

# Сравнительный анализ модулей и библиотек тестирования в Python

Модули/библиотек и	Описание	Достоинства	Недостатки
Assert	Используется для проверки истинности указанного утверждения. Если проверка не прошла, возбуждается исключение <code>AssertionError</code> .	1.Читаемость; 2.Нет необходимости подключать сторонние библиотеки;	1.Ручной запуск; 2.Приходится писать и отлаживать вручную.
Pytest	Модульное тестирование.	1.Независимость от API (no boilerplate); 2. Подробный отчет об ошибках. 3. Динамические фикстуры всех уровней, которые могут вызываться как автоматически, так и для конкретных тестов. 4. Дополнительные возможности фикстур (возвращаемое значение, финализаторы, область видимости, объект request, автоиспользование, вложенные фикстуры); 5. Плагины. Данный модуль имеет достаточно большой список дополнительных модулей, которые можно установить отдельно; 6. Возможность запуска тестов написанных на unittest и nose, то есть полная обратная совместимость с ними.	1. Необходимость отдельной установки модуля 2. Относительно высокий порог вхождения
nose	Девизом <i>nose</i> является фраза " <i>nose extends unittest to make testing easier</i> ", что можно перевести как " <i>nose</i> расширяет <i>unittest</i> , делая тестирование проще". <i>nose</i> идеален, когда нужно сделать тесты "по-быстрому", без предварительного планирования и выстраивания архитектуры приложения с тестами.	1.Функционал <i>nose</i> можно расширять и настраивать с помощью плагинов. 2. Имеет также собственное API, использовать которое необязательно Остальное см. в Pytest	см. в Pytest

DocTest	Вьет фрагменты текста, которые выглядят как интерактивные python сессии. Далее выполняет сеансы и проверяет, совпадает ли с тем что указано в docstring.	1.Читаемость; 2. Нет необходимости подключать сторонние библиотеки; 3. Находится в документации.	1.Длинные DocTests ухудшают читаемость программы; 2.Запускаются сразу все.
UnitTest	<i>Framework</i> для тестирования, входящий в стандартную библиотеку языка <i>Python</i> . Его архитектура выполнена в стиле <i>xUnit</i> . <i>xUnit</i> представляет собой семейство <i>framework'ов</i> для тестирования в разных языках программирования, в <i>Java</i> – это <i>JUnit</i> , <i>C#</i> – <i>NUnit</i> и т.д. Поддерживает автоматизацию тестирования, совместное использование кода установки и завершения работы тестов, агрегацию тестов в коллекции и независимость тестов от структуры отчетов.	1.Поддержка автоматизации тестов; 2.Использует общий код для настройки и завершения тестов, объединение тестов в групп 3.Позволяет отделять тесты от фреймворка для вывода информации.	1.Необходимость вручную писать много кода; 2. «инородный» стиль, унаследованный от Java; 3.Читаемость не такая хорошая, как у doctest и assert.