## 课程简介

## 学习准备

### 模块化(1)

### 模块化(2)

### 环境准备(mac)

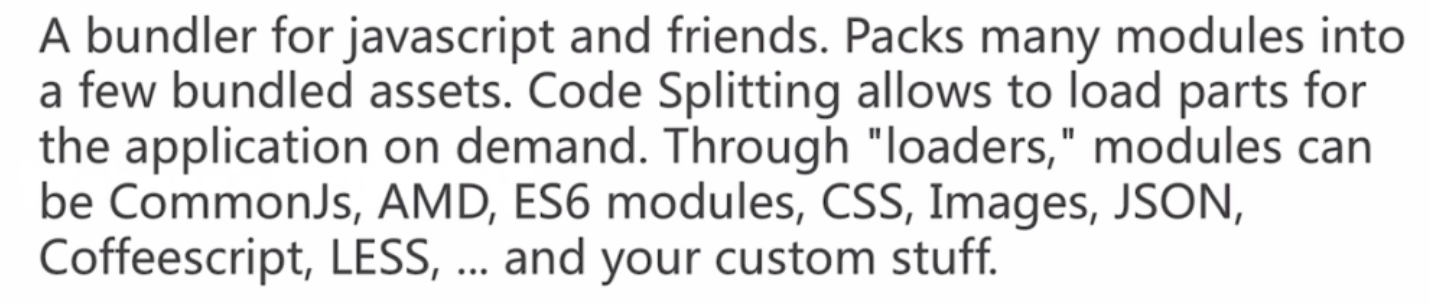
主要就是安装 webpack

**版本是3.10.0**

***npm install webpack@3.10.0 -g***

### webpack 简介

》》下面是官网对 webpack 的一个介绍



》》新官网地址

*https://webpack.js.org*

目前的版本是3.8.1

github 地址：

*https://github.com/webpack/webpack*

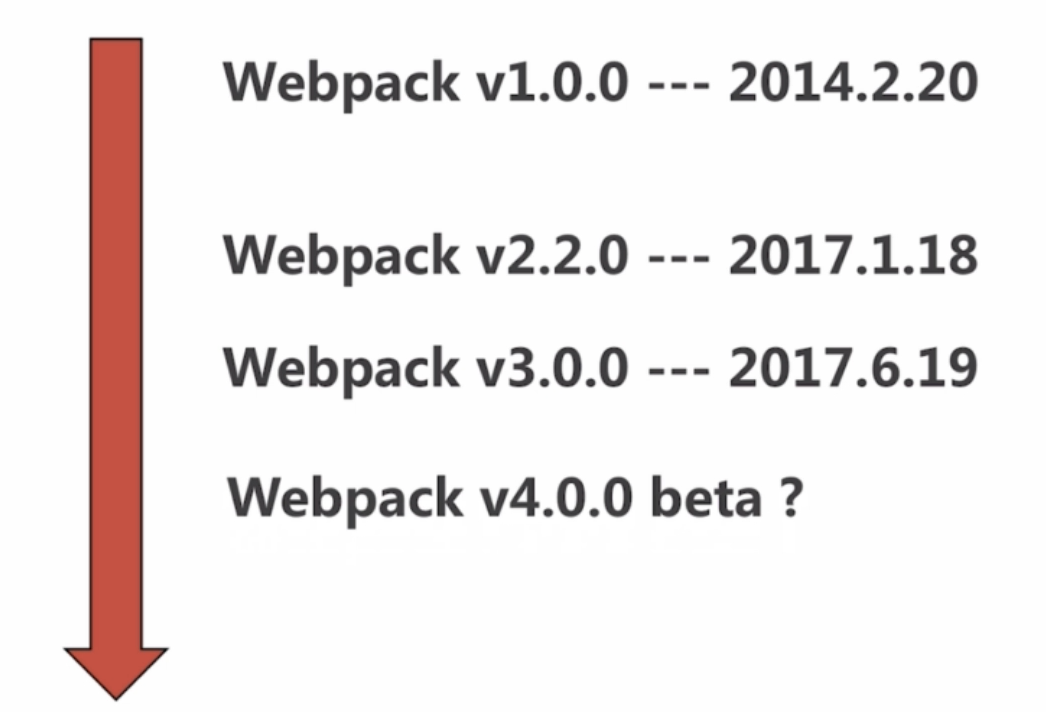
中文的官网：

*https://doc.webpack-china.org*

》》webpack 版本更迭

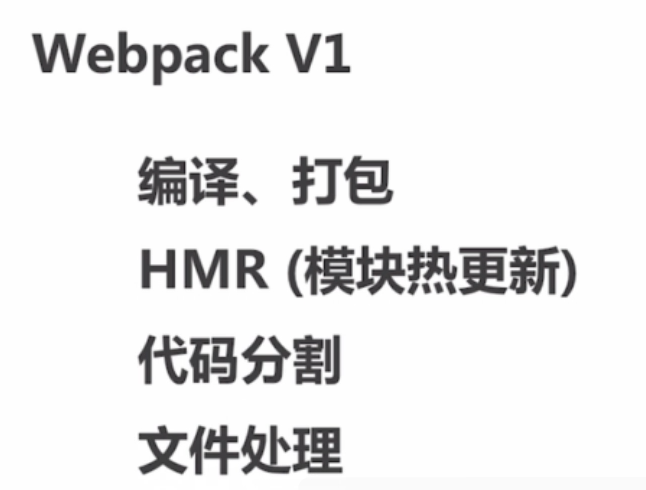
*https://github.com/webpack/webpack/release*

