IT CAREER HUB

Python

Кортежи

Преподаватель

Портрет

Имя Фамилия

Текущая должность

Количество лет опыта

Какой у Вас опыт - ключевые кейсы

Самые яркие проекты

Дополнительная информация по вашему усмотрению

Корпоративный e-mail

Социальные сети (по желанию)

Важно

- Камера должна быть включена на протяжении всего занятия
- В течение занятия вопросы задавать в чате или когда преподаватель спрашивает, есть ли у Вас вопросы
- Вести себя уважительно и этично по отношению к остальным участникам занятия
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя

Повторение

- Форматирование строк
- Задания для закрепления
- Метод format
- Форматирование чисел

План занятия

- Кортеж (tuple)
- Цикл for с кортежами
- Методы кортежа

ICH IT CAREER HUB

ОСНОВНОЙ БЛОК

IT CAREER HUB

Кортеж (tuple)

Полезно знать

Кортеж

Это упорядоченная неизменяемая (`immutable`) коллекция элементов.

Основные характеристики кортежа

- **Неизменяемость:** После создания кортежа его элементы нельзя изменить.
- **Упорядоченность:** Элементы в кортеже сохраняют порядок, в котором они были добавлены.
- **Поддержка дубликатов:** Список может содержать повторяющиеся элементы.
- **Поддержка различных типов данных:** Кортеж может содержать элементы разных типов.
- **Индексация:** К кортежам можно обращаться по индексу, как и к спискам.

Создание кортежа с несколькими элементами

Пояснения

Кортеж создаётся с использованием круглых скобок `()`, а элементы внутри разделяются запятыми.

Код

```
my_tuple = (1, 2, 3, "apple", True)

print(my_tuple)
```

Создание пустого кортежа

Пояснения

Пустой кортеж создаётся с помощью круглых скобок.

Код

```
empty_tuple = ()

print(empty_tuple)  # Вывод: ()
```

Создание кортежа с одним элементом

Пояснения

Для создания кортежа с одним элементом нужно добавить запятую после элемента, иначе Python воспримет это как математическое выражение, а не как кортеж.

Код

```
single_element_tuple = (5,)

print(single_element_tuple)
```

Доступ к элементам кортежа

Пояснения

К элементам кортежа можно обращаться с помощью индексов, так же как и к элементам списка. Индексация начинается с 0.

Код

```
my_tuple = (10, 20, 30, 40)

print(my_tuple[1])

print(my_tuple[-1])
```

Изменяемость кортежей

Пояснение

Кортежи в Python неизменяемы, поэтому после их создания элементы нельзя добавлять, изменять или удалять.

Код

```
my_tuple = (10, 20, 30)
# Попытка изменить элемент вызовет ошибку
# my_tuple[1] = 40
# TypeError: 'tuple' object does not
support item assignment
```

Операции с кортежами

```
tuple1 = (1, 2)
tuple2 = (3, 4)
print(tuple1 + tuple2)  # Конкатенация кортежей

my_tuple = (1, 2)
print(my_tuple * 3)  # Повторение кортежей

my_tuple = (10, 20, 30)
print(20 in my_tuple)  # Проверка на наличие элемента

my_tuple = (1, 2, 3, 4, 5)
print(len(my_tuple))  # Длина кортежа

# Сравнение кортежей происходит аналогично спискам, то есть поэлементно
tuple1 = (1, 2, 3)
tuple2 = (1, 2, 4)
print(tuple1 == tuple2)  # Сравнение кортежей
```

С кортежами можно проводить те же операции, что и со списками.

Преобразование кортежей

Пояснение

Можно преобразовать строку или другую коллекцию в кортеж с помощью функции `tuple()`.

Код

```
my_list = [1, 2, 3]

my_tuple = tuple(my_list)

print(my_tuple)

print(type(my_tuple))
```

Преобразование кортежей

Пояснение

А также можно преобразовывать кортеж в другую коллекцию, например в список.

