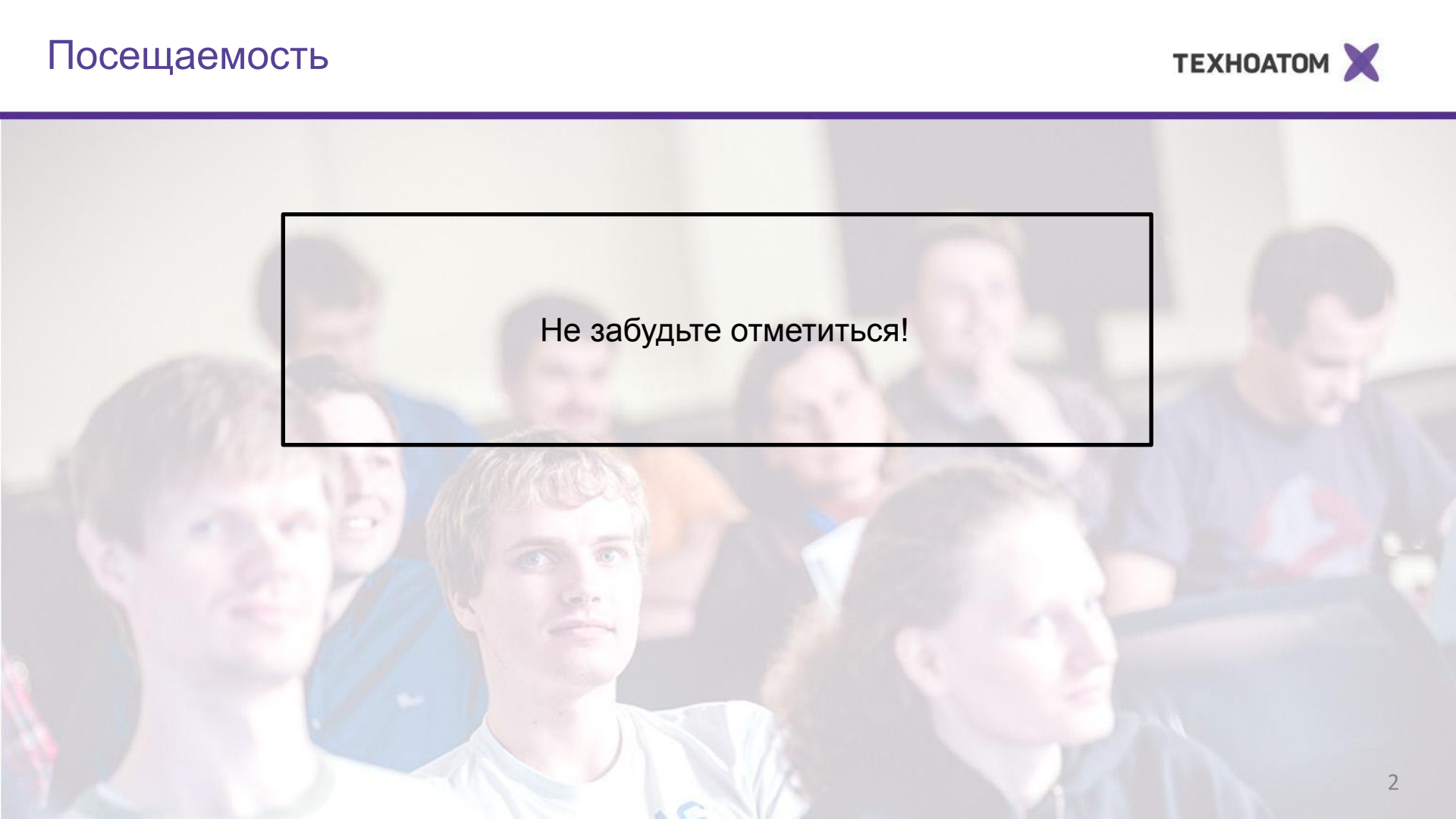


Вводное занятие

Урок первый

Антон Кухтичев



Не забудьте отметить!



