# Лекции/мастер-классы для участников

## Разработка задач для онлайн курсов - вторая половина сентября (22 или 29)

- Архитектура:
  - Как видит студент:
    - терминал
    - многократная отправка решения
  - Как видит автор задачи:
    - Bootstrap
    - Pytest, 2.7
    - Docker
  - Пробуем сделать простую задачу - заменить содержимое файла на заданное
  - Очевидные грабли
    - Pytest
      - Русский язык
      - Использование слешей
  - Неочевидные грабли
    - Права в Bootstrap
    - Использование репозиториев
    - Текущая директория
    - Преступная смекалка студентов курса
    - Разные условия задач
  - Решение - использовать заготовку
    - Генерация SSH-ключа
    - Заготовка bootstrap
    - Как писать генератор задания
    - Заготовка Pytest
  - Жульничество и точное выполение буквы задания
    - Цель задания - отработать определенный навык
    - Часто есть обходные пути
    - Как можно отсечь возможности для этого:
      - Как удалить sudo доступ дефолтного пользователя
      - Как проверить запуск определенных команд (история команд)
      - Посложнее:
        - кастомная libc и переопределение библиотечных вызовов
        - очень хитрая настройка прав
  - Известные проблемы https://support.stepik.org/en/support/solutions/articles/19000015512-linux-challenge-issues

## Отладка веб-приложений (29.09 или 06.09)

https://stepik.org/course/154

- Общая схема работы архитектуры клиент-сервер
- Коды ошибок HTTP
- Когда сломался JS/верстка
    - Режим инкогнито!!
    - Консоль
        - выполнение JS
        - проверка CSS свойств
- Когда сломался сервер:
    - Логи
        - апача
        - приложения в веб-интерфейсе
        - syslog
- Типичные грабли
    - HTML / JS
        - относиельные пути
        - внешние ресурсы (js, img)
        - фрагментированный JS (внутри HTML)
        - js, не обернутый в функции или $(document).ready
        - jquery вместо dom
        - console.log
        - поле id и class у тегов
    - Бакенд:
        - Используйте максимум возможностей БД (пример с list(...))
        - локальное имя в hosts для удобства
        - apache вместо run.py

# MongoDb (29.09 или 06.09)

https://bitbucket.org/mark_zaslavskiy/nosql_inroduction/src/353f681b18fbdd4b17d58fa062f51f8a0f6dc413/slides/3_document-oriented_databases.pdf?at=master&fileviewer=file-view-default

В перспективе будет заменено на презентацию и модуль в MOOC.

- Установка
    - apt-get install
    - как проверить успешность установки
    - как отключить журналироварние
- mongo shell -
    - как запустить,
    - подключится к БД,
    - как просмотреть список БД,
- Идея
    - базу не нужно создавать явно
    - таблицы = коллекции
    - нет ссылок между коллекциями
    - must have id
- CRUD
    - insert - создаем новый документ
    - update - меняем/удаляем поля
    - remove -

    - find
        - поиск по одному полю
        - or, and
        - поиск по подстроке и рег.выражению
        - поиск по вложенным полям
        - sort, limit, offset
    - как проделать запросы в python.

# Написание тестов (???)

К лекции нужно сделать репо с примерами.

- Какие бывают тесты и зачем
- Общая идея:
    - тестируем хорошее и плохое поведение
    - чистая тестовая среда
    - не подгоняем тестовую среду под тесты
- Пример юнит-теста на Python
- Как запускать много тестов одним скриптом
- Пример Selenium теста Python
    - Настройка selenium
    - Параметризованный тест
    - Скрипт для запуска всех тестов

# Докер и развертывание (??)

https://www.katacoda.com/courses/docker

- Очень примерно про архитектуру
- Устанавливаем докер (используем правильный репо)
- Запускаем простой образ
    - Смотрим как он выглядит в docker ps
    - Убиваем docker kill
    - Смотрим на образы в docker image
- Добавляем Vagrant
    - Установка
    - Настройка плагинов
    - пишем vagrant-file
        - Указываем образ ubuntu14:04 (или аналогичный)
        - Указываем проброс портов
        - Запускаем и убеждаемся, что проброс работает
        - Подключаемся по ссш
        - Смотрим статус vagrnt status
        - Убиваем конфигурацию vagrant destroy -f
- Создаем свой Docker-образ
    - Через Dockerfile
        - Идея
        - Заготовка Dockerfile - какие правила написания

        - Собираем образ
        - Отправляем образ в репозиторий
    - Через docker commit

From:
https://se.moevm.info/ - **МОЭВМ Вики [se.moevm.info]**

Permanent link:
**https://se.moevm.info/doku.php/courses:mse:lections**

Last update: