

Java-Web

一、环境配置

1. Java开发工具-JDK的配置

1.1 版本选择

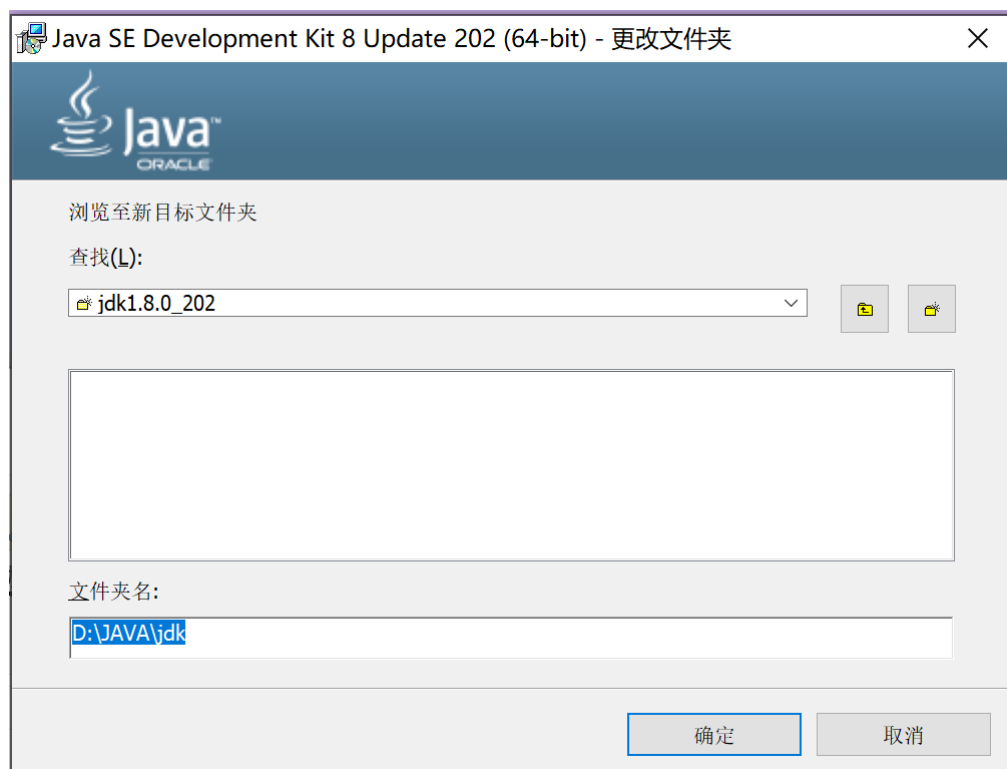
在JAVA-WEB开发中，我们通常使用基于Java8的JDK1.8.0，它是Oracle长期支持的版本，具有稳定和广泛支持的特点，接下来，我们将使用JDK1.8.0配置项目。

