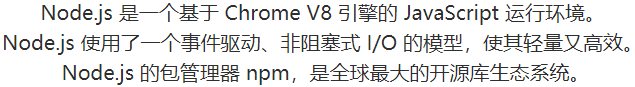
# 进击Node.js基础（一）

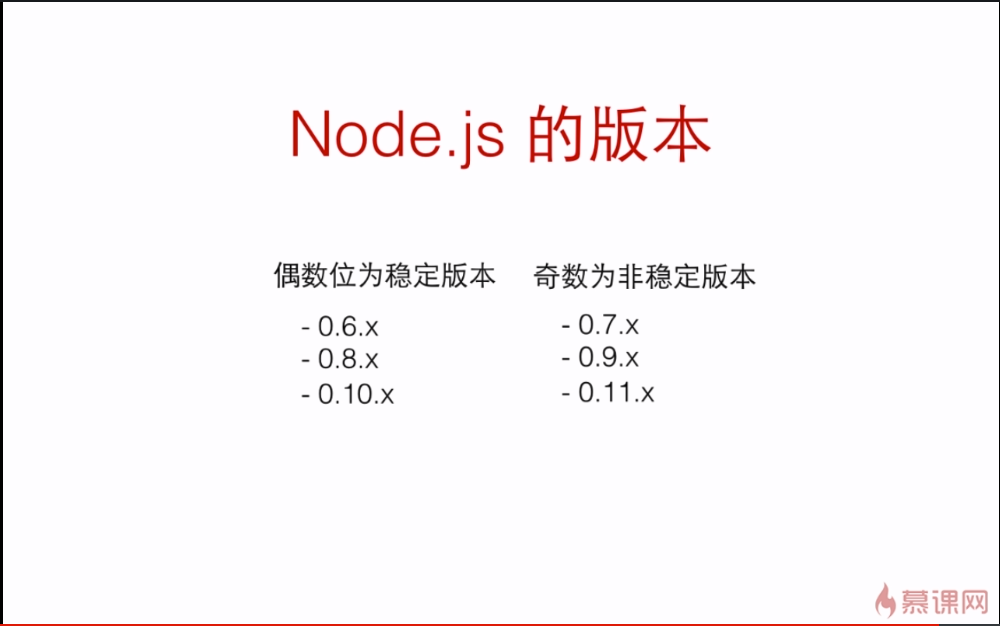
1. Nodejs是JavaScript的运行环境；



1. 基于Nodejs 的技术和完成的项目：



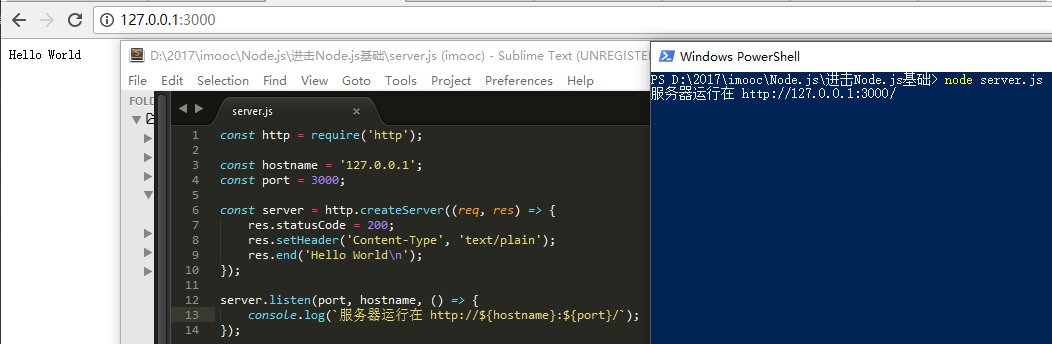
1. 网站推荐：<https://nodejs.org> ；<https://www.npmjs.com> ；<https://stackoverflow.com/> ；<https://github.com/> ；
2. Nodejs版本问题



1. 安装gitbash、安装nodejs；官网直接下载傻瓜式安装；查看版本号：node -v，npm-v
2. 事件冒泡：只要是能够触发事件冒泡的情况下，就考虑取消事件冒泡，或者直接利用事件冒泡



1. 实现一个简单web服务器



1. 浏览器执行环境中存在window全局变量，nodejs执行环境中存在process模块，这是两者的区别之一；
2. 大量js文件在页面引入时，很容易出现变量/方法被覆盖掉，特别是这些文件存在依赖关系的时候；容易导致页面出错；

因为javascript天生缺少一种模块管理机制 来隔离实现不同功能的JS片段；避免他们相互污染；

为此我们经常采用命名空间的方法，把变量和函数限制在某个特定的作用域内；人肉约定一套命名规范来约束代码，从而保证代码的安全执行。如jQuery必须通过$来调用其中的方法；

在commonjs中一个文件就是一个模块,就是一个作用域；

Node js借鉴了Commonjs的模块组织理念；并基于commonjs实现了一套模块管理系统；在Nodejs中每个js文件可以看做是一个独立的模块； 在里边不需要有命名空间，不用担心变量污染，整个文件中的代码可以非常自然的组织起来， 整个模块可以通过彼此的依赖引入组合起来从而形成更强大的功能/功能包； 安装nodejs时会安装npm这个包管理工具，通过该工具可以往项目中引入各种模块，这些模块的像是相互独立的 完整的；

