

Автоматизация тестирования консольных приложений Linux на Python

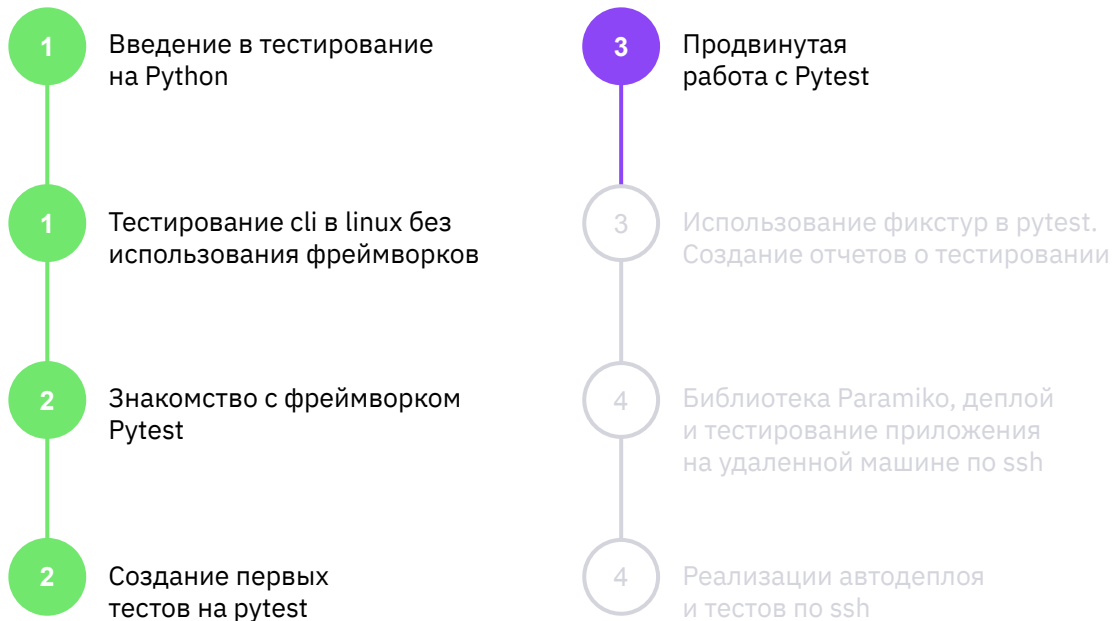
Лекция 3

Продвинутая работа с Pytest





План курса





Продвинутая работа с Pytest



Что будет на уроке сегодня

- 📌 отдельный файл для параметров теста
- 📌 тестовые фикстуры
- 📌 как вынести фикстуры в отдельный файл
- 📌 фикстуры с финализацией
- 📌 формирование отчетов в различных форматах
- 📌 конфигурирование запуска Pytest





Поехали!



Декораторы



Декораторы в python

Декораторы — это обёртки вокруг Python-функций (или классов), которые изменяют работу того, к чему они применяются.

В Python декораторы используются, в основном, для декорирования функций (или, соответственно, методов).

Декоратор это функция, которая принимает функцию и возвращает функцию.



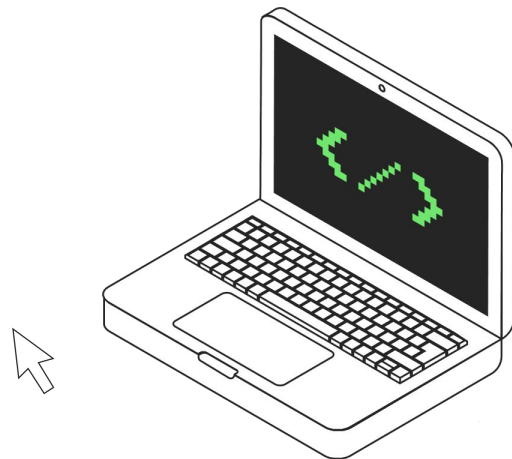


Фикстуры



Фикстуры Pytest

- ✓ Для того, чтобы зарегистрировать функцию как фикстуру, нужно использовать декоратор `@pytest.fixture`.
- ✓ Фикстуры позволяют тестовым функциям легко получать предварительно инициализированные объекты и работать с ними, не заботясь об импорте/установке/очистке.
- ✓ Фикстуры могут выполнять работу, а могут возвращать данные в тестовую функцию.





Написание кода
с использованием
фикстур



YAML

YAML — это язык для хранения информации в формате понятном человеку.

Его название расшифровывается как, «Ещё один язык разметки».

Однако, позже расшифровку изменили на — «YAML не язык разметки», чтобы отличать его от настоящих языков разметки.