Код

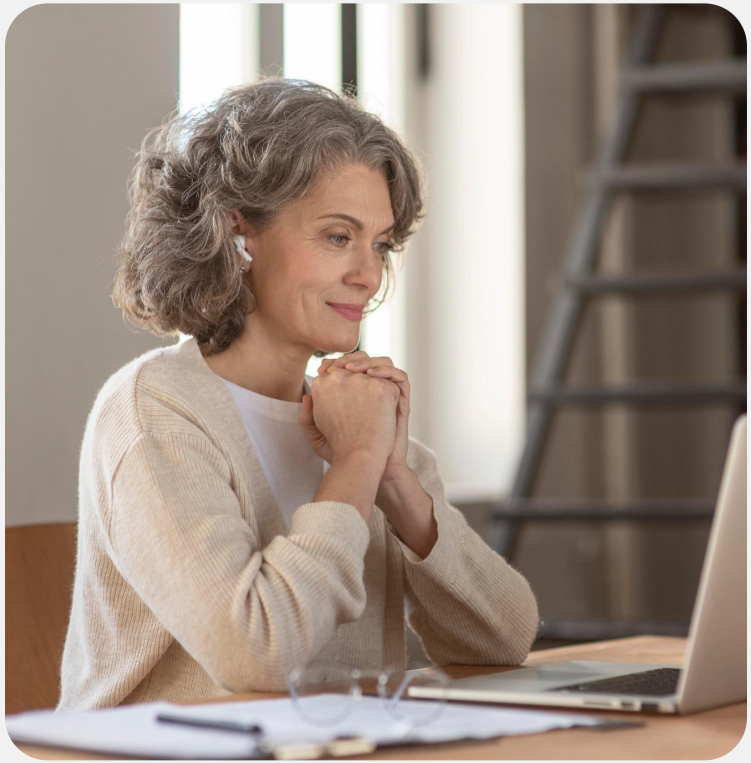
```
my_tuple = (1, 2, 3)

my_list = list(my_tuple)

print(my_list)

print(type(my_list))
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

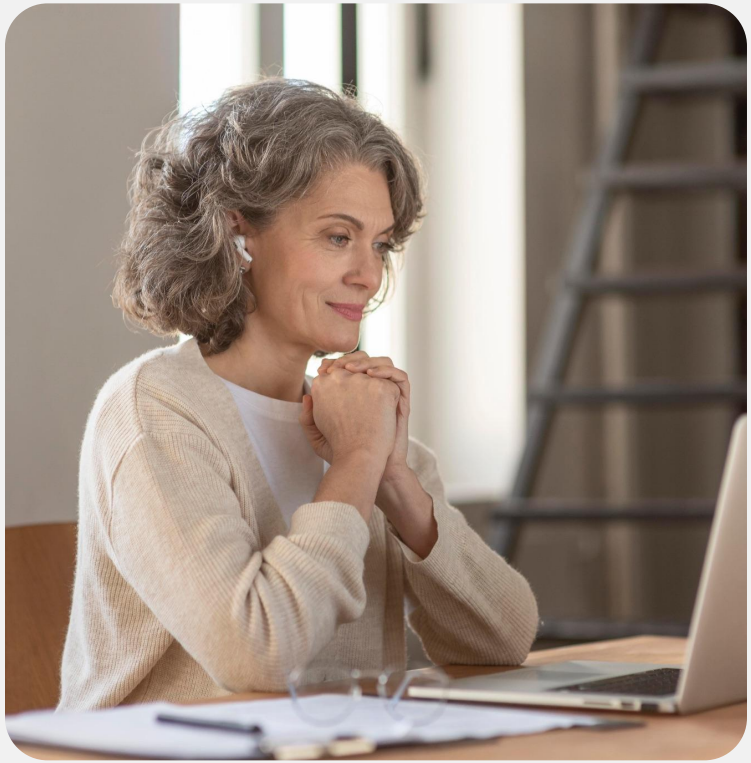


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
my_tuple = (1, 2, 3, 4, 5)

print(my_tuple[2])
```

a. 2
b. 3
c. (3)
d. (3,)

