Программа курса Python разработки

Получено: 16 Июля 2022 года

|  |  |
| --- | --- |
| Начало. Установка Python и PyCharm | Полезные материалы:   1. <https://younglinux.info/python/input> - Ввод и вывод данных 2. <https://pythontutor.ru/lessons/inout_and_arithmetic_operations/> - Ввод и вывод данных. Работа с числами   Домашнее задание: Написать программу выводящую ответ на пример: 2\*(5+637)/(628\*3)-89 |
| Начало. Операции с числами | Полезные материалы:   1. <https://pythonworld.ru/tipy-dannyx-v-python/chisla-int-float-complex.html> - Работа с числами и переменными 2. <https://metanit.com/python/tutorial/2.3.php> - Работа с числами и переменными   Домашнее задание:  >>Запрашиваем число x  <<Счиатем уравнение x^2 + 48x - 19(Вместо x число пользователя)  >>Выводим результат |
| Начало. Условные операторы(if, elif, else) | Полезные материалы:   1. <https://pythonworld.ru/osnovy/instrukciya-if-elif-else-proverka-istinnosti-trexmestnoe-vyrazhenie-ifelse.html> - Теория 2. <https://pythonru.com/uroki/uslovnye-vyrazhenija-i-konstrukcija-if-uroki-po-python-dlja-nachinajushhih> - Теория 3. <https://smartiqa.ru/python-workbook/if-else> - Задачник по условиям   Домашнее задание:   * >>Пользователь вводит число * <<Программа проверяет его на чётность * >>Выводит True есл оно чётное, False если не чётное |
| Начало. Условные операторы | Полезные материалы:   1. <https://letpy.com/python-guide/conditional-flow-control/> - Игра с угадыванием числа   Домашнее задание:   * >>Пользователь вводит здоровье игрока * >>Пользователь вводит наличие святой воды * >>Если у пользователя меньше либо 0 здоровья и нету святой воды, то печатаем что он мёртв * >>Иначе печатаем пользователь жив |
| Начало. Условные операторы | Домашнее задание:   * <<Пользователь вводит целое число. * >>Выведите его строку-описание вида "отрицательное четное число", "нулевое число", "положительное нечетное число", например, численным описанием числа 190 является строка "положительное четное число". |
| Циклы, списки | Полезные материалы:   1. <https://pythonworld.ru/osnovy/cikly-for-i-while-operatory-break-i-continue-volshebnoe-slovo-else.html> - Циклы 2. <https://pythontutor.ru/lessons/for_loop/> - Циклы 3. <https://pythonworld.ru/tipy-dannyx-v-python/spiski-list-funkcii-i-metody-spiskov.html> - Списки   Домашнее задание:   1. Пользователь вводит число. Определить, содержит ли список данное число x. Если содержит, то вывести на экран число 7145, если не содержит, то вывести на экран число 5741; 2. Найдите сумму и произведение элементов списка. Результаты вывести на экран; |
| Циклы | Полезные материалы:   1. <https://pythonru.com/osnovy/spiski-python-funkcii-i-metody-python-list> - Списки 2. https://python-scripts.com/loops-for-while - Циклы   Домашнее задание: Определить, какая цифра числа больше(Пример: в числе 7468737 наибольшая цифра 8) |
| Циклы. Срезы | Полезные материалы:   1. <https://pythonworld.ru/osnovy/indeksy-i-srezy.html> - Срезы 2. <https://ru.hexlet.io/courses/python-lists/lessons/slices/theory_unit> - Срезы   Домашнее задание: Заполнить список 1000 случайных чисел. И вернуть значения в промежутке с 371 до 782 |
| Задачи на циклы и условия | Домашнее задание: Бизнесмен взял ссуду m тысяч рублей в банке под 20% годовых. Через сколько лет его долг превысит s тысяч рублей, если за это время он не будет отдавать долг? |
| Задачи на циклы и условия | Полезные материалы:   1. <https://cleancitygroup.ru/zadachi-po-teme-cikly-s-usloviem/> - Сборник задач для практики   Домашнее задание: Каждый год урожайность повышается на 5%. Через сколько лет урожай удвоится? |
| Установка и работа с библиотеками. PIP | Полезные материалы:   1. <https://pythonworld.ru/osnovy/pip.html> - Работа с PIP 2. <https://tproger.ru/explain/pip-kak-ustanavlivat-pakety-v-python/> - Работа с PIP 3. <https://pypi.org> - Список библиотек для PIP 4. <https://pythonworld.ru/moduli/modul-time.html> - Библиотека time   Домашнее задание:   * << Пользователь вводит от какого числа считать * << Пользователь вводи до какого числа считать * >>Программа выводит РАЗ в секунду число(Сколько уже прошло секунд) |
| Практика работы с библиотеками | Домашнее задание:   * <<Программа 'придумывает' число в заданной уровнем сложности последовательности  1. Новичок(1) - от 0 до 10 2. Спортсмен(2) - от 0 до 25 3. Мастер(3) - от 0 до 50 4. Зверь(4) - от 0 до 100  * >>Пользователь вводит число и программа печатает угадал он или нет * <<Программа выполняется до тех пор пока пользователь не угадает число * Если пользователь справился за менее 3 попыток, вывести: Удача на твоей стороне! В любом другом случае вывести число потраченных попыток |
