

eventLoop

示例

```
console.log(1) //同步
setTimeout(() => { //宏任务
  console.log(2)
}, 0)
new Promise(resolve => {
  console.log(3) //同步
  resolve()
}).then(() => { //微任务
  console.log(4)
})
setTimeout(() => { //宏任务
  console.log(5)
  new Promise(resolve => {
    console.log(6)
    setTimeout(() => { //宏任务
      console.log(7)
    })
    resolve()
  }).then(() => { //微任务
    console.log(8)
  })
}, 500)
new Promise(resolve => {
  console.log(9) //同步代码
  resolve()
}).then(() => { //微任务
  console.log(10)
  setTimeout(() => { //宏任务
    console.log(11)
  }, 0)
})
console.log(12) //同步
```

- 先找同步代码 1 3 9 12
- 再找对应的script中的微任务 4 10
- 再找宏任务 2 11 5 6 8 7

执行顺序为 1 3 9 12 4 10 2 11 5 6 8 7

回顾

前端知识体系

- 优化

资源优化

减少资源（剔除非必要资源（cdn 图床 精灵图））

渲染优化

减少渲染（减少重绘回流（减少dom操作）减少渲染量（懒加载））

加快渲染（进行内存加载 减少请求）

...

• 渲染

浏览器渲染原理

减少非必要渲染（虚拟dom技术 diff算法（进行比对渲染的））

...

• 应用

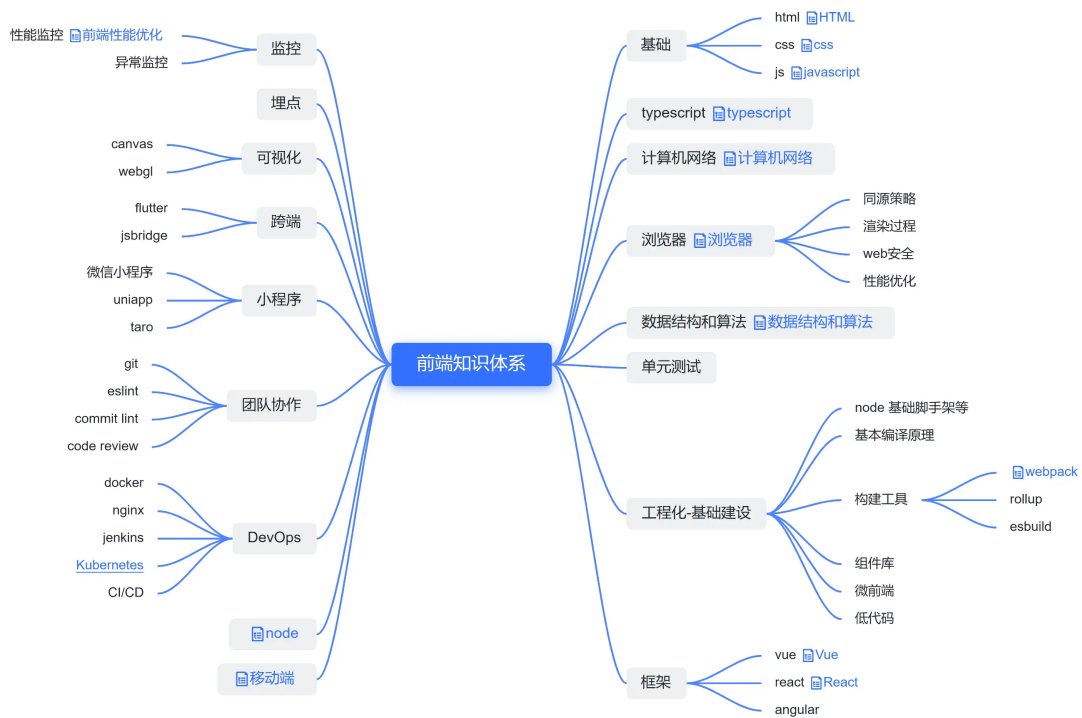
第三方插件使用（lodash.js、moment.js、tree.js...）

• 网络（锚点 抓包...）

• 前端工程化

webpack、gulp、vite....

• ...



