**Vue的生命周期**

beforeCreate：组件实例初始化时，这时组件的属性还未被计算（如data和methods）

created:组件实例创建完成，属性已计算完成，但还未挂载到dom中，不能访问$el

beforeMount：组件挂载前，此时是首次挂载前最后一次修改data的机会

mounted:组件完成首次挂载，此时可以ajax获取数据

beforeUpdate:组件即将更新

updated:组件更新完毕

beforeDestory:组件销毁前

destoryed:组件销毁后

**Vue组件的通信**

1: 父传子：在父组件中设置属性 v-bind:a=‘data’**;**子组件通过this.props获取属性

2：子传父：父组件自定义事件v-on:recive=’dosthing’，子组件触发this.$emit("recive ",a,..)

3：父子双向绑定 ，结合父传子和子传父

**Vue的组件包含的属性**

**1：**data(数据)--对象/方法 在模块化中它是一个方法函数 返回一个数据对象

2：模板(component) --对象 其中每一个属性的value也是一个对象，它有自己的template data methods等属性

3：挂载元素(el) --字符串属性

4：方法(methods) --对象 和计算属性一样 其中的每一个方法都是函数 在模板和视图调用方法时执行函数

5：生命周期(mounted...)

6：计算属性(computed)--对象 每一个计算属性是一个方法 返回根据data中数据计算和演绎之后的数据 它自动根据data数据的改变而改变

7：监听( watch)--对象 键为监听的数据名 值为函数 当数据发生变更时函数将会执行

8：指令(directives)--对象 每一个指令又是一个对象 对象里是钩子函数 key是钩子 value是函数

**Vue自定义组件：**

除了核心功能默认内置的指令 (v-model 和 v-show)，Vue 也允许注册自定义指令。注意，在 Vue2.0 中，代码复用和抽象的主要形式是组件。然而，有的情况下，你仍然需要对普通 DOM 元素进行底层操作，这时候就会用到自定义指令。

钩子： bind:第一次绑定时调用。只会调用一次。

inserted:在绑定元素插入父节点时调用，

updata；在包含指令的组件更新后调用

directive:{

"A":{

bind:function(el,binding){

el--指令绑定的元素 可用于直接操作dom

binding--一个对象包括value(向指令传入的值)

}

}

}

**Vue响应式的原理：**

Vue的响应式使开发中在修改组件data里的数据时自动更新视图和触发生命周期。当我们定于一个Vue实例的data属性的时候，Vue将遍历此对象所有的属性，并使用Object.defineProperty()把这些属性转化为含有set和get。这使得属性被修改和访问时实例可以追踪到，并通知变化。每个组件实例都有相应的 **watcher** （观察者）实例对象，它会在组件渲染的过程中把属性记录为依赖，之后当依赖项的 setter 被调用时，会通知 watcher 重新计算，从而致使它关联的组件得以更新。

Vue和React的对比

区别：

1： React拥有更强大更开放的社区 , facebook团队维护在国外开发社区中拥有更多的开发中和维护者

2： JSX语法让dom&css in js ，Vue采用templact包装dom ,dom js css分割

3: React的组件更新原则—组件更新默认会重新渲染整个组件子树，如需优化可采用PureComponent和shouldComponentUpdate优化

Vue 组件更新时会自动追踪依赖变化数据的子组件 只重新渲染需要更新的子组件，这依赖于Vue的响应式的原理。

**算法题：**

1. length为n-1的数组，包换值为1~n的不重复的元素，求不在数组中的元素是多少。

解法1：累加数组元素，和1~n的和比较，差为所求元素

解法2：异或位运算符，数组所有元素的^结果 与1~n元素的^结果 相异或

二：树的遍历：

BFS广度优先：从顶点出发访，横向访问其所有相邻节点，在依次横向访问其相邻节点的相邻节点，直到所有节点被访问。

DFS深度优先：从任选顶点v起 若当前顶点问的相邻顶点，访问与其中一个相邻的顶点，依次类推向下访问，若其相邻订点都访问过或没有则返回上一次访问的顶点，直到所有能与v顶点想通的顶点都被访问到。若图中还存在为访问的顶点则已该顶点起重复上述步骤。