| Работа с файлами | Полезные материалы:   1. <https://pythonworld.ru/tipy-dannyx-v-python/fajly-rabota-s-fajlami.html> 2. <https://istories.media/workshops/2021/01/29/vvedenie-v-python-chast-11-rabota-s-failami/> 3. <https://tproger.ru/articles/files-in-python/>   Домашнее задание: Напишите программу, которая будет открывать определенный файл и выводить на печать построчно последние строки в количестве lines (на всякий случай проверим, что задано положительное целое число).  *\*Строки(lines) передаёт пользователь* |
| Файлы - практика #1 | Полезные ссылки:   1. <https://pythonru.com/osnovy/fajly-v-python-vvod-vyvod> 2. https://python-scripts.com/work-with-files-python   Домашнее задание: Напишите проверку на то, является ли строка палиндромом. Палиндром — это слово или фраза, которые одинаково читаются слева направо и справа налево. Слова находится в файле |
| Файлы - практика #2 | Полезные ссылки:   1. <https://codernet.ru/books/software_development/grokaem_algoritmy/> - Книга 'Грокаем Алгоритмы'   Домашнее задание: Написать программу, принимающая 2 аргумента: строку, которую нужно зашифровать, и ключ шифрования. Программа возвращает строку, зашифрованную путем применения алгоритма XOR (^) над символами строки с ключом(Зашифрованную строку хранить в файле). Написать также возможность, по зашифрованной строке и ключу восстанавливать исходную строку. |
| Практика по алгоритма #1 | Домашнее задание:   1. Доделать программу по получению все возможных комбинаций из заданных чисел. 2. Узнать о "Большом О"/Оценке сложности алгоритма, по книге "Грокаем Алгоритмы" или аналогичным статьям(<https://habr.com/ru/post/104219/>) |
| Исключения в Python. Создание собственных функций | Полезные ссылки:   1. <https://pythonworld.ru/tipy-dannyx-v-python/isklyucheniya-v-python-konstrukciya-try-except-dlya-obrabotki-isklyuchenij.html> - Исключения 2. <https://pythonworld.ru/tipy-dannyx-v-python/vse-o-funkciyax-i-ix-argumentax.html> - Функции 3. <https://pythonru.com/osnovy/funkcii-v-python> - Функции   Домашнее задание: Посчитать количество одинаковых элементов в списке(Список берётся из файла(Пример: в файле записанно число 172727, программа должна преобразовать это число в список и найти кол-во одинаковых аргументов)) |
| Функции - Практика #1. if \_\_name\_\_ == '\_\_main\_\_' | Полезные материалы:   1. <https://coderoad.ru/419163/Что-делает-if-__name__-__main__-do>   Напишите функцию, которая принимает имя файла и выводит его расширение. Если расширение у файла определить невозможно, выбросите исключение. Также выведите все данные из файла в консоль |
| Функции - Практика #2. Анотации типов | Полезные материалы:   1. <https://fastapi.tiangolo.com/ru/python-types/>   Домашнее задание: Напишите функции для расчета площади и периметра окружности. Итоговые значения(Площади и периметра) сохранить в файл |
| Функции - Практика #3 | Полезные материалы:   1. <https://pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html> - Стандарты кода PEP8   Домашнее задание: Дмитрий считает, что когда текст пишут в скобках (как вот тут, например), его читать не нужно. Вот и надумал он существенно укоротить время чтения, написав функцию, которая будет удалять все, что расположено внутри скобок. Помогите ленивому Диме разработать функцию shortener(st), которая будет удалять все, что внутри скобок и сами эти скобки, возвращая очищенный текст (скобки могут быть вложенными). |
| Функции - Практика #4 | **Домашнее задание**: Николай решил вспомнить старые времена. В свое время было модно писать сообщения с чередующимися заглавной и малой буквами. Он захотел изобрести функцию, которая будет делать с любой предоставленной строкой аналогичное. Ваша задача: повторить труд студента camel(st) с учетом того, что пробелы и знаки препинания не должны портить чередование регистра символов (они в этом процессе не учитываются, но возвращаются в итоговой строке). |
| Функции - Практика #5 | Домашнее задание:  Напишите программу на Python для расчета количества дней между двумя датами.   * *Даты выборки* : (2014, 7, 2), (2014, 7, 11) * *Ожидаемый выход* : 9 дней |
| Словари и рекурсия | Полезные материалы:   1. <https://pythonworld.ru/tipy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html> - Словари 2. <https://pythontutor.ru/lessons/functions/> - Рекурсия 3. <https://foxford.ru/wiki/informatika/rekursiya-v-python> - Рекурсия   Домашнее задание: (Фотография в прикреплённом файле) |
| Задача - Пирамида | Дополнительных материалов нету! |
