Js基础

变量提升

变量定义： 定义会提升，赋值不会

函数声明会提示

类型转换

显式转换

隐式转换

相等比较

两个非原生值之间的相等，就是检查引用地址是否相等

原生值与非原生值比较，会将非原生值转换为原生值

Css基础

适配方案

<https://juejin.im/post/5b0a9f266fb9a07aa114a908>

rem + simple-flexible

微任务与宏任务

堆： 存放对象的空间

栈： 存放基础变量: 数字，字符串， Boolean， 引用的变量地址

闭包

bind

函数节流与防抖

节流： throttle

调用动作的时刻大于等于执行周期则执行该动作，然后进入下一下新周期。

防抖： debounce

调用动作过了n毫秒后才执行，若在n毫秒内又调用此动作，则重新计算执行时间。实际上是针对输入框响应请求，多次请求只有最后一个通过

Promise async & await

原型和class

1. 所有引用类型有自己的隐式原型 (\_\_proto\_\_)
2. 函数都有自己的显示原型（prototype） 这个对象中有{constructor: f}，可以重置这个显示原型。
3. 所有引用类型的隐式原型都指向对应的构造函数的显示原型。
4. 使用引用类型的某个属性时，如果没有这个属性，会在\_\_proto\_\_中去找。

Classs类声明：

不可以提升，不能重复声明, sayHi()不可以被枚举（与es5不同）

Class Person {

Constructor() {

}

sayHi() {

}

}

全面的深度复制（fn, Date, reg）

高阶函数

https://juejin.im/post/5ad6b34a6fb9a028cc61bfb3

Vue相关

Vue-router

<https://juejin.im/post/5b10b46df265da6e2a08a724>

构建一个store

webpack相关

输入网址发生了什么

渲染

设计模式

算法与数据结构

http

DNS优化

DNS预获取 DNS prefetch

DNS请求次数

浏览器默认会对当前页面中与当前域名不在同一个域的域名进行预获取，并且缓存结果（隐式DNS Prefetch）

<meta http-equiv="x-dns-prefetch-control" content="on">

<link rel="dns-prefetch" href="//www.itechzero.com">

<link rel="dns-prefetch" href="//api.share.baidu.com">

<link rel="dns-prefetch" href="//bdimg.share.baidu.com">