Commonjs规范包括模块、包、二进制、控制台、编码等规范，来约定Javascript怎么来组织编写；

同时大部分标准也是在拟定和讨论之中的；

把不同功能的代码看做是独立的模块，每个模块看做是独立的作用域；

但并不是孤立的，可能存在依赖关系；

每个模块可以分为三个关键的部分，分别是模块的定义，标识，引用；

Commonjs是一套规范呐！

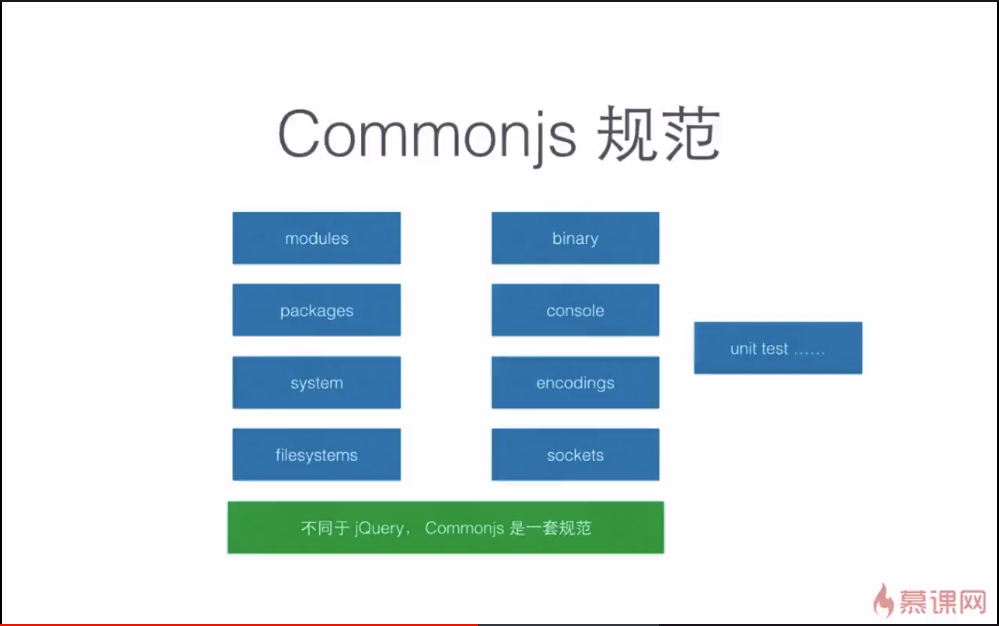
CommonJS模块的特点如下：

所有代码都运行在模块作用域，不会污染全局作用域。

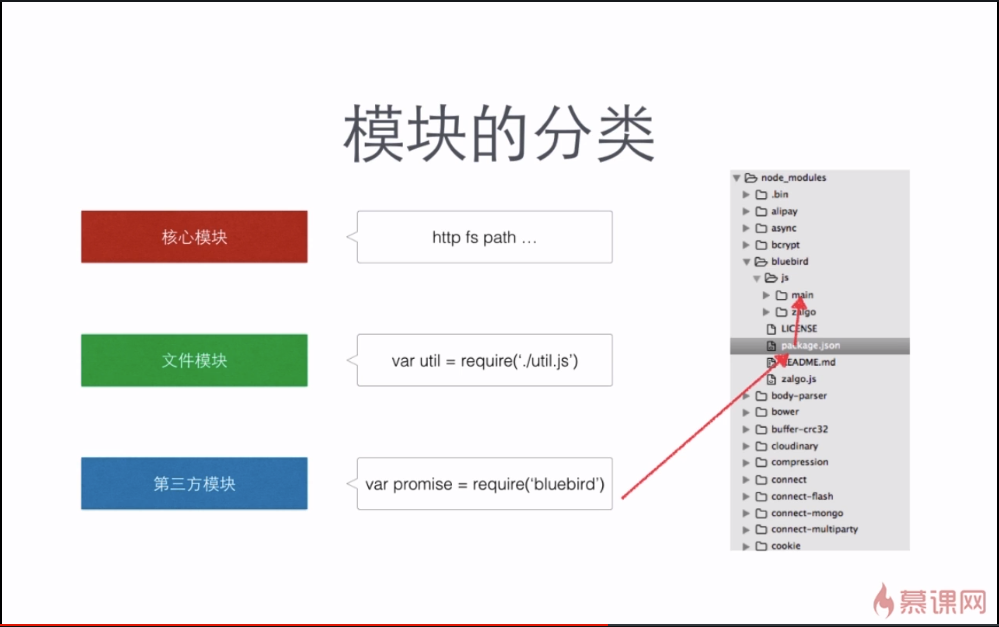
模块可以多次加载，但是只会在第一次加载时运行一次，然后运行结果就被缓存了，以后再加载，就

直接读取缓存结果。要想让模块再次运行，必须清除缓存。

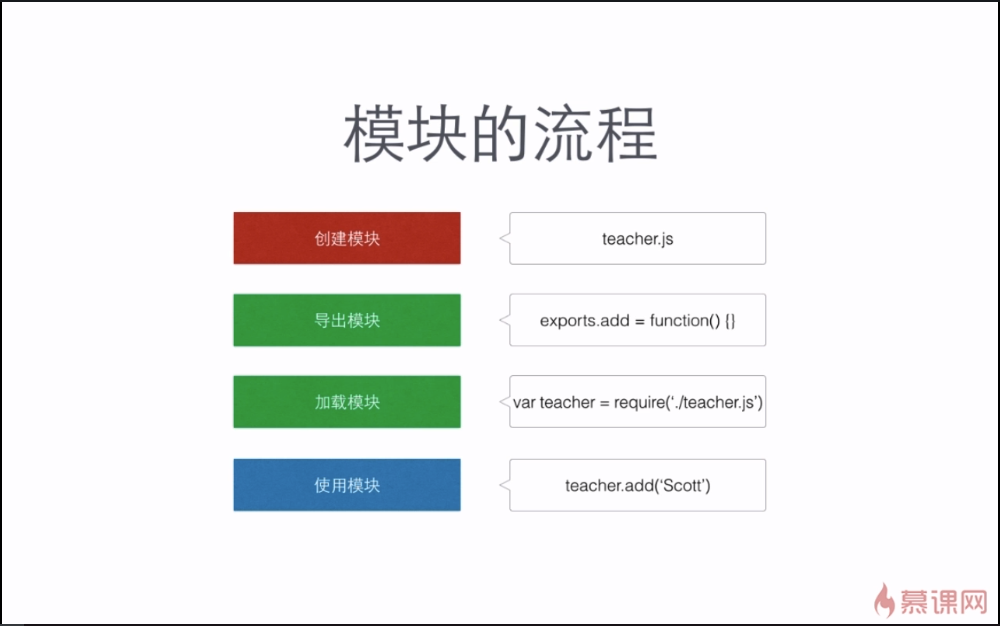
模块加载的顺序，按照其在代码中出现的顺序。



1. 在nodejs中文件和模块是一一对应的，如下是引用模块的方法：



1. 创建简单的模块：



1. Nodejs模块系统，module.exports = Hello 和exports.world = function(){}的区别：



1. 打开命令行，输入node，回车，进入nodejs运行环境；
2. URL网址解析：



1. QueryString参数处理小利器
2. HTTP源码解析：
3. 1
4. 1
5. 1
6. 1
7. 1
8. 1

# 进击Node.js基础（二）

1. 1
2. 1
3. 1
4. 1
5. 1
6. 1

# node+mongodb 建站攻略（一期）

1. Node+Express快速搭建电影网站 --- 开发框架介绍



1. 项目开发流程



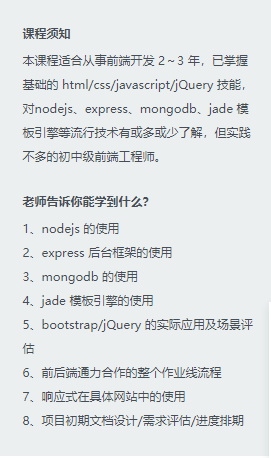
1. 项目结构初始化 → 入口文件编写



1. <http://momentjs.cn/> : Moment.js 是一个 JavaScript 日期处理类库,用于解析、检验、操作、以及显示日期



1. 课程须知



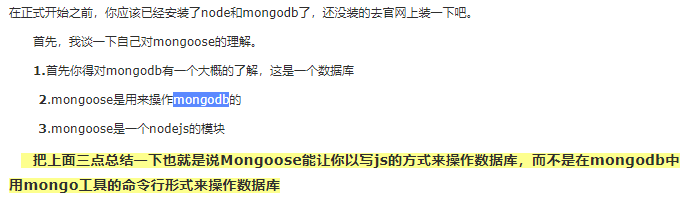
1. 设计数据库模型：数据库MongoDB!!!

