## Урок 1. Основы клиент-серверного взаимодействия. Работа с API

### Текст ДЗ:

| ### 1. Посмотреть документацию к API GitHub, разобраться как вывести список репозиториев для конкретного пользователя, сохранить JSON-вывод в файле \*.json.  ### 2. Изучить список открытых API (https://www.programmableweb.com/category/all/apis). Найти среди них любое, требующее авторизацию (любого типа). Выполнить запросы к нему, пройдя авторизацию. Ответ сервера записать в файл.  Если нет желания заморачиваться с поиском, возьмите API вконтакте (https://vk.com/dev/first\_guide). Сделайте запрос, чтобы получить список всех сообществ на которые вы подписаны.  \*\*\*  ### Инструкция к сдаче  Настоятельно рекомендуем сдавать практическое задание в виде ссылки на pull request.  Рекомендуемый способ организации данных в репозитории: создать отдельные папки по темам и помещать в них отдельные файлы для каждой задачи с правильным расширением.  #### Ссылка на инструкцию по работе с git и сдачу практики:  https://docs.google.com/document/d/1RAT\_ukE39iOfbz1xa39QXae2hBUEZ4U6Fko\_wFDdrsM/edit  #### Ссылка на видеокурс по Git:  <https://geekbrains.ru/courses/1117>  Очень интересный сайт для простого изучения Git:  <https://learngitbranching.js.org/?locale=ru_RU>  #### Если остались сложности с системой git, обратитесь к преподавателю или наставнику. |
| --- |

### Выкладываемые материалы:

| * [Методичка 1. Основы клиент-серверного взаимодействия. Работа с API](https://docs.google.com/document/d/1ks8ps8guTJ9dcnqpaNcsVsH1uYykW-w1rSF16uso20o/edit) * [Презентация 1. Основы клиент-серверного взаимодействия. Работа с API](https://docs.google.com/presentation/d/1EDJB5V82uC85CqhzhUrNdO1zCP1C4na8/edit?usp=sharing&ouid=113515512031729575917&rtpof=true&sd=true) * [Методичка 1. Основы компьютерных сетей](https://docs.google.com/document/d/1cjllr8m36iG68ba8ovD40On21ZBoOh0a/edit?usp=sharing&ouid=113515512031729575917&rtpof=true&sd=true) |
| --- |

## Урок 2. Парсинг HTML. Библиотека Beautiful soup.

### Текст ДЗ:

| ## Вариант 1  ###Необходимо собрать информацию о вакансиях на вводимую должность (используем input или через аргументы получаем должность) с сайтов HH(обязательно) и/или Superjob(по желанию). Приложение должно анализировать несколько страниц сайта (также вводим через input или аргументы). Получившийся список должен содержать в себе минимум:  1. Наименование вакансии.  1. Предлагаемую зарплату (разносим в три поля: минимальная и максимальная и валюта. цифры преобразуем к цифрам).  1. Ссылку на саму вакансию.  1. Сайт, откуда собрана вакансия.  По желанию можно добавить ещё параметры вакансии (например, работодателя и расположение). Структура должна быть одинаковая для вакансий с обоих сайтов. Общий результат можно вывести с помощью dataFrame через pandas. Сохраните в json либо csv.  ## Вариант 2  **(не выдаем, росконтроль перешел на динамику)**  ### Необходимо собрать информацию по продуктам питания с сайта: [Список протестированных продуктов на сайте Роскачество](https://rskrf.ru/ratings/produkty-pitaniya/) Приложение должно анализировать несколько страниц сайта (вводим через input или аргументы).  Получившийся список должен содержать:  1. Наименование продукта.  1. Все параметры (Безопасность, Натуральность, Пищевая ценность, Качество) Не забываем преобразовать к цифрам  1. Общую оценку  1. Сайт, откуда получена информация.  Общий результат можно вывести с помощью dataFrame через Pandas. Сохраните в json либо csv. |
| --- |

