

开始一个项目的构建：

- 1、确定项目中有哪些组件，如[用户的登录](#)，[注册](#)等
- 2、创建组件

使用命令来完成创建：**ng g c 组件名** 如：ng g c login

这个命令会生成四个文件：新组件的html、ts、测试ts、css

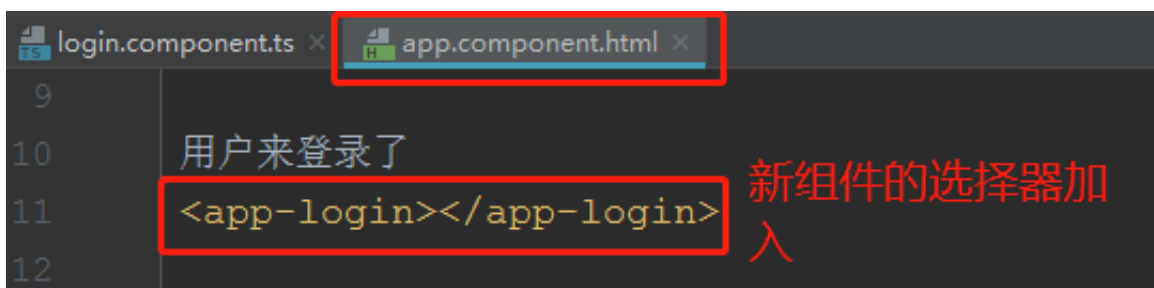
- 3、注册组件

在app.module.ts的 **declarations**配置中 注册新创建的组件，默认用命令创建的组件会自动注册

```
declarations: [  
  AppComponent,  
  LoginComponent  
],
```