| Задача - Пирамида | Дополнительных материалов нету! |
| Задача - Пирамида(Конец) | Домашнее задание: Красиво оформить программу Пирамида |
| Генераторы списков, Декораторы | Полезные материалы:   1. <https://pythoner.name/list-comprehension> - Генераторы списков 2. <https://tproger.ru/translations/demystifying-decorators-in-python/> - Декораторы 3. <https://pythonworld.ru/osnovy/dekoratory.html> - Декораторы   Домашнее задание: Написать свой декоратор для функции(У каждого разный). Не писать декоратор подсчёта времени выполнения функции и проверки на чётность/нечётность |
| Повторение | Дополнительных материалов нету! |
| Итоговая работа за первый СверхМодуль(Старт) | Домашнее задание: доделать итоговую работу! |
| ООП - Начало #1 | Полезные материалы:   1. <https://itproger.com/course/python/17> - Статья 2. <https://clck.ru/YTmX5> - Книга 3. <https://www.youtube.com/watch?v=irf2ekfkK0Q> - Видео(2 часа) |
| ООП - Начало. Метод \_\_init\_\_ #2 | Полезные материалы:   1. <https://itproger.com/course/python/17> - Статья 2. <https://clck.ru/YTmX5> - Книга 3. <https://www.youtube.com/watch?v=irf2ekfkK0Q> - Видео(2 часа) |
| ООП - Начало. Инкапсуляция и Наследование #3 | Домашнее задание:   1. <https://www.youtube.com/watch?v=M58eiYbM6AE> 2. <https://www.youtube.com/watch?v=EvGi6XDgV7w> 3. <https://www.youtube.com/watch?v=eI0XzQw3V0Q&t=335s> 4. Подготовится к опросу по этой теме |
| ООП - Начало. Полиморфизм #4 | Полезные материалы:   1. <https://www.youtube.com/watch?v=Ay_GwOQWPs8&t=7s>   Домашнее задание: Переписать практическую работу с функций на классы и подготовиться к опросу по Полиморфизму |
| Python ООП - Средний уровень | Полезные материалы:   1. <https://www.youtube.com/watch?v=F7KxC6JTmsc> 2. <https://habr.com/ru/post/186608/>   Домашнее задание: Применить(Если возможно) магические методы к практической работе |
| Доработка программы. Создание собственных исключений | Полезные материалы:   1. <https://habr.com/ru/company/piter/blog/537642/> - Создание своих исключений   Домашнее задание: Сделать шифрование паролей при регистрации и авторизации, нарисовать Класс Диаграмму своей программы |
| Доработка программы. Приведение структуры к требованию UML диаграммы | Домашнее задание: Доделать программу и попрактиковаться с созданием UML диаграмм |
| ООП - Абстракция, агреция, композиция | Полезные материалы:   1. <https://habr.com/ru/post/354046/> 2. <https://pythobyte.com/abstraction-in-python-2759e6c7/>   Домашнее задание: доделать программу применив все новые знания по принципам ООП |
| Базы данных. Начало - CREATE, DROP, ALTER | Полезные материалы:   1. <https://www.oracle.com/ru/database/what-is-database/> - Что такое БД и какими они бывают 2. <https://russianblogs.com/article/73441278572/> - Запросы   Домашнее задание: Выучить показанные на уроке запросы(Создание таблицы, удаление таблицы, добавление и удаление столбцов из таблицы), скачать и установить на свой компьютер базу данных MySQL |
| Базы данных. CRUD. Create, Select, Update, Delete/Drop | Полезные материалы:   1. <https://bestprogrammer.ru/izuchenie/chto-takoe-crud-operatsii> 2. <https://clck.ru/dXfrt>     Домашнее задание:   1. Выучить все запросы представленные на уроке 2. Сделать такие SQL запросы:  * Создать любую таблицу. 4-5 столбцов * Заполнить данными (Минимум 3 поля) * Написать запрос для выбора всех столбцов если id в промежутке от 2 до 5 и отсортировать по любому поля кроме id Удалить любой столбец и добавить любой столбец * Удалить таблицу |
| Базы данных. Связи, Индексы. Работа с БД из Python | Полезные материалы:   1. <https://habr.com/ru/post/488054/> - Связи таблиц 2. https://im-cloud.ru/blog/chto-takoe-indeksy-bazy-dannyh-dlja-nachinajushhih/ - Индексы 3. <https://tproger.ru/articles/indeksy-v-postgresql/> - Индексы в PostgreSQL   Домашнее задание: Выучить все запросы представленные на уроке. Применив библиотеку sqlite3/pymysql подключиться к БД из Python |
| База данных. Псевдонимы, GROUP BY | Полезные материалы:   1. <https://pythonworld.ru/osnovy/formatirovanie-strok-operator.html> Форматирование строк. %s 2. <https://tproger.ru/translations/sql-recap/> Стандартные SQL запросы   Домашнее задание: Придумать идею для ВК бота, достаточно интересную. Доделать программу начатую на уроке. Познакомится с Git |
| Бот для ВК. GIT | Полезные материалы:   1. https://vk-api.readthedocs.io/en/latest/ 2. <https://habr.com/ru/post/319178/> 3. <https://javarush.ru/groups/posts/2683-nachalo-rabotih-s-git-podrobnihy-gayd-dlja-novichkov> GIT   Домашнее задание: Понять всё пройденное на уроке(Про работу с GIT и шаблоном для создания ботов) |
