



TRAINEE. ПРАКТИКА

Данный этап курса предполагает периодическую **обратную связь от наших экспертов**.

После прохождения теста этого модуля откроется возможность присылать результаты выполнения практических заданий экспертам на проверку.

Результатом выполнения этого практического модуля является web-проект, при написании которого тебе понадобятся все ранее полученные навыки. Ссылку на репозиторий своего проекта можно отправить

здесь: <https://education.nixsolutions.com/mod/quiz/view.php?id=104>

Все работы отправленные **до 10:00 понедельника** будут проверяться в течении недели (в зависимости от загрузки экспертов).

По результатам проверки тебе будет дана обратная связь или рекомендации.

Ждем твоих работ!

№	ЗАДАНИЕ
1.	Ответьте на вопросы: Как удалить дубли строк из файла? Как дописать содержимое одного текстового файла в конец второго? Как разбить текстовый файл на несколько по 100 строк в каждом? Как вывести на экран первые 30 строк файла? Как вывести на экран последние 30 строк файла? Как вывести на экран строки текстового файла /tmp/file.txt начинающиеся со слова 'START'? Как среди нескольких файлов в директории найти те, которые содержат слово 'test'?
2.	Дан список из словарей в формате [{'name': 'Oleg', 'age': 23}, {'name': 'Vasya', 'age': 19}] Отсортируйте список по возрасту людей, чтобы получилось [{'name': 'Vasya', 'age': 19}, {'name': 'Oleg', 'age': 23}] Используйте sorted и lambda
3.	Написать свой декоратор, который будет отлавливать ошибки, полученные в ходе выполнения обёрнутой функции, логгировать их и делать raise отловленной ошибки
4.	есть 2 списка: list1 = ['Oleg', 'Vasya', 'Anna'] list2 = ['Ivanov', 'Sidorov', 'Petrova'] используя zip соедините 2 списка так, чтобы получился список из tuple, в формате [('Oleg', 'Ivanov'),]
5.	напишите lambda функцию, которая будет складывать 2 числа: my_lambda = <ваша лямбда функция> >>> my_lambda(3, 5) 8
6.	Дан список из словарей: list_ = [{'name': 'Alex', 'age': 25}, {'name': 'Oleg', 'age': 23}, {'name': 'Anna', 'age': 32}, {'name': 'Igor', 'age': 50}, {'name': 'Anton', 'age': 17},] Отфильтруйте его так, чтобы в нём остались только люди, имена которых начинаются с буквы "A", и возраст между 18 и 30 годами включительно.

№	ЗАДАНИЕ
7.	<p>Напишите функцию, которая делает return строки приветствия людей из разных стран на разных языках, в зависимости от страны человека.</p> <p>Вызовите функцию используя map</p> <p>Требование: используйте map</p> <p>Для примера можете взять этот список юзеров:</p> <pre>users_list = [('Александр', 'ru'), ('James', 'us'), ('Daniella', 'es'), ('Someone', 'unknown country'),]</pre> <p>На выходе с использованием users_list, после использования map вы должны получить результат iterable с такими приветствиями:</p> <p>Привет, Александр!</p> <p>Hello, James!</p> <p>Hola, Daniella!</p> <p>Hello, Someone, but I do not know where are you from!</p>
8.	<p>создайте файл с таким содержанием:</p> <pre>""" some line blablabla you dont need to catch this line try to catch me but not me I'm here, catch me!!! """</pre> <p>откройте данный файл при помощи контекстного менеджера в режиме чтения и соберите список всех строк, которые содержат текст "catch me" в один список после чего выведите в консоль количество найденных в файле строк</p>
9.	<p>создайте функцию-генератор, которая принимает на вход два числа, первое - старт, второе - end.</p> <p>генератор в каждом цикле должен возвращать число и увеличивать его на 1</p> <p>при итерации генератор должен начать с числа start и закончить итерации на числе end</p> <p>т.е. при вызове</p> <pre>for i in my_generator(1, 3): print(i)</pre> <p>в консоли должно быть:</p> <pre>1 2 3</pre>
10.	<p>Напишите функцию, которая принимает число - таймзону от GMT (например, Киев - таймзона +2, соответственно передаёте 2, или Гавайи - таймзона = -10, соответственно передаёте -10), и возвращает текущую дату и время в указанной таймзоне.</p> <p>Формат:</p> <p>[<часов>:<минут>:<секунд>] - <день месяца>, <полное название месяца> of <год></p> <p>например: [16:22:26] - 11, March of 2021</p>
11.	<p>Сгенерируйте рандомное число при помощи модуля random:</p> <pre>import random</pre> <p>random_number = random.randint(1, 10)</p> <p>При помощи модуля time засеките время ожидания программы и выведите его в консоль</p> <p>start_time = <ваш код здесь></p> <p>"усыпите" выполнение программы на <random_number> секунд</p>

