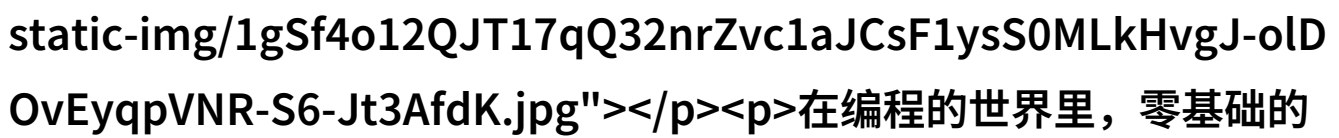


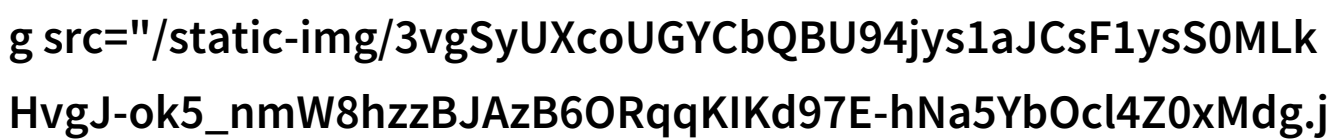
技术指南-初学者的完美起步第一次给0开

初学者的完美起步：第一次给0开的手法图解



在编程的世界里，零基础的新手往往面临着许多挑战。他们需要从一片空白中开始构建自己的知识体系，而这其中最重要的一环就是学会如何正确地给项目“开门”。也就是说，让一个完全没有任何内容、代码或功能的项目变得可运行。这一步对于每个初学者来说都至关重要，因为它标志着他们正式迈入了实践阶段。

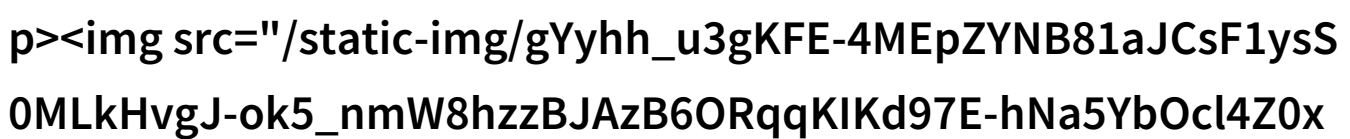
在这个过程中，“第一次给0开的手法图”就成为了许多新手的宝贵指南。这种图可以帮助开发者理解整个项目结构，以及各部分之间如何协同工作，从而确保项目能够顺利启动。



首先，我们需要明确的是，什么是“第一次给0开”的概念？简单来说，就是为一个全新的项目创建初始文件和目录结构，这通常包括但不限于以下几个方面：

确定框架：选择合适的框架或者工具来作为你的工程基础。这可能是一个基于Node.js、Python、Java等语言的框架，它会提供一系列预设好的配置和模板，以加速开发进度。

初始化仓库：如果你打算将你的项目托管到GitHub或其他版本控制平台上，你应该在开始之前初始化一个Git仓库。这有助于你跟踪所有更改，并且让团队成员（如果有的话）可以协作。

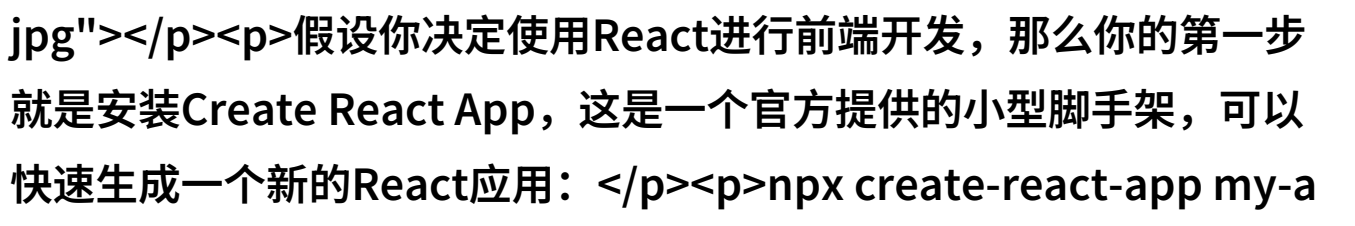


建立基本结构：根据所选框架，创建必要的文件夹和子目录，如src/main/java（用于存放源代码）、src/test/java（用于存放测试代码）等。在这个阶段，你还可能需要设置一些配置文件，比如package.json（npm包管理器）、requirements.txt（依赖项列表）等。

接下来，让我们通过一些真实案例来看看如何实际操作

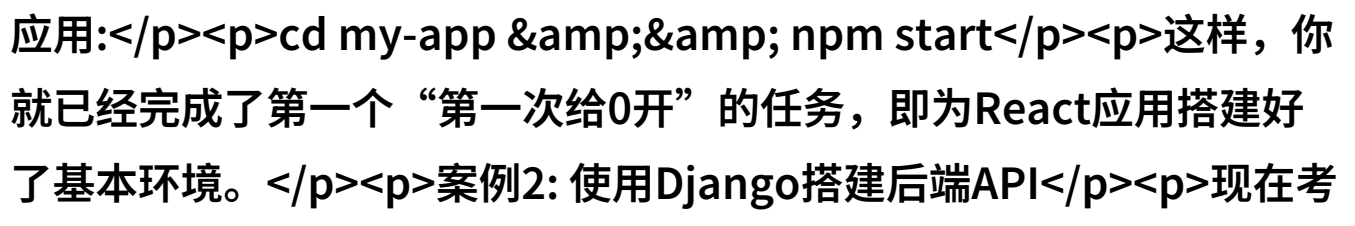
作一下这些步骤：

案例1: 使用React搭建前端应用



假设你决定使用React进行前端开发，那么你的第一步就是安装Create React App，这是一个官方提供的小型脚手架，可以快速生成一个新的React应用：

```
npx create-react-app my-app
```



然后进入刚创建的my-app目录并启动应用：

```
cd my-app && npm start
```

这样，你就已经完成了第一个“第一次给0开”的任务，即为React应用搭建好了基本环境。

案例2: 使用Django搭建后端API

现在考虑另一种情况，如果你想要用Python和Django构建后端API服务，那么你的流程会稍微不同。你首先要安装pipenv，并通过它来管理依赖项，然后使用django-admin命令行工具来创建新的Django项目：

```
pip install pipenv django==3.2.5 # 安装pipenv及指定版本django
```

```
mkdir my_django_app && cd $_ # 创建并进入我的djangoapp目录
```

```
pipenv install django # 在虚拟环境中安装django3.2.5
```

```
python -m django --version 3.2 --extension=rest_framework,graphene_django # 初始化Django并添加扩展包rest_framework,graphene_django
```

```
# 运行数据库迁移以设置好数据库表格：
```

```
python manage.py migrate --noinput
```

```
# 启动服务器：
```

```
python manage.py runserver
```

以上两个案例展示了不同的技术栈下的“第一次给0开”方法。无论是哪种方式，只要遵循上述指导原则，你都能成功地为自己实现这一关键一步。如果遇到困难，不妨查看相应技术社区中的相关教程或者咨询经验丰富的大佬，他们总有一些建议能帮你走出迷雾。

学者的完美起步第一次给0开的手法图解.pdf" target="_blank">下载
本文pdf文件</p>