用代码来变现的: 广度优先：每次获取顶点下的所有的可访问节点，顺序插入数组，循环访问（while）,先进先出，并循环上述操作，直到数组为空，

深度优先：每次获取顶点下的所有的可访问节点，倒序插入数组，后进先出，每次取最后一次插入的新元素

Java视频教程目录：\* M8 d( u, k7 M; Q4 v, ~+ A, d  
        │  zhufeng.zip2 ?' }\* |  U( y: V  
        │  zhufengpeixun-201812-master.zip6 o0 B% k/ w4 M7 K+ F/ J' x  
        │  $ \$ `; r; \_$ l9 |$ w% f6 l- w  
        ├─01公开课内容  
        │      01-webpack简易版实现.mp4  
        │      02-多种跨域方案详解.mp4  
        │      03-react-router.mp4+ [7 O! H+ ?2 z& q; P, g  
        │      04-oauth.mp4  
        │      05-VUE及REACT中的DOM-DIFF算法.mp4  
        │      06-VUE框架中MVVM实现原理深度解读.mp4$ d$ r: |/ s/ X' {7 w7 x  
        │      07-WEBPACK-全面进阶4.0版本.mp4  
        │        w% ]2 Y8 N) y  
        ├─02MVVM原理实现# s! s: S" ?9 C\* C% u9 P3 z  
        │      01-Object.defineProperty.mp4% w  w- R7 G7 G4 N; z. Q: }" Q  
        │      02-数据劫持Observe.flv  
        │      03-数据代理.flv  
        │      04-编译模板Compile.flv  
        │      05-发布订阅模式.flv8 m/ R2 V' Z) y/ x7 {9 ^; ~, z  
        │      06-连接视图与数据.flv  
        │      07-双向数据绑定的实现.flv$ l7 C6 I7 ?9 o: V; @. ?  
        │      08-实现computed.flv  
        │      9 ~5 z( s( h& Y- l, ~  B  
        ├─03git基础用法( k& u\* e4 O( X\* O/ {  
        │      01-什么是git.mp4  
        │      02-linux命令.flv" ?3 }( W. B0 U" A- ]\* x  
        │      03-提交到历史区.flv  
        │      04-撤销文件.flv  
        │      05-分支.flv& Q+ e% N& P7 s  
        │      06-分支合并.flv  c; R) U, V- l3 r, C) G9 ^- U) i  
        │      07-远程仓库.flv  
        │      08-拉取线上最新代码.flv  
        │      09-代码fork-pull-request.flv- o, a3 p( ]( a3 M7 [9 B  
        │      10-界面操作.flv  
        │        
        ├─04node基础  
        │  ├─01第一天  
        │  │      01-什么是node.mp48 X& s; Y2 D( a+ E- ]. s: i: R  
        │  │      02-区分环境变量.flv  \_1 c, W: Q( B; y9 \_2 j  
        │  │      03-什么是模块.flv: s1 \! o) h$ G7 y1 y  
        │  │      04-模块的使用.flv1 j2 v$ A2 o' K5 N9 Z  
        │  │      05-安装.flv  
        │  │      06-发布包.flv: L& e2 s\* f! u9 \_  
        │  │      07-核心模块.flv/ p- K. |/ D7 @4 M- X1 d) u9 y  
        │  │      08-promisify.flv  
        │  │      # T$ c) J% c\* \. H' g) R/ H" M( T/ g  
        │  ├─02第二天$ [1 y  ^, x" G% \& T\* @) T! m+ O  D6 B  
        │  │      02-concat方法.flv  
        │  │      03-进制转化.flv$ s2 U/ J7 B  Y% {  
        │  │      04-promise链.flv  
        │  │      05-promise.flv9 A# a  Z0 B  r2 O& \_! K$ q( P  
        │  │      06-递归创建目录.flv/ p0 u$ K/ {; d$ t0 h. {) i0 q  
        │  │      07-流.flv  
        │  │        
        │  ├─03第三天  
        │  │      01-pipe实现.flv6 B3 j: a. b: l: P+ Q& ?' y, x  
        │  │      02-http服务.flv  
        │  │      03-实现静态服务.flv  Z; B2 R; c/ h\* K+ ]9 D  
        │  │      04-静态html返回.flv  
        │  │      05-fetch的用法.flv# F# X4 p7 H! j0 W4 l$ A: e  
        │  │      06-查询所有.flv1 [& `+ c7 |; g5 R/ ^; w  