Алёна Елизарова

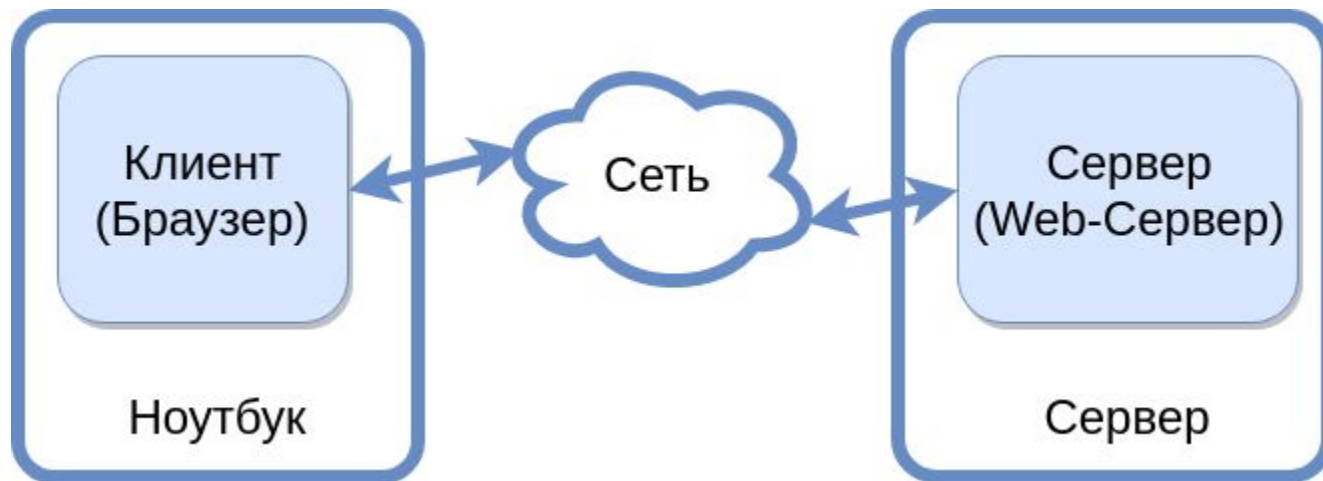


Антон Кухтичев

Основные тезисы

- Цель курса - разработать современный и молодёжный мессенджер (a-la Slack, Telegram);
- Разработка - индивидуальная, без команд;
- 10 смешанных занятий (лекция + семинар);
- 10 домашних заданий (до 100 баллов в сумме);
- 9 пятиминутных тестов;
- 1 экзамен;
- 1 пересдача;
- сдать за раз можно максимум 2 (две) ДЗ;
- скринкасты пишутся и выкладываются.

1. Интенсив по Python;
2. HTTP, Web сервера;
3. Django, Application Server;
4. JSON, API, REST;
5. Работа с СУБД;
6. Авторизация;
7. Работа с файлами;
8. Контейнеризация.



- Библиотеки в ЯП: libcurl, urllib и т.д.;
- Консольные утилиты: wget, curl, telnet;
- Роботы: поисковики, вредоносные скрипты;
- Браузеры:
 - Полноценные: firefox, chrome и т.д.
 - Встроенные: web-view, webkit и т.д.

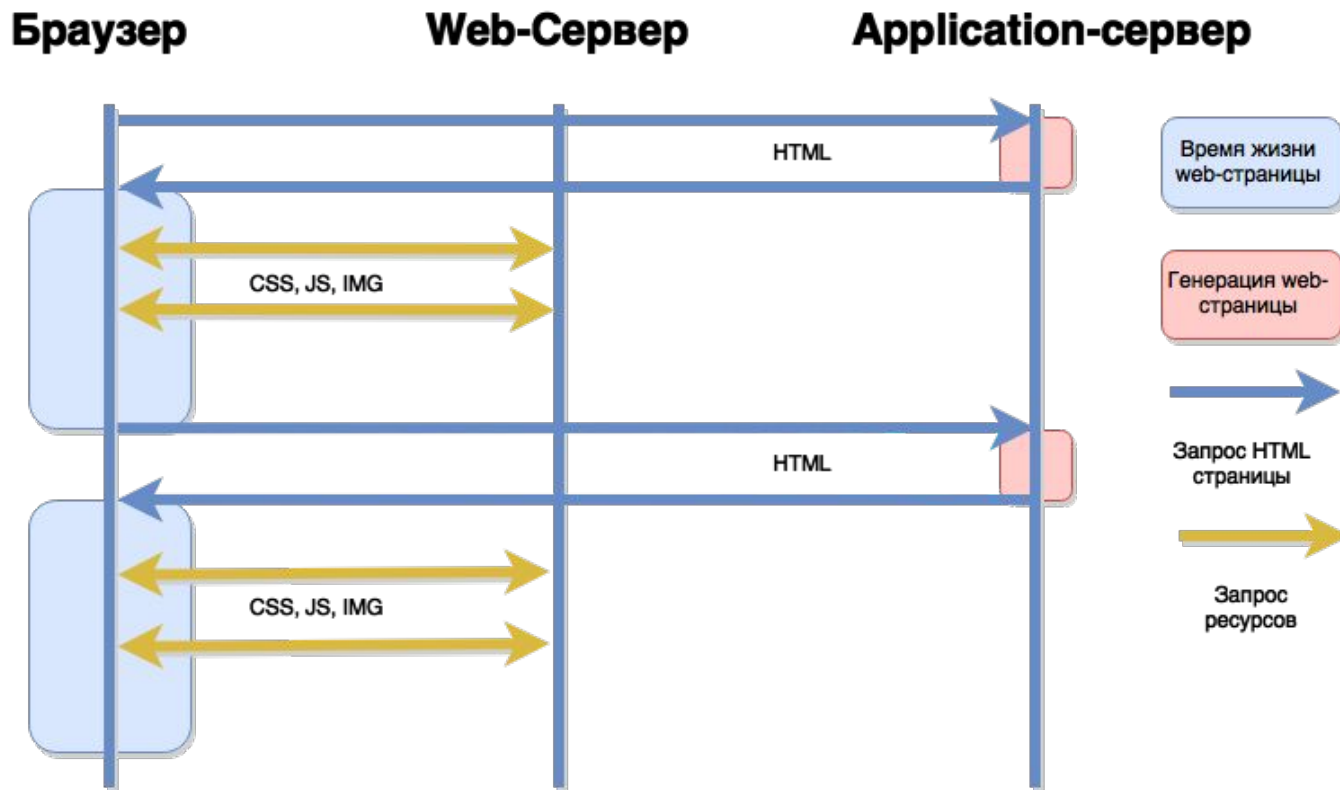
Основное назначение браузера - отображение HTML страниц.