@峰主圈技术社区

内容回顾

第一周内容

- js基础内容（变量声明 基础语法 运算符及表达式 基础数据类型（number string boolean undefined null））
- 条件控制语句（if else switch case）
- 循环控制语句（for while do while）
- 函数（使用function来定义 函数的预编译 函数的封装思想 函数的arguments）、
- 数组（数组的定义 数组的属性及方法 length属性 方法 push pop shift unshift concat join sort reverse slice splice..）

第二周内容

- 字符串 (字符串的属性及方法 length属性 方法 indexOf lastIndexOf search match replace split substring substr slice charAt charCodeAt toUppercase toLowercase trim ... 静态方法 String.fromCharCode)
- 日期时间对象 (Date 对应的方法 get方法和set方法 getFullYear getMonth getDay getDate.....)
- BOM (Browser object model 浏览器对象模型 主要对象 (window, location (href、port、hash、search...))、history (go back forward pushState replaceState...), screen、navigator、frames、document))
- DOM (document object model 文档对象模型 (三类节点 element、attribute、text) (nodeType nodeName、nodeValue))
节点相关操作方法 (createElement createAttribute createTextNode append appendChild insertBefore removeChild clone replaceChild...) (获取元素 getElementById getElementsByClassName document.querySelector...) (属性 getAttribute setAttribute removeAttribute)) (部分属性 innerHTML innerText title id className tagName..) 获取样式 (style属性获取内嵌样式) getComputedStyle e.currentStyle 兼容性)

第三周内容

- 事件 (常用事件 鼠标 (click,dblclick、mousedown、mouseup、mousemove、mouseenter、mouseleave、mouseover、mouseout、contextmenu) 键盘事件 (keydown、keyup、keypress) html事件 (load unload close beforeunload focus blur select change scroll ...) event (事件源对象 offsetX offsetY pageX pageY clientX clientY screenX screenY ctrlKey shiftKey altKey target currentTarget button type keyCode charCode....) 事件监听器 (addEventListener removeEventListener) 事件代理 (将事件加给对应的父元素 利用e.target来区别目标元素) 事件流 (捕获-目标-冒泡) 取消事件冒泡 (cancelBubble=true 兼容性 stopPropagation) 阻止默认行为 (preventDefault 兼容ie的returnValue=false))
- cookie (cookie的产生为了解决http无状态 cookie是随请求发送 (组成由 name=value;expires=时间;domain=跨域;path=地址;secure 是否使用https) localStorage sessionStorage (本地存储) json数据格式 (用于传输的格式 [] 数组 {}对象) JSON.stringify JSON.parse 序列化方法)
- 正则表达式 (声明 RegExp (g 全局匹配 i 不区分大小写 m 换行) (元字符 [] {} () + * ? \s \d \w | ^ \$)) 相关方法 (test 是否匹配 exec 获取匹配的数组 支持正则的字符串方法 (search match replace split))
- es5 es6新增 (ES5新增 严格模式 (use strict) 新增数组高阶函数 (forEach map reduce reduceRight filter some every) for in、JSON.parse JSON.stringify、this绑定的方法 (bind call apply)、对象新增 (getter 和 setter 以及对应的方法)、数组新增 Array.isArray ..) (ES6新增 字符串模板 (` `) 新增值类型 (symbol、BigInt) 新增变量修饰关键词 (let const) 字符串方法 (includes startsWith endsWith repeat) 数组方法 (findIndex find 静态方法 Array.from Array.of) 函数新增 (默认参数 箭头函数 (没有this 没有arguments 没有原型不能被new)) 对象简写、解构赋值、扩展运算符 (...) 模块化 (export import (AMD 预加载 CMD 按需加载)) class (类 constructor 构造器) promise、generator函数 (function* (yield next))、set (weakSet (add delete clear forEach keys values entries has)) map (weakMap (set get delete clear forEach keys has ...)))

第四周内容

- 运动 (动画 (强烈推荐C3动画) 使用对应的setInterval来控制对应的元素样式变化)
- 面向对象 (找有对应方法的对象做对应的事情 特性 (封装 (抽取对应的属性和方法)、继承 (extends关键词)、多态 (重写)) (new 关键词调用 class类 构造函数) 工厂函数 拖拽实现 (mousedown mousemove mouseup))

- 原型和继承（原型（构造函数的prototype 对象的 `__proto__`（原型只声明一次 是一个公共空间 一般用于存储方法）原型链（在 `__proto__` 上查找对应的属性的过程））继承（实现方式（class的extends、原型链继承、对象冒充、组合继承、寄生组合继承）））
- 闭包和promise（闭包（函数嵌套函数 内部函数拥有外部函数的参数或者变量 不会被gc回收、优缺点、应用（缓存、防抖、节流、函数柯里化））promise（es6新增的一个类它有三种状态（pending、fulfilled、rejected）原型方法（then、catch、finally）静态方法（all、reject、resolve、allsettle、race）））
- eventLoop（事件轮询）宏任务（script主体、setTimeout、setInterval...）微任务队列（promise.then、promise.catch promise.finally、nextTick（如果有dom操作 它就是宏任务（等待dom操作完成再执行））...）

事件轮询流程（先执行同步内容 ---- 再进行事件轮询（先宏后微（script 微任务先走...）））