

Автоматизация тестирования консольных приложений Linux на Python

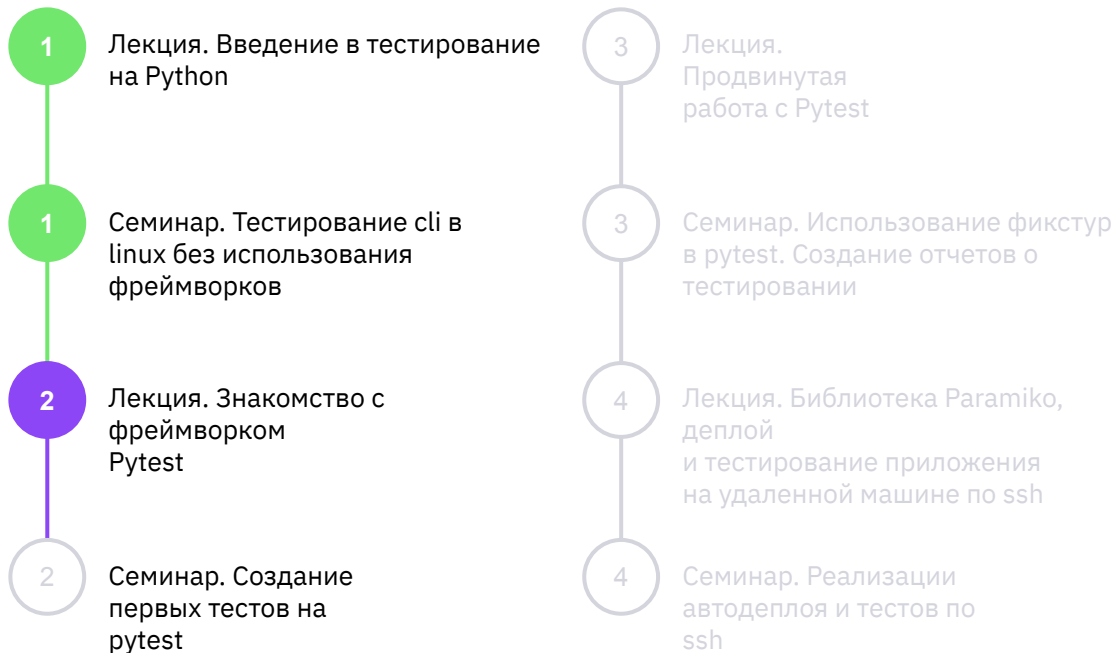
Лекция 2

Знакомство с фреймворком Pytest





План курса









Знакомство с фреймворком Pytest





Поехали!

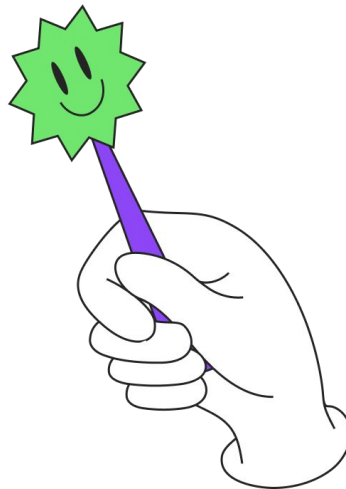
Что будет на уроке сегодня

-  оператор assert
-  фреймворки для тестирования на Python
-  использование Pytest
-  ключи и параметры Pytest



