

## 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])
```

i

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

Список — структура данных, предназначенная для хранения упорядоченных наборов элементов. Слово «упорядоченные» в этом случае означает, что элементы структуры хранятся в том порядке, в котором они были добавлены. Элементы списка индексируются, то есть имеют порядковый номер.

Кортеж — это неизменяемый список. Кортеж не может быть изменён никаким способом после его создания. Кортеж определяется так же, как список, за исключением того, что набор элементов заключается в круглые скобки, а не в квадратные.

Список является динамическим, тогда как кортеж имеет статические характеристики. Это означает, что списки могут быть изменены, тогда как

# SKILLFACTORY

кортежи не могут быть изменены, кортеж быстрее списка из-за статичности по своей природе. Списки обозначаются квадратными скобками, а кортежи обозначаются скобками.

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

Миграция базы данных Django – это способ применения изменений, которые мы внесли в модель, в схему базы данных. Django создает файл миграции внутри папки миграции для каждой модели, чтобы создать схему таблицы, и каждая таблица сопоставляется с моделью, для которой создается миграция.

Django предоставляет различные команды, которые используются для выполнения задач, связанных с миграцией в Python. После создания модели мы можем использовать эти команды.

**makemigrations:** используется для создания файла миграции, который содержит код для табличной схемы модели.

**migrate:** создает таблицу в соответствии со схемой, определенной в файле миграции.

**sqlmigrate:** используется для отображения необработанного SQL-запроса примененной миграции.

**showmigrations:** в ней перечислены все миграции и их статус.

# SKILLFACTORY

**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.

```
k = ["key_1", "key_2", "key_3", "key_4", "key_5"]
```

```
v = ["val_1", "val_2", "val_3"]
```

```
make_dict = {}
```

```
x = 0
```

```
while x < len(k):
```

```
    try:
```

```
        make_dict[k[x]] = v[x]
```

```
        x = x + 1
```

```
    except:
```

```
        make_dict[k[x]] = "None"
```

```
        x = x + 1
```

```
print(make_dict)
```

# SKILLFACTORY

**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, sign):  
    if sign == "+":  
        return num1 + num2  
    elif sign == "-":  
        return num1 - num2  
    elif sign == "*":  
        return num1 * num2  
    elif sign == "/":  
        return num1 / num2  
    else:  
        return "Неизвестная операция"
```