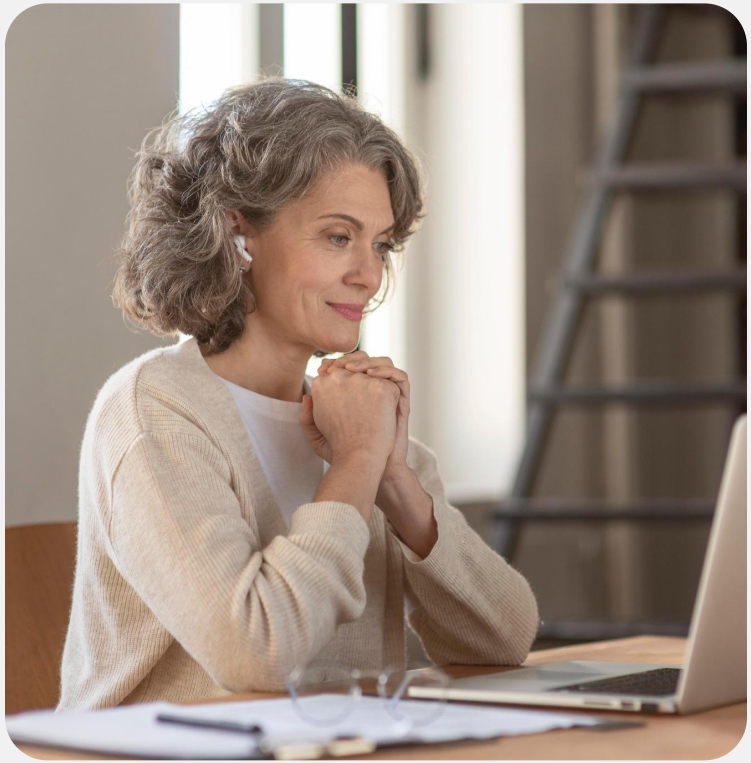


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
my_tuple = (1, 2, 3, 4, 5)

print(my_tuple[2])
```

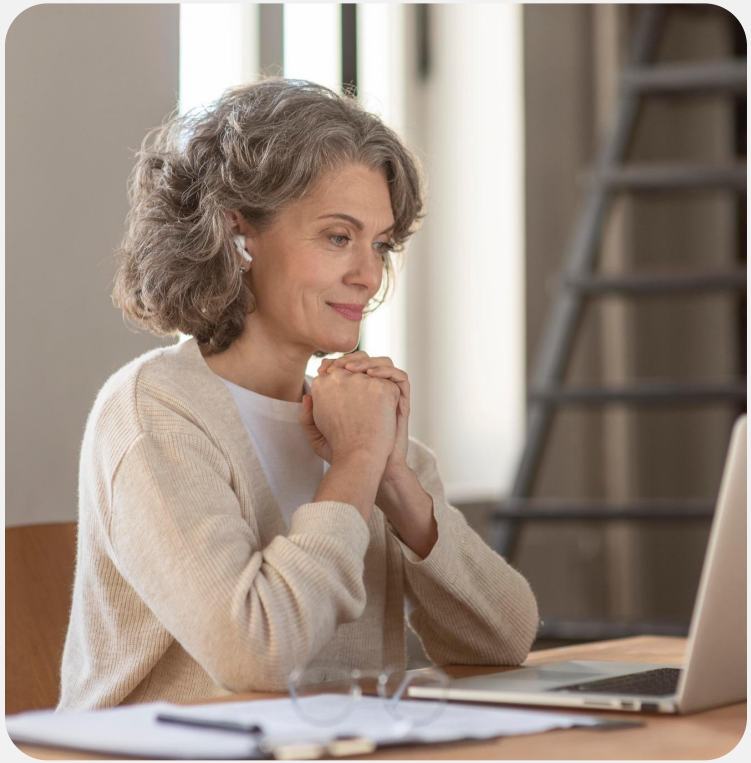
a. 2
b. 3
c. (3)
d. (3,)



Выберите правильный вариант ответа

Как создать кортеж с одним элементом?

a. `(5)`
b. `(5,)`
c. `tuple([5])`
d. `tuple(5)`



Выберите правильный вариант ответа

Как создать кортеж с одним элементом?

a. `(5)`
b. `(5,)`
c. `tuple([5])`
d. `tuple(5)`

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Цикл for с кортежами

Цикл for с кортежами

Пояснение

Цикл `for` работает с кортежами так же, как и со списками.

Код

```
my_tuple = (10, 20, 30, 40)

for item in my_tuple:

    print(item)
```

ICH IT CAREER HUB

Упаковка и распаковка коллекций

Полезно знать

ICH
IT CAREER HUB

Упаковка и распаковка коллекций

Это операции, которые применяются для того, чтобы собрать несколько значений в одну коллекцию (упаковка) или разобрать коллекцию на отдельные переменные (распаковка).

Полезно знать

ICH
IT CAREER HUB

Упаковка (Packing)

Это процесс, при котором несколько значений собираются в одну коллекцию.

Пример упаковки

Пояснение

Когда мы присваиваем несколько значений одной переменной, Python автоматически упаковывает их в кортеж.

Код