| Бот в функциональном стиле. Ограничения при создании ботов в ВК(Макс. кол-во символов и тд) | Домашнее задание: Разобраться с написанным на уроке кодом |
| Создание ботов в ООП стиле | Домашнее задание: Разобраться с написанным на уроке кодом |
| Боты для ВК №1. Генераторы функций, итераторы | Дополнительных материалы:   1. <https://devpractice.ru/python-lesson-15-iterators-and-generators/> 2. <https://habr.com/ru/post/488112/>   Домашнее задание: Доделать команды не требующие параметров(По типу /help, /погода и тп), изучить генераторы и итераторы |
| Парсинг сайтов. Requests BeautifulSoup | Полезные материалы:   1. <https://pythonru.com/biblioteki/parsing-na-python-s-beautiful-soup> Статья про парсинг 2. <https://habr.com/ru/post/544828/> Статья про парсинг 3. <https://www.youtube.com/watch?v=zKuBDil5dlw> Видео урок №1 4. <https://www.youtube.com/watch?v=yJcCKuxfb2o&list=PLM6XATa8CAG4F9nAIYNS5oAiPotxwLFIr> Курс HTML/CSS   Домашнее задание: Начать смотреть серию видео про HTML/CSS, разобраться с темой парсинга сайтов |
| Создание бота для ВК №2. Создание своего парсера | Полезные материалы:   1. <https://nuancesprog.ru/p/12524/> Интерпритатор/Компилятор   Домашнее задание: доделать парсер для своего проекта |
| Создание ботов для ВК №3 | Домашнее задание: Работать над кодом парсера |
| Асинхронность, декораторы staticmethod, classmethod | Полезные материалы:   1. <https://highload.today/asinhronnost-v-python-kak-twitter-obrabatyvaet-milliardy-seansov-v-den/> - Асинхронность 2. <https://habr.com/ru/post/421625/> - Асинхронность 3. <https://coderoad.ru/136097/Разница-между-staticmethod-и-classmethod> - Декораторы staticmethod, classmethod   Домашнее задание: Доделать парсер, разобраться с декораторами и асинхронностью |
| Многопоточность. Создание бота для ВК №4 | Полезные материалы:   1. https://dev-gang.ru/article/mnogopotocznost-v-python-t2bkyunvku/ - Многопоточность 2. <https://thecode.media/python-time-2/> - Многопоточность   Домашнее задание: Разобраться в многопочности и понять её отличие от асинхронности |
| Функция super, \*args и \*\*kwargs, lambda. Создание бота для ВК №5 | Полезные материалы:   1. <https://habr.com/ru/company/ruvds/blog/482464/> - args, kwargs 2. https://docs-python.ru/tutorial/vstroennye-funktsii-interpretatora-python/funktsija-super/ - super() 3. <https://webdevblog.ru/kak-ispolzovat-v-python-lyambda-funkcii/> - lambda   Домашнее задание: повторить теорию изученную на уроке. Продолжить создание бота |
| Создание бота ВК №6 | Дополнительных материалов нету! |
| Создание бота ВК №7 | Дополнительных материалов нету! |
| Создание бота ВК №8 | Дополнительных материалов нету! |
| Создание бота ВК №9 | Дополнительных материалов нету! |
| Создание бота ВК №10 | Дополнительных материалов нету! |
| Создание бота ВК №11 | Дополнительных материалов нету! |
| Создание бота ВК №12 | Дополнительных материалов нету! |
| Создание бота ВК №13 | Дополнительных материалов нету! |
| Создание бота ВК №14. Клавиатуры ВК. VkKeyboard | Полезные материалы:   1. https://vk-api.readthedocs.io/en/latest/keyboard.html   Домашнее задание: добавить клавиатуру ВК в свой проект |
| Работа в линукс терминале. Редактор VIM | Полезные материалы:   1. <https://selectel.ru/blog/basic-linux-commands/> - Основные команды Линукс 2. <https://jino.ru/journal/articles/proframmirovanie-v-vim/> - Как работать с vim   Домашнее задание: выучить команды cd, ls, sudo, vim. Просмотреть видеокурс по HTML, CSS на канале "Фрилансер по жизни" |
| Создание бота ВК №15 | Домашнее задание: Закончить создания ±50% функциональности бота |
| Создание бота ВК №16 | Дополнительных материалов нету! |
| Урок по JavaScript №1 | Полезные материалы:   1. <https://learn.javascript.ru> 2. <https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps>   Домашнее задание: Закрепить пройденное на уроке |
| Урок по JavaScript №2. Работа с DOM деревом | Полезные материалы:   1. <https://learn.javascript.ru/dom-nodes> 2. <https://tproger.ru/translations/dom-javascript/>   Домашнее задание: Попробовать написать свой сайт с использованием изученных материалов |
| Создание бота ВК №17 | Дополнительных материалов нету! |
| Создание бота ВК №18 | Дополнительных материалов нету! |
| Создание бота ВК №19 | Домашнее задание: Завершить создание бота. Отправить ссылку на код своего проекта(GitHub) до 30.12.21 12:00 |
| Создание бота ВК №20 | Дополнительных материалов нету! |
| Code-Review ВК Бота | Дополнительных материалов нету! |
