

## Итоговая аттестация по программе «Профессия "Веб-разработчик на Python"»

Создайте копию этого шаблона у себя на диске. После того, как вы ответите на все задания, скачайте документ в формате .pdf и приложите его в Форму для сдачи ответа на учебной платформе.

### Приступим к тесту

**Задание 1.** Что выведется на экран?

```
quote = "Do not give up, the beginning is always the hardest."  
print(quote [8])
```

Ответ: i

**Задание 2.** Дайте определение понятиям «Список» и «Кортеж». Опишите, чем они отличаются.

Это две структуры данных которые используются для хранения в коллекции.

Основные отличия:

1. Кортеж - это упорядоченная коллекция элементов

Он определяется круглыми скобками ()

Элементы кортежа являются неизменяемыми, после создания кортежа нет возможности изменить его элементы

Пример: some\_tuple = (1, 2, 3, 4)

Есть возможность опустить скобки

2. Список - тоже считается упорядоченной коллекцией элементов  
Элементы списка могут быть изменены(изменяемость)

Можно добавлять, удалять и менять

# SKILLFACTORY

Используются квадратные скобки [ ]

Отсюда вывод: Они отличаются: 1) Синтаксисом у списка квадратные скобки [ ], а у кортежа круглые ( ).

2) Изменяемостью, кортежи неизменяемые, а списки да

3) Кортежи более производительны, потому как меньше потребляют память

4) списки имеют больше методов для изменения и управления данными

**Задание 3.** Что такое миграции в Django и какие команды для работы с ними существуют в стандартном инструментарии Django?

Миграции в django позволяют автоматически управлять структурой базы данных. Они согласуют модель данных написанных с помощью python с фактической структурой данных.

В стандартном инструментарии django существуют такие команды:

`makemigrations` - она создает файлы миграции на основе моделей(`python manage.py makemigrations`)

`migrate` - команда которая применяет миграции по моделям приложения(`python manage.py migrate`)

`showmigrations` - команда которая показывает список всех доступных миграций, а так же выводит статус миграции(`python manage.py showmigrations`)

`sqlmigrate` - показывает sql запрос который будет выполнен для конкретной миграции(`python manage.py sqlmigrate app_name migration_name`)

**Задание 4.** Есть два списка разной длины. В первом содержатся ключи, а во втором — значения. Напишите функцию `make_dict`, которая создаёт из этих ключей и значений словарь.

Если ключу не хватило значения, в словаре должно быть значение `None`. Значения, которым не хватило ключей, нужно игнорировать.

**Обратите внимание!** Функция должна возвращать созданный словарь.

Приложите код решения.

# SKILLFACTORY

```
def make_dict(keys, values):  
    result = {}  
    for i, key in enumerate(keys):  
        if i < len(values):  
            result[key] = values[i]  
        else:  
            result[key] = None  
    return result
```

**Задание 5.** Напишите простой калькулятор. У вас должна быть функция `calculate`, которая принимает три аргумента: первые два — числа, третий — операция, которая должна быть произведена над ними.

**Доступные операции:** сложение, вычитание, умножение и деление.

Функция возвращает результат операции. В остальных случаях функция должна вернуть строку «Неизвестная операция».

```
def calculate(num1, num2, operation):  
    operations = {  
        '+': lambda x, y: x + y,  
        '-': lambda x, y: x - y,  
        '*': lambda x, y: x * y,  
        '/': lambda x, y: x / y if y != 0 else "Деление на ноль  
невозможно"  
    }  
    result = operations.get(operation)  
    if result:  
        return result(num1, num2)  
    else: return "Неизвестная операция"
```

