# 1.趋势

## 以后前端开发用ts会越来越多，ts的语法比较严格，对我们开发来说反而更好，可以在编写代码的时候就知道错误。

# 2.搭建ts开发环境

## 1.全局安装ts，注意最好以管理员权限打开cmd，否则安装失败

|  |
| --- |
| npm i -g typescript |

## 2.查看ts版本号：tsc -v

|  |
| --- |
|  |

# 3.编写ts代码

## 我们新建一个文件夹：ts-node-projs，然后用webstorm打开，新建一个tsdemo1.ts文件

|  |
| --- |
|  |

## 编译ts文件：tsc 文件名.ts,会输出同名的js文件

|  |
| --- |
|  |
|  |

## 然后就可以使用node 文件名.js来运行

## 每一次都这么做比较烦，有一种自动编译的方法，步骤如下:

### 1.生成配置文件

我们先把新生成的js文件删除，在终端中定位到ts文件所在的路径，然后运行tsc –init，然后就会在这个路径下面生成一个tsconfig.json文件

|  |
| --- |
|  |

### 2.配置tsconfig.json文件，打开outDir的注释，可以根据需要修改路径

|  |
| --- |
|  |

### 3.启动ts监视器

#### 如果你是使用vscode编辑器，你可以点击终端菜单-运行任务-点击typescript-选择tsc-监视tsconfig.js,然后终端会启动一个进程一直阻塞在那里，其实他现在已经把ts编译成js了，以后一旦修改了ts文件它会自动编译，还有你可以安装一个叫做Error lens的创建来在错误的代码后面显示错误信息

#### webstorm的配置：

#### 点击file-setting，在弹出的对话框中点击language&frameworks-typescript,在右边把2个复选框都勾上

|  |
| --- |
|  |

#### 然后点击tools-File Watchers，在右边点击+，选择选择custom，此时会弹出一个窗口

|  |
| --- |
|  |

#### 输入如下信息，点击ok

**Name:TypeScript**

**File Type:TypeScript**

**Scope:All Places**

**Program:** **C:\Program Files\nodejs\node\_modules\typescript\bin\tsc（也就是安装typescript的路径下的tsc文件）**

**Arguments:--sourcemap --target "ES5"**

**Output paths to refresh:$FileNameWithoutExtension$.js:$FileNameWithoutExtension$.js.map**

**WOrking directory:$FileDir$**

|  |
| --- |
|  |

#### 点击ok，然后我们编辑一下ts文件，它就会自动编译为js文件

# 需要注意的是，上面的配置只是为了学习ts的。真正的项目开发中是不需要这么做的因为项目的构建工具会帮我们做这些工作，但是全局安装ts是需要的