| Принципы SOLID | Полезные материалы:   1. <https://medium.com/webbdev/solid-4ffc018077da> - Статья про SOLID   Домашнее задание: Придумать идею для оконного приложения |
| Создание оконного приложения. Tkinter | Полезные материалы:   1. <https://pythonru.com/uroki/obuchenie-python-gui-uroki-po-tkinter> 2. <https://younglinux.info/tkinter/tkinter> |
| Создание оконного приложения №2. Tkinter | Полезные материалы:   1. https://python-scripts.com/tkinter 2. <http://rassyhaev.ru/статьи/python-3/модуль-tkinter-урок-1-введение/> |
| Создание оконного приложения №3 Tkinter | Дополнительных материалов нету! |
| Создание оконного приложения №4 Tkinter | Дополнительных материалов нету! |
| Создание оконного приложения №5 Tkinter | Дополнительных материалов нету! |
| Создание оконного приложения №6 Tkinter | Дополнительных материалов нету! |
| Создание оконного приложения №7 Tkinter. UML Диаграммы | Полезные материалы:   1. [https://habr.com/ru/post/511798](https://habr.com/ru/post/511798/) - Что такое UML диаграммы и как их создавать 2. [https://creately.com/ru](https://creately.com/ru/lp/инструмент-создания-диаграмм-UML-онлайн/) - Сайт для создания UML диаграмм   Домашнее задание: Создать UML диаграмму своего проекта |
| Создание оконного приложения №8 Tkinter | Дополнительных материалов нету! |
| Создание оконного приложения №9 Tkinter | Дополнительных материалов нету! |
| Создание оконного приложения №10 Tkinter. многоуровневая архитектура | Полезные материалы:   1. <https://ru.wikipedia.org/wiki/Многоуровневая_архитектура> - Википедия 2. <https://habr.com/ru/post/319126/> - Чистая архитектура в Python(Статья не для новичка) |
| Создание оконного приложения №11 Tkinter. Многоуровневая и многослойная архитектура | Полезные материалы:   1. <https://studfile.net/preview/7071621/> |
| Создание оконного приложения №12 Tkinter | Дополнительных материалов нету! |
| Создание оконного приложения №13 Tkinter. Паттерн Синглтон | Полезные материалы:   1. <https://refactoring.guru/ru/design-patterns/singleton> 2. <https://pythobyte.com/how-to-create-a-singleton-in-python-4dfcc720/> |
| Создание оконного приложения №14 Tkinter | Дополнительных материалов нету! |
| Создание оконного приложения №15 Tkinter. README, Лицензии GitHub | Полезные материалы:   1. <https://docs.microsoft.com/ru-ru/contribute/markdown-reference> - README файл 2. <https://tproger.ru/articles/whats-difference-between-licenses/> - Отличия популярных лицензий |
| Логирование. Паттерн "Стратегия" | Полезные материалы:   1. <https://russianblogs.com/article/29271459871/> - Loguru(Логирование) 2. <https://habr.com/ru/post/210288/> - Шпаргалка по паттернам 3. <https://refactoring.guru/ru/design-patterns/strategy/python/example> - Паттерн "Стратегия"   Домашнее задание: Попрактиковаться с легированием. Запомнить и понять паттерн "Стратегия" |
| Создание оконного приложения №16 Tkinter | Дополнительных материалов нету! |
| Создание оконного приложения №17 Tkinter. Финал | Домашнее задание: Закончить проект и отправить проект учителю для Code-Review |
| Code-Review приложения Tkinter. Паттерн Адаптер | Полезные материалы:   1. <https://refactoring.guru/ru/design-patterns/adapter/python/example#lang-features> - Паттерн Адаптер |
| Создание игры на PyGame №1. | Полезные материалы:   1. <https://pythonru.com/uroki/biblioteka-pygame-chast-1-vvedenie> - PyGame 2. <https://habr.com/ru/post/347138/> - PyGame |
| Создание игры на PyGame №2(Работа с картинками). Паттерн Фасад | Полезные материалы:   1. https://python-course.readthedocs.io/projects/elementary/en/latest/lessons/18-pygame.html - Доки PyGame 2. <https://refactoring.guru/ru/design-patterns/facade/python/example> - Паттерн Фасад |
| Создание игры на PyGame №3(Спрайты, музыка) | Полезные материалы:   1. <https://pythonru.com/uroki/biblioteka-pygame-chast-2-rabota-so-sprajtami> |
| Создание игры на PyGame №4. Паттерн Прокси | Полезные материалы:   1. <https://refactoring.guru/ru/design-patterns/proxy> |
| Создание игры на PyGame №5. Паттерн Декоратор | Полезные материалы:   1. <https://refactoring.guru/ru/design-patterns/decorator> 2. <https://habr.com/ru/post/86255/> |
| Создание игры на PyGame №6 | Дополнительных материалов нету! |
| Просмотр проучившихся проектов PyGame | Дополнительных материалов нету! |
| Завершение PyGame | Домашнее задание: Оформить получившуюся игру на GitHub(Добавить README, закоментирвать и задокументировать код) |
| Как работает интернет? Теория | Полезные материалы:   1. <https://developer.mozilla.org/ru/docs/Learn/Common_questions/How_does_the_Internet_work> 2. <https://thecode.media/how-internet-works-1/> 3. [Теория от меня(\*кликни\*](/media/uploads/2022/03/07/ppzavy.pdf)) |