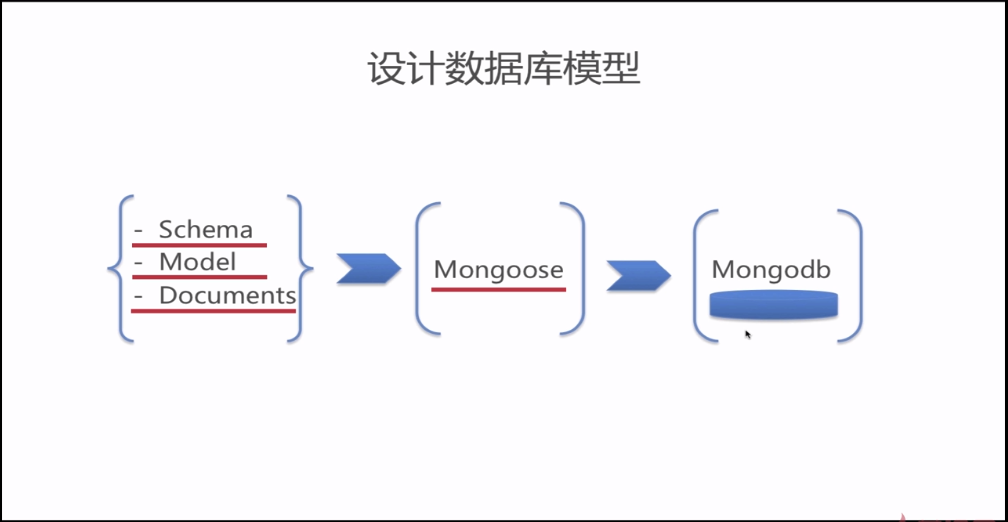
schema模式 – 字段定义（示例图，核心知识体系）、model模型 – 对模式进行编译生成构造函数（）、documents文档 – 文档实例化，可进行增删改；

Mongoose工具可以对MongoDB进行建模；

初识Mongoose：<http://www.cnblogs.com/bax-life/p/7795061.html> ；

MongoDB学习笔记之Mongoose的使用： <http://blog.csdn.net/sinat_25127047/article/details/50560167> ；