YAML

```
simple-property: a simple value

object-property:
  a-property: a value
  another-property: another value

array-property:
  - item-1-property-1: one
    item-1-property-2: 2
  - item-2-property-1: three
    item-2-property-2: 4

# no comment in JSON
```

JSON

```
{
  "simple-property": "a simple value",
  "object-property": {
    "a-property": "a value",
    "another-property": "another value"
  },
  "array-of-objects": [
    { "item-1-property-1": "one",
      "item-1-property-2": 2 },
    { "item-2-property-1": "three",
      "item-2-property-2": 4 }
  ]
}
```

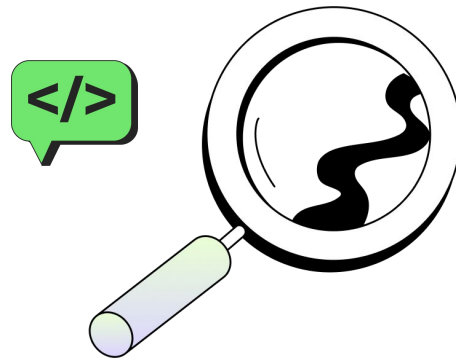


Работа с yaml



Если вы планируете использовать фикстуру в нескольких тестах (файлах с тестами), то можно объявить ее в специальном файле `conftest.py`.

При этом импортировать ее не нужно — `pytest` найдет ее автоматически.



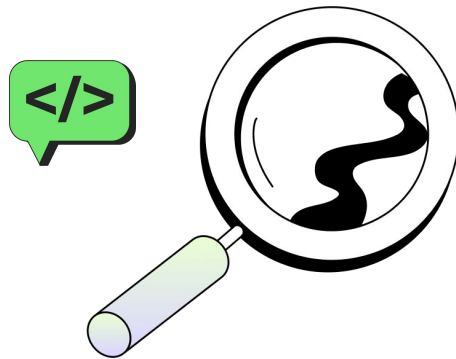


Если вы планируете использовать фикстуру в нескольких тестах (файлах с тестами), то можно объявить ее в специальном файле `conftest.py`.

При этом импортировать ее не нужно — `pytest` найдет ее автоматически.

Поиск фикстур начинается с тестовых классов, затем они ищутся:

- в тестовых модулях (файлах с тестами)
- в файлах `conftest.py`,
- и, в последнюю очередь, во встроенных и сторонних плагинах.





Работа с conftest.py



Pytest поддерживает выполнение фикстурами специфического завершающего кода при выходе из области действия.



Если вы используете оператор `yield` вместо `return`, то весь код после `yield` выполняет роль после завершения работы тестовой функции.



Финализаторы в фикстуре





Отчетность

pytest test.py --junitxml=report.xml

Java ?! Не перепутал ли преподаватель курс?

JUnit
Программное
обеспечение

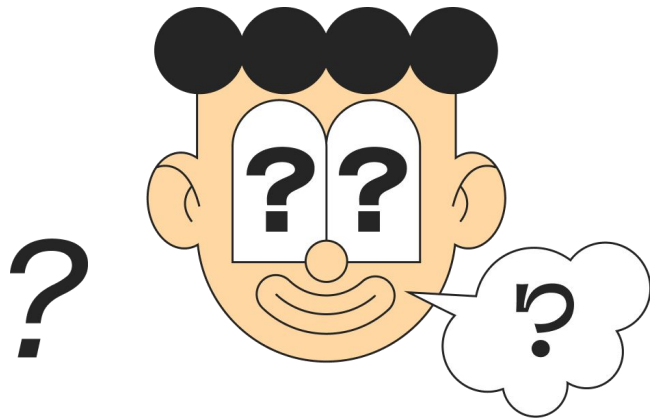
JUnit — фреймворк для модульного тестирования программного обеспечения на языке Java. Созданный Кентом Бекем и Эриком Гаммой, JUnit принадлежит семье фреймворков xUnit для разных языков программирования, берущей начало в SUnit Кента Бека для Smalltalk. [Википедия](#)

Аппаратная платформа: [Java Virtual Machine](#)

Лицензия: [Common Public License](#)

Последняя версия: 5.8.2 (28 ноября 2021; 13 месяцев назад)

Язык программирования: [Java](#)





Отчетность в Pytest





Конфигурационный файл pytest.ini



pytest.ini



Что мы узнали сегодня

- 📌 как сделать отдельный файл для параметров теста
- 📌 как писать тестовые фикстуры
- 📌 как вынести фикстуры в отдельный файл
- 📌 что такое фикстуры с финализацией
- 📌 как формировать отчеты в различных форматах
- 📌 как конфигурировать запуск Pytest





Спасибо за внимание!