        │  │      07-添加用户.flv  
        │  │      08-删除用户.flv  
        │  │      09-跨域.flv1 v2 s" o; r. y9 D1 b\* z3 g\* }( v  
        │  │      & {% b, K- C! E- d) F# H. k4 U  
        │  └─04第四天  
        │          01-express的使用.flv  
        │          02-请求参数.flv$ R8 N) [\* ~7 O$ b3 ]$ p  
        │          03-路径参数实现.flv  `( O7 E\* y% \# ^+ J  
        │          04-中间件.flv  
        │          05-中间件装饰模式.flv  
        │          06-res上新增的方法.flv  
        │          07-bodyParse中间件.flv  
        │          08-ejs模板的使用.flv" y; V\* l# E\* C0 ^- S! L+ M+ W! {  
        │          09-express静态文件中间件.flv" y\* h) l: d# N$ T  `) T9 ]$ g  
        │          10-更改express的写法.flv  
        │          3 i. s\* t\* [6 ]: `8 k/ M  
        ├─05正式课实录  
        │  ├─01ES  
        │  │      01-es61.flv8 X1 |. m/ L" \_3 ^' l  
        │  │      02-es62.flv  
        │  │      03-es63.flv# f0 j, v4 m& P2 G  d5 a: S# o3 P  
        │  │      04-es64.flv  
        │  │      05-es65.flv3 [! X8 {4 K1 |6 J, n  
        │  │        
        │  ├─02promise  
        │  │      01-generator.flv\* f8 [; D; l0 d  
        │  │      02-promise1.flv  
        │  │      03-promse2.flv  
        │  │      04-promise3.flv  
        │  │      05-promise4.flv6 h/ g/ O6 A7 N  
        │  │      $ B+ B. R% G; [. w\* m# y  
        │  ├─03node基础7 p; s. M' H, w, X3 e7 n  u  
        │  │      01-单线程1.flv+ X+ v! `3 M5 U6 o7 Q! F\* O4 x  
        │  │      02-单线程2.flv  
        │  │      03-repl.flv3 r6 ]2 @) \2 {( |  
        │  │      04-global.flv  
        │  │      05-event.flv  
        │  │      06-module.flv  K0 h% t. }3 B% h; z. [  
        │  │      07-npm.flv3 [% p! f% ^\* \_# [+ r3 o  
        │  │      08-buffer.flv  
        │  │      09-fs.flv  
        │  │      10-fs.flv& `0 k, |) X5 I( G  
        │  │      11-fs.flv. p' ~$ A5 t1 ]7 b$ ^' N6 N  
        │  │      12-stream.flv  
        │  │      13-stream.flv" P/ \_  Q+ G' {6 I0 A) A( s  
        │  │      14-stream.flv  
        │  │      15-stream.flv  
        │  │      16-stream.flv  
        │  │      17-tcp.flv! p\* A! v  Z7 U" U4 P  
        │  │      18-tcp.flv  
        │  │      19-tcp.flv, s! ~& ~" `$ N/ T: g- }3 u  
        │  │      20-tcp.flv2 \_! X% Z  }9 p0 v  
        │  │      21-tcp.flv  
        │  │      22-tcp.flv  H6 W4 z5 z" @7 K& [5 C: d  
        │  │      23-http.flv  
        │  │      24-http.flv  
        │  │      25-http.flv; ~; K5 y2 ]' ?/ D; r  
        │  │      26-http.flv6 {! y2 B9 g2 T% B. x$ o- C0 k  
        │  │      27-http.flv! r  i8 `\* \( {  
        │  │      28-http.flv  
        │  │      29-http.flv  
        │  │      30-http.flv  
        │  │      31-http.flv  
        │  │      32-express.flv' M. {) y6 K\* {4 i  
        │  │      33-express.flv, e$ t2 M. t2 R" n  
        │  │      34-express.flv  
