### Webpack

#### 1.模块打包工具

Web应用开发规模太大，人工维护代码成本难以承受，可以让开发效率成倍的提升。

模块打包工具的任务就是解决模块间的依赖，使其打包后的结果能在浏览器上运行，他的工作方式主要分为两种：

1. 将存在依赖关系的模块按照特定规则合并为单个JS文件，一次性全部加载进页面中。
2. 在页面初始时加载一个入口模块，其他模块异步的进行加载。

目前比较流行的模块打包工具有Webpack、Parcel、Rollup等。

#### 2.Webpack概述

Webpack是一个开源的JavaScript模块打包工具，和同类模块打包工具，Webpack具备以下几点优势：

1. Webpack默认支持多种模块标准，包括AMD、CommonJS，以及最新的ES6模。
2. Webpack有完备的代码（code spliting）分割解决方案。可以分割打包后的资源，按需加载，提升渲染速度。
3. Webpack可以处理各种类型的资源（样式、模板、图片）。
4. Webpack拥有庞大的社区支持。

#### 3.Webpack安装

安装模块的方式有两种：全局安装、本地安装

与他人进行项目协作时，每个人的Webpack版本不同，导致输出的结果不一样。并且部分依赖于webpack的插件会调用项目中webpack内部模块，这种情况下仍需要在本地安装。

从命令行进入安装目录，执行npm的初始化，生成npm的说明书，记录项目名称、版本、仓库地址信息等：

npm init

执行安装webpack命令，webpack是核心模块，webpack-cli是命令行工具：

npm install webpack webpack-cli --save-dev

查看版本号：

npx webpack-v

#### 4.打包之前的知识点。

Webpack五大核心概念：

1. Entry：指示webpack以哪个文件作为入口起点开始打包。
2. Output：指示webpack打包后的资源bundle输出到哪里去，以及如何命名。
3. Loader：让webpack去处理那些非JavaScript文件（webpack自身只理解JavaScript）。
4. Plugins：插件可以用与执行范围更广的任务（包括打包优化和压缩），一直到重新定义环境中的变量。
5. Mode：指示webpack使用相应的模式。

development（开发模式）能让代码在本地调试运行的环境。

production（生产模式）能让代码优化上线运行的环境。

|  |  |  |
| --- | --- | --- |
| 打包 | 下载安装 | 命令行 |
| 样式资源 | loader包 | npm i css-loader style-loader less-loader less -D |
| HTML资源 | plugin包 | npm istall –-save-dev html-webpack-plugin |
| 图片资源 | loader包 | npm install –-save-dev html url-loader file-loader |
|  |  |  |

为了满足不同应用场景的需求，我们可以把webpack配置项以及相对应的参数放在webpack.config.js文件里，在Webpack每次打包的时候读取该配置文件即可知道怎么工作。

所有的构建工具都是基于node.js平台运行的，模块化默认采用CommonJS。

通常情况会设置源码目录与资源输出目录。工程源代码放在/src中，输出资源放在/dist中。

Webpack社区已经为我们提供了一个便捷的开发工具——webpack-dev-server。使用以下命令安装：

npm install webpack-dev-server --save-dev

//其中--save-dev参数是将webpack-dev-server作为工程的devDependencies(开发环境依赖)记录到package.json中。

Webpack-dev-server的两大职能：

1. 令webpack进行模块打包，并处理打包结果的资源请求。
2. 作为普通的web server，处理静态资源文件请求。

直接使用webpack开发和webpack-dev-server有一个很大的区别，前者每次都会生成一个budnle

#### 二级标题

##### 三级标题