1. This
   1. 定义
   2. call apply 作用、区别
   3. bind 实现
   4. arguments相关
      1. 如何判断arguments是重定义的还是传入的
   5. 类Array对象转换成Array对象
   6. array.of ES6
   7. array构造函数参数个数不同时的区别
   8. var arr = new Array() 与var arr = []区别
2. 闭包
   1. 含义: 上下文,可见性,生命周期
   2. 常见用法
   3. 内存泄露问题
      1. 手动设置为null
      2. 垃圾回收机制
      3. chrome的profile工具
         1. 如何才能更容易找到:使用命名函数
      4. v8堆内存上限,可以通过命令行提升
3. 作用域
   1. for while 函数
      1. 变量提升
      2. let ES6
   2. 跨级作用域
4. 原型链
   1. prototype
   2. constructor
   3. prototype和this都挂着,访问哪个
   4. object.prototype.a function.prototypr.b 函数可以调用a和b
5. css
   1. 盒模型
      1. 不同浏览器盒模型不同
      2. box-sizing
      3. 为什么有的人会在reset里设置border-box
   2. 定位
   3. 自适应
   4. 居中
      1. flex
6. table-cell
7. absolute; top,left,right,bottom:0;margin:auto
8. top 50%;left 50% transform:translate (-50%,-50%)
   1. 外边距合并
9. 兼容性
   1. html 根据不同IE版本,引不同的文件
   2. css hack
      1. “+” “\_” “\*” “!import”
      2. -webkit- -o- -ms- -moz-
   3. js
      1. polyfill/shim
         1. 实现object.create()
      2. try catch
10. 函数式编程
    1. 函数可作为参数传递
11. html5/css3
    1. html5
       1. webSocket
       2. canves
          1. 2D上下文/webGL
       3. 本地存储
          1. localStorage sessionStorage webdb 区别 (容量 存储时间等)
    2. css3
       1. brother child 选择器
       2. animation
       3. transition
       4. transform
12. Ajax
    1. W3C和IE区别
    2. 怎么用
    3. 判断超时
    4. 判断成功
    5. form表单
    6. 状态码
13. 跨域
    1. 同源策略 原因
    2. CORS
       1. 简单和多步骤
       2. 服务器端的头
    3. JSONP
       1. 优缺点
    4. Iframe
    5. Flash跨域 安全问题
    6. 回传数据不能Array开头 原因
14. HTTP协议
    1. http请求组成
    2. 常见方法 区别
    3. 请求构成
    4. post提交数据的方式
    5. url多长
    6. domain 并行请求
    7. 头部信息
    8. 缓存 cache-control
       1. 有哪些值
    9. https握手方式 为什么慢
    10. http2比http提升了什么
    11. 怎么升到的webSocket
15. 异步处理的演变 异步处理的库
16. angularJS解决了什么样的问题
17. 为什么有模板
18. 为什么前后端分离
19. 模板化 组件化 是什么 原因
20. 安全性: AST分析
21. 优化
    1. css
       1. 选择器
    2. js
       1. 加载时间优化
          1. 打包工具有哪些
       2. 执行时间优化
          1. reflow repaint