1.2 安装JDK

首先我们下载JDK：<https://www.123pan.com/s/sVSxjv-9LSXv.html>

下载完成后，运行 "jdk-8u202-windows-x64.exe" JDK安装程序，要记住JDK的存放路径：

在这里我们推荐在D盘根目录新建一个'JAVA'文件夹，以此为安装路径，如图：



安装JDK后，会提示安装JRE，JRE是JDK的分支插件，建议在JAVA目录下新建一个文件夹，名为 'jre'，如图：



如图安装好后，JAVA的开发工具的结构应该如下：

```
D:  ->  | JAVA  ->  | jdk
        | jre
```

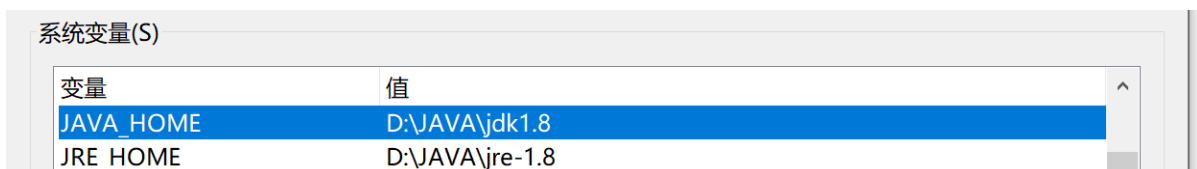
1.3 配置JAVA环境变量

1.3.1 步骤

为了让计算机上所有利用JAVA的开发工具能够找到JDK的位置，我们需要配置系统变量

首先：按下Win键+S，搜索高级系统设置，打开后选择环境变量，在系统变量中找到 'JAVA_HOME' 和 'JRE_HOME' 变量，如果没有则新建一个，分别将 'JAVA_HOME' 和 'JRE_HOME' 的值更改为JDK和JRE的存放路径。

配置好后，你的系统变量应该是类似于这样的：



接着：在系统变量选择Path，点击编辑：新建一条：

```
%JAVA_HOME%\bin
```

环境变量配置好后一定要点击确定来保存配置好的环境变量。

1.3.2 测试

为了测试环境变量是否配置完成，我们可以打开命令控制符(CMD)，我们可以按下Win键+S，搜索 'cmd' 或者 “命令控制符”

打开它，输入以下代码：

```
java -version
```

如果控制台正确输出了JAVA的版本，则代表JDK已被正确安装并且环境变量已被正确配置：

```
C:\Users\Liuyuexiang>java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
```

2.Tomcat的配置

2.1 Tomcat的介绍

Tomcat是一个Java Servlet容器，用于运行Java Servlets和呈现JavaServer Pages(JSP)。它在计算机本地提供了一个Web服务器环境，可部署和管理基于Java的Web应用程序。