### Выкладываемые материалы:

| * [Методичка 2. Парсинг HTML. Библиотека Beautiful Soup](https://docs.google.com/document/d/1mFxcdjP0NG6cFWDgNsPfzZxt_Nd9D0wWuJCf4ES-fJk/edit) * [Презентация 2. Парсинг HTML.](https://docs.google.com/presentation/d/1TFQ8R5V-Ad4m-VWRwIE-lyYxJ1HYLfHH) [Библиотека](https://docs.google.com/document/d/1mFxcdjP0NG6cFWDgNsPfzZxt_Nd9D0wWuJCf4ES-fJk/edit) [Beautiful Soup](https://docs.google.com/presentation/d/1TFQ8R5V-Ad4m-VWRwIE-lyYxJ1HYLfHH) * [Информация по CSS селекторам](https://msiter.ru/tutorials/css-nachalnogo-urovnya/selektory-svoistva-znacheniya) * [Дополнительные возможности CSS](https://naikom.ru/blog/archives/2306) * [Еще про CSS](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Selectors) |
| --- |

## Урок 3. Система управления базами данных MongoDB в Python

### Текст ДЗ:

| ### 1. Развернуть у себя на компьютере/виртуальной машине/хостинге MongoDB и реализовать функцию, которая будет добавлять только новые вакансии/продукты в вашу базу.  ### 2. Написать функцию, которая производит поиск и выводит на экран вакансии с заработной платой больше введённой суммы (необходимо анализировать оба поля зарплаты). Для тех, кто выполнил задание с Росконтролем - напишите запрос для поиска продуктов с рейтингом не ниже введенного или качеством не ниже введенного (то есть цифра вводится одна, а запрос проверяет оба поля) |
| --- |

### Выкладываемые материалы:

| * [Методичка 3. СУБД MongoDB](https://docs.google.com/document/d/1OvtCj78OpTQDOmfoTGHMIZ2x-UonCT-PYiyRJ9--IO8/edit#heading=h.gjdgxs) * [Презентация 3. СУБД MongoDB](https://docs.google.com/presentation/d/159hpZw0GvGSgpFN35P2zhUZ-jrfAzHWR/edit?usp=sharing&ouid=113515512031729575917&rtpof=true&sd=true) * [Инструкция по установке и подключению к облачному кластеру MongoDB Atlas](https://youtu.be/lE3xnfQtjVk) (пока введены санкции, Атлас не работает. Так что бесполезно) |
| --- |

## Урок 4. Парсинг HTML. Xpath.

### Текст ДЗ:

| 1. Написать приложение, которое собирает основные новости с сайта на выбор [news.mail.ru](https://news.mail.ru/), [lenta.ru](https://lenta.ru/), [yandex-новости](https://yandex.ru/news/). Для парсинга использовать XPath. Структура данных должна содержать:  \* название источника;  \* наименование новости;  \* ссылку на новость;  \* дата публикации.  2. Сложить собранные новости в БД  Минимум один сайт, максимум - все три |
| --- |

### Выкладываемые материалы:

| * [Методичка 4. Парсинг HTML, XPath](https://docs.google.com/document/d/1EptKAOCS2mjcBsGORzTFdUX_WnV1OxxrsJIZx_m-XPo/edit) * [Презентация 4. Парсинг HTML. XPath](https://docs.google.com/presentation/d/1laOzR-ZetZiXRQ6Hn7Wa2cBDqVFVIXG5/edit?usp=sharing&ouid=113515512031729575917&rtpof=true&sd=true) |
| --- |

## Урок 5. Selenium в Python.

### Текст ДЗ:

| ###Вариант I  Написать программу, которая собирает входящие письма из своего или тестового почтового ящика и сложить данные о письмах в базу данных (от кого, дата отправки, тема письма, текст письма полный)  Логин тестового ящика: study.ai\_172@mail.ru  Пароль тестового ящика: NextPassword172#  ###Вариант II  2) Написать программу, которая собирает товары «В тренде» с сайта техники mvideo и складывает данные в БД. Сайт можно выбрать и свой. Главный критерий выбора: динамически загружаемые товары |
| --- |

### Выкладываемые материалы:

| * [Методичка 5. Selenium в python](https://docs.google.com/document/d/1RSBog41yeYs6glCJsEkQh2_ME8XXw_M5MWGU5wwRXJ8/edit) * [Презентация 5. Selenium в python](https://docs.google.com/presentation/d/1bPoUToLO3kuvt19bOzQo5gxQLl625GPB/edit?usp=sharing&ouid=113515512031729575917&rtpof=true&sd=true) |
| --- |

