

## JC2002 Java 编程 - 实践 1（第 1 天）

本实践的目的是熟悉 Codio，并创建一个 GitHub 学生账户（如果还没有的话），这在大学学习期间将非常有用。

1. 在本课程中，我们将使用 Codio 作为您运行应用程序和编程的在线场所。每次从 Codio Practical Box 访问时，您都会在同一空间中找到您的文件。如果您在另一门课程（可能是 Python 课程）中有一个 Codio 实用宝盒，那么它将是另一个宝盒。您可以在 "我的阿伯丁" 课程页面找到本课程的 Codio Practical Box 链接。Codio Box 运行 Ubuntu Linux，因此如果您对 Linux 还不熟悉，可以查看他们的[命令行初级教程](#)。
2. GitHub 是一个源代码云存储库，既可以是公共的，也可以是私有的。Git 是一个版本控制系统，用于跟踪应用程序的成长，而 GitHub 则是一个在线仓库，用于存储个人、组织和公司的应用程序。您现在就应该[创建一个学生账户](#)，以便在大学学习期间一直使用。这将允许您使用私人（而非公共）资源库，您应始终将其用于评估工作。

定期使用 GitHub 来存储您的所有评估工作，以避免在关键时刻笔记本电脑出现故障而带来的麻烦。你是计算机专业的学生，现在就开始养成良好的备份习惯吧。

在 GitHub 上学习 "hello world" 教程。您可以在 Codio 设置页面设置 SSH 密钥，如下所示：

- 1) 进入 JC2002 Codio Practical Box。查看右上角，您会看到一个带有您姓名的图标。点击该图标进入您的 Codio 面板页面。
- 2) 在 Codio 面板页面的左下方，您会看到您的账户名。点击它进入 "设置"。现在您会看到左侧有两列菜单。在最靠近页面中间的菜单中，您会看到 **"应用程序"**。在这个页面，你可以登录自己的 GitHub 账户，将 Codio 账户与 GitHub 绑定。登录之后，就可以把 Codio SSH 密钥上传到 GitHub。
3. 接下来，您应该熟悉在 Codio 中编辑源代码、编译和运行 Java 程序。为方便起见，您可以选择 **"视图" -> "布局" -> "2 列"**，然后激活其中一列并从 **"工具" -> "终端"** 中选择终端，将 Codio 视图分为文件编辑区和终端区。

选择 " 文件 "-> " 新建文件 "，创建一个新的 Java 源文件。将文件命名为 "HelloWorld.java"。在该文件中，您应编写下面的 Java 代码（您可以粘贴本文档中的代码，也可以手动编写，代码并不长）：

```
公共类 HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

在 Codio 中，源文件会在您编辑时自动保存。文件准备就绪后，您可以在终端窗口编写以下命令尝试编译：

```
javac HelloWorld.java
```

如果文件编译成功，目录中就会出现文件 HelloWorld.class。然后，就可以通过编写命令来执行该文件了：

```
java HelloWorld
```

如果一切顺利，终端窗口中应出现文本 "Hello World"。您可以尝试将源代码中的文本 "Hello World "修改为其他内容，然后重新编译并运行程序。