| Протоколы, модель OSI. Теория | Полезные материалы:   1. <https://otus.ru/nest/post/1919/> - Протоколы 2. <https://selectel.ru/blog/osi-for-beginners/> - Модель OSI 3. [Теория от меня(\*кликни\*)](/media/uploads/2022/03/07/jqdasa.pdf)   Домашнее задание: Выучить теорию за этот и предыдущий урок |
| Читаем код библиотек. CodeWars | Полезные материалы:   1. <https://techrocks.ru/2017/10/13/improving-code-reading-skills/> - Как читать чужой код 2. <https://ru.hexlet.io/blog/posts/how-to-read-code-8-principles> - 8 принципов для чтения чужого кода 3. <https://ru.hexlet.io/blog/posts/pervyy-opyt-na-codewars> - Про CodeWars   Домашнее задание: Решить 3 задачи(7кю) или 5 задач(8кю) с codewars.com |
| Паттерн Memento. Решение задач с CodeWars | Полезные материалы:   1. <https://refactoring.guru/ru/design-patterns/memento> - Паттерн Memento 2. <https://tproger.ru/articles/pattern-oop-hranitel/> - Паттерн Memento   Домашнее задание: Скачать Flask |
| Решение задач CodeWars | Домашнее задание: Сделать 2-3 задания на CodeWars(7-8кю) |
| Flask. Первое приложение | Полезные материалы:   1. <https://pythonru.com/uroki/3-osnovy-flask> 2. <https://itproger.com/course/flask> |
| Flask №1. Шаблоны, url\_for, abort, redirect | Полезные материалы:   1. <https://habr.com/ru/post/346340/> - Шаблоны 2. <https://pythonru.com/uroki/6-shablony-vo-flask> - Шаблоны 3. https://docs-python.ru/packages/veb-frejmvork-flask-python/funktsija-redirect/ - Основные функции |
| Flask №2. ORM, Flask-SQLAlchemy | Полезные материалы:   1. <https://pythonru.com/uroki/15-osnovy-orm-sqlalchemy> 2. <https://habr.com/ru/post/346344/>   Домашнее задание: Создать модели БД для своих проектов и написать документацию к моделям |
| Flask №3. Вёрстка страницы | Полезные материалы:   1. https://webdesign-master.ru/blog/html-css/ 2. <https://www.youtube.com/playlist?list=PLM6XATa8CAG4F9nAIYNS5oAiPotxwLFIr>   Домашнее задание: Сверстать любую страницу своего проекта(Желательно регистрацию/авторизацию) |
| Паттерн Наблюдатель(Observer) | Полезные материалы:   1. <https://refactoring.guru/ru/design-patterns/observer> 2. http://cpp-reference.ru/patterns/behavioral-patterns/observer/ |
| Flask №4. Регистрация | Полезные материалы:   1. <https://www.kaspersky.ru/blog/identification-authentication-authorization-difference/29123/> |
| Flask №5. Авторизация, Flask-Login | Полезные материалы:   1. <https://www.digitalocean.com/community/tutorials/how-to-add-authentication-to-your-app-with-flask-login-ru> 2. <https://pythonru.com/uroki/18-autentifikacija-vo-flask>   Домашнее задание: Сделать регистрацию и авторизацию для своего проекта |
| Основа безопастности. Виды атак, самые известные хакеры | Полезные материалы:   1. [Мой материал](/media/uploads/2022/03/22/qnbyvr.pdf) |
| Flask №6. Миграции, Flask-Migrate | Полезные материалы:   1. <https://habr.com/ru/post/346344/> 2. https://flask-migrate.readthedocs.io/en/latest/index.html - Доки   Домашнее задание: Создать миграции своего приложения и зафиксировать изменения на GitHub |
| Flask №7 | Дополнительных материалов нету! |
| Flask №8. Цели SMART | Полезные материалы:   1. <https://www.carrotquest.io/blog/smart-goal/> |
| Flask №9. Паттерн Фабричный Метод | Полезные материалы:   1. <https://webdevblog.ru/shablon-fabrichnogo-metoda-i-ego-realizaciya-v-python/> 2. <https://proglib.io/p/factory-method-python> |
| Flask №10 | Дополнительных материалов нету! |
| Flask №11, Интерпретаторы Python | Полезные материалы:   1. https://dev-gang.ru/article/7-luczshih-perevodczikov-python-obnovleno-2019-auvtihzcfi/ - Какие есть интерпретаторы 2. <https://proglib.io/p/pypy-uskoryaem-python-s-minimalnymi-usiliyami-2020-11-26> - Интерпретатор PyPy |
| Flask №12. Паттерн Строитель | Полезные материалы:   1. https://refactoring.guru/ru/design-patterns/builder - VPN 2. <http://www.managepy.ru/паттерн-строитель-builder-на-python/> |
| Flask №13 | Дополнительных материалов нету! |
| Flask №14. Дата-Классы | Полезные материалы:   1. <https://habr.com/ru/post/415829/> 2. https://pythonru-com.turbopages.org/turbo/pythonru.com/s/osnovy/dataclass-v-python |
| Flask №15 | Дополнительных материалов нету! |
| Flask №16 | Дополнительных материалов нету! |
| Тестирование. Модуль unittest | Полезные материалы:   1. <https://devpractice.ru/unit-testing-in-python-part-1/> 2. <https://pythonworld.ru/moduli/modul-unittest.html> 3. <https://ru.hexlet.io/courses/advanced_python/lessons/python_testing_unittest/theory_unit>   Домашнее задание: Начать покрывать свой код тестами |
