# Vue

## Es6和js 的关系

Es6 是一种规范 ，js严格执行这种规范

## ES6用法

### 关键字

Let 局部作用域

const 声明常量 一定要初始化 且不能更改

var 全局

### 解构赋值

//1、数组解构

// 传统

let a = 1, b = 2, c = 3

console.log(a, b, c)

// ES6

let [x, y, z] = [1, 2, 3]

console.log(x, y, z)

//2、对象解构

let user = {name: 'Helen', age: 18}

// 传统

let name1 = user.name

let age1 = user.age

console.log(name1, age1)

// ES6

let { name, age } = user//注意：结构的变量必须是user中的属性

console.log(name, age)

### 反引号 `

定义多行字符串，还可以在字符串中加入变量和表达式。

let age = 27

let info = ` I am ${age+1} years old next year.`

### 拓展运算符（...）

// 1、拷贝对象

let person1 = {name: "Amy", age: 15}

let someone = { ...person1 }

console.log(someone) //{name: "Amy", age: 15}

// 2、合并对象

let age = {age: 15}

let name = {name: "Amy"}

let person2 = {...age, ...name}

console.log(person2) //{age: 15, name: "Amy"}

### 箭头函数

## Vue

### 路由跳转

1 <router-link :to="'orderDetail'">订单详情</router-link>

2 this.$router.push({name: "orderDetail"});

this.$router.push({path: "/orderDetail"});

## 生命周期

重要的两个方法：

created 数据渲染之前

Mounted 数据渲染之后

## Node.js

1 js的运行环境，类似java的jdk环境

2 模拟服务器效果，类似tomcat

## NPM

类似maven

安装了node.js 默认就装了npm

### 项目初始化

#建立一个空文件夹，在命令提示符进入该文件夹 执行命令初始化

npm init

#按照提示输入相关信息，如果是用默认值则直接回车即可。

#name: 项目名称

#version: 项目版本号

#description: 项目描述

#keywords: {Array}关键词，便于用户搜索到我们的项目

#最后会生成package.json文件，这个是包的配置文件，相当于maven的pom.xml

#我们之后也可以根据需要进行修改。

如果想直接生成 package.json 文件，那么可以使用命令

npm init -y

### npm install命令的使用

1#使用 npm install 安装依赖包的最新版，

#安装会自动在项目目录下添加 package-lock.json文件，这个文件帮助锁定安装包的版本

#同时package.json 文件中，依赖包会被添加到dependencies节点下，类似maven中的 <dependencies>

如 npm install jquery

#如果安装时想指定特定的版本

npm install jquery@2.1.x

#devDependencies节点：开发时的依赖包，项目打包到生产环境的时候不包含的依赖

#使用 -D参数将依赖添加到devDependencies节点

npm install --save-dev eslint

#或

npm install -D eslint

#全局安装

#Node.js全局安装的npm包和工具的位置：用户目录\AppData\Roaming\npm\node\_modules

#一些命令行工具常使用全局安装的方式

npm install -g webpack

2 根据配置文件package.json 下载依赖 命令

终端命令 npm install

### 其他命令

#更新包（更新到最新版本）

npm update 包名

#全局更新

npm update -g 包名

#卸载包

npm uninstall 包名

#全局卸载

npm uninstall -g 包名

## Babel

转码器，可以将ES6代码转为ES5代码

1npm install --global babel-cli

#查看是否安装成功

babel –version

2配置

Babel的配置文件是.babelrc，存放在项目的根目录下 文件内容

{

"presets": ["es2015"],

"plugins": []

}

3在项目中安装

npm install --save-dev babel-preset-es2015

4、转码

文件转文件

babel src/example.js -o dist1/compiled.js

文件夹转文件夹

babel 要转换的文件夹路径-d 转换的文件夹路径

如babel src -d dist2

## 模块化

后端 类与类之间的调用为模块化操作

前端 js 与js调用 为模块化操作

CommonJS使用 exports 和require 来导出、导入模块。

**运行程序**

node common-js模块化/引入模块.js

ES6使用 export 和 import 来导出、导入模块。

ES6**的模块化无法在Node.js中执行，需要用Babel编辑成ES5后再执行。**

## Webpack

可以将多种静态资源 js、css、less 转换成一个静态文件

## debugger 设置断点调试

## ==比较的是值 ===比较的是值和类型