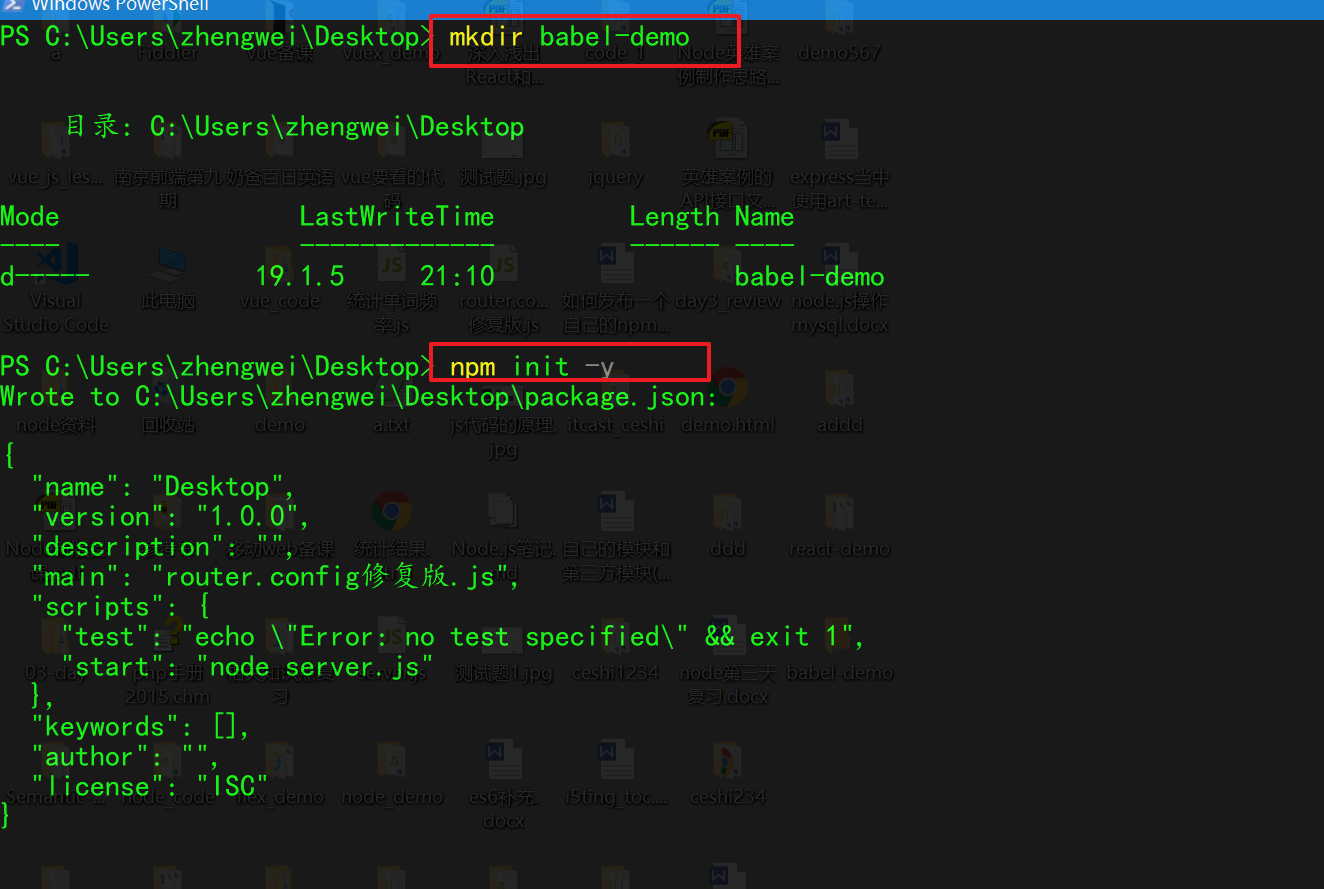
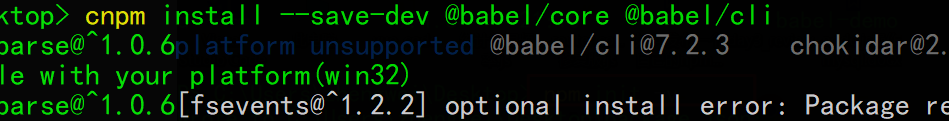
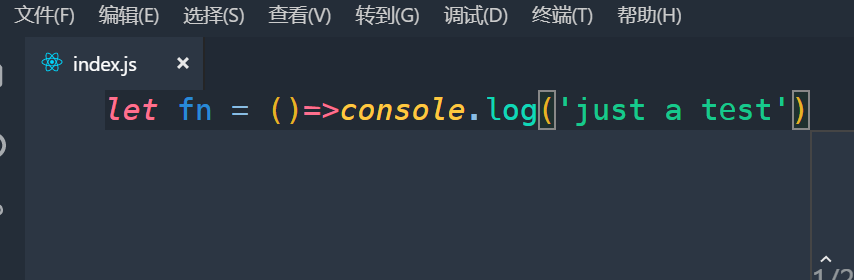
第一步、创建项目文件夹，初始化



