

Создание таблицы со сравнительным анализом библиотек для тестирования.

Критерии: читаемость, доступность, запуск и написание

Библиотека	Описание	Достоинства	Недостатки
Assert	Используется для проверки истинности указанного утверждения.	1.Легко читаемы; 2.Не используют ничего кроме стандартных средств (библиотеки также подключать не надо);	1.Запуск тестов вручную; 2.Сложно писать и отлаживать, потому что всё необходимо писать вручную (в том числе и сообщение об ошибке).
DocTest	Модуль doctest ищет куски текста, которые выглядят как интерактивные сессии Python и затем выполняет эти сессии, чтобы проверить, что они работают точно так же, как показано.	1.Доступность (не нужно скачивать дополнительные библиотеки); 2.Легко читается; 3. Легко проверяется, так как находится в документации.	1.Длинные докесты ухудшают читаемость программы; 2.Нет возможности определять какие именно докесты будут запущены, запускаются все.
UnitTest	Поддерживает автоматизацию тестирования, совместное использование кода установки и завершения работы тестов, агрегацию тестов в коллекции и независимость тестов от структуры отчетов.	1.Удобный запуск тестов (поиск тестов в директории, запуск отдельных тестов, методов и т. д.); 2.Понятные сообщения об ошибках; 3. Доступен в стандартной библиотеке.	1.Громоздкий, заставляет писать много кода; 2.API унаследован от Java («инородный» стиль); 3.Понятность тестов уступает видам тестирования с помощью doctest, assert.
Py.test	Позволяет легко писать небольшие тесты, но масштабируется для поддержки сложного функционального тестирования приложений и библиотек.	1.Практически нет API; 2.Тесты реализуются с помощью assert; 3.Детализированный вывод; 4.Богатые возможности по запуску тестов; 5.Параметризация тестов; 6.Переиспользуемость тестов; 7.Богатый собственный функционал и множество расширений.	1.Сложность внутренней реализации 2.Несколько более непривычный внешний вид, если есть привязанность к традиционным (например, UnitTest) тестам.