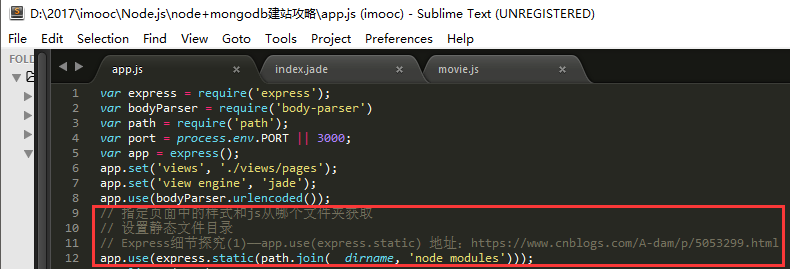




1. JavaScript标准规范JavaScript Standard Style：<https://standardjs.com/rules-zhcn.html#javascript-standard-style>



1. Express细节探究(1)——app.use(express.static)：<https://www.cnblogs.com/A-dam/p/5053299.html>



1. Underscore模块是一个 JavaScript 工具库；



1. 1
2. 1
3. 1
4. 1
5. 1

# node建站攻略(二期)——网站升级

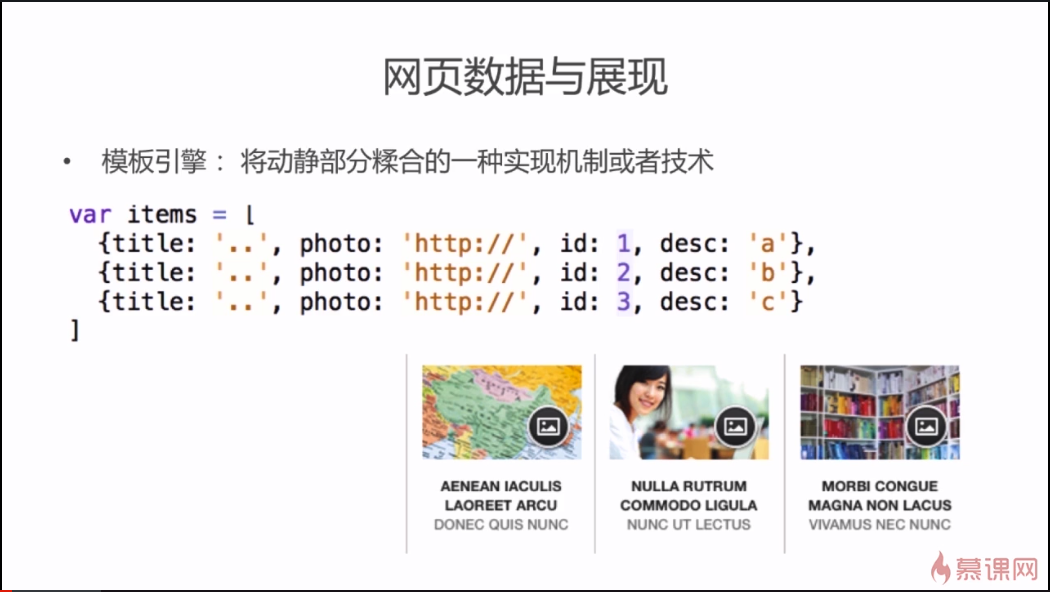
1. 1
2. 1
3. 1
4. 1
5. 1
6. 1

# 创业公司的Nodejs工程师

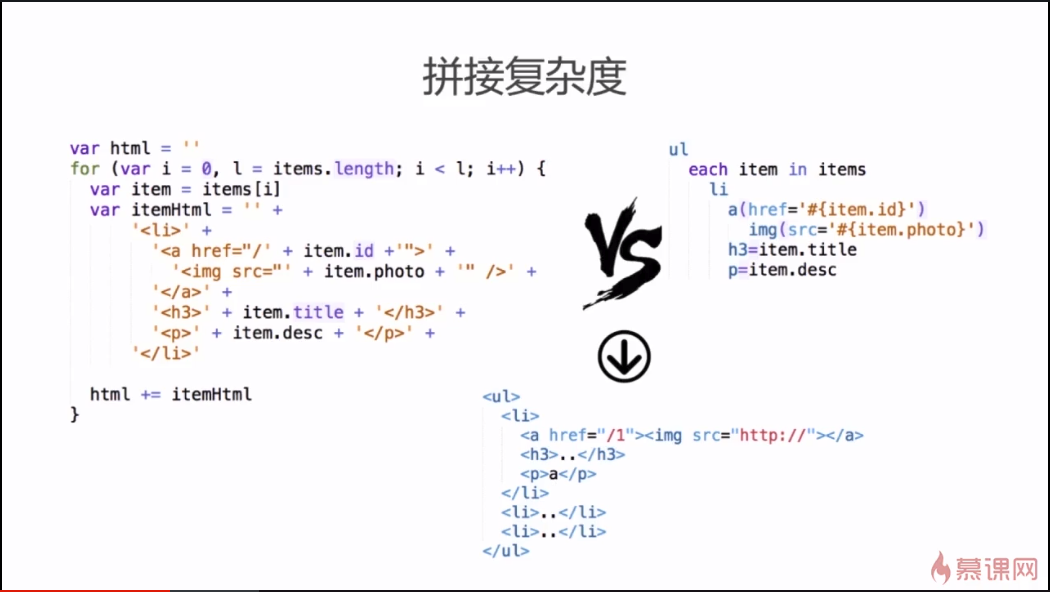
1. 1
2. 1
3. 1
4. 1
5. 1
6. 1

# 带你学习Jade模板引擎

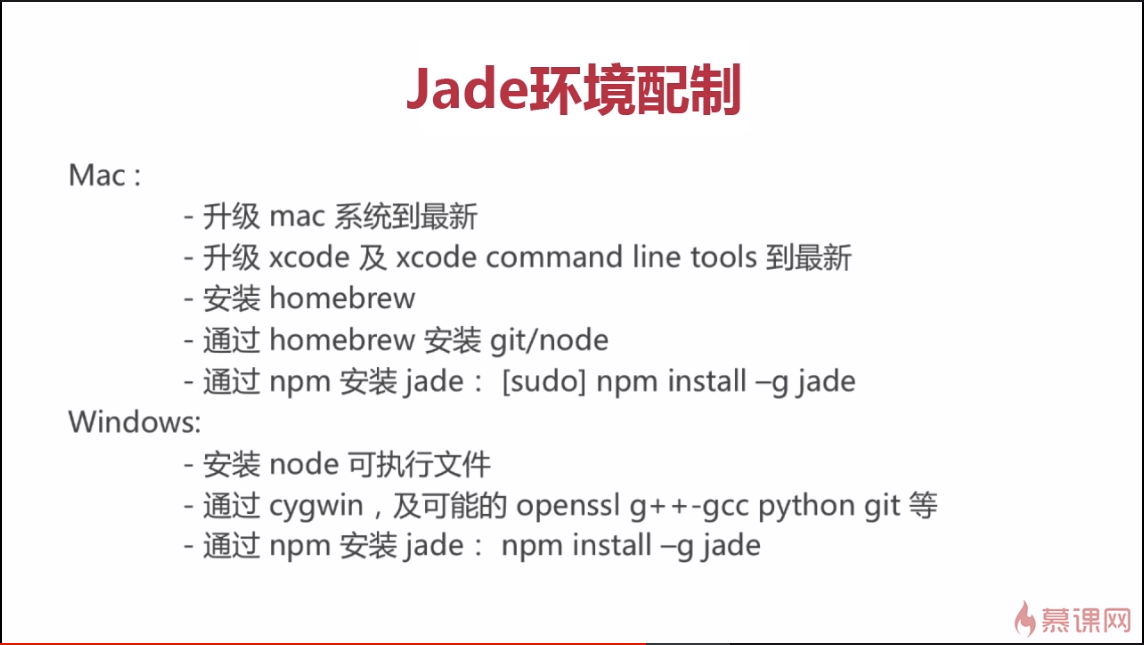
1. 模板引擎是什么：将静态部分糅合的一种实现机制或者技术



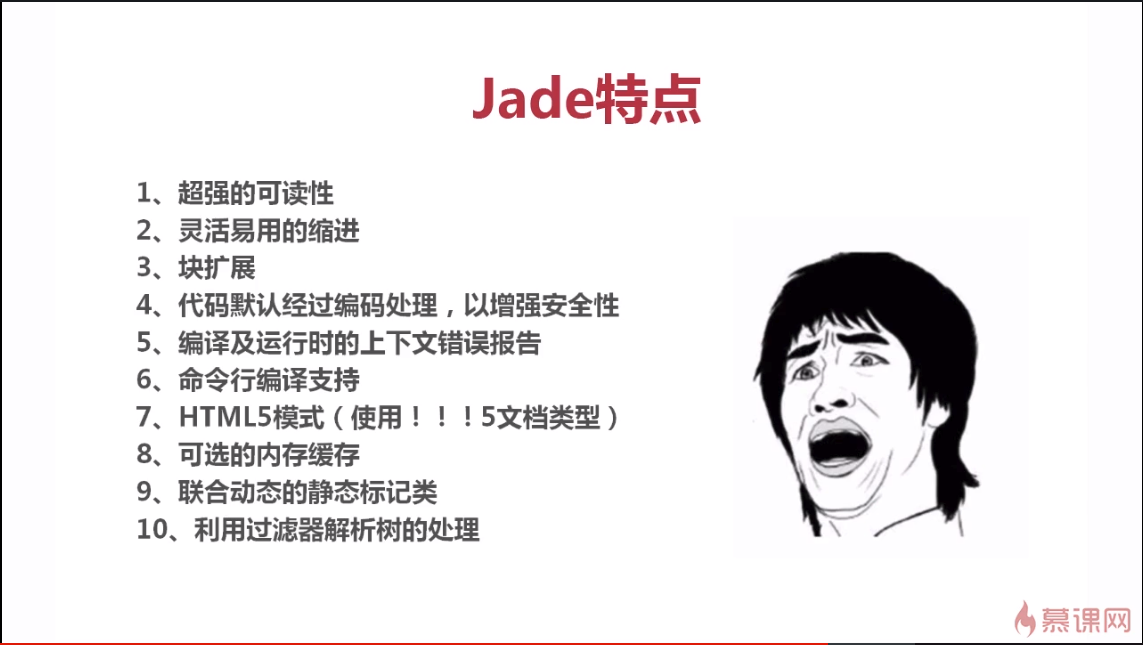
1. 模板引擎 VS 简单字符串拼接



1. Jade环境配置



1. Jade特点：Jade是express（基于 Node.js（基于 Chrome V8 引擎的 JavaScript 运行环境） 平台，快速、开放、极简的 web 开发框架）默认的模板引擎；Jade不局限于Node.js,用其他后台语言也可以用Jade来开发页面再编译成HTML，而不是像ejs这样的模板引擎里面本来就是HTML标签；