```
# Упаковка нескольких значений в кортеж

packed_tuple = 1, 2, 3

print(packed_tuple)

print(type(packed_tuple))
```

Полезно знать

Распаковка

Это процесс, при котором значения из коллекции присваиваются отдельным переменным.

Пример распаковки

Пояснение

Если коллекция (список или кортеж) содержит столько же элементов, сколько переменных, Python позволяет легко распаковать её.

Код

```
# Кортеж с тремя элементами
packed_tuple = (1, 2, 3)

# Распаковка значений кортежа в переменные
a, b, c = packed_tuple
print(a, b, c)
```

Распаковка с помощью оператора *

Пояснение

Когда необходимо вывести элементы коллекции по отдельности, из можно распаковать с помощью оператора * и передать в `print()`.

Код

```
numbers = [1, 2, 3, 4, 5]
# Вывод коллекции
print(numbers)
# Вывод элементов по отдельности
print(*numbers)
```

Упаковка с помощью оператора *

Пояснение

Иногда можно упаковать остаток элементов после распаковки в одну переменную с помощью оператора *.

Это удобно, когда количество переменных для распаковки меньше, чем количество элементов в коллекции.

Код

```
numbers = [1, 2, 3, 4, 5]
# Распаковка первого элемента в a,
# последнего в b, остальные элементы в other
a, *other, b = numbers
print(a)
print(b)
print(other)
```

Распаковка в цикле

Пояснение

Распаковка особенно полезна при работе с циклами, когда нужно перебирать элементы коллекций с несколькими значениями.

Код

```
pairs = [(1, 'apple'), (2, 'banana'), (3, 'cherry')]
for number, fruit in pairs:
# Распаковка элементов кортежа в переменные
    print(f"Число: {number}, Фрукт: {fruit}")
```

Ошибки при распаковке

Пояснение

Количество переменных должно соответствовать количеству элементов в коллекции, за исключением случаев, когда используется оператор *. В противном случае возникнет ошибка.

Код

```
data = (1, 2, 3)
# Попытка распаковать больше
# переменных, чем элементов
a, b, c, d = data
# Ошибка, т.к. слишком мало элементов
a, b = data
# Ошибка, т.к. слишком много элементов
```

Полезно знать

ICH
IT CAREER HUB

Функция enumerate

Это встроенная функция Python, которая добавляет счётчик к итерируемому объекту, таким как список, кортеж или строка, и возвращает результат в виде объекта `enumerate`.

Функция enumerate

Синтаксис

```
enumerate(iterable, start=0)
```

Пояснение

- `iterable` — это коллекция, которую нужно перебирать (например, список или кортеж).
- `start` (необязательно) — значение, с которого начинается счётчик. По умолчанию — 0.

Отображения содержимого enumerate

Определение

Чтобы увидеть содержимое объекта `enumerate`, необходимо преобразовать его в список, кортеж или другую структуру данных, которая может быть напечатана, т.к. содержимое объекта `enumerate` не отображается напрямую при выводе.

Код

```
fruits = ["apple", "banana", "cherry"]

enumerated_fruits = enumerate(fruits)

# Чтобы увидеть содержимое, преобразуем
# объект enumerate в список

print(list(enumerated_fruits))
```

Пример: Вывод индекса и элемента

```
fruits = ["apple", "banana", "cherry"]


for index, fruit in enumerate(fruits):
    print(f"{index}: {fruit}")
```

Пример 1: Редактирование списка по индексу

```
numbers = [10, 20, 30, 40]


for index, value in enumerate(numbers):
    numbers[index] = value * 10


print(numbers)
```

Пример 2: Доступ к соседним элементам

Функция `enumerate()` также может быть полезна, когда нужно получить доступ к текущему элементу и его соседним элементам в последовательности. Это часто применяется при анализе последовательных данных, где требуется сравнить соседние элементы.