Однако, возможности современных браузеров огромны.

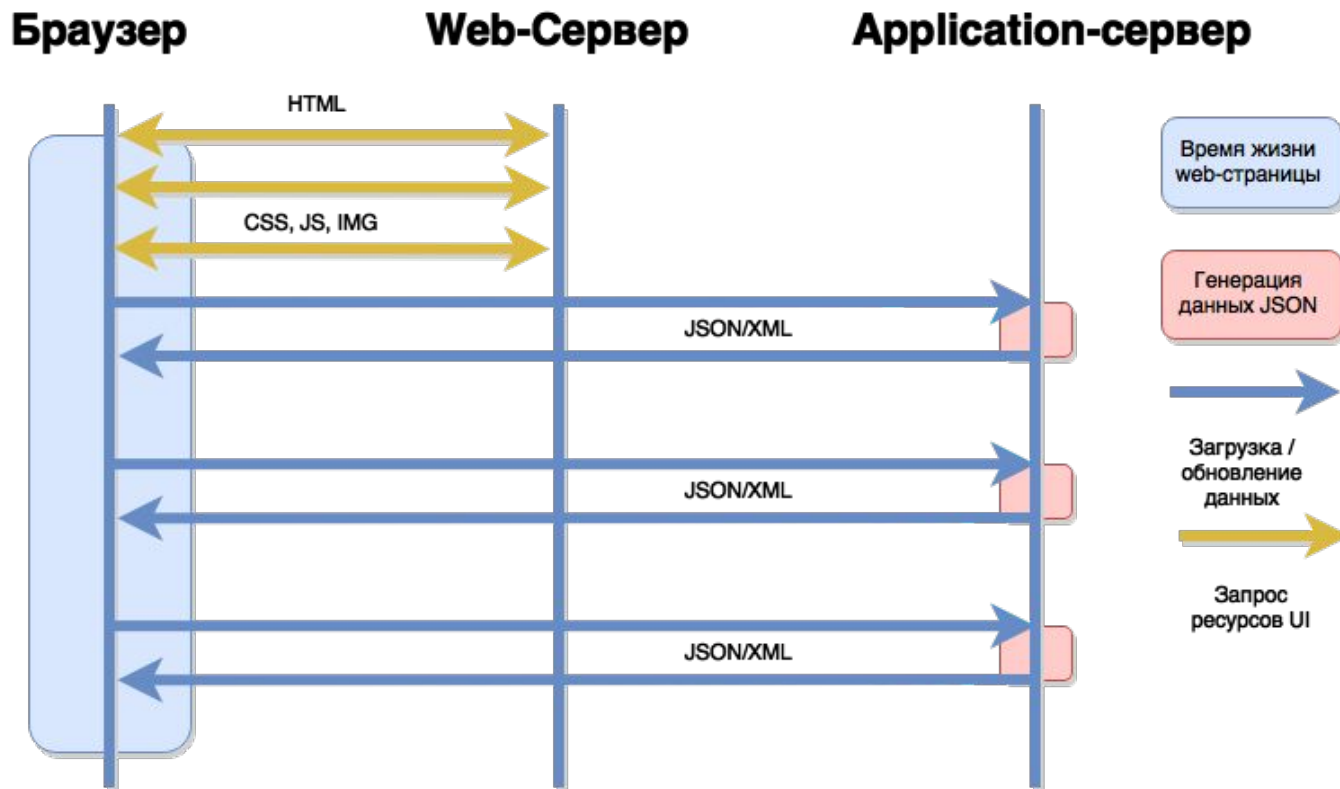
Существуют операционные системы и 3D-игры, работающие внутри браузеров!