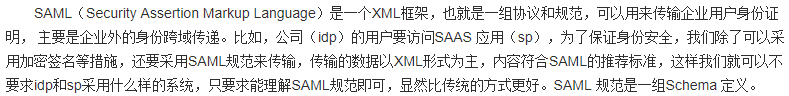
1. 为什么需要用到Jade这样的模板引擎，需要编译，还需要配置那么麻烦，直接写HTML标签不也一样吗？尤其是使用一些集成的ide或者结合一些插件，比如emmet来写代码也很快；

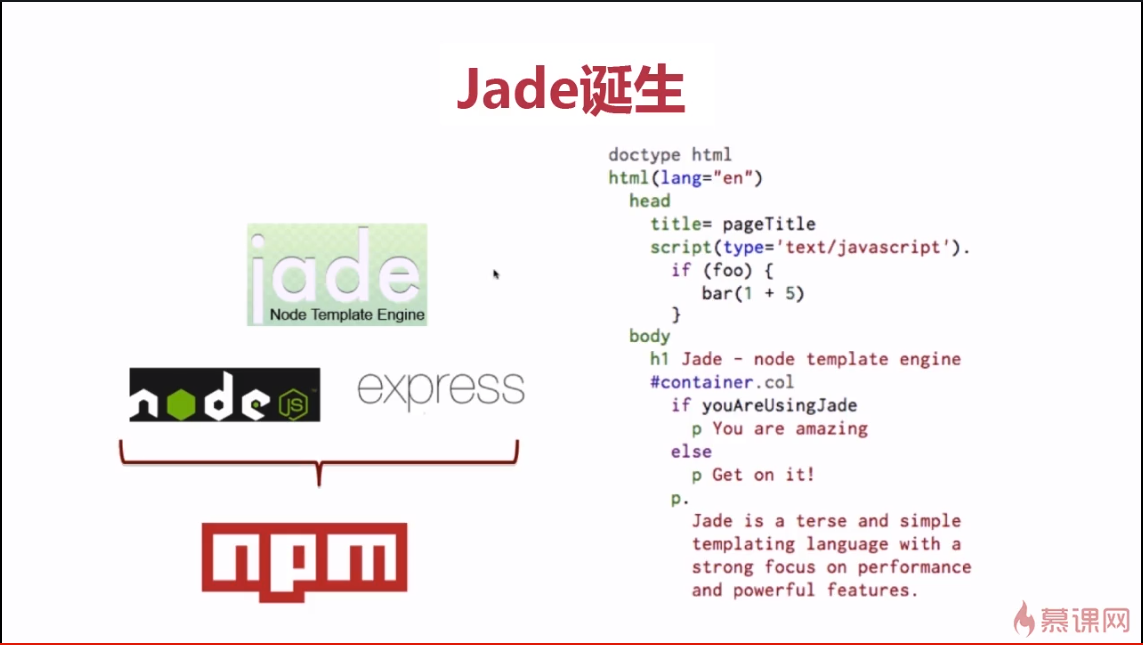
在开发HTML页面的时候会涉及很多复杂区块，会涉及深层次的DOM嵌套，这就是著名的巢状DOM结构嵌套，在后期维护修改时容易出错（一不小心少了<符号，或者标签未闭合）；



1. Jade的诞生

SAML用一套新的语法来描述HTML元素，最大特点是不像HTML使用成对的标签描述，而是使用缩进排列那样的json关系来解决嵌套和成对标签等问题；Jade就是由这样的概念启发；



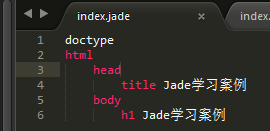


1. Jade版本问题；



1. 编写Jade；

新建index.jade文件，输入；

；

在命令行进行编译：jade index.jade，目录中会生成index.html文件（该文件经过压缩）





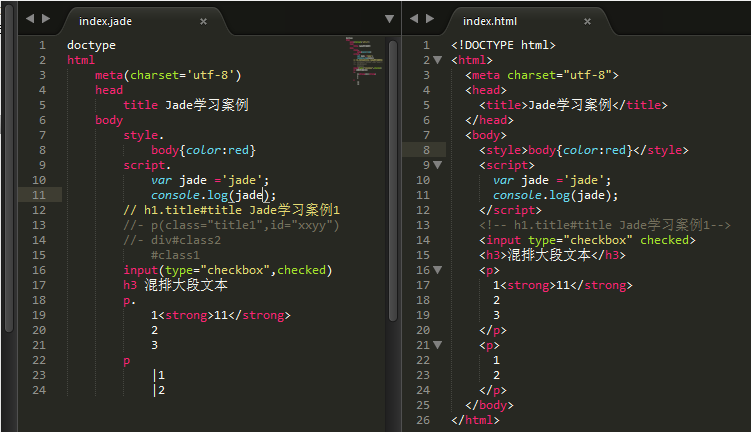
jade -P index.jade，生成格式化未压缩的文件



Jade -P -w index.jade，生成格式化未压缩的文件，同时实时编译



1. 元素和标签傻傻分不清楚：div是一个块状元素，<div></div>，尖括号包起来的div（如<div>、</div>）是标签，用来标记这个元素，尖括号里面是元素名，元素是由开始和结束标签组成，用来包含某些内容的整段代码；整个HTML文档就是由许许多多的HTML元素组成的；所以应该说是元素语法，而不是标签语法；
2. 标签语法、属性文本和值、混合的成段文本和标签、注释和条件注释；



