

语言基础篇

JS 语言本身:

1. 数据类型 总

typeof 各类型的输出是什么 为什么?

让你判断一个变量是不是空对象 有哪些判断方式?

2. 数据类型 对象

1. 深拷贝 阅读一下lodash的deepclone实现

2. this关键字

3. new 操作符原理

4. 原型链

1. 我们自己定义的话 如何?

2. 语言已有的原型链是怎样的?

3. 特殊对象 数组

1. JS中, 数组的本质是什么? 牵扯到两个概念 array-like 和 iterable

2. iterable是如何通过iterator便利的。你能否自己写一个iterable, 让其被js原生语法遍历? 如for of

4. 特殊对象 函数

1. 闭包 思考一下这东西能解决什么问题 什么应用场景该用闭包

2. 箭头函数特性 除书写简洁 还能解决什么问题?

3. decorator 装饰器是干嘛的? 自己写一个装饰器函数

5. es6 class

1. 类如何用function实现? 考虑到类继承、类警惕啊函数等。

2. super关键字

3. instanceof 原理

6. Promise

1. new Promise(...).then(...) 两个... 里 分别是同步的还是异步的?

2. Promise API自己实现一下。比如race、all、allsettled

3. promise的异常处理和普通的try catch有何不同? 思考下 如果没有处理promise的异常一直向上抛 会怎么样?

7. 进阶: Proxy、[Object properties configuration](#)

1. 其应用场景一般是写一些框架 对语言本身进行一些魔改。由于你用的是vue, 那需要挖掘下vue里面哪些特性/功能是通过proxy来实现的。

2. getter setter实现数据、dom的双向绑定 写一下。

8. 进阶: generator

浏览器:

1. 浏览器架构

1. 想一下之前分享的文档, 浏览器作为一个多进程的应用程序, 其由哪些部分组成, 各部分如何相互配合, 实现了哪些功能。

2. 浏览器提供的实用工具。即开发者工具里的各个tab页是干啥的, 咋用

2. 浏览器API

1. dom操作

2. 元素位置大小获取

3. 事件捕获冒泡机理 事件捕获的应用场景 事件委托是什么 坏处是？
4. 文档加载流程 及 对应的全局事件有哪些 我们为什么要关注各事件
5. `setTimeout` `setInterval` `requestAnimationFrame` `requestIdleTime` 各子功能 适用场景 优劣对比
6. `window.postMessage` 这个 API 了解下其应用场景。
7. `fetch` API 尤其是其中和跨域相关的参数实际有什么作用。这个要了解 http 协议中的一些基本约定。浏览器和服务端按照这个约定，在请求和响应里携带哪些信息？而 js 能控制的有哪些？体现在 `fetch` api 里又是什么？
8. `cookie` `storage` 基本使用
9. `script` 标签 `type=module` 后 `import` 和 `export`； es6 `dynamic import`
3. 进阶 `service worker`
4. 进阶 `intersectionobserver`
5. 进阶 `mutationobserver` （各框架或多或少都用到 功能很强大 了解下）
6. 进阶 `web component` （不那么重要）

NodeJs:

1. `commonjs` 风格 `require`。这里面门道很多，包括 `require` 的算法是什么，`require.cache` 是怎么做的。循环引用是怎么不出问题的。
2. 剩下的很多 如 `stream`、多进程多线程程序如何写、`nodejs` 做了哪些事情能让我们直接使用一些东西？比如 `module` 全局对象如何塞入的 其属性有哪些等。都非常有必要学习 按照这个教程从头捋：<https://github.com/samerbuna/efficient-node>
3. `nodejs` 如何调试？思考下调试的本质是什么？