## Урок 6. Фреймворк Scrapy. Знакомство

### Текст ДЗ:

| \*\*I вариант\*\*  1) Доработать паука в имеющемся проекте, чтобы он формировал item по структуре:  \*Наименование вакансии  \*Зарплата от  \*Зарплата до  \*Ссылку на саму вакансию  И складывал все записи в БД(любую)  2) Создать в имеющемся проекте второго паука по сбору вакансий с сайта superjob. Паук должен формировать item'ы по аналогичной структуре и складывать данные также в БД  \*\*II вариант\*\*  1) Создать пауков по сбору данных о книгах с сайтов labirint.ru и/или book24.ru  2) Каждый паук должен собирать:  \* Ссылку на книгу  \* Наименование книги  \* Автор(ы)  \* Основную цену  \* Цену со скидкой  \* Рейтинг книги  3) Собранная информация должна складываться в базу данных |
| --- |

### Выкладываемые материалы:

| * [Методичка 6/7. Парсинг данных: Scrapy](https://docs.google.com/document/d/1Uzd2IqNUrfSDQ8M-PqzoGzfjJhxT_5p-UVQUmxISlNs/edit) * [Видео по функциям-генераторам и yield](https://youtu.be/FfuhhNZ1oHY) |
| --- |

## Урок 7. Фреймворк Scrapy. Скачивание файлов и фото

### Текст ДЗ:

| 1) Взять любую категорию товаров на сайте Леруа Мерлен. Собрать следующие данные:  ● название;  ● все фото;  ● ссылка;  ● цена.  Реализуйте очистку и преобразование данных с помощью ItemLoader. Цены должны быть в виде числового значения.  \*\*\*  Дополнительно:  2)Написать универсальный обработчик характеристик товаров, который будет формировать данные вне зависимости от их типа и количества.  3)Реализовать хранение скачиваемых файлов в отдельных папках, каждая из которых должна соответствовать собираемому товару |
| --- |

### Выкладываемые материалы:

| * [Методичка 6/7. Парсинг данных: Scrapy](https://docs.google.com/document/d/1Uzd2IqNUrfSDQ8M-PqzoGzfjJhxT_5p-UVQUmxISlNs/edit) * [Работа с файлами и фото в scrapy](https://docs.scrapy.org/en/latest/topics/media-pipeline.html) * [Видео по сбору данных с сайта недвижимости cian (scrapy+selenium)](https://youtu.be/p0Lx7DOaaok) * [Проект для сбора недвижимости с сайта cian](https://drive.google.com/file/d/1nst2QEjpPBKcCLC0ZPF2l2jnWR6LFdDz/view?usp=sharing) |
| --- |

## Урок 8. Фреймворк Scrapy. Реализация механизмов клиент-серверного взаимодействия

### Текст ДЗ:

| 1) Написать приложение, которое будет проходиться по указанному списку двух и/или более пользователей и собирать данные об их подписчиках и подписках.  2) По каждому пользователю, который является подписчиком или на которого подписан исследуемый объект нужно извлечь имя, id, фото (остальные данные по  желанию). Фото можно дополнительно скачать.  4) Собранные данные необходимо сложить в базу данных. Структуру данных нужно заранее продумать, чтобы:  5) Написать запрос к базе, который вернет список подписчиков только указанного пользователя  6) Написать запрос к базе, который вернет список профилей, на кого подписан указанный пользователь  \*\*\*  Для выполнения данной работы необходимо делать запросы в апи инстаграм с указанием заголовка User-Agent : 'Instagram 155.0.0.37.107' |
| --- |

### Выкладываемые материалы:

| * [Методичка 8.](https://docs.google.com/document/d/1SKS9YrlvUA8OEx0PkqRge2AVipxsO1N3RhDwEKXQeNE/edit#) Фреймворк Scrapy. Реализация механизмов клиент-серверного взаимодействия(не обновлена. Внутри информация о OpenData)) * [8 урок по opendata и парсингу Telegram](https://gbcdn.mrgcdn.ru/uploads/record/59179/attachment/b36578e7480d42eb12674ef85481380a.mp4) * [Пайплайн для автоматического создания собственного csv-датасета на основе собранных данных в scrapy](https://drive.google.com/file/d/1xmz5qsVp_UvHmhXbr3rXTaBckIqQO8ds/view?usp=sharing) * [Код скрапера Telegram](https://drive.google.com/file/d/1eo7omfs1LXS71R-1AQN5W-heMswgi80m/view?usp=sharing) * [Бонусный 9 урок. Middleware и сбор данных с помощью модуля scrapy-selenium](https://gbcdn.mrgcdn.ru/uploads/record/126276/attachment/8a86b082c35618410f8164111371f987.mp4) |
| --- |