| Flask №17 | Дополнительных материалов нету! |
| Flask №18 | Дополнительных материалов нету! |
| Flask №19 | Дополнительных материалов нету! |
| Flask №20 | Дополнительных материалов нету! |
| Паттерн Компоновщик (Composite) | Полезные материалы:   1. <http://www.managepy.ru/паттерн-компоновщик-на-python/> 2. <https://proglib.io/p/py-patterns> |
| Flask №21 | Дополнительных материалов нету! |
| Flask №22 | Дополнительных материалов нету! |
| Flask №23. Нотации типов (typing) | Полезные материалы:   1. <https://habr.com/ru/company/lamoda/blog/432656/> 2. https://docs-python.ru/standart-library/modul-typing-python/ |
| Flask №24 | Дополнительных материалов нету! |
| Создание соц. сети №1 | Дополнительных материалов нету! |
| Создание соц. сети №2 | Дополнительных материалов нету! |
| Создание соц. сети №3 | Дополнительных материалов нету! |
| Создание соц. сети №4 | Дополнительных материалов нету! |
| Создание соц. сети №5 | Дополнительных материалов нету! |
| Создание соц. сети №6 | Дополнительных материалов нету! |
| Создание соц. сети №7 | Дополнительных материалов нету! |
| Создание соц. сети №8 | Дополнительных материалов нету! |
| Создание соц. сети №9 | Дополнительных материалов нету! |
| Создание соц. сети №10 | Дополнительных материалов нету! |
| Создание соц. сети №11 | Дополнительных материалов нету! |
| Создание соц. сети №12 | Дополнительных материалов нету! |
| Создание соц. сети №13 | Дополнительных материалов нету! |
| Создание соц. сети №14 | Дополнительных материалов нету! |
| Создание соц. сети №15 | Дополнительных материалов нету! |
| Создание соц. сети №16 | Дополнительных материалов нету! |
| Создание соц. сети №17 | Дополнительных материалов нету! |
| Создание соц. сети №18 | Дополнительных материалов нету! |
| Создание соц. сети №19 | Дополнительных материалов нету! |
| Создание соц. сети №20 | Дополнительных материалов нету! |
| Создание соц. сети №21 | Дополнительных материалов нету! |
| Создание приложения на Django №1 | Полезные материалы:   1. <https://itproger.com/course/django> 2. <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django>   Домашнее задание: Придумать идею для своего сайта |
| Создание приложения на Django №2. Приложения, структура проекта | Полезные материалы:   1. <https://habr.com/ru/company/vivid_money/blog/544856/> 2. <https://ru.hexlet.io/blog/posts/pochemu-django-luchshiy-freymvork-dlya-razrabotki-saytov> |
| Нововведения Python 3.10 | Полезные материалы:   1. <https://habr.com/ru/company/skillfactory/blog/550540/> 2. <https://www.youtube.com/watch?v=0kyy_zKO86U> |
| Моржовый оператор в Python | Полезные материалы:   1. <https://habr.com/ru/company/otus/blog/555924/> 2. <https://nuancesprog.ru/p/10634/> |
| Создание приложения на Django №3. Шаблоны, маршруты | Полезные материалы:   1. <https://www.youtube.com/watch?v=rLIcE7-FuBQ&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=11> 2. <https://www.youtube.com/watch?v=V8-pJfyO3Jk&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=17> |
| Создание приложения на Django №4. Модели | Полезные материалы:   1. <https://www.youtube.com/watch?v=erefMlNL0Vw&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=7> 2. <https://www.youtube.com/watch?v=moR7m2-tkxs&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=8> |
| Создание приложения на Django №5. Модели, погружение. Миграции | Полезные материалы:   1. <https://www.youtube.com/watch?v=moR7m2-tkxs&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=8> - Миграции 2. <https://www.youtube.com/watch?v=7ToZ08JVqig&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=9> - Модели 3. <https://www.youtube.com/watch?v=1T1s4eWdHtA&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=10> - Модели |
| Создание приложения на Django №6 | Дополнительных материалов нету! |
| Создание приложения на Django №7 | Дополнительных материалов нету! |
| Создание приложения на Django №8. Админка | Полезные материалы:   1. <https://www.youtube.com/watch?v=uZZrK4v3Ddo&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=12> |