        │  │      35-express.flv' x; a+ z: w" `9 E\* q9 z4 W; M5 \! M  
        │  │      36-express.flv: Q2 U# D( X% v% o  
        │  │      37-express.flv  
        │  │      38-cookie.flv8 r9 o# M9 \6 Y! E; a  
        │  │      39-session.flv  
        │  │      40-koa.flv" G2 f2 E' a; ?) @  
        │  │      41-koa.flv  
        │  │      42-koa.flv  
        │  │      43-koa.flv  
        │  │      44-koa.flv+ p$ ~! R; d7 w  
        │  │      45-移动端布局.flv% \6 b8 N: e) k$ @6 C  f# C  
        │  │     , D, M" }0 }6 H( N) y  
        │  ├─04webpack$ |( |0 n. a. S# N+ {; i; G  
        │  │      01-webpack.flv: z# R2 o7 a- f' Q/ U- F& J& F: K  
        │  │      02-webpack.flv  
        │  │      03-webpack.flv  
        │  │      04-webpack.flv  
        │  │      05-ast.flv' ~) L, `6 @- j, H0 Z  
        │  │      06-webpack源码.flv  
        │  │      07-loader.flv  
        │  │      08-plugin.flv) e\* z( d\* n9 E: n0 F. I  
        │  │      09-webpack.flv6 D4 A# u. d2 ?  
        │  │      , w4 \_0 C0 w4 \_\* L" p& t  
        │  ├─05react) i9 g# \' f  u( a) R; `7 j  
        │  │      01-react-props.flv+ v. q8 M# Q3 Y: b, H1 Q( y  
        │  │      02-react-keys.flv  
        │  │      03-react-state.flv0 a% r9 K" p+ N4 @) K6 A( A. G0 |# f  
        │  │      04-react-lift.flv0 a: q1 ]& a$ \' g0 ?  
        │  │      05-生命周期.flv$ c. `5 @# o8 l2 n8 D# x, w  
        │  │      06-属性校验.flv  
        │  │      07-dom-diff.flv  
        │  │      08-fragment.flv  
        │  │      09-路由.flv, L9 \_1 w: r\* e  
        │  │      10-手写路由.flv  
        │  │      11-domdiff.flv  
        │  │      12-domdiff.flv  
        │  │        
        │  ├─06antdesign9 V  w& }' ?  c- F% ]5 z) t& \_  
        │  │      01-项目配置.flv  
        │  │      02-登录表单.flv5 ?5 }0 v( h& S! ?& {9 ~5 Y/ p  
        │  │      03-封装请求方法.flv  
        │  │      04-分页.flv  
        │  │      05-菜单.flv+ k9 K' g9 D; N9 W4 s1 Z: m  
        │  │  ! A7 l1 H0 P\* C; n; |) l6 q& l  O  
        │  ├─07redux# C2 t3 l1 {9 `& L  
        │  │      02-react-redux.flv5 x5 H6 L  p; w" a! J  
        │  │      03-redux中间件.flv( v9 U8 W5 i/ {: l/ R  
        │  │      04-redux-saga.flv4 h+ w3 u4 Z\* e0 N' p0 c& B/ m0 m  
        │  │       
        │  ├─08监控3 b' p, z' O  ^9 k+ k3 C$ ~, d  
        │  │      01-监控.flv  
        │  │      02-监控.flv  
        │  │      03-监控.flv  
        │  │      04-监控.flv  
        │  │      05-监控5.flv# k- ^9 ]7 R7 A1 }9 \0 \. I- `5 h  
        │  │  $ `; n' U9 K6 ]\* u5 F( D) h# g+ e  
        │  ├─09mongo  
        │  │      01-mongo1.flv  
        │  │      02-mongo2——看了40分钟.flv. `0 D) Q. X3 i" [7 u( c$ u' D  
        │  │      03-mongo3.flv/ b" b9 P! {4 k# h' `0 H  V% ~  
        │  │      04-mongo4.flv1 i- o: X, t, O3 ~) {8 P  
        │  │      05-mongo5.flv  
        │  │      06-mongo6.flv  
