**前端主干**

* **前端工程化工具**

作用

模块化文件的组织、ES6 JS 文件的编译、打包压缩所有的 JS 代码、优化和合并图片静态资源

前端工程化，就是在使用工具处理那些与实际业务无关的内容，比如处理 JS 编译、打包、压缩、图片合并优化等等各个方面的工程性代码

具体类目

包管理工具

npm

node自带

yarn

源代码静态检查和格式化工具

eslint

通过配置 .eslintrc (runtime config) 就可以使用了, 配合 eslint 的编 辑器插件，我们就可以在编辑代码时 eslint 对我们的代码进行提示和修复。

prettier

.prettierrc 里面也可以 进行配置，最终搭配 prettier 的 编辑器插件，我们同样能够实现代码编辑状态下的提示和修复。

ES6 及其他泛 JS 语言的编译

我们不能直接在线上使用 ES6 语法规范的 JS 代码，我们就需要通过工具对 JS 进行编译. 有些项目我们可能会使用 coffeescript，typescript，flow，elm，ocaml 等可以编译为 JS 语言 的泛 JS 语言书写代码，这就需要在调试或发布时，使用编译工具将对应代码编译为 JS 代码才能直接运行。

编译工具：

babel

而其他的语言则对应有自己的编译器，例如 coffeescript 使用 coffeescript 编译器进行编译为 js，typescript 使用 typescript 编译器编译为 js。

js压缩工具

uglify

流式处理工具

grunt

通过 gruntfile.js 中的配置来让 grunt 做不同的操作

gulp

这是一类脚本工具，通过写脚本的形式将不同的工具安排执行流程

通用处理工具

通用处理工具这里我们的分类是从功能上来讲，具备上面列举类目的多种功能的集合。他们有个最大的特点就是，他们已经不再是一个普通工具，而是一个具有插件化的系统，有着丰富和完善的社区环境，他们属于前端解决方案 这么一个领域。理论上他们可以做非常多的事情，而不像上面介绍的大部分工具，只能处理某一个垂直分类下的内容。

有哪些

webpack

webpack 通过 webpack.config.js 配置

Webpack 提供丰富配置和插件，需复杂配置，先打包再运行

vite

Vite 开发时不打包，直接用 ES Module 运行，部署时打包，开箱即用

* **主题**
* **主题**
* **主题**