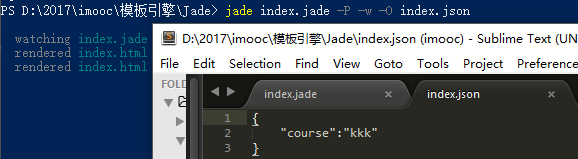
1. 声明变量和数据传递；



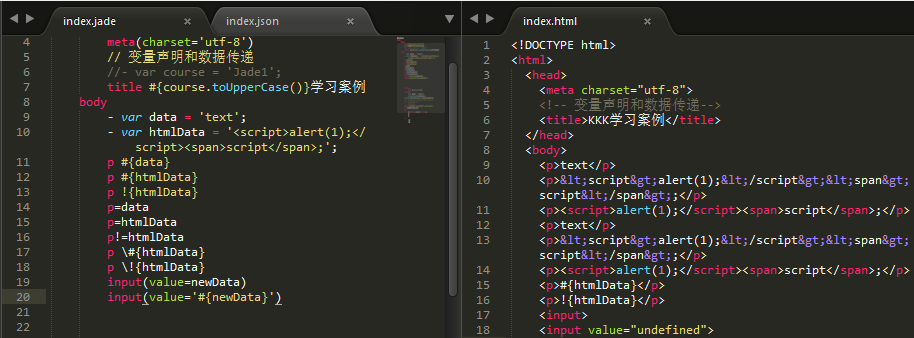
命令行传递数据：



json文件传递参数：



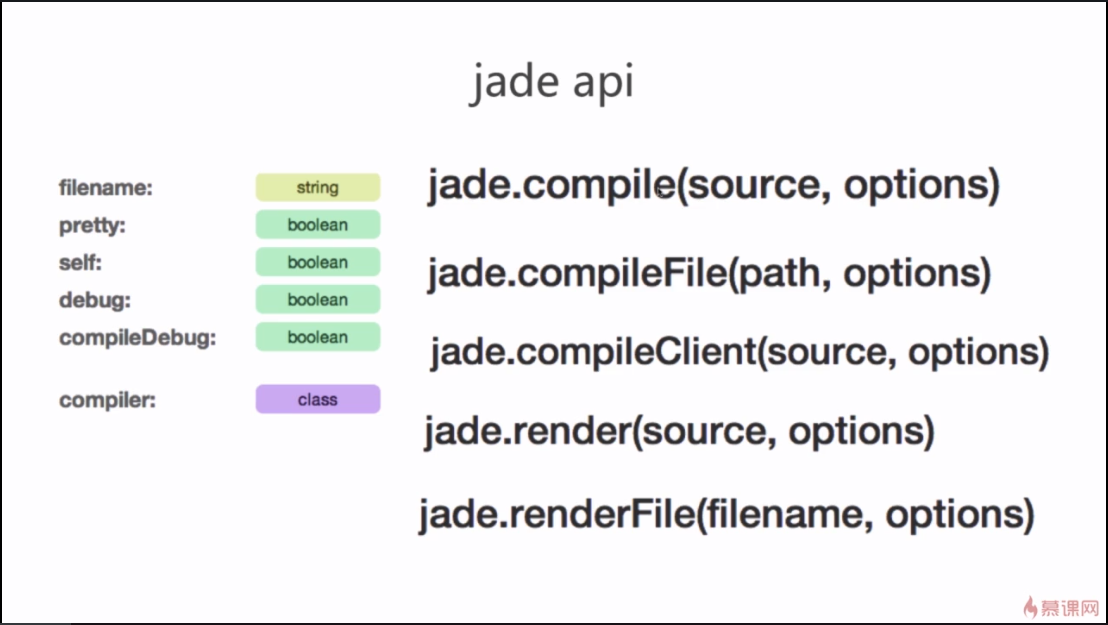
1. 安全转义与非转义；



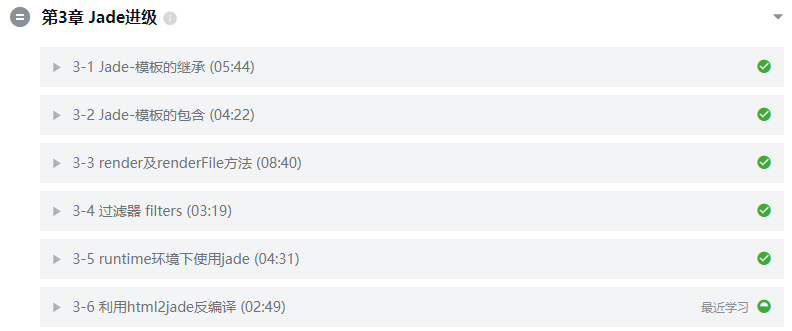
1. 流程代码-for-each-while；

流程代码-if-else-unless；

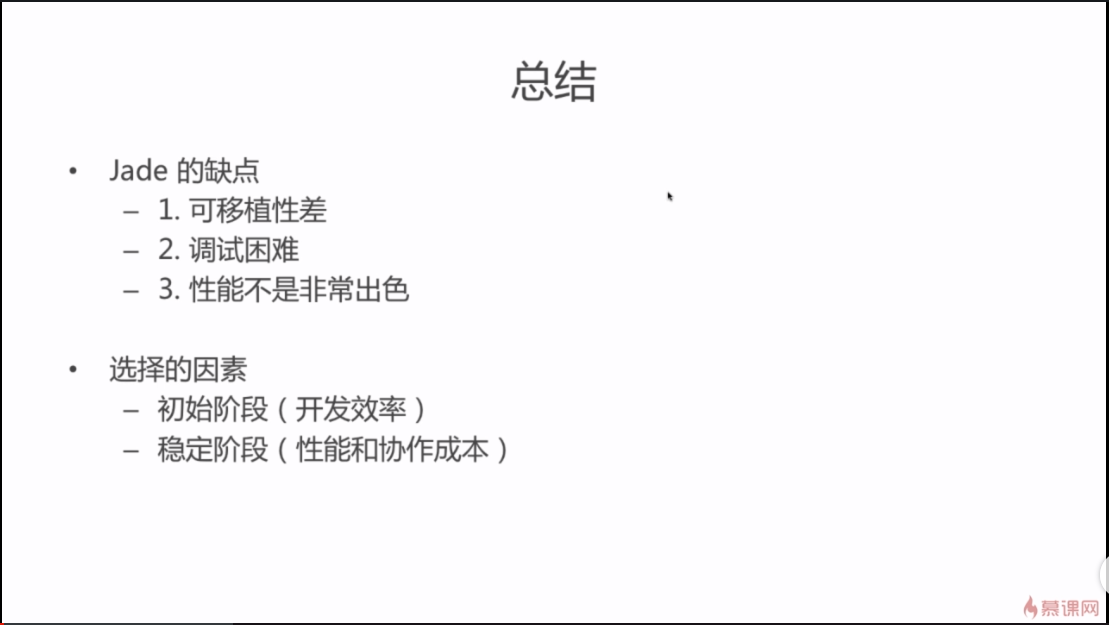
1. 有时候要面向好多区块，有时列表区块，但是他们的代码结构又是一摸一样的，那在Jade里面也是各写各的吗？；
2. 在 Scss 和 Jade 中，混合宏（mixins）都是举足轻重的语法。混合宏具有复用、解耦、可读、可扩。可维护等等优势。<https://www.imooc.com/video/5299> ；
3. 模板继承解决的是子文件和父文件之间继承和代码复用的问题，关键字extends，block；模板包含解决的是文件和文件之间和区块之间的内嵌问题，关键字include；
4. Jade api；



