

一、组件component

1. 什么是组件？

组件（Component）是 Vue.js 最强大的功能之一。组件可以扩展 HTML 元素，封装可重用的代码
组件是自定义元素（对象）

2. 定义组件的方式

方式1：先创建组件构造器，然后由组件构造器创建组件
方式2：直接创建组件

3. 组件的分类

分类：全局组件、局部组件

4. 引用模板

将组件内容放到模板<template>中并引用

5. 动态组件

```
<component :is="">组件
```

多个组件使用同一个挂载点，然后动态的在它们之间切换

```
<keep-alive>组件
```

二、组件间数据传递

1. 父子组件

在一个组件内部定义另一个组件，称为父子组件
子组件只能在父组件内部使用
默认情况下，子组件无法访问父组件中的数据，每个组件实例的作用域是独立的

2. 组件间数据传递（通信）

2.1 子组件访问父组件的数据

a) 在调用子组件时，绑定想要获取的父组件中的数据
b) 在子组件内部，使用`props`选项声明获取的数据，即接收来自父组件的数据
总结：父组件通过`props`向下传递数据给子组件
注：组件中的数据共有三种形式：`data`、`props`、`computed`

2.2 父组件访问子组件的数据

a) 在子组件中使用`vm.$emit(事件名, 数据)`触发一个自定义事件，事件名自定义
b) 父组件在使用子组件的地方监听子组件触发的事件，并在父组件中定义方法，用来获取数据
总结：子组件通过`events`给父组件发送消息，实际上就是子组件把自己的数据发送到父组件

3. 单向数据流

`props`是单向绑定的，当父组件的属性变化时，将传导给子组件，但是不会反过来
而且不允许子组件直接修改父组件中的数据，报错

解决方式：

方式1：如果子组件想把它作为局部数据来使用，可以将数据存入另一个变量中再操作，不影响父组件中的数据

方式2：如果子组件想修改数据并且同步更新到父组件，两个方法：

a. 使用`.sync`（1.0版本中支持，2.0版本中不支持，2.3版本又开始支持）

需要显式地触发一个更新事件

b. 可以将父组件中的数据包装成对象，然后在子组件中修改对象的属性（因为对象是引用类型，指向同一个内存空间），推荐

4. 非父子组件间的通信

非父子组件间的通信，可以通过一个空的`Vue`实例作为中央事件总线（事件中心），用它来触发事件和监听事件

```
var Event=new Vue();  
Event.$emit(事件名,数据);  
Event.$on(事件名,data => {});
```

三、slot内容分发

本意：位置、槽

作业：用来获取组件中的原内容，类似`angular`中的`transclude`指令

四、vue-router路由

1. 简介

使用`Vue.js`开发SPA（Single Page Application）单页面应用

根据不同`url`地址，显示不同的内容，但显示在同一个页面中，称为单页面应用

参考

```
bower info vue-router  
cnpm install vue-router -S
```

2. 基本用法

```
a. 布局  
b. 配置路由
```

3. 路由嵌套和参数传递

传参的两种形式：

```
a. 查询字符串: login?name=tom&pwd=123  
    {{$route.query}}  
b. rest风格url: regist/alice/456  
    {{$route.params}}
```

4. 路由实例的方法

```
router.push() 添加路由，功能上与<route-link>相同  
router.replace() 替换路由，不产生历史记录
```

5. 路由结合动画

五、 单文件组件

1. .vue文件

.vue文件，称为单文件组件，是Vue.js自定义的一种文件格式，一个.vue文件就是一个单独的组件，在文件内封装了组件相关的代码：html、css、js

.vue文件由三部分组成: <template>、<style>、<script>

```
<template>
  html
</template>

<style>
  css
</style>

<script>
  js
</script>
```

2. vue-loader

浏览器本身并不认为.vue文件，所以必须对.vue文件进行加载解析，此时需要vue-loader
类似的loader还有许多，如: html-loader、css-loader、style-loader、babel-loader等
需要注意的是vue-loader是基于webpack的

3. webpack

webpack是一个前端资源模板化加载器和打包工具，它能够把各种资源都作为模块来使用和处理
实际上，webpack是通过不同的loader将这些资源加载后打包，然后输出打包后文件
简单来说，webpack就是一个模块加载器，所有资源都可以作为模块来加载，最后打包输出

[官网] (<http://webpack.github.io/>)

webpack版本: v1.x v2.x

webpack有一个核心配置文件: webpack.config.js，必须放在项目根目录下

4. 示例，步骤:

4.1 创建项目，目录结构 如下:

webpack-demo

```
| -index.html
| -main.js   入口文件
| -App.vue   vue文件
| -package.json  工程文件
| -webpack.config.js  webpack配置文件
| -.babelrc   Babel配置文件
```

4.2 编写App.vue

4.3 安装相关模板

```
cnpm install vue -S
```

```
cnpm install webpack -D
```

```
cnpm install webpack-dev-server -D
```

```
cnpm install vue-loader -D
```

```
cnpm install vue-html-loader -D
```

```
cnpm install css-loader -D
```

```
cnpm install vue-style-loader -D
```

```
cnpm install file-loader -D
```

```
cnpm install babel-loader -D
```

```
cnpm install babel-core -D
```

```
cnpm install babel-preset-env -D //根据配置的运行环境自动启用需要的babel插件
```

```
cnpm install vue-template-compiler -D //预编译模板
```

```
合并: cnpm install -D webpack webpack-dev-server vue-loader vue-html-loader css-loader  
vue-style-loader file-loader babel-loader babel-core babel-preset-env vue-template-  
compiler
```

4.4 编写main.js

4.5 编写webpack.config.js

4.6 编写.babelrc

4.7 编写package.json

4.8 运行测试

```
npm run dev
```

六、vue-cli脚手架

1. 简介

```
vue-cli是一个vue脚手架，可以快速构造项目结构
vue-cli本身集成了多种项目模板：
  simple  很少简单
  webpack 包含ESLint代码规范检查和unit单元测试等
  webpack-simple 没有代码规范检查和单元测试
  browserify 使用的也比较多
  browserify-simple
```

2. 示例，步骤：

2.1 安装vue-cli，配置vue命令环境

```
cnpm install vue-cli -g
vue --version
vue list
```

2.2 初始化项目，生成项目模板

语法：vue init 模板名 项目名

2.3 进入生成的项目目录，安装模块包

```
cd vue-cli-demo
cnpm install
```

2.4 运行

```
npm run dev //启动测试服务
npm run build //将项目打包输出dist目录，项目上线的话要将dist目录拷贝到服务器上
```

3. 使用webpack模板

```
vue init webpack vue-cli-demo2
```

ESLint是用来统一代码规范和风格的工具，如缩进、空格、符号等，要求比较严格

[官网](#)

问题Bug: 如果版本升级到node 8.0 和 npm 5.0, 控制台会报错:

```
GET http://localhost:8080/__webpack_hmr net::ERR_INCOMPLETE_CHUNKED_ENCODING
```

解决方法:

a) 降低Node版本到7.9或以下

b) 修改build/dev-server.js文件, 如下:

```
var hotMiddleware = require('webpack-hot-middleware')(compiler, {  
  log: () => {},  
  heartbeat:2000 //添加此行  
})
```

参考: <https://github.com/vuejs-templates/webpack/issues/731>