        │  │      07-mongo7.flv+ z4 L7 h\* d8 y$ M) m  
        │  │        
        │  ├─10浏览器渲染  
        │  │      01-浏览器解析.flv  
        │  │        
        │  ├─11mysql% Z+ q) L8 @4 W  
        │  │      01-mysql1.flv  
        │  │      02-mysql2.flv  
        │  │      03-mysql3.flv' I1 P3 V# f, G\* Q8 f9 U, {1 @  
        │  │      
        │  ├─12珠峰博客/ d  f" A2 ~% V. U/ ^3 F  
        │  │      01-博客1.flv\* x0 }, N% c& W2 X, g  
        │  │      02-博客2.flv+ ^# T' X; ?6 P& n  
        │  │      03-博客3.flv. c$ ]% U3 V. w: |0 j4 W3 Q  
        │  │      04-博客4.flv  
        │  │      05-博客5.flv  
        │  │      06-博客6.flv- q9 ?( `. n! D- @, N& f  
        │  │      07-博客7.flv  
        │  │      08-博客08.flv- g4 z2 \( a& r  
        │  │      09-博客09.flv  
        │  │      10-博客10.flv  
        │  │      11-博客11.flv  
        │  │      12-博客12.flv4 [9 q! k. x3 d  
        │  │      3 ?8 D( t7 \_8 j# q% J  
        │  ├─13redis& h+ }+ K( c: N/ P) r  
        │  │      01-redis.flv  
        │  │      
        │  ├─14测试  
        │  │      01-单元测试.flv# M" z4 S3 i; \" h  
        │  │      02-jwt.flv  
        │  │      03-单元测试.flv  
        │  │      04-单元测试.flv/ ], b/ |4 f( h6 w  
        │  │      05-单元测试.flv  
        │  │     $ y9 ~6 T1 c8 Z4 Y  
        │  ├─15canvas2 S3 I4 Y, k! Y& s) h+ J  
        │  │      01-画布画图.flv  
        │  │      02-background实现.flv! w. p3 T: r8 A4 h\* Z  
        │  │      03-land.flv  
        │  │      04-pipe.flv  
        │  │      05-bird.flv  
        │  │      06-碰撞检测.flv  
        │  │      07-场景管理之0,1.flv  
        │  │      08-场景管理之2,3.flv\* \% g4 m; n& P  
        │  │      09-颁奖界面.flv  
        │  │      % @) ?; F" u+ X1 M2 B! Y  
        │  ├─16linux. h4 @$ B+ M$ @- Z3 i. L0 q; I6 L  
        │  │      01-linux.flv& Z2 y# D) F! L: e! I  
        │  │      02-linux.flv  
        │  │      03-linux.flv  
        │  │      04-linux.flv  
        │  │      05-linux.flv  
        │  │    
        │  ├─17react源码  
        │  │      01-React源码.flv  
        │  │  % `& T- g: [! v  u( ~4 K  
        │  ├─18websocket  
        │  │      01-websocket.flv& m5 C5 r6 m8 L. @6 v% B  
        │  │      02-socket.flv  
        │  │      03-dva.flv  
        │  │      04-聊天室.flv  
        │  │      05-聊天室.flv5 ^7 y: \1 E5 v# F' ^8 K  
        │  │      8 J0 y' h0 J- ]0 K  
        │  ├─19跳一跳4 z% O' x\* t7 e# ~0 i$ G8 \  
        │  │      01-webGL介绍以及three.flv% b4 g\* J) [8 D  
        │  │      03-几何体创建以及相机镜头位置改变.flv  
        │  │      04-更新相机坐标实现视觉动画.flv5 R3 {0 T' R4 t& @  
        │  │      05-绑定事件实现jumper跳跃功能.flv  
        │  │      06-回顾思路梳理逻辑.flv8 ?/ i& I2 x% ], t- I3 r  z5 v  
        │  │      07-最终完成实现成功和失败的处理和重置操作.flv) B( y3 F1 e0 f8 T  
        │  │      6 u. Q9 C+ M% q& d6 S0 b  Z  
        │  ├─20珠峰爬虫  
        │  │      01-爬虫1.flv  