》》大版本变化



讲课的时候 4.0还没有发布正式版

》》功能进化





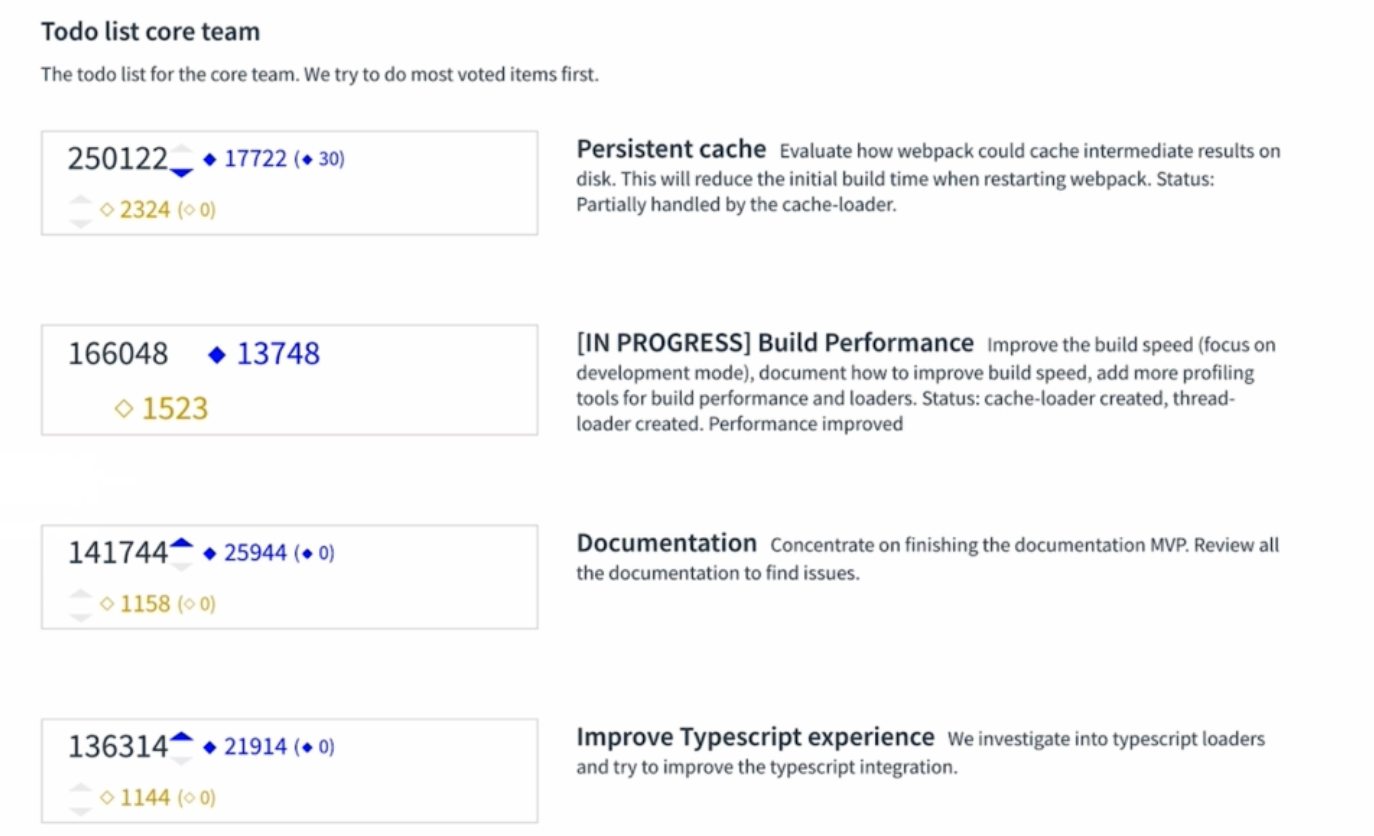


》》版本迁移



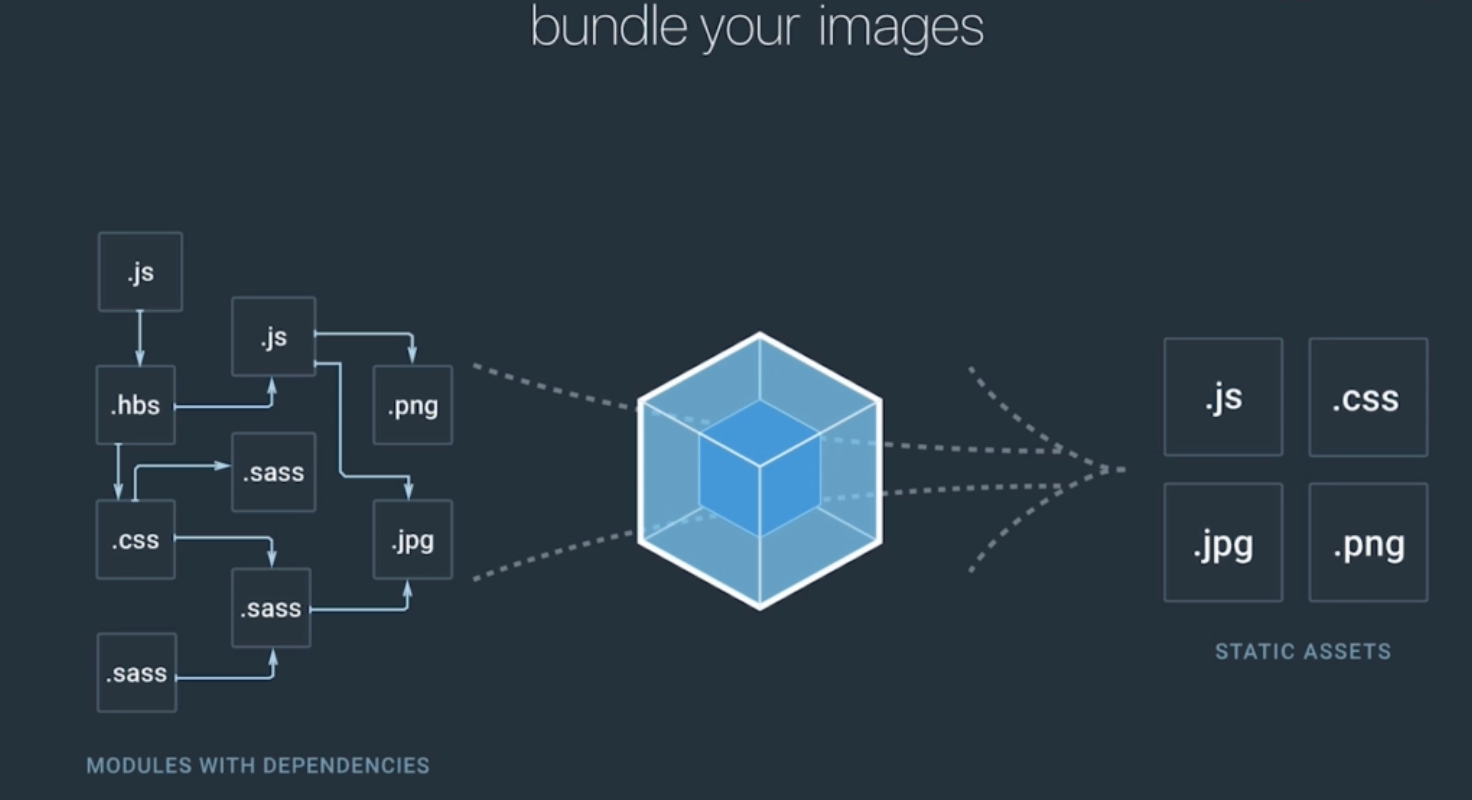
》》参与社区投票

可以看见核心团队的 TodoList



vote: https://webpack.js.org/vote

》》新官网



### webpack 核心概念

从 1.0 版本开始的核心概念就没有变过

四个：**Entry**、**Output**、**Loaders**、**Plugins**

1、Entry

主入口，理解某一段**代码的入口**，需要告诉 webpack，import 了什么，有什么初始化的代码，可以间接的找到依赖的一些模块

其实也是**打包的入口**，依赖找依赖。找全所有的

Entry 可以是**一个**入口，也可以是**多个入口**的。多入口：多页面的应用 ，单页面的应用读到的代码是一个 Entry，框架的是一个 Entry

module.exorts 是一个对象，就是 webpack 的对象，里面有一个 entry 的属性，这个属性的值可以是路径，也可以是单独的 js 文件。这个值也是可以是数组的。

|  |
| --- |
| module.exports{  entry: 'index.js'  }  module.exports{  entry: ['index.js', 'vendor.js']  }  // 这里的每个 entry 是有名字的，就会知道是哪个 entry，推荐这种写法的  // 想要继续添加一个新的入口，可以灵活的添加  module.exports{  entry: {  index: 'index.js',  vendor: 'vendor.js' // 添加新的入口  }  } |

2、Output

表示输出

**打包生成的文件的 bundle** 的一个描述

也有是**一个或多个**

当多个的时候**自定义规则**，这个 output 对应的规则名

代码演示：