| Тестовый созвон с HR | Полезные материалы:   1. <https://m.youtube.com/watch?v=k6SpcnyhK2Y> - Что спрашивать на собесе 2. <https://m.youtube.com/watch?v=BZy28F4tetM&t=39s> - Что спрашивать на собесе   Вопросы которые я рекомендую задавать:   * Чем ваша компания отличается от других? * Как вы будете оценивать что я хорошо в чем-то (например в програмировании, шарпе, джаве и тд) через 1-2-3 месяца? * Как вы собираете обратную связь от сотрудников? * Какой карьерный рост предполагается с данной позиции, когда и как он будет? * Логирование рабочего времени? * Переработки и как они оплачиваются? * Вопрос про рабочее оборудование и программное обеспечение? * В какой команде я буду работать? * Какие технологии используются на проекте? * Вопрос про испытательный срок (тестовый период)? * Рабочий график в компании. Во сколько начинается работа и во сколько заканчивается? Можно ли работать удаленно если болеешь? * Как часто повышают зарплату? Какие условия повышения? * Обучение для сотрудников? * Будут ли командировки? * Какой социальный пакет: страховка здоровья, зубной, спортзал? |
| Тестовое техническое собеседование №1 | Что не помешает повторить:   1. Повторить терминологию (Пакет, модуль) 2. Потоки/процессы/асинхронность 3. Вспомнить про лямбду 4. Вспомнить как Python представляет себе код (100 это объект?) 5. Виды полиморфизма 6. Вспомнить про @staticmethod/@classmethod 7. WSGI |
| Тестовое техническое собеседование №2 | Что не помешает повторить:   1. Куки/сессии 2. Многопоточность и асинхронность 3. Контекстные менеджеры 4. Итераторы/генераторы 5. Английские термины (comprehension, value, reference) 6. Большое О (Из алгоритмов) 7. Компиляция/интерпретация |
| Создание приложения на Django №9 | Дополнительных материалов нету! |
| Строки Python - более глубокое погружение | Полезные материалы:   1. <https://pythonru.com/osnovy/stroki-python> |
| Составляем резюме №1 | Полезные материалы:   1. <https://vc.ru/hr/148577-kak-podgotovit-horoshee-rezyume-dlya-it-specialista> 2. <https://www.youtube.com/watch?v=6jgFuTpWKOQ> 3. <https://www.youtube.com/watch?v=XM-SR1uc3b8>   Домашнее задание: Составить своё резюме |
| Составляем резюме №2 | Дополнительных материалов нету! |
| Создание приложения на Django №10 | Дополнительных материалов нету! |
| Практика решения алгоритмических задач CodeWars | Дополнительных материалов нету! |
| Встроенные теги и фильтры шаблонов Django | Полезные материалы:   1. <https://django.fun/docs/django/ru/4.0/ref/templates/builtins/> |
| Создание декораторов с параметрами | Полезные материалы:   1. https://docs-python.ru/tutorial/dekoratory-python/dekoratory-argumentami/ 2. <https://russianblogs.com/article/3435303742/> |
| Создание регистрации и авторизации. Django-allauth | Полезные материалы:   1. <https://m.youtube.com/watch?v=5DWgaSVEREQ> 2. <https://pythobyte.com/the-complete-django-allauth-guide-la3-ceddf3d3/> 3. <https://russianblogs.com/article/6833594772/>   Домашнее задание: Добавить регистрацию и авторизацию в свой проект |
| Создание приложения на Django №11 | Дополнительных материалов нету! |
| Рекурсия. Вспоминаем и углубляемся | Полезные материалы:   1. <https://foxford.ru/wiki/informatika/rekursiya-v-python> - Теория 2. https://code-enjoy.ru/ege\_po\_informatike\_2022\_zadanie\_16\_rekursiya/ - Для практика 3. <https://server.179.ru/tasks/training/recursion.html> - Для практика |
| Микросервисы. Создание своего микросервисного приложения | Полезные материалы:   1. <https://habr.com/ru/post/249183/> 2. <https://proglib.io/p/mikroservisnaya-arhitektura-na-primere-python-i-grpc-2021-02-12> 3. <https://django.fun/python/postroenie-mikroservisa-v-python/>   Домашнее задание: Понять теорию, попробовать создать своё мини-приложение на микросервисах |
| Декораторы с аргументами. Создание декоратора для проверки типов | Полезные материалы:   1. <https://tirinox.ru/parametric-decorator/>   Домашнее задание: Попробовать создать свой декоратор с параметрами |
| Создание приложения на Django №12. Рынок IT сейчас, как искать работу | Полезные материалы:   1. <https://timeweb.com/ru/community/articles/luchshie-servisy-i-sayty-dlya-poiska-raboty-v-it> 2. <https://www.youtube.com/watch?v=LPdvfRzVoAg>   Домашнее задание: Зарегестрироваться на бирже для трудоустройства (Например: hh.ru) и попробовать пооткликаться на вакансии |
| Создание приложения на Django №13 | Дополнительных материалов нету! |
| Django ORM №1 | Полезные материалы:   1. <https://docs.djangoproject.com/en/3.0/ref/models/querysets/> - Статься в доке 2. <https://www.youtube.com/watch?v=gDpUOCwh6JE&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=32> - Видеоурок 3. <https://www.youtube.com/watch?v=MODCU-eKxxw&list=PLmC7X4gkQWCeyIdLxHZdts-3tkcrxP4-o&index=33> - Видеоурок |
| Большое О, сложность алгоритмов | Полезные материалы:   1. <https://webdevblog.ru/bolshoe-o-chto-eto-takoe-pochemu-eto-vazhno-i-pochemu-eto-ne-vazhno/> 2. <https://russianblogs.com/article/650740266/> |