```
numbers = [10, 20, 15, 30, 25, 35]

for index, value in enumerate(numbers[:-1]):    # Проходим по списку, кроме последнего элемента
    if value > numbers[index + 1]:
        print(f"{value} больше, чем {numbers[index + 1]}")
```

Изменение стартового значения enumerate()

```
languages = ("Python", "Java", "C++")


for index, language in enumerate(languages, start=1):
    print(f"{index}: {language}")
```

Использование enumerate() со строкой

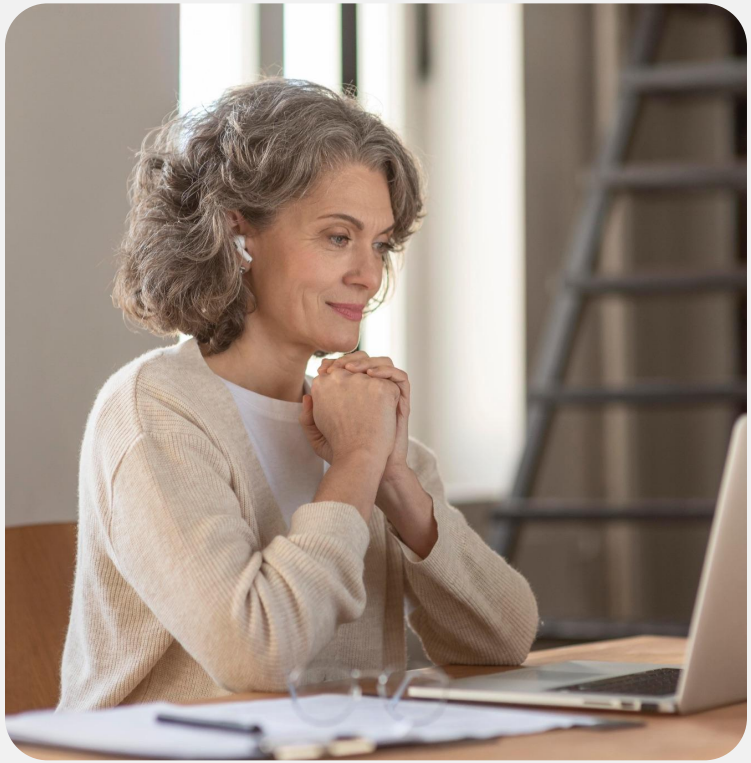
```
text = "Python"


for index, letter in enumerate(text):
    print(f"{index}: {letter}")
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

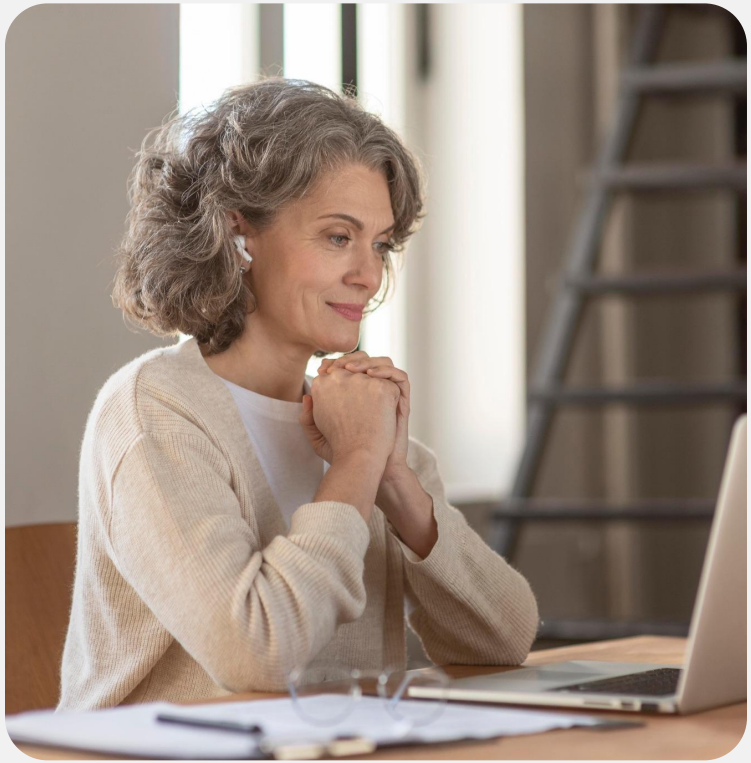


Выберите правильный вариант ответа

Что произойдет при попытке распаковать кортеж с тремя элементами в две переменные?

```
my_tuple = (1, 2, 3)

a, b = my_tuple
```