|  |
| --- |
| module.exports={  entry: 'index.js'  output: {  filename: 'index.min.js' // 指定输出的文件名  }  }  module.exports={  entry: {  index: 'index.js',  vendor: 'vendor.js' // 添加新的入口  }  output: {  filename: '[name].min.[hash:5].js' // name 就是 entry 的 name，hash 是 md5 过后的一个字符串，这就是自定义规则的  }  }  // 还可以配合 cdn，后面会讲到 |

3、Loaders

可以处理 js 之外的东西

使用来**处理文件**的

把我们所说的**文件转化成模块**，如下面的代码就是，可以当成 js 的一个模块，引入进来

|  |
| --- |
| module.exports={  **module**: {  rules: {  test: /\.css$/,  use: 'css-loader'  }  }  } |

哪些常用的 loader

》》编译相关的：babel-loader, ts-loader

》》样式相关的：style-loader, css-loader, less-loader, postcss-loader

》》文件相关的：file-loader, url-loader

4、Plugins

强大的插件系统

是参与整个打包过程的

打包优化和压缩

配置编译时的变量

极其灵活

|  |
| --- |
| const webpack = require('webpack')  module.exports={  plugins: {  new webpack.optimize.UglifyJsPlugin()  }  } |

常用 Plugins

》》优化相关的：CommonsChunkPlugin，UglifyjsWebpackPlugin

》》功能相关的：ExtractTextWebpackPlugin，HtmlWebpackPlugin，HotModuleReplacementPlugin