2.2 版本选择

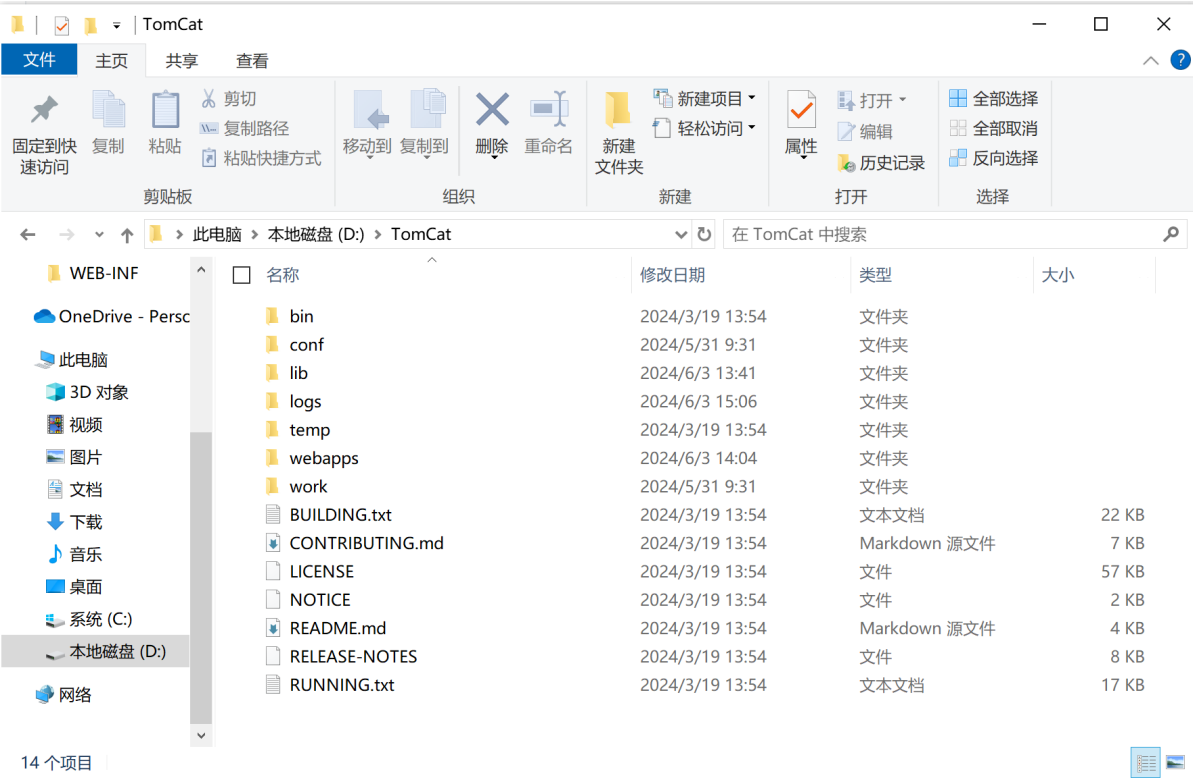
我们选择较为稳定且广泛使用的Tomcat8.5.100版本

下载链接：<https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.100/bin/apache-tomcat-8.5.100-windows-x64.zip>