1. 不明白的章节；



1. Jade总结；



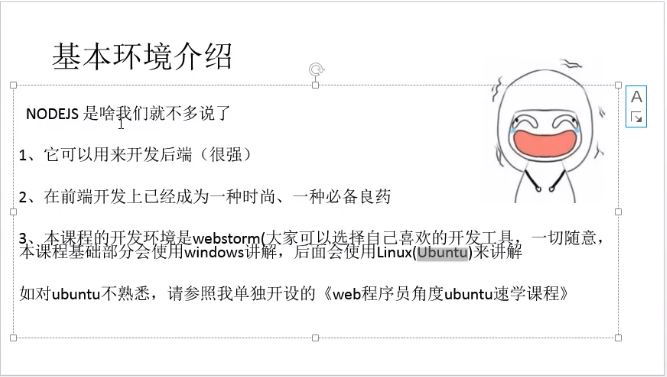
# 前端开发速学成财:Nodejs+Gulp+Webpack基础实战篇视频课程

1. <http://edu.51cto.com/course/7276.html> ；

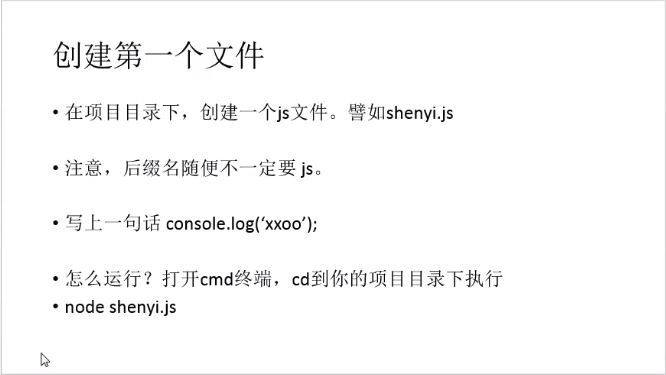


1. 基本环境介绍；

对Linux（ubuntu）不熟悉，需要学习《web程序员角度ubuntu速学课程》

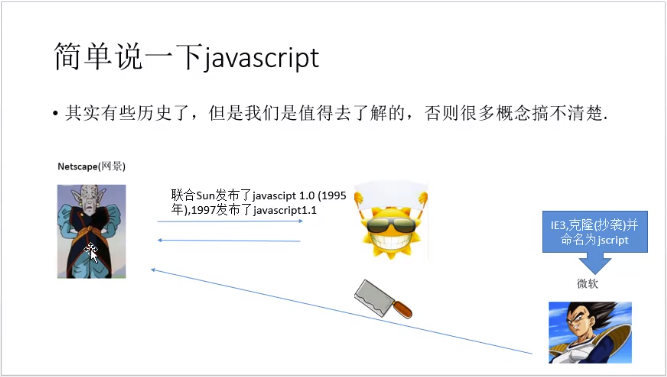


1. 创建第一个文件；

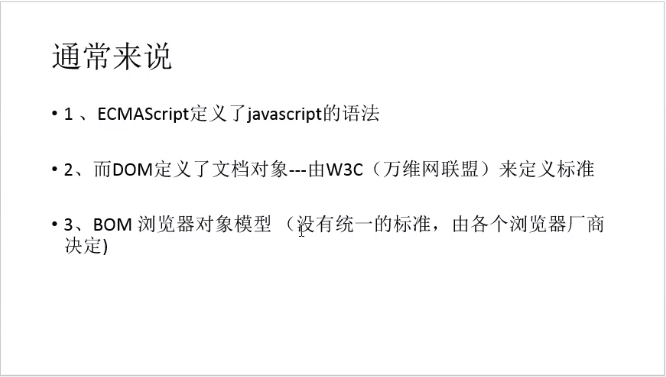
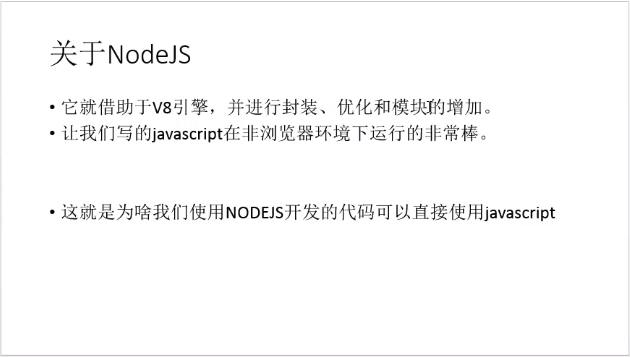
1. 简单说一下JavaScript；

JavaScript可以专门下载一个引擎，这个引擎可以像运行控制台程序一样去运行JavaScript，Nodejs就是一个是一个基于 Chrome V8 引擎的 JavaScript 运行环境；JavaScript并不是一定要在浏览器运行，有引擎就够了；

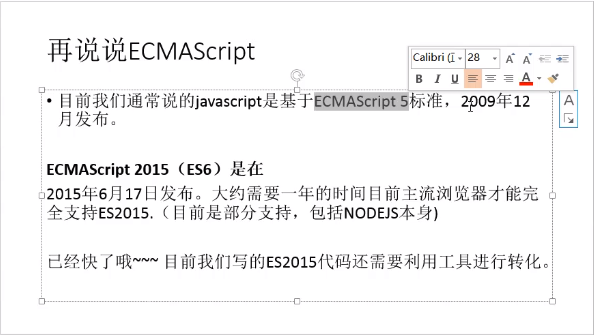




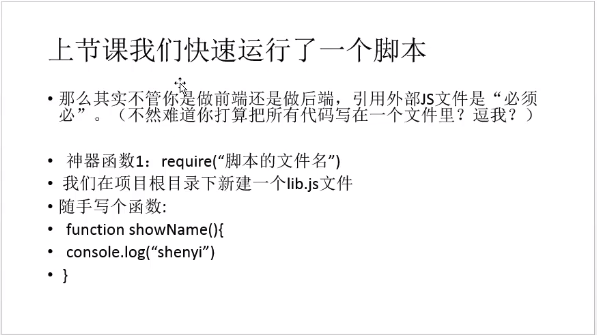
1. JavaScript包含以下三点；Nodejs处理的是ECMAScript；