Работа с выводом в subprocess

- ✓ `subprocess.PIPE`
- ✓ `subprocess.STDOUT`
- ✓ `subprocess.DEVNULL`





Написание тестов для архиватора 7z в Linux





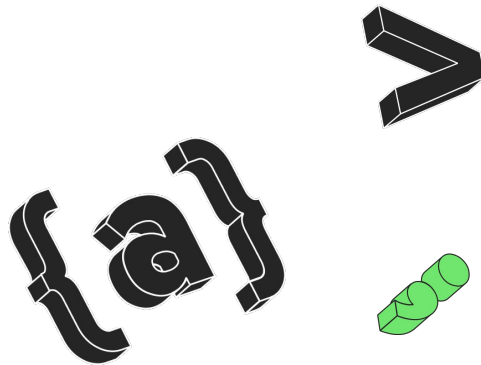
Оператор assert



Оператор assert

- ✓ Оператор assert работает как логическое выражение, проверяя, является ли заданное условие истинным или ложным.
- ✓ Если условие истинно, то ничего не происходит и выполняется следующая строка кода.
- ✓ Если же условие ложно, оператор assert останавливает выполнение программы и выдает ошибку. В этом случае assert работает как ключевое слово raise и выводит исключение.
- ✓ Исключение, вызванное оператором assert, также называется AssertionError

```
assert <condition>, <message>
```





Редактирование кода
для использования `assert`



Фреймворк — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.





Какие вы знаете фреймворки для Python?

Напишите ваши ответы в комментариях к видео.
Время на размышление — 1 минута.





Фреймворки для тестирования на Python

	Лицензия	Установка	Применение	Особенности
 Robot	Free software (ASF License)	Библиотека	Приемочное тестирование	Управление ключевыми словами
 PyTest	Free software (MIT License)	Отдельная	Модульное, приемочное, функциональное тестирование	Классы и фикстуры для упрощения тестирования
 Unitest	Free software (MIT License)	Часть стандартной библиотеки	Модульное тестирование	Быстрота и гибкость



Фреймворки для тестирования на Python

	Лицензия	Установка	Применение	Особенности
 DocTest	Free software (MIT License)	Часть стандартной библиотеки	Модульное тестирование	Использование интерактивной командной строки Python при работе с тестируемым приложением
 Nose2	Free software (MIT License)	Плагин для unittest	Продвинутое модульное тестирование	Большое число плагинов
 Testify	Free software (ASF License)	Плагин для unittest	Продвинутое модульное тестирование	Расширяет функциональность unittest и nose2



Pytest — это даже больше, чем фреймворк, это платформа для тестирования программного обеспечения.

Pytest — это программа командной строки, она предоставляет инструмент, который автоматически находит написанные тесты, запускает тесты и пишет отчеты с результатом.



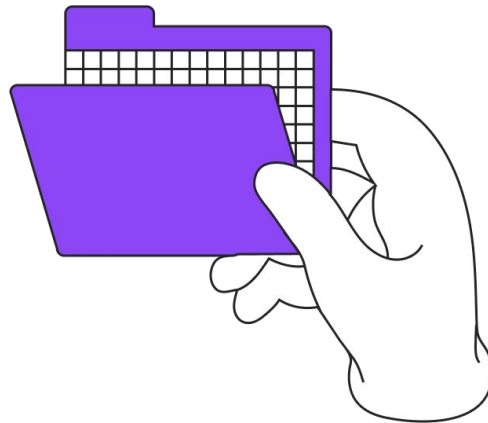
Достоинства Pytest

- ✓ Простые тесты легко написать в pytest.
- ✓ Сложные тесты тоже просто писать.
- ✓ Тесты очень легко читаются.
- ✓ Вы можете начать работу в pytest очень быстро.
- ✓ Можно использовать pytest для запуска тестов, написанных для unittest или nose.



Требования к именованию

- ✓ Имя файла должно начинаться с **«test»** или заканчиваться на **«test.py»**.
- ✓ Имя тестовой функции внутри файла должно начинаться с **«test_»**, например **«test_gb»**.





Первый тест на pytest





Как вы думаете: `assert` из предыдущего примера
и `assert` из последнего примера один и тот же?

Напишите ваши ответы в комментариях к видео.
Время на размышление — 1 минута.





Ответ

Нет, так как первый прерывает выполнение остальных тестов, а второй — нет.



Почему мы не импортировали модуль pytest в файле с тестами?

Напишите ваши ответы в комментариях к видео.
Время на размышление — 1 минута.





Ответ

В этом нет необходимости, т.к. файл запускается не напрямую, а передается в pytest.







Ключи и параметры Pytest





Что мы узнали сегодня

-  оператор assert
-  фреймворки для тестирования на Python
-  использование Pytest
-  ключи и параметры Pytest





Спасибо за внимание!