www.evolutionoftheweb.com

- Пользователь вводит URL;
- Браузер загружает Web страницу - HTML документ;
- Браузер анализирует (parse) HTML и загружает доп. ресурсы;
- Браузер отображает (rendering) HTML страницу;
- Пользователь переходит по гиперссылке или отправляет форму;
- Цикл повторяется.



- Браузер загружает Web страницу, ресурсы и отображает её;
- JavaScript загружает данные с помощью AJAX запросов;
- JavaScript обеспечивает полноценный UI на странице;
- Пользователь взаимодействует с UI, что приводит к вызову JavaScript обработчиков;
- JavaScript обновляет данные на сервере или загружает новые данные, используя AJAX.



- UI находится на 1 или нескольких страницах (single page)
- UI полностью статичен: HTML, CSS, JS - статические файлы
- Логика UI полностью работает на стороне клиента
- Используется шаблонизация в JavaScript
- Application сервер возвращает чистые данные (JSON или XML, а не HTML)

Скачиваем VirtualBox

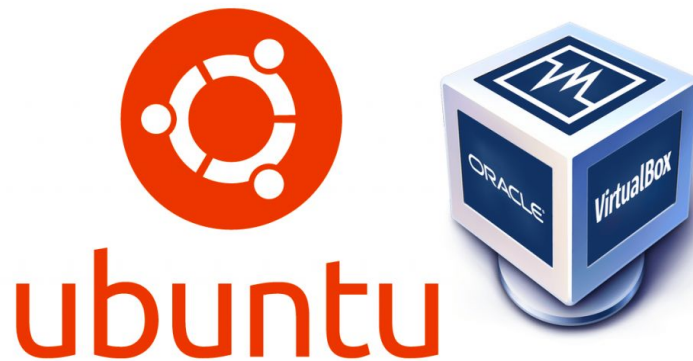
<https://www.virtualbox.org/wiki/Downloads>

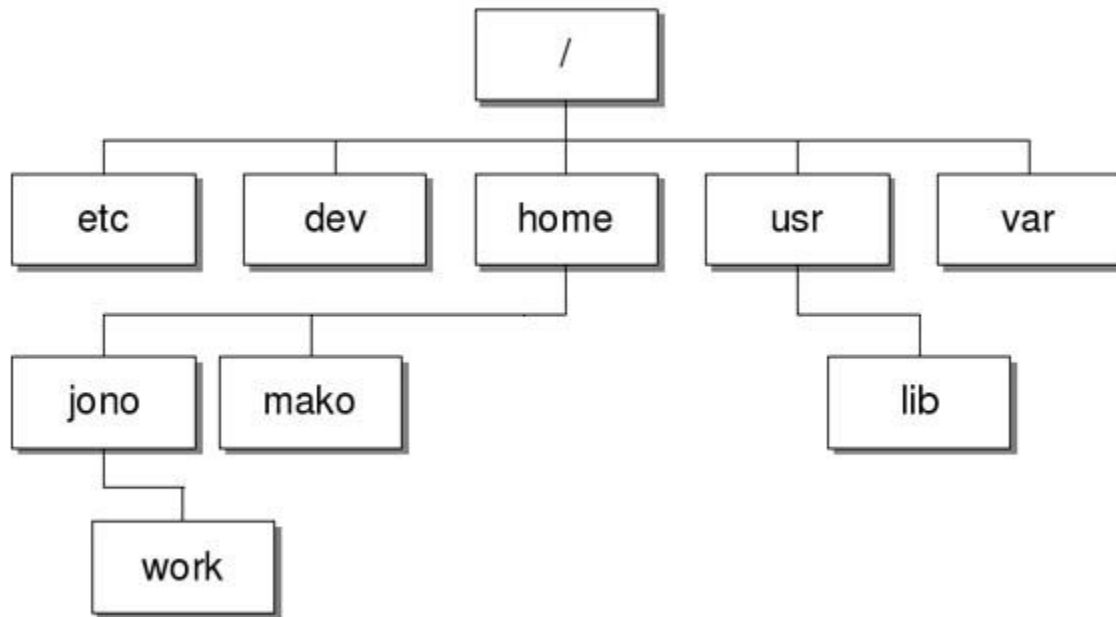
Скачиваем образ Ubuntu

<https://www.ubuntu.com/download/desktop>

...