- 4、使用渲染新组件页面

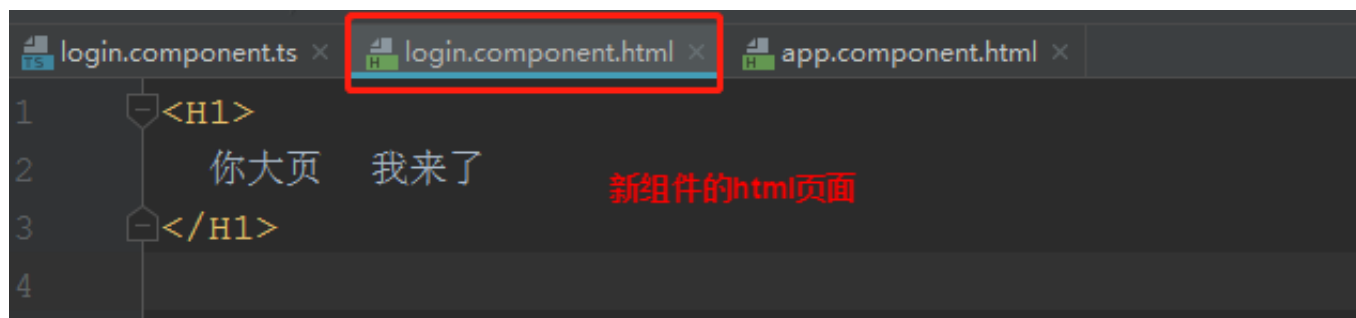
在app.component.html页面中加上新组件的选择器即可



新组件的选择在：login.component.ts文件中



加入选择器后真正渲染出的页面是新组件的html页面

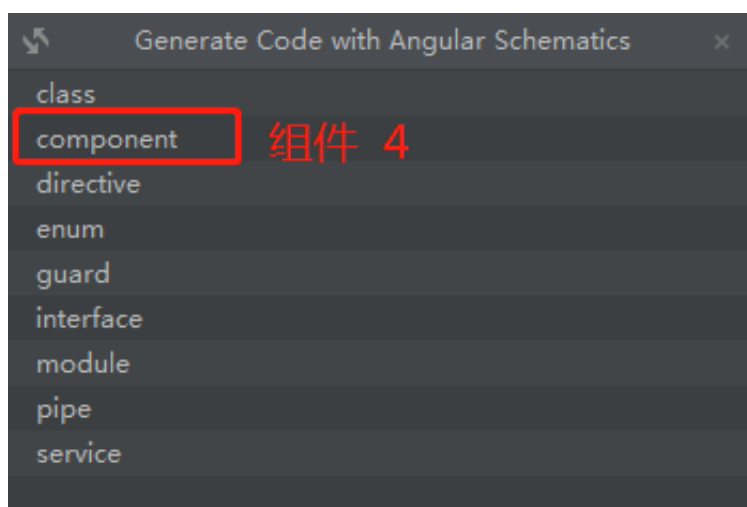
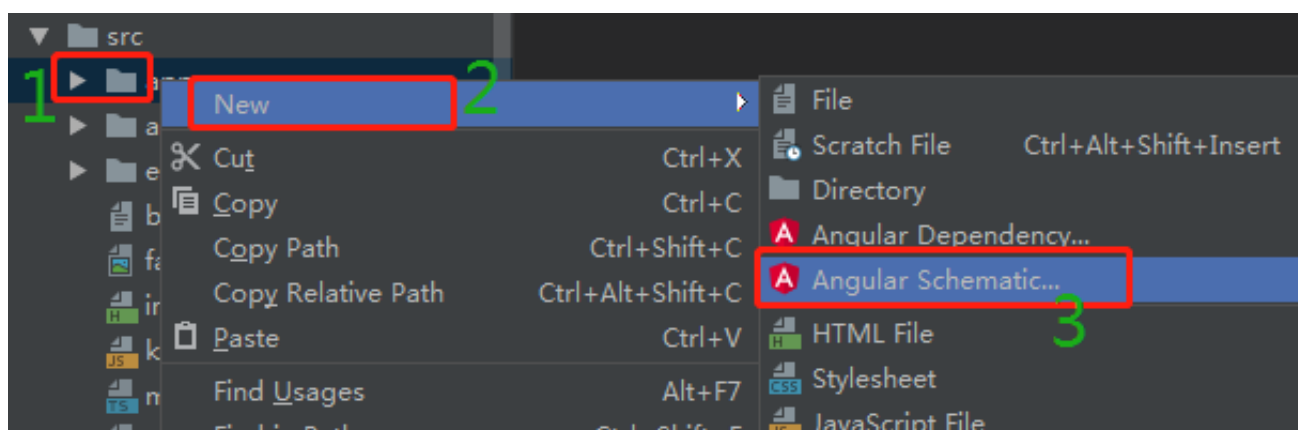


创建新组件的可视化方法:

在项目中

选中app目录，右键new----->Angular Schematic----->选择组件
component----->

回车----->输入组件名----->回车



路由：用于在不同视图之间导航

The screenshot shows the Kaikba.com login and registration interface. At the top is the logo and the text '互联网人学习成长社群'. Below it are two links: '登录' (Login) and '注册' (Register), both enclosed in a red rectangular box. Below these links are two input fields: '请输入注册时的手机号码' (Please enter the mobile number used for registration) and '请输入密码' (Please enter password). Below the password field is a checkbox for '7天自动登录' (7-day automatic login) and a link for '忘记密码' (Forgot password). At the bottom is a large blue button labeled '登录' (Login). Red annotations are present: '如 页面中有注册和登录两个功能' (If there are two functions, registration and login, on the page) above the input fields; '当点击登录 就不能操作 注册' (When clicking login, you cannot operate registration) below the first input field; '这个时候就需要用 路由 来切换' (At this time, you need to use routing to switch) below the second input field; and '组件实现' (Component implementation) below the checkbox.

1、在app.module.ts配置声明路由

```
const routes: Routes = [{path,component},....]
```

The screenshot shows the app.module.ts file in a code editor. The routes configuration is as follows:

```
import {RouterModule, Routes} from '@angular/router';  
// 配置路由 地址是什么的时候 由哪个去处理  
const routes: Routes = [  
  {path: 'login', component: LoginComponent},  
  {path: 'register', component: RegisterComponent},  
];
```

The routes configuration is highlighted with a red rectangular box. There are also some green and red icons on the left side of the code editor.

另外配置路由地址为空以及地址不正确的处理：

// 处理没有匹配的地址重定向到指定的页面

```
{path: '**', redirectTo: '/login'},
```

// 如果path为空时 要加一个pathMatch: 'full', 表示确定地址就是空

后面什么都没有

```
{path: "", pathMatch: 'full', redirectTo: '/login'},
```

2、导入路由配置模块，在app.module.ts的@NgModule的imports中

```
@NgModule({  
  declarations: [  
    AppComponent,  
    LoginComponent,  
    RegisterComponent  
  ],  
  imports: [  
    BrowserModule,  
    // 导入路由配置模块  
    RouterModule.forRoot(routes)  
  ],  
})
```

3、设置路由出口，放入路由插座在app.component.html中（让路由起作用能真正可以渲染正确的页面）

```
app.component.html x  
1  
2  
3 <!--想要看到路由内容，需要放置路由出口-->  
4 <router-outlet></router-outlet>  
5
```

4、路由链接app.component.html中

```
<a routerLink="login" routerLinkActive="active">登录</a>
```

```
<a routerLink="register" routerLinkActive="active">注册</a>
```

```
<!--路由导航链接-->
<div class="logo">
  <a href="http://www.kaikeba.com/">
    
  </a>
</div>
<div class="nav">
  <a routerLink="login" routerLinkActive="active">登录</a>
  <a routerLink="register" routerLinkActive="active">注册</a>
</div>

<!--想要看到路由内容，需要放置路由出口-->
<router-outlet></router-outlet>
```