如果下载出现问题，选择国内链接：<https://www.123pan.com/s/sVSxjv-mLSXv.html>

2.3 部署Tomcat

我们将Tomcat解压，可以在D盘根目录新建文件夹，名为 'Tomcat'，解压到此，应当是这样的：



2.4 配置Catalina环境变量

Catalina是Tomcat的核心Servlet容器，负责处理HTTP请求和响应，我们打开高级系统设置，在系统变量新增一个变量：

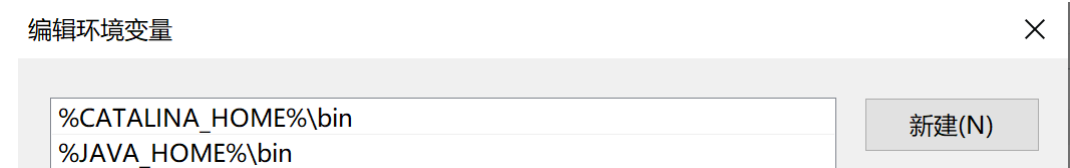
变量名：CATALINA_HOME

变量值：Tomcat存放路径，如果我们按照上文所做的将Tomcat安装到D:\Tomcat中，变量值应该为：D:\Tomcat

在系统变量中的 ' Path ' 变量中新增一条：

```
%CATALINA_HOME%\bin
```

结合前面我们在 ' Path ' 中新建的JAVA_HOME，我们的Path变量应该是这样的：



环境变量配置好后一定要点击确定来保存配置好的环境变量。

2.5 Tomcat服务的启动与关闭

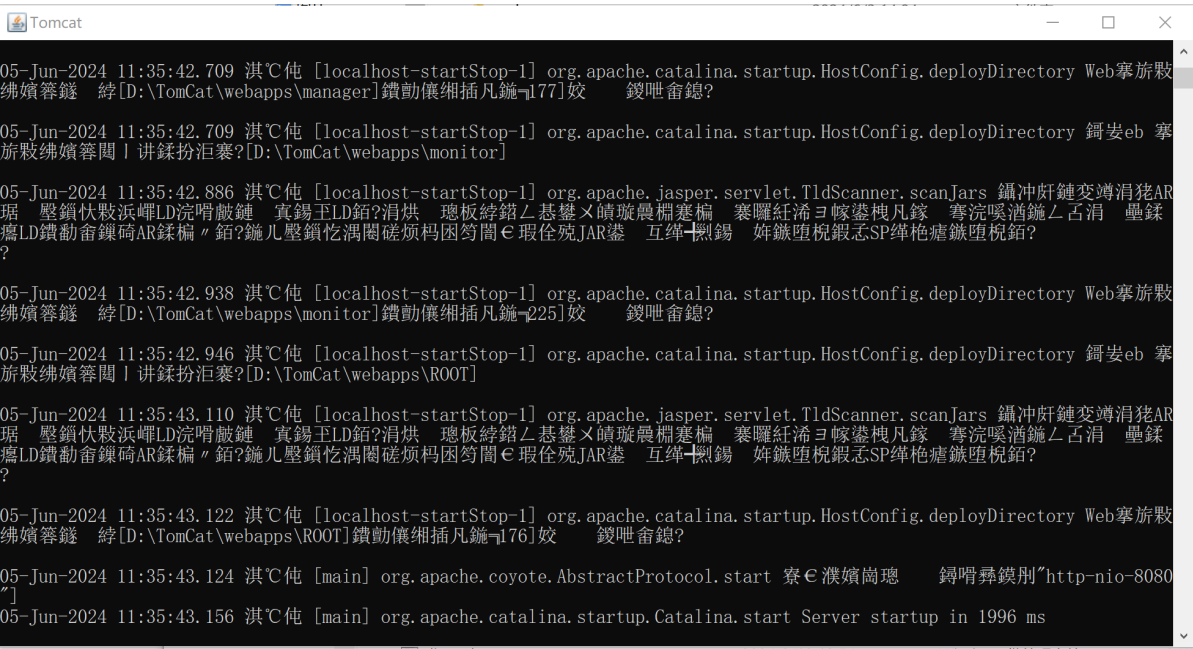
2.5.1 如何启动和关闭Tomcat服务

只有启动Tomcat服务，Tomcat才能正常运行

打开Tomcat存放路径中的 ' bin ' 文件夹，其中的 ' startup.bat ' 文件是启动Tomcat服务的批处理文件；

' shutdown.bat ' 是关闭Tomcat服务的批处理文件；

我们试着打开Tomcat服务，运行 ' startup.bat '，打开后会弹出一个窗口，类似于这样：



由于编码问题，Tomcat的控制台程序的中文会乱码，但并不影响使用。

如果你有强迫症，想要解决这个问题，可以打开Tomcat存放路径中的conf文件夹，找到其中的 ' logging.properties ' 文件

以文本模式编辑它，找到这一行：

```
java.util.logging.ConsoleHandler.encoding = UTF-8
```

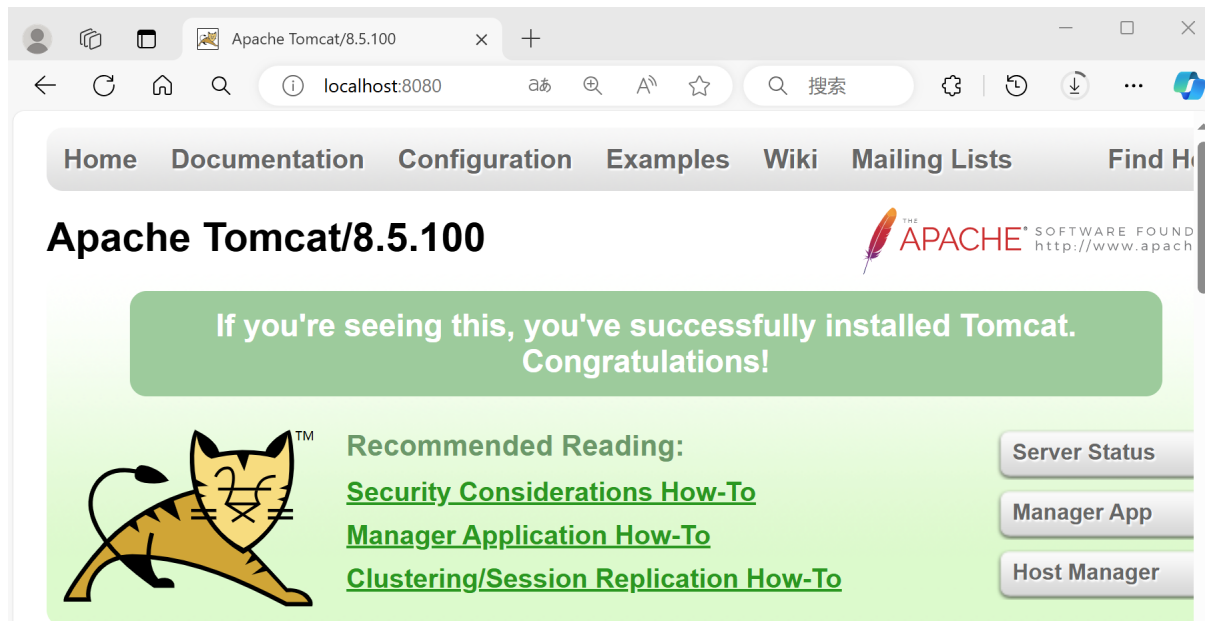
改为：

```
java.util.logging.ConsoleHandler.encoding = GBK
```