Profit!





1. `sudo apt install git` # установить известную программу
2. `sudo apt search Django` # поискать среди доступных
3. `sudo apt remove git` # удалить программу
4. `sudo apt purge nginx` # удалить программу и все ее файлы

apt - пакетный менеджер Ubuntu (yum, pacman, emerge...)

sudo - временное повышение привелегий до root

В репозиториях ОС обычно не самые свежие программы.

1. `git init` # создать новый репозиторий
2. `#` или
3. `git clone git@github.com:nuf/quack.git` # клонировать
4. `git status` # посмотреть статус файлов
5. `git add some_file some_dir` # добавить файлы в индекс
6. `git commit` # сформировать новый коммит
7. `git push` # отправить изменения в github
8. `git pull` # получить последние изменения

Шаблон имени:

YYYY-HALF_YEAR-Atom-Backend-N-LAST_NAME

YYYY - год

HALF_YEAR - половина года. 1, если сейчас янв-июн, 2, если июл-дек

N - первая буква имени

LAST_NAME - фамилия

Пример: 2019-2-Atom-Backend-A-Kukhtichev

1. venv/
2. *.pyc
3. *.swp
4. *.swp

1. `# Устанавливаем pip3`
2. `sudo apt install python3-pip`
3. `# А затем уже библиотеки Python`
4. `sudo pip3 install Django==2.0 # в систему`
5. `pip3 install --user py.test # или только для себя`
6. `# Или ...`

```
1. # Переходим в директорию проекта
2. cd /home/nuf/quack
3. # Возможно потребуется сделать ещё вот это вот
4. sudo apt-get install python3-venv
5. # Создаем виртуальное окружение в директории venv
6. python3 -m venv venv
7. # "Активируем" его
8. source ./venv/bin/activate
```

1. `# Устанавливаем необходимые библиотеки в venv`
2. `pip install Django==2.2.5 pytest psycopg2`
3. `# "Запоминаем" версии установленных библиотек`
4. `pip freeze > requirements.txt`

Не забываем закоммитить файл **requirements.txt**, но не директорию **venv**! (она должна быть в **.gitignore**)

1. Завести репозиторий на github - 1 балл;
2. Установить Python (3.6) - 2 балла;
3. Создать виртуальное окружение для Python и установить Django (2.2.5) - 2 балла;
4. Описать зависимости в requirements.txt - 1 балл;
5. Создать простой Hello-World скрипт на Python - 2 балла;
6. Создать правильный .gitignore файл - 1 балл;
7. Оформить изменения в виде отдельных осмысленных коммитов - 1 балл.

Полезные ссылки

[Виртуальное окружение](#)

[Github](#)

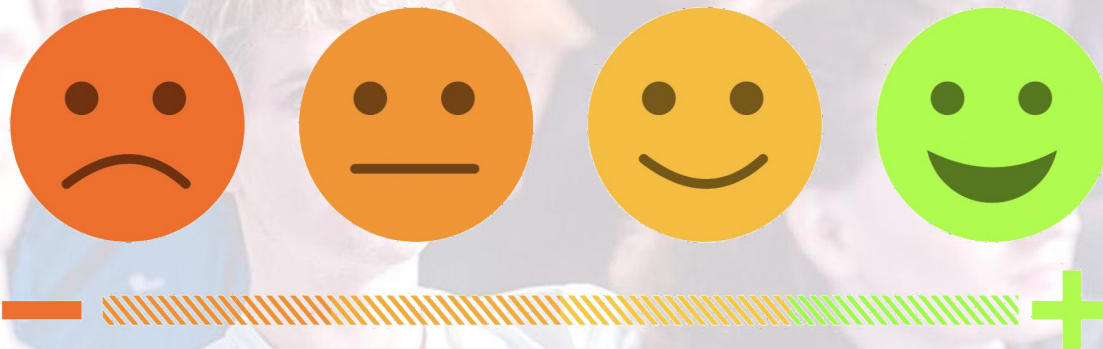
[Virtual Box](#)

[Тут берём LTS Ubuntu \(например 18.04\)](#)

Для саморазвития (опционально)

[Чтобы не набирать двумя пальчиками](#)

Не забудьте оценить занятие!



Спасибо
за внимание!

Антон Кухтичев

a.kukhtichev@corp.mail.ru