在本节课开始之前，我们安装了test组件，对代码进行了一些修改，我们需要对git仓库的代码进行一下提交。在以后的课程中，如果对代码进行了修改，在每节课之后我会自动的把代码进行一次提交，并且提交到我的github主页上。

我们正在使用Symfony开发一套博客系统，博客系统最重要的就是文章数据的管理，Symfony作为MVC框架，我们需要使用一个model类对文章的数据进行封装。上节课我们已经安装了test组件，我们可以使用make命令行来创建一个单元测试类。

在底部控制台中我们输入`symfony console make:test`来创建一个单元测试类，make命令提供了以下几种测试类型： `TestCase`应用于最基础的单元测试。
`KernelTestCase`通常用于集成测试。如果我们想在测试用例中使用Symfony提供的一些服务，我们就可以使用KernelTestCase。 剩下的其他三种通常用于功能测试。

本节课我们将编写的是最基础的单元测试，这里类型我们选择`TestCase`。下一步就是输入类名，这里的示例是`BlogPostTest`，就是博客的文章测试。正好我们开发的也是一个博客系统，我们可以输入`PostTest`。使用Post类对文章数据进行封装，Test后缀说明我们当前的类是一个测试类。

回车，在tests目录中创建了PostTest类。我们打开，这里有个最基础的测试方法。测试方法都以test单词作为开头，后面可以根据需要自己写。测试方法不需要返回值，都是void。

一般情况下，我们不需要传入参数，在某些情况下，我们测试方法可能需要依赖于上一个测试的结果，那时需要我们传入参数，后面课程我们会讲。

现在我们给我们的文章添加测试代码。修改这个方法，我们希望有一个Post类来对文章的数据进行封装，Post类它应该是一个`POPO对象`，就是简单的PHP对象。最常见的POPO对象都有getters和setters方法，我们希望Post类中有一个属性，`title`用来保存文章的标题，可以通过setters方法设置文章的标题，可以通过getters方法来获取文章的标题。

编写代码，$post->setTitle()设置标题，然后我们对标题进行验证。单元测试提供了一些验证方法，我们使用assertSame()。 assertSame()第一个参数是期待的值，第二个参数是实际的值，第三个参数是验证失败时显示的消息，我们期待`这是一个标题`。第二个参数，我们使用$post的getters方法来获取文章的标题，`getTitle()`。

```
class PostTest extends TestCase
{
    public function testPostTitle(): void
    {
        $post = new Post();

        $post->setTitle("这是一个标题");

        $this->assertSame("这是一个标题", $post->getTitle());
    }
}
```

这是一个很简单的测试代码，我们执行测试代码，输入`php bin/phpunit`，报错了，它提示Post类不存在。现在我们要创建我们Post类，我们期望Post类作为我们MVC框架的Model，Symfony提供了另外一个make命令，来辅助我们创建这个model类。

我们查看make命令，`make:entity`创建一个doctrine的实体类，输入`symfony console make:entity`，类名就是Post。现在我们需要输入Post类的属性，这里属性我们输入title，这里要输入字段的类型。

我们使用make命令行创建的是doctrine的实体类，这个类用于orm框架，实体类与数据库表的关系对应，实体类的每个属性都对应了数据库表的一列。下面我们选择title属性在数据库表中的类型，，我们文章的标题是字符串，这里可以直接使用string类型，255是代表该列在数据库中可以存储的最长长度，我们这里标题默认255就可以了，这个字段在数据库表中可以为空吗？文章的标题都不能为空，这里我们直接回车，选择no就可以了。 还要添加另外一个属性吗？我们不需要添加另外属性，直接回车。

打开src目录下Entity目录，命令行帮助我们创建了一个Post类，Post类中有一个title属性和getTitle()、setTitle()方法。刚才我们选择的类型string和长度255都以ORM注解的形式添加到了注释中，代表title属性在数据库表中列的类型是string类型，最大长度是255。

现在我们修改测试代码，我们需要使用Post类。再次执行单元测试，OK单元测试通过了，这就是一个最简单的测试驱动开发的一个流程。当然Post类是一个简单的PHP类，我们并不需要对代码进行重构。博客文章只需要标题是不够的，还需要正文、摘要、发布状态，发布日期，更新日期，封面图像等等属性，你可以继续编写单元测试代码，但是Post类只是一个简单的PHP对象类，并不需要这样麻烦编写测试代码，这样太无聊了。我们只会再增加一些功能性的代码时，再编写测试代码来节省开发的时间。

下面我将继续使用make:entity命令添加额外的属性在控制台中输入`symfony console make:entity`，类名我们仍然输入Post类。Post类已经存在了，然后我们可以额外的添加一些属性。

添加文章正文body，我们可以使用问号查看所有可用的类型。text类型和string类型在数据库中的存储形式也是不一样的 `string`类型在数据库中是`VARCHAR`，`text`类型在数据库中是`TEXT`，body属性我们不限制长度，所以我们选择text。在数据库中可以为空吗？可以为空。

文章摘要，摘要类型输入text，在数据库中可以为空吗？可以为空。

status，我们文章通常发布之后有草稿状态和已发布状态，我们使用string类型来区分文章的状态，长度输入20可以使用单词`draft`代表草稿状态、`published`代表已发布状态。该字段在数据库可以为空吗？可以为空。

然后是文章的创建时间，数据类型，我们选择`datatime`，日期类型，可以为空吗？可以为空。

然后是更新时间，暂时先添加这些属性回车。我们查看代码，命令行帮助我们完成了Post类的创建。

当然博客还有评论功能，在下一节我们将使用命令行快速的创建评论的model类。