接着关闭并重新启动Tomcat服务，乱码的问题就解决了。

2.5.2 Tomcat的管理后台

在浏览器访问 'localhost:8080/ '



如果出现了如下界面，说明你的Tomcat已经配置好了，我们需要使用这个管理后台的功能，以后也许会提到。

二、第一个JavaWeb网页

1.Tomcat的项目部署

想要把网页部署在Tomcat服务器，通常有两种方法，我们先讲述第一种，第二种在后续会讲到。

我们通常把项目部署在Tomcat路径下的webapps文件夹，文件结构如下：

```
D: -> | Tomcat -> | ...
      | ...      | webapps | -> | docs      ---
      |           |         | | examples   ...
      |           |         | | host-manager 示例项目
      |           |         | | manager      ...
      |           |         | | ROOT         ---
      |           |         | | Your-Project-Name-1 你的项目1
      |           |         | | Your-Project-Name-2 你的项目2
```

可见，我们的项目分别独立地存放在 'webapps' 文件夹中，它们互不干扰；

2.项目的文件结构

当我们在IntelliJ IDEA、Eclipse IDE上创建一个新的webapp项目，它会创建一个固定的项目（文件）框架结构，为了能够对项目框架结构有一定的理解，我们需要自己来搭建如下的项目结构：

```
Your-Project-name -> | WEB-INF -> | classes -> | com -> | pkg -> | Class1.class
```



1.WEB-INF

WEB-INF 是一个特殊目录，包含了Web应用的配置信息、类文件和库文件。这个目录对外部访问是隐藏的，即客户端无法直接请求 这个目录中的内容。

a. classes

classes 目录包含了编译后的Java类文件。按照包名创建目录结构，在这个示例中，com.pkg 目录中包含了编译后的类文件（如 Class1.class）

```
WEB-INF/classes/com/pkg/Class1.class
```

这些类文件通常包括Servlet类、过滤器类和其他业务逻辑类。

b. lib

lib 目录包含了Web应用所需的所有依赖库（JAR文件）。这些依赖库可以是外部第三方库或你自己编写的库。

```
WEB-INF/lib/dependence1.jar
```