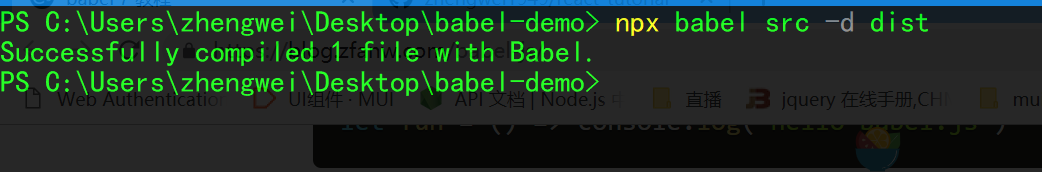
第二步、安装babel/core babel/cli



第三步、创建src目录，创建index.js代码



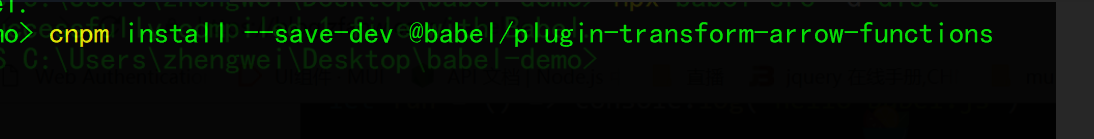
第四步、执行编译



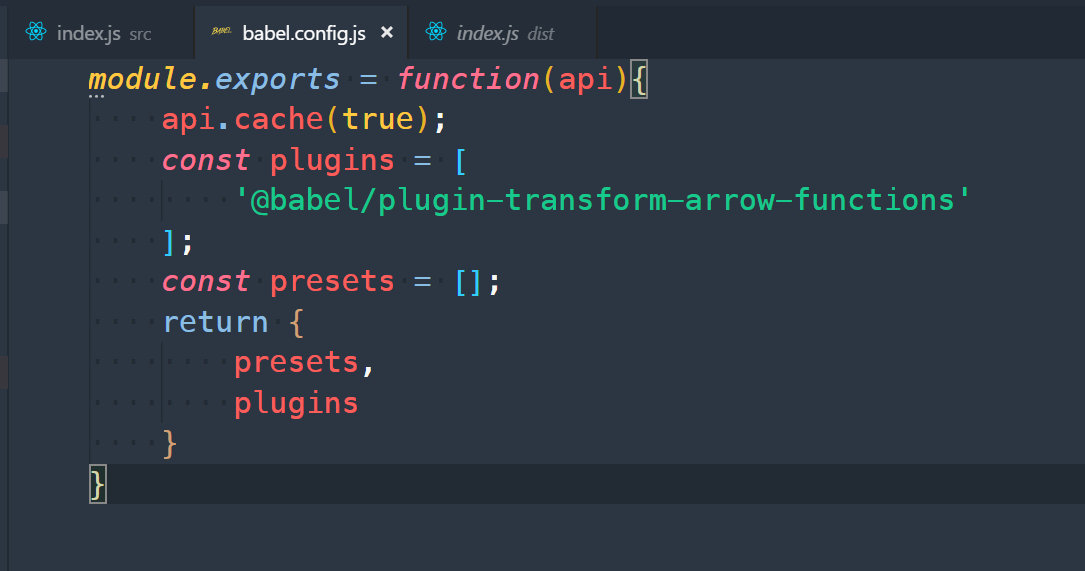
这时候，大家会发现，dist中的代码并没有任何区别，这是因为babel/core默认不会引任何的插件，如果你想编译，还必须要装相应的插件