a. a = 1, b = 2
b. a = (1, 2), b = 3
c. a = 1, b = (2, 3)
d. Ошибка

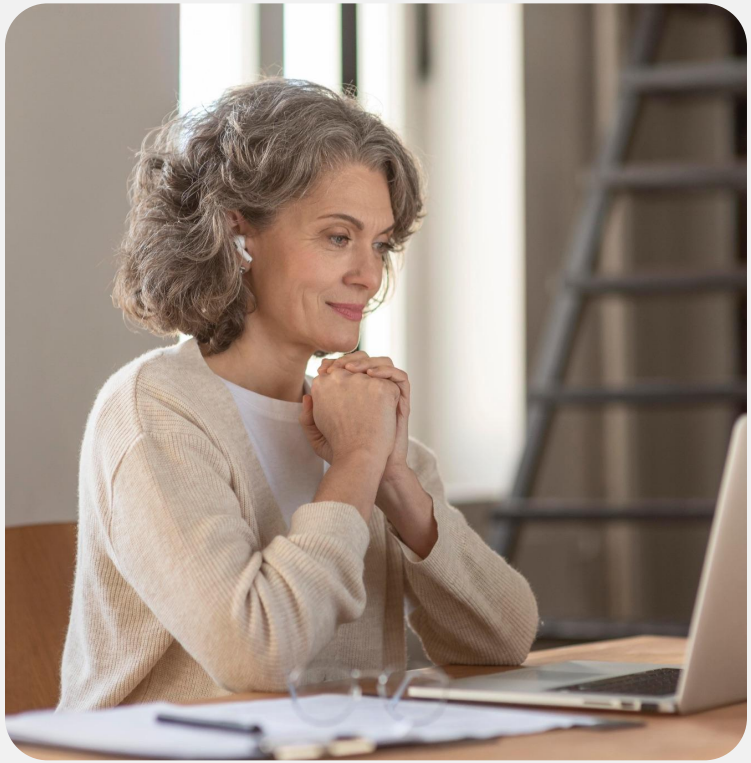


Выберите правильный вариант ответа

Что произойдет при попытке распаковать кортеж с тремя элементами в две переменные?

```
my_tuple = (1, 2, 3)

a, b = my_tuple
```

a. a = 1, b = 2
b. a = (1, 2), b = 3
c. a = 1, b = (2, 3)
d. Ошибка

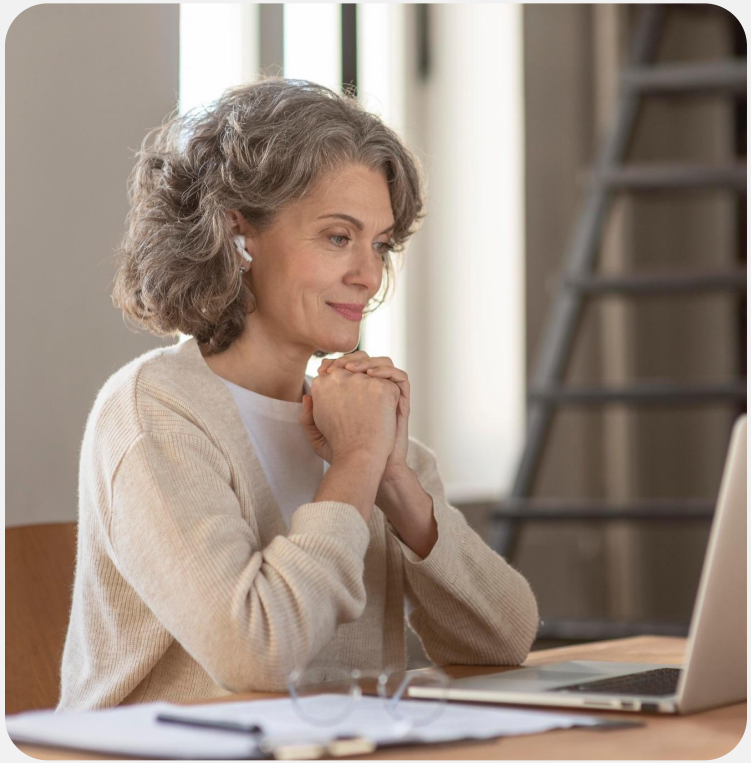


Выберите правильный вариант ответа

Какой результат будет выведен при выполнении следующего кода?

```
fruits = ["apple", "banana", "cherry"]

for index, fruit in enumerate(fruits):

    print(f"{index}: {fruit}", end=' ')
```

a. `0: apple 1: banana 2: cherry`
b. `1: apple 2: banana 3: cherry`
c. `Ошибка`
d. `0: apple 1: banana 2: cherry на разных строках`