这些JAR文件在运行时会被加载到应用的类路径中，供应用程序使用。

c. web.xml

web.xml 是Web应用的部署描述符，包含了Servlet配置、过滤器配置、监听器配置、欢迎文件列表和其他配置信息。

2.assets

assets 目录包含了静态资源，如CSS样式表、JavaScript脚本和其他静态文件（如图片）。这些文件通常用于前端页面的展示和交互。

a. CSS

css 目录包含了所有的CSS样式文件，用于定义Web应用的样式和布局。

```
assets/css/style.css
```

b. js

js 目录包含了所有的JavaScript脚本文件，用于实现Web应用的动态交互功能。

```
assets/js/script.js
```

c. 其他资源

assets 目录还可以包含其他静态资源，如图片、字体文件等。

3.index.jsp

index.jsp 是Web应用的入口页面。当用户访问Web应用的根URL时，这个页面通常是第一个被显示的页面。

4.其他JSP页面

项目根目录下还可能包含其他JSP页面，用于处理不同的功能和展示不同的内容。

3. 第一个项目：数据库视图-静态网页

3.1 项目介绍

为了更容易理解Java Web项目，我们从最简单的项目开始：

这个网页将连接MySQL数据库，显示数据库中某个表的全部内容。

3.2 数据库的准备

MySQL在后续的项目会常常用到，下载MySQL： <https://www.123pan.com/s/sVSxjv-NLSXv.html>

运行下载好的 'mysql-8.4.0-winx64.msi'，如果提示 'requires Visual Studio 2019' 我们需要安装运行库：

下载链接： <https://www.123pan.com/s/sVSxjv-OLSXv.html>

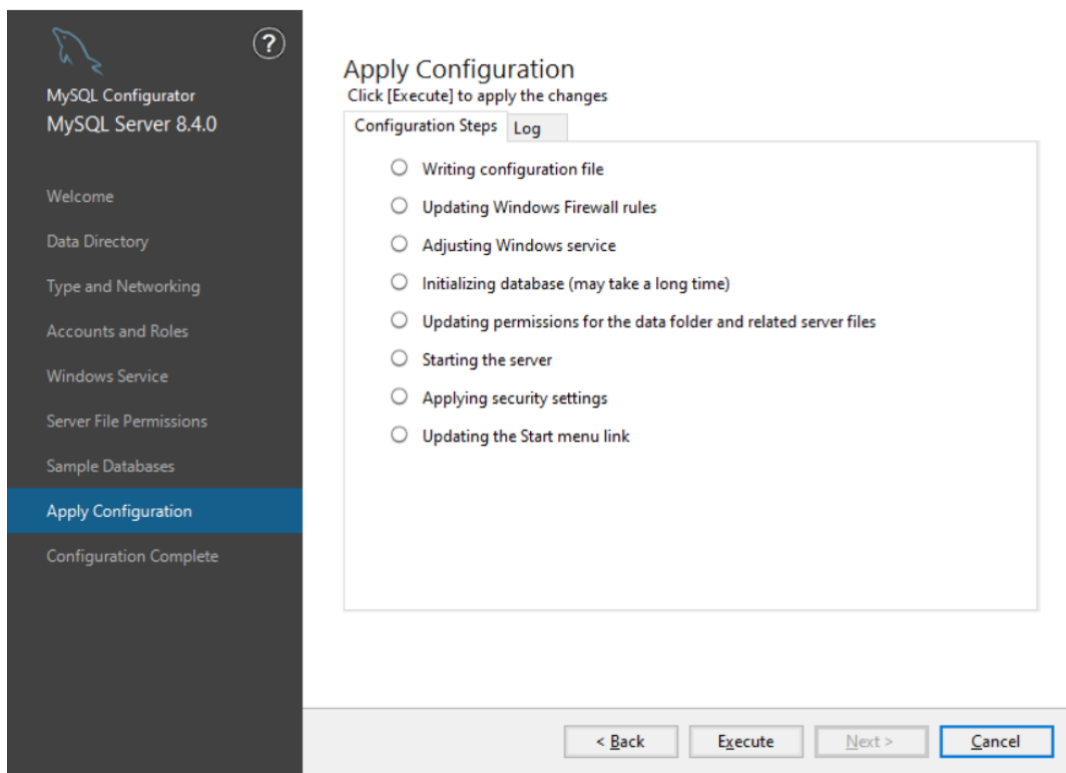
安装运行库后，我们就可以正常安装了，在 'Choose Setup Type' 中选择 'Typical'，最后勾选 'Run MySQL Configurator'

点击 'Finish'

在MySQL Configurator中全程下一步，需要修改的部分：

The screenshot shows the MySQL Configurator interface for MySQL Server 8.4.0. The 'Accounts and Roles' tab is active. In the 'Root Account Password' section, the 'MySQL Root Password' and 'Repeat Password' fields are both filled with six dots. The 'Password strength' is indicated as 'Weak'. Below this, the 'MySQL User Accounts' section is visible, featuring a table with columns for 'MySQL User Name', 'Host', and 'User Role'. To the right of the table are buttons for 'Add User', 'Edit User', and 'Delete'. The left sidebar contains a navigation menu with options like 'Welcome', 'Data Directory', 'Type and Networking', 'Accounts and Roles' (selected), 'Windows Service', 'Server File Permissions', 'Sample Databases', 'Apply Configuration', and 'Configuration Complete'. At the bottom of the window are navigation buttons: '< Back', 'Next >', and 'Cancel'.

