# webpack干货总结

# 前言：

html dom以及html5控件拖拽总结  
1.<script>标签可以放在任何一个元素里面  
2.返回服务器的域名：document.domain  
3.使用锚点来定位页面位置的时候，name属性只能适用于a标签，其他标签使用id属性  
4.返回当前页面的完整url：document.URL（URL是大写）  
5.html顶部加了<!DOCTYPE html>声明之后，在body标签上加事件，点击空白地方无法触发，只能点击在document元素上才能触发  
6.拖拽对象：DataTransfer——effectAllowed和dropEffect通常应用的事件方法名不一样，effectAllowed多用在dragstart事件中，而dropEffect属性的设置主要用在dragenter和dragover事件中；effectAllowed和dropEffect的彼此间是有制约关系，当我们给effectAllowed设置了对应的属性值，则dropEffect只能设置为effectAllowed允许的值，否则是无效的。  
7.DataTransfer对象出现在拖拽事件中，具体包括开始拖拽dragstart事件，拖拽进入dragenter事件，拖拽离开dragleave事件，拖拽经过dragover事件，拖拽释放drop事件以及拖拽结束dragend事件。  
8.document元素的属性可以直接通过document.getElementById("#id").style.color="blue" 这种方式修改

## webpack细节解释

1. loader 支持链式传递。能够对资源使用流水线(pipeline)。一组链式的 loader 将按照相反的顺序执行。loader 链中的第一个 loader 返回值给下一个 loader。在最后一个 loader，返回 webpack 所预期的 JavaScript。  
   2.webpack.config.js 里面的resolve对象用来生成解析说明，其中alias用来创建路径别名，alias{xyz$:path.resolve(\_\_dirname, 'path/to/file.js')}，对象的键后的末尾添加 $，以表示精准匹配；import Test1 from 'xyz'; // 精确匹配，所以 path/to/file.js 被解析和导入；import Test2 from 'xyz/file.js'; // 非精确匹配，触发普通解析  
   3.webpack.config.js 的target对象，使用 node webpack 会编译为用于「类 Node.js」环境（使用 Node.js 的 require ，而不是使用任意内置模块（如 fs 或 path）来加载 chunk）。module.exports = {target: 'node'};可以通过打包两份分离的配置来创建同构的库  
   4.npx webpack --config webpack.config.js 通用config命令可以传递任何名称的配置文件  
   5.配置步骤： 1、创建工程目录； 2、初始化工程目录：npm init -y。 3、全局安装webpack-cli。4、全局安装webpack。5、webpack –mode development/production进行打包，可在package.json中配置dev和build的脚本，便只需运行npm run dev/build，作用相同。 6、在webpack –mode development/production可串联设置其他参数。  
   6.注意事项： 1、webpack-cli必须要全局安装，否则不能使用webpack指令； 2、webpack也必须要全局安装，否则也不能使用webpack指令。 3、webpack4.x中webpack.config.js这样的配置文件不是必须的。 4、默认入口文件是./src/index.js，默认输出文件./dist/main.js。

## webpack相关配置

1.html-webpack-plugin这个插件可以让webpack自动输出一个index.html页面。不用手动再去添加绑定  
2.clean-webpack-plugin这个插件可以删除自动输出目录下的缓存文件，但是最新用法跟webpack描述有出入：  
const { CleanWebpackPlugin } = require("clean-webpack-plugin");  
plugins: [  
 new CleanWebpackPlugin()  
]  
3.使用devtool: 'inline-source-map' 选项可以让我们可以在浏览器中调试源文件  
4.webpack --watch可以在命令行窗口打开的时候，自动检测文件的改变保存时，主动编译  
5.webpack-dev-server --open可以在 localhost:8080 下建立服务，将 dist 目录下的文件，作为可访问文件，如果修改和保存任意源文件，web 服务器就会自动重新加载编译后的代码  
7.创建library,需要在output里面表明使用对象libraryTarget，可以通过以下方式暴露 library：  
变量：作为一个全局变量，通过 script 标签来访问（libraryTarget:'var'）。  
this：通过 this 对象访问（libraryTarget:'this'）。  
window：通过 window 对象访问，在浏览器中（libraryTarget:'window'）。  
UMD：在 AMD 或 CommonJS 的 require 之后可访问（libraryTarget:'umd'）。  
8.在webpack.config.js里面添加externals对象，表明当前library的相关依赖

## 三、webpack优化相关

1.Vue Loader：此 loader 支持用于 vue 组件的 HMR，提供开箱即用体验。

2.设置mode为production会让webpack自动启用 uglifyjs 压缩插件

3.在webpack的配置文件中设置了mode之后，process.env.NODE\_ENV的值会被设置成相应的值，这个对象可以在脚本中作为全局对象使用

4.提取共享模块：

optimization: {

splitChunks: {

cacheGroups: {

commons: {

name: "commons",

chunks: "initial",

minChunks: 2

}

}

}

},

官方文档的 new webpack.optimize.CommonsChunkPlugin()是会报错的的

5.懒加载，第一次打开页面的时候不会去加载对应的js资源文件，当触发事件的时候才去加载资源文件， webpackChunkName 这个注释的作用是命名懒加载的资源文件，否则会按照数字递增命名