用bigint

# 性能优化概述

# 浏览器渲染的过程

# JS垃圾回收

# Cookie、session、sessionStorage、localStorage的区别

# 伪元素的作用

# http1.0 http2.0

# http和https

# http缓存

# react diff算法

# react useEffect、useMemo、useCallback的作用和区别

# react setState是异步还是同步

# 为什么setState是异步的

# 什么是hooks闭包陷阱

# react性能优化

# react 类组件和函数组件的区别 （多次问到）

# react 什么是高阶组件

# react 有没有像vue-router中的路由守卫

# react fiber

# ts type和interface区别

# react 组件是什么ts类型

# vue 响应式原理

# vue与react的区别

# 设计模式概述

# 作为前端架构及管理者的职责