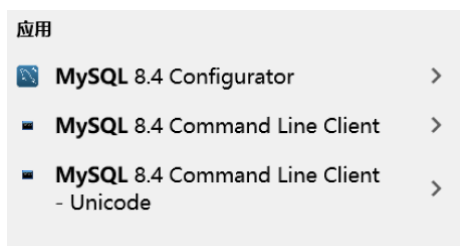
设置MySQL的管理员密码，我们通常设置为 '123456'；



在该页面点击 'Execute' , 等待配置完成点击Next, 接着点击Finish。

3.3 建库建表

我们按下Win键+S, 搜索 'MySQL' , 在搜索记录中找到 'MySQL 8.4 Command Line Client'



MySQL的使用与SQL Server有部分差异, MySQL提供了两种命令行客户端, 它们仅仅是编码不同

我们选择MySQL 8.4 Command Line Client即可

打开MySQL命令行后, 需要输入我们在Configurator中设置的密码, 我们输入密码后出现如下的输出, MySQL服务就正确地被配置好了:

```
Enter password: *****
welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.4.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```



```
mysql>
```

我们可以输入MySQL语句对数据库进行操作，接下来我们在为项目配置数据库的过程中演示如何操作。

a. 新建数据库 'book_db'

```
mysql> create database book_db    /* SQL语句的';'不可舍弃，只有用';'结束一个语句后按下回车才会执行SQL语句 */
    -> \c    /*当你未结束语句后，按下回车可以换行，不影响sql语句的结构，如果你上一行输错了，不能返回上一行更改
                但你可以输入 '\c'后按下回车取消本次输入
*/
mysql> create database book_db
    -> ;    /*我们可以看到换行并不影响SQL语句结构*/
Query OK, 1 row affected (0.06 sec)
```

当我们出现Query OK的字样，则代表这句SQL语句执行成功了。

b.选择数据库 'book_db'

MySQL中是多数据库的数据库管理系统，我们需要先选择需要操作的数据库：

```
use book_db;
```

执行后结果应是这样的：

```
mysql> use book_db;
Database changed
```

c. 新建表'book_tb'

选择数据库 'book_db'后，我们可以新建表：

```
create table book_tb(
id int,
name varchar(20),
author varchar(10)
);
```

执行后结果应该是这样的：

```
mysql> create table book_tb(
    -> id int,
    -> name varchar(20),
    -> author varchar(10)
    -> );
Query OK, 0 rows affected (0.10 sec)
```

d.在表中插入几条数据

```
insert into book_tb values (1,'沉默的大多数','王小波');
insert into book_tb values (2,'白色鸟','何立伟');
insert into book_tb values (3,'额尔古纳河右岸','迟子建');
insert into book_tb values (4,'送你一颗子弹','刘瑜');
```

执行后结果应该是这样的:

```
mysql> insert into book_tb values (1,'沉默的大多数','王小波');
Query OK, 1 row affected (0.02 sec)

mysql> insert into book_tb values (2,'白色鸟','何立伟');
Query OK, 1 row affected (0.00 sec)

mysql> insert into book_tb values (3,'额尔古纳河右岸','迟子建');
Query OK, 1 row affected (0.00 sec)

mysql> insert into book_tb values (4,'送你一颗子弹','刘瑜');
Query OK, 1 row affected (0.02 sec)
```

我们可以执行select语句查看表:

```
mysql> select * from book_tb;
+-----+-----+-----+
| id    | name          | author |
+-----+-----+-----+
| 1     | 沉默的大多数  | 王小波 |
| 2     | 白色鸟        | 何立伟 |
| 3     | 额尔古纳河右岸 | 迟子建 |
| 4     | 送你一颗子弹  | 刘瑜   |
+-----+-----+-----+
4 rows in set (0.02 sec)
```