end_time = <ваш код здесь>	
№	ЗАДАНИЕ
12.	<p>создайте тестовую директорию с несколькими файлами разных названий при помощи модуля os создайте функцию для очистки директории от файлов определённого расширения, принимающую на вход: путь к директории, где нужно удалить файлы строку - формат файла, по которой будет определяться, нужно ли удалять файл пример вызова: delete_files_func(path='/path/to/folder', file_extension='.txt') после удаления файлов функция должна вернуть список названий всех удалённых файлов</p> <p>результат - в директории /path/to/folder удалены все файлы, расширение которых == .txt</p>
13.	<p>Напишите скрипт, который пишет в консоль "Hello, <имя> <фамилия>! Nice to see you here!" На вход скрипт принимает имя и фамилию, которые в последствии и используются в скрипте.</p>
14.	<p>Создайте несколько классов конструктора автомобиля: Engine -> Car -> Honda Реализуйте наследование этих классов и базовые методы для каждого из них, например start/stop для engine, open/close doors для Car и т.д. Используйте __init__ для инициализации необходимых параметров (максимальная скорость, объем двигателя, и т.п.) Примените инкапсуляцию для сокрытия вспомогательных методов По необходимости используйте staticmethod / classmethod / property</p>
15.	<p>Создайте несколько классов: Animal (абстрактный класс), Cat, Dog Создайте абстрактные методы say, run и jump в классе Animal (abc.abstractmethod) Реализуйте полиморфизм поведения животных для методов: say, run, jump</p>
16.	<p>Напишите скрипт, который вычисляет площадь фигур - прямоугольник, квадрат, треугольник На входе скрипт принимает название фигуры и длины сторон. Требования: используйте классы, наследование, модуль sys</p>
17.	<p>Для выполнения задания воспользуйтесь БД, созданной во время выполнения задач уровня Beginner.</p> <p>Использовать транзакции для insert, update, delete на 3х таблицах.</p>
18.	<p>Для выполнения задания воспользуйтесь БД, созданной во время выполнения задач уровня Beginner.</p> <p>Придумать 3 запроса, которые можно оптимизировать с помощью индекса (проверять стоит ли оптимизировать запрос оператором explain) и оптимизировать их используя индекс. Результат проверять также оператором explain.</p>
19.	<p>Для выполнения задания воспользуйтесь БД, созданной во время выполнения задач уровня Beginner.</p> <p>Написать представление для таблицы products, для таблиц order_status и order , для таблиц products и category</p>
20.	<p>Для выполнения задания воспользуйтесь БД, созданной во время выполнения задач уровня Beginner.</p> <p>Написать 2 любые хранимые процедуры.</p>

№	ЗАДАНИЕ
21.	<p>Для выполнения задания воспользуйтесь БД, созданной во время выполнения задач уровня Beginner.</p> <p>Создать функцию, которая сетит shipping_total = 0 в таблице order, если город юзера равен x. Использовать IF clause.</p>
22.	<p>Для выполнения задания воспользуйтесь БД, созданной во время выполнения задач уровня Beginner.</p> <p>Сравнить цену каждого продукта n с средней ценой продуктов в категории продукта n. Использовать window function. Таблица результата должна содержать такие колонки: category_title, product_title, price, avg. Пример для решения можно найти в теории.</p>
23.	Протестируйте данные функции .

№	ЗАДАНИЕ	ЧТО ОЖИДАЕТСЯ В РЕЗУЛЬТАТЕ
24.	<p>Реализовать проект с использованием всех полученных ранее знаний и технологий.</p> <p>Детальные требования указаны ниже:</p> <p>Требования к проекту</p>	<p>Ссылка на ваш публичный репозиторий с проектом.</p> <p>Оставить её можно</p> <p>здесь: https://education.nixsolutions.com/mod/quiz/view.php?id=104</p>