CopyWebpackPlugin

》》名词

Chunk：这个是什么了？英文解释就是块，在 webpack 里面就是代码块。

Bundle：英文解释就是绑定，webpack 就是打包的意思了

Module：图片，通过 loader 处理完以后就一个模块

## 由浅入深 webpack(1)

### 使用 webpack

》》使用 webpack 的方式

1、webpack 命令的方式

2、使用 webpack 配置的，平时工作中，多使用这个方式，配合 node，npm 一起使用

3、第三方脚手架，很常见的，如 vue-cli。等等， vue 等这些框架都会有

下面就是详细的讲解：

第一种： webpack 命令

webpack -h ：帮助

webpack -v ：版本

webpack <entry>[<entry>] <output> : 打包命令

下面介绍一个很厉害的，很惊讶的东西，**webpack-cli**

***npm install webpack-cli@3.3.10 -g***

这是额外插入的介绍的： webpack-cli 不过最后也没怎么用这个东东

1、可以创建一个交互式的初始化项目，后面填写的是模板的名称

webpack-cli init wepack-addons-demo

2、迁移项目：v1 -> v2

webpack-migrate 迁移文章，就进行迁移了

刚开始还是很惊艳的，很激动的，但是细细一想，或者脚手架的确实，也没有官方的脚手架，所以还是慢慢丰富，所以 cli 的功能还是不太好用。

目前来看，也很久没更新了，v4 版本都是alpha。

第二种： webpack 配置

webpack

webpack --config webpack.conf.dev.js

可以不指定配置文件，就会使用默认的配置文件，也可以手动指定配置文件

第三种： 使用第三方的脚手架

vue-cli,angular-cli,react-starter

### 打包 JS

## 由浅入深 webpack(2)