        │  │      02-爬虫2.flv0 c; `9 R4 g+ `4 e5 I- B1 k  
        │  │      03-爬虫3.flv  
        │  │      04-爬虫4.flv  
        │  │      05-爬虫5.flv7 C' B% A, ]8 l+ X: u  
        │  │      06-爬虫6.flv  
        │  │      07-布署.flv1 t8 A& t9 ^\* S% D& M  
        │  │      08-布署.flv2 Y# i2 L& Y5 t3 F: \) y  
        │  │        
        │  ├─21web安全  
        │  │      01-xss.flv' p+ w0 M\* J' G9 v/ m  
        │  │      02-xss.flv: O' c& ^6 O4 j1 W3 \_  
        │  │      03-xss.flv5 [. Y5 i( g; A/ e; c3 ]  
        │  │      04-xss.flv  
        │  │      05-xss.flv  
        │  │        
        │  ├─22nginx  
        │  │      01-nginx.flv6 f5 g( @% f: t" E  
        │  │      02-nginx.flv3 t! A' M& b9 f% x  
        │  │        
        │  ├─23react测试  
        │  │      01-react测试.flv6 W2 f; I$ [- B, }; l  
        │  │      02-react测试.flv  
        │  │        I' ~; w& K: |  
        │  ├─24docker  
        │  │      01-docker.flv  
        │  │      02-docker.flv6 Y/ \! E+ {$ D  
        │  │        
        │  ├─25https( i/ x$ M1 s" E4 x\* ]& L: |  
        │  │      01-https.flv. }\* z6 ^. R2 u7 |) e  
        │  │      02-https.flv( A$ X9 T& A( G; T# l  
        │  │        
        │  └─26ssr% ]1 Y7 V) O  G5 k/ F8 y4 c  
        │          02-nuxt.flv! g! l: b% Y# A6 ]/ p  
        │          02-nuxt2.flv  
        │          ( B' l' R$ J7 e, ^7 \_  
        ├─06算法/ p& z3 [: ]# a. W: q  
        │  │  01-直接大厂的资深前端算法修炼.flv1 s) B/ O5 T+ ^4 `( }2 ~  
        │  │ . e) T( j+ B' L& P  
        │  ├─01第一次开课  
        │  │      01-算法1.flv, K; S: F0 q, D: u3 d  
        │  │      02-算法2.flv; H" E\* g/ h6 J7 q- \  
        │  │      03-算法3.flv6 O+ y! N% {3 ?; P  
        │  │      04-算法4.flv! S1 E' ]$ n! t( g  
        │  │      05-算法5.flv  
        │  │        
        │  └─02第二次开课/ w) O5 o3 s\* A5 S, n  
        │          01-时间复杂度1.flv  
        │          02-时间复杂度2.flv4 g2 N$ [+ t2 \_; M  ^6 p  
        │          03-前端算法场景.flv  
        │          04-前端算法场景.flv  
        │          05-前端算法场景.flv  
        │          06-数据结构.flv6 m- \. I4 p\* K3 |- g  
        │          07-数据结构.flv8 h% o% U3 E3 \_; U$ I  e  
        │          08-数据结构.flv' V. |; O6 O; I( I3 \_  
        │          09-堆栈和队列.flv  
        │          10-堆栈和队列.flv  
        │          11-散列.flv6 I2 d: m' K\* A7 D$ D\* B$ X\* J0 O  
        │          12-堆.flv3 M3 c+ x" X- h. \_9 R( o( @; }  
        │          13-堆.flv  
        │          14-堆.flv9 |, j2 ~" m( Y2 a% H5 k0 E" q  
        │          15-最优解.flv7 d5 @1 R) G# t  
        │          16-最优解.flv! o\* h& Z. N, ^" q\* P0 X! }( W7 b( h  
        │          17-最优解.flv- {$ R. i4 h! `+ C\* z% y  
        │          0 c6 ]. L$ C4 @2 f  
        └─07公开课5 I' }  p9 Z3 S8 M% {  
                        01-弹幕.flv( Z3 ~0 I, @' E; C  R3 s+ J  
                        02-跨域.flv