一、.d.ts是干嘛的

.d.ts文件是ts用来声明变量，模块，type，interface等等的，那在这种后缀的ts文件声明这些东西和在纯ts文件声明这些东西又什么区别呢？

在.d.ts声明变量或者模块等东西之后，在其他地方可以不用import导入这些东西就可以直接用，用，而且有语法提示。

但是也不是说创建了.d.ts文件，里面声明的东西就能生效了，毕竟归根到底也是.ts文件，需要预编译，所以需要在tsconfig.json文件里面的include数组里面添加这个文件

include数组里面可以不用写.d.ts文件的绝对路径，可以通过glob通配符，匹配这个文件所在的文件夹或者是“祖宗级别”文件夹。

支持的glob通配符有：

\* 匹配0或多个字符（不包括目录分隔符）

? 匹配一个任意字符（不包括目录分隔符）

\*\*/ 递归匹配任意子目录

具体tsconfig.json配置请看这tsconfig.json · TypeScript中文网 · TypeScript——JavaScript的超集

二、declare是干嘛的

.d.ts 文件中的顶级声明必须以 "declare" 或 "export" 修饰符开头。

通过declare声明的类型或者变量或者模块，在include包含的文件范围内，都可以直接引用而不用去import或者import type相应的变量或者类型。

1.declare声明一个类型

declare type Asd {

name: string;

}

在include包含的文件范围内可以直接使用Asd这个type

2.declare声明一个模块

最经典的声明模块应该是这样了

declare module '\*.css';

declare module '\*.less';

declare module '\*.png';

在编辑ts文件的时候，如果你想导入一个.css/.less/.png格式的文件，如果没有经过declare的话是会提示语法错误的

3.declare声明一个变量

这个什么情况下会用到呢？假如我在项目中引入了一个sdk，这个sdk（我们以微信的sdk为例）里面有一些全局的对象（比如wx），但是如果不经过任何的声明，在ts文件里面直接用wx.config()的话，肯定会报错。

有一句说法我蛮喜欢的：declare就是告诉TS编译器你担保这些变量和模块存在，并声明了相应类型，编译的时候不需要提示错误！

4.declare声明一个作用域

declare namespace API {

interface ResponseList {}

}

声明完之后在其他地方的ts就可以直接API.ResponseList引用到这个接口类型

三、注意

1..d.ts文件顶级声明declare最好不要跟export同级使用，不然在其他ts引用这个.d.ts的内容的时候，就需要手动import导入了

2.在.d.ts文件里如果顶级声明不用export的话，declare和直接写type、interface效果是一样的，在其他地方都可以直接引用

declare type Ass = {

a: string;

}

type Bss = {

b: string;

};

可以直接使用Ass和Bss作为某个变量的类型

————————————————

版权声明：本文为CSDN博主「曩昔方」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_34551390/article/details/118800743