复习：

1. 父组件往子组件传递

传递方： 父组件； 接收方：子组件

<son mytitle="this is tilte">

props: ['mytitle']

2. 子组件往父组件传递

① 传递：子组件 ② 接收： 父组件

<son @customEv="hC"></son>

this.$emit('customEv', 123)

3. 直接从子组件中获取数据

① 给子组件起别名 <son ref="com1">

② this.$refs.com1.uname

4. 直接从父组件获取数据

this.$parent.uname

5. 兄弟组件之间的通信

① 创建总线var bus = new Vue();

② 接收方 准备事件

bus.$on('customE', (msg)=> { this })

③ 发送方 触发事件

bus.$emit('customE','hello')

6. 创建组件的时候，把模板单独拿出来

Vue.component('my-com',template:`#myc`)

创建一个模板标签

<script type="text/x-template" id="myc">

//大量的标签

</script>

二、路由

单页应用 SPA ---> single page application

建立的是，路径和组件之间的对应关系

1. 路由的执行过程

① 对浏览器中的路径进行解析

② 通过解析的路径来找对应的组件

③ 在组件中，通过ajax获取数据

④ 把获取的数据渲染到页面上

2. 路由的使用步骤

① 引入路由插件

② 创建组件

③ 实例化路由

④ 在实例化的路由中配置路由 (routes)

⑤ 把创建好的路由实例注册到Vue实例（router）

⑥ 在视图中，创建router-view，作为组件渲染的位置

练习：(: -- 10: 27)

创建一个路由，实现 所有邮件(/all)、收件箱(/inbox)、重要邮件(/imp)三个对应关系

3. 路由配置注意事项

① 路由配置 redirect 改变路径的指向

② path中的 \*，如果以上都没有匹配，自动指向的组件

4. 路由跳转

① 通过路由标签 ，会被渲染成a标签

<router-link to="/login">登录</router-link>

② 通过js

通过一个点击事件，执行 this.$router.push('/login')

5. 路由中传递参数

① 传递 /detail/2

② 接收 在路由配置中 path:'/detail/:id'

就表示是要通过id属性来接收所传递的参数

this.$route.params.id

6. router单词汇总

routes 配置路由

router 注册路由实例

$router 路由跳转 this.$router.push('/')

$route 获取参数 this.$route.params.id

7. 子路由配置(children)

routes:[

{path: '/login', component: login, children:[ {},{} ]}

]

三、动画

前提： 切换效果（v-show v-if）

步骤：

① 把切换的部分放入到 transition标签之间

② 设置动画的样式

.v-enter, .v-leave-to{ //准备进场、已经离场 }

.v-enter-active,.v-leave-active{//准备离场，已经进场 }

四、vue-resource插件的使用

1. 引入文件

2. 在要请求的位置

this.$http.get('请求的url').then((result)=>{

//result就是请求的结果

})

3. 在php中设置允许跨域

header('Access-Control-Allow-Origin:\*');

五、Vue命令行使用

1. 安装

① 进入tpls文件夹

② 按照shift，单机鼠标右键，进入命令行工具

③ cnpm install 安装依赖的node包

④ 启动项目 npm start

2. 认识目录结构

src 项目目录（src- resource）

①assets 用于存放一些调用的文件（css, js,img）

②components，用于放置项目中的组件，文件后缀.vue

③ router 用于放置 路由实例化，配置等

④ App.vue 根组件，所有的路由切换都是在这里边执行

⑤ main.js 入口文件，项目启动的时候，从这开始

作业：

路由嵌套