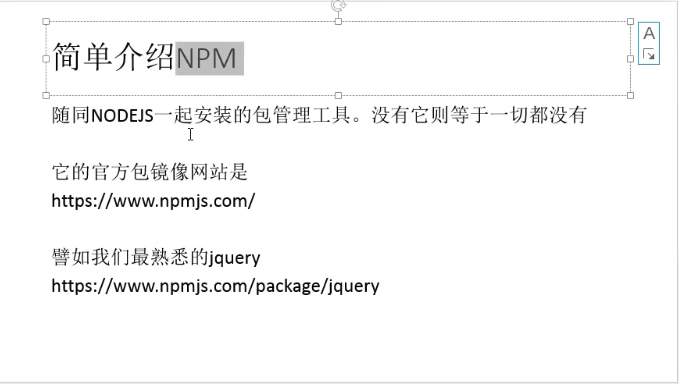
1. ECMAScript；



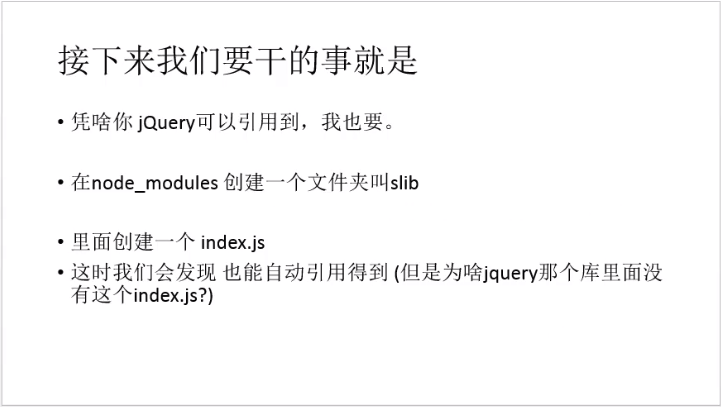
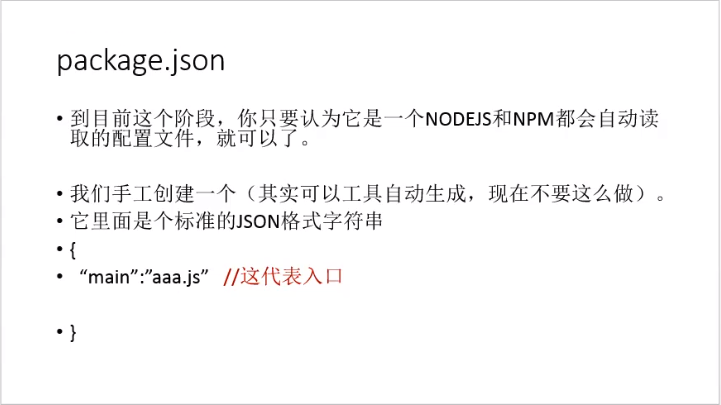
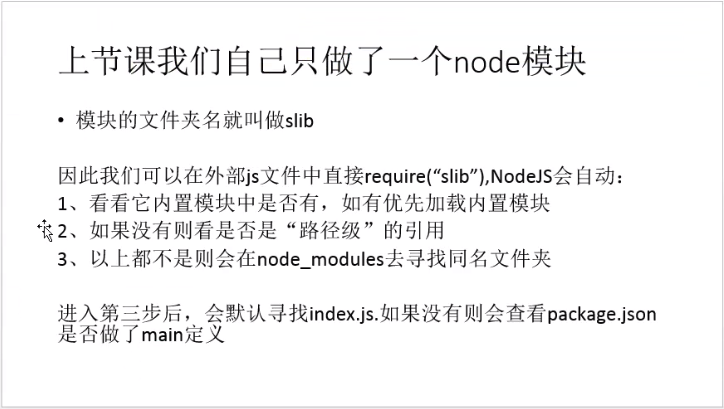
1. 引用外部文件；

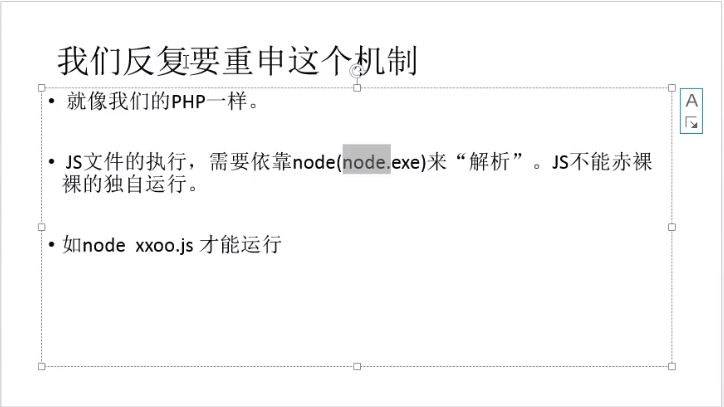
1. npm包管理工具；

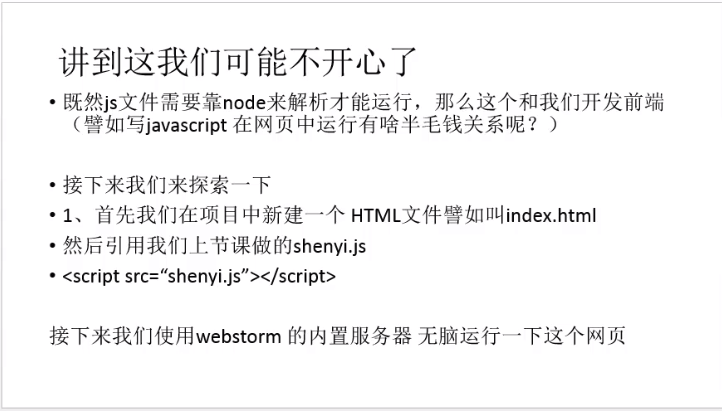
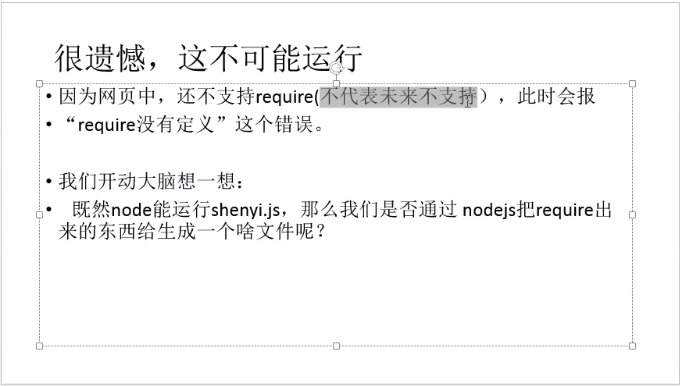


1. node模块理解；

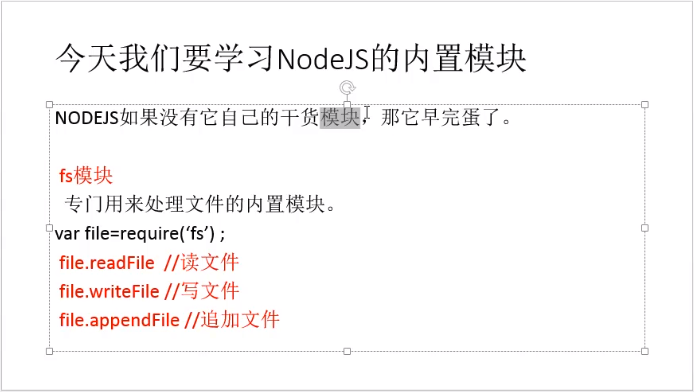
  

1. ；



1. Nodejs内置模块；了解编译思路；

1. ；
2. ；
3. ；
4. ；
5. ；
6. ；
7. ；
8. ；

# ；webpack深入与实战

1. <https://segmentfault.com/a/1190000006178770> ；
2. 1
3. 1
4. 1
5. 1
6. 1