# VueJS 高级

# Layout布局

1. 创建layout文件，vue一定要有根节点，根节点为src/main.js
2. 编写layout文件

<template>

<div class=”app-header”>header</div>

<div class=”app-content”>content</div>

<div class=”app-footer”>footer</div>

</template>

1. 在main.js中引入layout文件

//位置自定义

Import layout from “./components/layout”

1. 将layout直接渲染到根节点

New Vue ({

El : “#app”,

Template : “<Layout/>”

Components : { layout }

)}

1. 编写其他layout文件
2. 通过vue-router进行路径跳转

# 模板循环

//template中的内容都将循环显示，也可用<div>标签代替

<template v-for=”product in productList”>

<h1>{{....}}</h1>

<ul>{{.....}}</ul>

</templeate>

# 给特定位置添加样式

//line-last : true时添加样式

:class = “{‘line-last’ : index % 2 !== 0 }”

# Vue-resource实现ajax

1. 安装vue-resource

Cnpm install vue-resource --save

1. 在main.js引入

Import vueRourse from “vue-resource”

1. 使用全局方法use，注册

Vue.ues(vueRourse)

1. 使用get

Export default {

//在生命周期创建完毕后触发

Created: function () {

This.$http.get().

then(function( data ) {

..成功回调

},function( err ){

..失败回调

}

}

1. 使用post

Export default {

Created: function () {

This.$http.post(“地址”,{ data : value }).then(function(){...},function(){})

}

}

-------------------------------------

前端使用json-service进行模拟后端服务器

安装json-service

-------------------------------------

使用ES6语法的箭头函数 =>

Export default {

Created () {

This.$http.get(....).then( (res) => {

//重新赋值

this.newsList = res.data } , (data ) => {...}

}

}

------------------------

.new-item {

//显示状态

Display : inline-block;

//固定宽度

Width : 80px;

//超出隐藏

Overflow : hidden;

//超出部分显示”...”

Text-overflow : ellipsis;

//不允许换行

White-space : nowrap

}

# 引入图片

Data () {

Return {

Slides: {

//通过js引入的文件，需要require

Src : require(‘....地址....’)

}

}

}

# 调用mothod方法

This.$option.mothods.方法名（）

# 修改class样式方法

使用v-bind:class绑定class类，建议使用缩写：

Style样式与class类使用方法相同，:style;

1. <div :class=”{class:value}”></div>

:class可以使用一个表达式，当value值为true时，显示class类

1. <div :class=”{class:value,’class2’:value}”></div>

也可以绑定多个值

-----------------以上是直接写入对象--------------------------------

1. <div :class=”class”></div>

绑定一个类，使用变量里是一个类

New vue({

Data{

Class:{

One:true,two:true

}

}

1. <div :class=”class”></div>

使用计算属性

New vue({

Computed:{

Class:function(){

Return {

.......

}

}

}

})

---------------------------------------------

1. <div :class=”[class1,class2]”></div>

使用数组传递变量，同时显示class1,class2。class1,class2位于data中的数据

1. <div :class=”[aaa?aaa:bbb]”></div>

也可以是用绑定一个三元表达式，不能绑定到对象中，