https://babel.docschina.org/docs/en/plugins

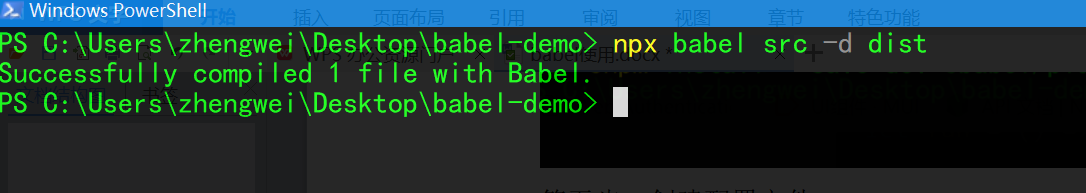
我们用到了箭头函数，所以必须安装对应的插件



第五步、创建配置文件babel.config.js

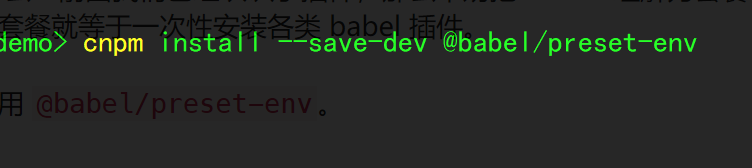


第六步、再次编译成功

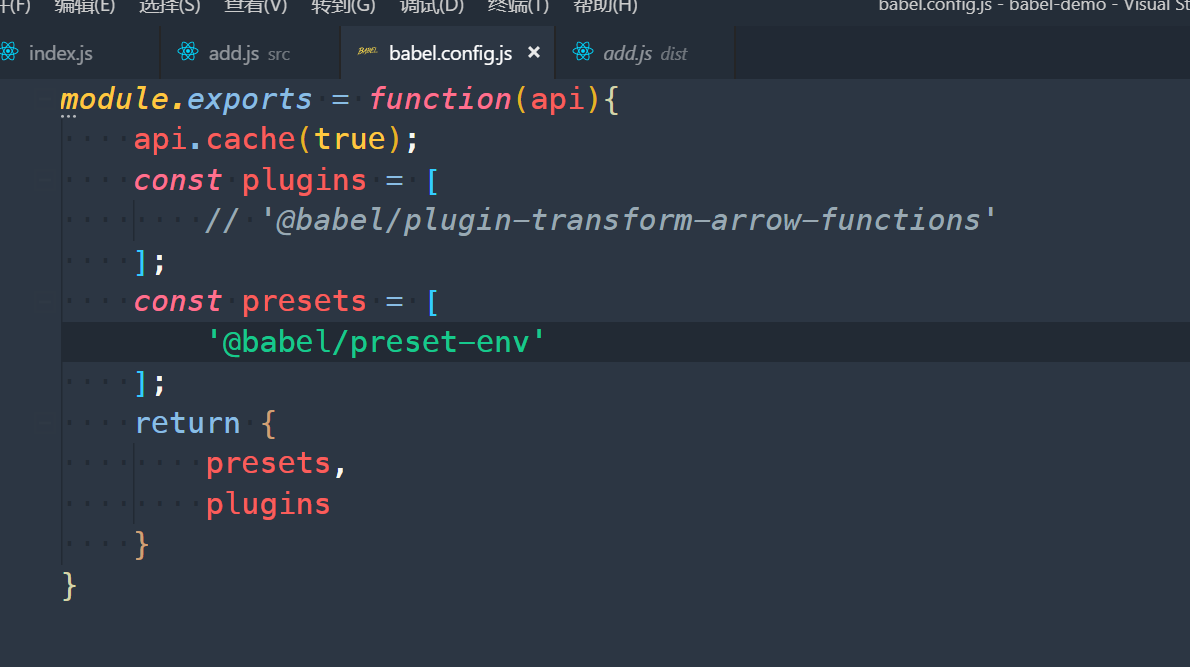


如果我们用到一个es6的新特性，就必须安装插件，太麻烦了， 所以我们可以考虑使用presets

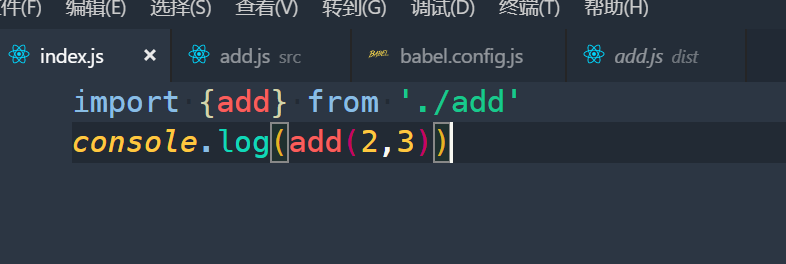
第七步、



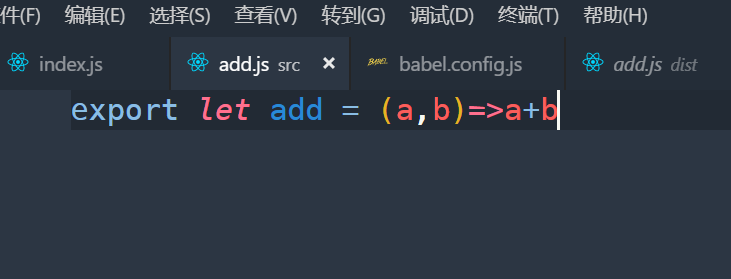
接下来，修改配置文件



index.js修改为



添加add.js



编译之后，我们发现这里用到的还是require方法，在浏览器端并不支持



cnpm install --save @babel/runtime

浏览器兼容性的配置



|  |
| --- |
| module.exports = function(api){  api.cache(true);  const plugins = [  [  "@babel/plugin-transform-runtime"  ]  ];  const presets = [  [  '@babel/env',  {  "targets": {  "browsers":[  // "chrome 59"  "ie 10"  ]  },  "modules": 'amd',  "useBuiltIns": 'usage', // 是否开启自动支持 polyfill  }  ]  ];  return {  presets,  plugins  }  } |