

06.Loader 打包图片文件

学习要点：

- 1.打包思路
- 2.安装部署

本节课我们来开始学习一下在 `module` 下设置打包图片文件的功能。

一. 打包思路

1. 本节课，我们继续打包文件：图片文件，分为 `css` 引入图片和 `html` 插入图片；
2. 如果是 `css` 加载图片，都是背景图，总所周知小图片采用 `base64` 转换字符串；
3. 而大图片和 `html` 插图图片，则都需要单独的 `loader` 插件来完成；
4. 本节课用到的 `loader` 插件如下：
 - (1) `.file-loader`：解析 JavaScript 和 `css` 插入的图片；
 - (2) `.url-loader`：将图片转换为 `base64` 编码字符串；
 - (3) `.html-loader`：将 `.html` 进行打包，从而解析 `img` 插入问题；

二. 安装部署

1. 首先，先研究第一个插件：`file-loader`，安装并配置；

```
npm i file-loader -D
```

```
{
  test: /\..(png|jpg|gif)/,
  use : ['file-loader']
}

#loading {
  width: 780px;
  height: 422px;
  background-image: url("./loading.gif");
}

#ts {
  width: 609px;
  height: 609px;
  background-image: url("./ts.png");
}
```

PS：也可以在 `.js` 文件引入图片，会被自动解析；

```
const img = require('./ts.png');
console.log(img);
```

2. 现在，我们要扩展这个插件内容，要求自定义目录和名称：

```
{
  test: /\..(png|jpg|gif)/,
  loader: 'file-loader',           //加载一个 loader
  options: {
    name: './img/[name].[ext]',   //写入 img 目录，且按原名和后缀，
  },
},
```

3. 第二个插件，我们限定小于 10kb 的，直接用 base64 编码保存图片；
4. 具体为何要用这么做：扩展要探讨很多，直接搜索相关关键字了解；
5. 总结就是一句话：小图片用 base64 性价比收益最高，反之一样；

```
npm i url-loader -D
```

```
{
  test: /\..(png|jpg|gif)$/ ,
  //这里使用 url-loader，它依赖 file-loader
  //通过先判断图片大小，小的采用 base64，大的采用 file-loader
  loader: 'url-loader',
  options: {
    limit: 10240,                 //限定 10kg 一下采用 base64 编码，
    name: './img/[name].[ext]'
  },
},
```

6. 第三种，就是插入到 html 的图片，需要使用 html-loader 进行解析；

```
npm i html-loader -D
```

```
{
  test: /\.html$/,
  use: ['html-loader']           //自动交给 url-loader 处理
},
```