1.

v-bind动态绑定的本地图片src需要放在static这个文件夹中

2.

<!-- elementui组件的prop不是string类型的时候需要v-bind绑定 不然会报错 -->

3.

 <!-- props有标题、背景颜色等 通过一个对象传进去-->

    <index-content v-bind="content\_01"></index-content>

    <index-content v-bind="content\_02"></index-content>

 data() {

    return {

      content\_01: {

        title\_01: "第一阶段课程",

        title\_02: "Java设计和编程思想",

        container\_bg: "#fff",

        title\_bg: "url('/static/image/strong02.png')",

        title\_01\_col: "#000",

        title\_02\_col: "#000"

      }

}

4.

<!-- 通过props、v-bind、style和data里的样式对象 控制组件的样式(也可写行内style解决) -->

    <div class="index-content-container" :style="bgColor">

 data() {

        return {

            bgColor: {

                backgroundColor: this.container\_bg

            }

        }

    }

 props: {

        container\_bg: {

            type: String,

            default: '#fff'

        }

1. 使用/deep/或>>>深度选择器改变element-ui的默认样式；使用scoped 后，父组件的样式将不会渗透到子组件中，使用深度选择器则可以从父组件渗透到子组件

/deep/ .el-carousel\_\_indicator--horizontal:hover

              .el-carousel\_\_button

                transition: all 0.2s

                height: 7px

6.

### **二、关于子组件的根元素**

　　使用了scoped属性之后，父组件的style样式将不会渗透到子组件中，然而**子组件的根节点元素**会**同时被设置了scoped的父css样式和设置了scoped的子css样式影响**，这么设计的**目的是父组件可以对子组件根元素进行布局**。

//子组件

<template>

    <div class="bottom-container">

        <p>Copyright </p>

    </div>

</template>

//调用子组件

<bottom></bottom>

//父组件中的样式修改子组件样式

.bottom-container

    background-color #ccc

    /deep/ p

            font-size: 30px

7.当使用了一些插件在ie会打不开，需要在webpack.base.conf.js 的include中做处理

{

        test: /\.js$/,

        loader: 'babel-loader',

        include: [resolve('src'), resolve('test'), resolve('node\_modules/webpack-dev-server/client'),

          //有些插件需要写进来兼容ie

          resolve('node\_modules/vue-baidu-map/components')

        ]

      }

1. es6语法转换成es5 安装babel-polyfill 并在main.js里

import 'babel-polyfill';

1. 在 vue 中，实现 Tab 切换主要有三种方式：使用动态组件，使用 vue-router 路由，使用第三方插件。

<https://blog.csdn.net/zhumengzj/article/details/79300784>

1. em和rem

原文链接：<http://caibaojian.com/rem-vs-em.html>

· rem 和 em 单位是由浏览器基于你的设计中的字体大小计算得到的像素值。

· em 单位基于使用他们的元素的字体大小。

· rem 单位基于 html 元素的字体大小。

· em 单位可能受任何继承的父元素字体大小影响

· rem 单位可以从浏览器字体设置中继承字体大小。

· 使用 em 单位应根据组件的字体大小而不是根元素的字体大小。

· 在不需要使用em单位，并且需要根据浏览器的字体大小设置缩放的情况下使用rem。

· 使用rem单位，除非你确定你需要 em 单位，包括对字体大小。

· 媒体查询中使用 rem 单位

· 不要在多列布局中使用 em 或 rem -改用 %。

· 不要使用 em 或 rem，如果缩放会不可避免地导致要打破布局元素。