Выберите правильный вариант ответа

Какой результат будет выведен при выполнении следующего кода?

```
fruits = ["apple", "banana", "cherry"]

for index, fruit in enumerate(fruits):

    print(f"{index}: {fruit}", end=' ')
```

a. 0: apple 1: banana 2: cherry
b. 1: apple 2: banana 3: cherry
c. Ошибка
d. 0: apple 1: banana 2: cherry на разных строках

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Методы кортежа

Полезно знать

IT CAREER HUB

Кортежи

Это неизменяемые коллекции, поэтому они поддерживают ограниченное количество методов.

Метод count()

Пояснение

Метод `count()` используется для подсчёта, сколько раз определённый элемент встречается в кортеже.

`value` — элемент, количество вхождений которого нужно подсчитать.

Синтаксис

```
tuple.count(value)
```

Пример использования count()

```
my_tuple = (1, 2, 3, 2, 4, 2)

count_of_twos = my_tuple.count(2)

print(count_of_twos)
```

Полезно знать

ICH
IT CAREER HUB

Метод index()

Возвращает индекс первого вхождения указанного элемента в кортеже. Если элемент не найден, будет вызвана ошибка `ValueError`.

Метод index()

Пояснение

- `value` — элемент, индекс которого нужно найти.
- `start` (необязательно) — начальная позиция для поиска.
- `end` (необязательно) — конечная позиция для поиска.

Синтаксис

```
tuple.index(value, start=0, end=None)
```

Пример использования index()

```
my_tuple = (10, 20, 30, 20, 40)

index_of_first_twenty = my_tuple.index(20)

print(index_of_first_twenty)
```

Пример с указанием диапазона поиска

```
my_tuple = (10, 20, 30, 20, 40)

index_of_twenty_in_range = my_tuple.index(20, 2)  # Начинаем поиск с индекса 2

print(index_of_twenty_in_range)
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ПРАКТИЧЕСКАЯ РАБОТА

1. Кортеж уникальных

Напишите программу, которая обрабатывает кортеж чисел. Программа должна вернуть кортеж, в котором будут только уникальные элементы.

Не используйте неизученные коллекции.

Данные:

```
numbers = (1, 2, 3, 2, 1, 4, 5, 3, 6)
```

Пример вывода:

```
Уникальные элементы: (1, 2, 3, 4, 5, 6)
```

2. Кортеж выше среднего

Напишите программу, которая обрабатывает кортеж чисел. Программа должна вернуть кортеж, состоящий из элементов, которые больше среднего значения всех элементов исходного кортежа.

Данные:

```
numbers = (10, 15, 20, 25, 30)
```

Пример вывода:

```
Элементы больше среднего: (25, 30)
```

IT CAREER HUB

ДОМАШНЕЕ ЗАДАНИЕ

Домашнее задание

1. Прогрессия увеличения

Напишите программу, которая создаёт новый кортеж, состоящий из элементов изначального в том же порядке. Добавить в него только элементы, которые больше всех предыдущих значений в исходном кортеже.

Данные:

```
numbers = (3, 7, 2, 8, 5, 10, 1)
```

Пример вывода:

```
Кортеж по возрастанию: (3, 7, 8, 10)
```

Домашнее задание

2. Повторяющиеся элементы

Напишите программу, которая находит индексы элементов кортежа, встречающихся более одного раза. Вывести сами элементы и их индексы.

Данные:

```
numbers = (1, 2, 3, 4, 2, 5, 3, 6, 4, 2, 9)
```

Пример вывода:

```
Индексы элемента 2:   1 4 9
Индексы элемента 3:   2 6
Индексы элемента 4:   3 8
```

Заключение