

# 单元测试是什么？

## 1. 单元测试的基本概念

单元测试是软件工程中降低开发成本，提高软件质量常用方式之一，单元测试是一项由开发人员或者测试人员来对程序模块的正确性进行检验测试的工作，用于检查被测试代码的功能是否正确，养成单元测试的习惯，不但可以提高代码的质量，还可以提升自己的编程和技巧。Django 单元测试使用的是 Python 标准库 unittest 模块来定义相应的测试用例。在本节我们先了解什么是单元测试以及作为开发人员在这个环节应该注意什么。

单元测试其实就是对模块、类、函数实现的功能执行检测，看看是否满足预期，是否达到功能要求，它是一次检查检验的过程。如果某个模块或者函数满足预期，则表示测试通过，否则表示失败，比如工厂在组装一台电视机之前，会对每个元件都进行测试看是否合格，这就是单元测试，是不是很好理解。如果不局限于 Django 框架的话，除了单元测试以外，还有系功能测试、组件测试、系统集成测试等、测试是一项很重要的工作，有专门的职位比如软件测试人员、测试工程师等。

## 2. 为什么编写单元测试

其实这个问题就像你问为什么有质量检测员一样，所以可以看出编写单元测试的重要性，俗话说“人无完人，金无足赤”，开发人员也有犯错误的时候，这就需要编写单元测试来完成对自己代码的检查工作，那编写单元测试对于整个项目以及个人而言到底有哪些益处呢？下面就给大家进行一下总结：

### 1) 降低开发成本

单元测试可以快速的提供反馈，将问题在开发阶段就暴露出来。这样就可以减少向下游传递的问题，比如说在系统集成阶段出现了问题，这不就很麻烦吗？因为模块、组件太多，会给你无从下手的感觉。所以做好单元测试无疑是降低了软件的开发成本。

有数据表明尽早地对软件产品进行测试将使效率和质量都得到最好的保证。Bug 发现的越晚，修改它所需的费用就越高，因此从费用成本角度来看，应该尽可能早的查找和修改 Bug。在修改费用变的过高之前，单元测试是一种在早期抓住 Bug 的有效途径。

## 2) 边界检测提高代码质量

边界值检测方法是单元测试中常用的方法之一，举个例子，比如  $1 < x < 10$ ，那么边界值就是 0、2、9、11，因为在实际的使用中，我们不容易发现边界问题，比如注册用户名一般使用字母数字下划线等，那么如果使用特殊的符号又会怎么样呢？这就需要我们去做检验。通过边界值检测的方法就能很好的解决这个问题，从而提高代码的质量。

## 3) 提高开发人员职业素养

由于互联网的发展，促使了软件行业的发展，企业在遵守行业标椎的同时也制定相关的企业标准使软件开发集成上线更加精细化，形成了一套完成的流程。对于部分开发者来说一旦编码完成，他们总是会迫切希望进行软件的集成工作，这样就能够看到系统的启动了。这在外表上看来确实是明显的进步，但实际而言只是为了进度而进步，严重忽略了单元测试的重要性，若出现问题只会浪费更多的时间成本。

所以编写单元测试，不仅可以提升企业的效率，而且可以使开发人员站在用户的角度重新调整自己的代码，进行代码重构，代码的优化等，这也培养了开发人员的职业素养，提升的他们的编码能力，对于企业的长远发展是非常有利的。

# 3. 单元测试对象有哪些

通常情况下，项目开发的周期是比较紧张的，要想 100% 全覆盖地编写单元测试代码需要花费大量的时间。这时，就需要有选择地编写测试代码，对于功能实现很简单的函数，就可以考虑忽略。那么哪些代码应该做测试呢？在这里我们进行了简单的总结，如下所示：

## 1) 核心的业务层代码

对于项目中最核心的部分也就是业务层代码，这是单元测试不可忽视，且非常重要的一项，所以开发人员要保证不会错误。

## 2) 需要经常修改的代码

对于产品迭代或代码重构而言，可能有些模块需要经常被修改，它们也需要被单元测试所覆盖，这样可以检验修改后的代码是否存在问题。

## 3) 逻辑复杂的代码

涉及到逻辑层代码，也需要开发人员进行单元测试，做到零纰漏。尤其对于那些至关重要的逻辑代码，需要使用单元测试验证它的可用性，易用性。

总之在单元测试有一个原则，即能够有效的完成有效的检验，又能把珍贵时间合理利用，不至于浪费，这就需要在单元测试环节中，做到灵活变通，不死板。作为开发人员，当我们完成一个功能的时候，就应该针对性的考虑是否要测试该功能，这样就避免了项目后期工作中的应接不暇。才能够更好的完成单元测试环节。