IT CAREER HUB

Python

Цикл for

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

- Неизменяемость строк
- Кодировка
- Функции ord, chr
- Сравнение строк
- Функция len
- Индексация строк
- Срезы строк
- Оператор принадлежности in

План занятия

- Итерируемый объект
- Цикл for
- Функция range
- Операторы break, continue, else в цикле for
- Вложенные циклы
- Вложенные циклы с использованием while и for

ICH IT CAREER HUB

ОСНОВНОЙ БЛОК

IT CAREER HUB

Итерируемый объект

Полезно знать

ICH
IT CAREER HUB

Итерируемый объект

Это объект, который может состоять из множества элементов и предоставляет их по одному, когда это необходимо

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Цикл for

Полезно знать

IT CAREER HUB

Цикл for

Это конструкция, которая позволяет повторно выполнять блок кода (итерацию) для каждого элемента в последовательности.

Синтаксис

```
for переменная in последовательность:  # условие цикла

    # блок кода, выполняющийся для каждого элемента
```

Преимущества тернарного оператора:

- **Переменная** — это переменная, которая на каждом шаге цикла присваивает значение текущего элемента из последовательности.
- **Последовательность** — это итерируемый объект (например, строка), по которому цикл проходит.
- **Блок кода (или тело цикла)** — это инструкции, которые будут выполняться для каждого элемента последовательности.

Цикл for

Условие цикла

Цикл `for` будет работать до тех пор, пока есть элементы в последовательности, которые можно обработать. Как только все элементы будут пройдены, цикл завершится.

В каждой итерации переменная `letter` получает следующий символ строки, а затем выполняется блок кода — в данном случае это команда `print(letter)`, которая выводит этот символ.

Код

```
text = "Python"

for letter in text:

    print(letter)
```

Полезно знать

IT CAREER HUB

Итерация

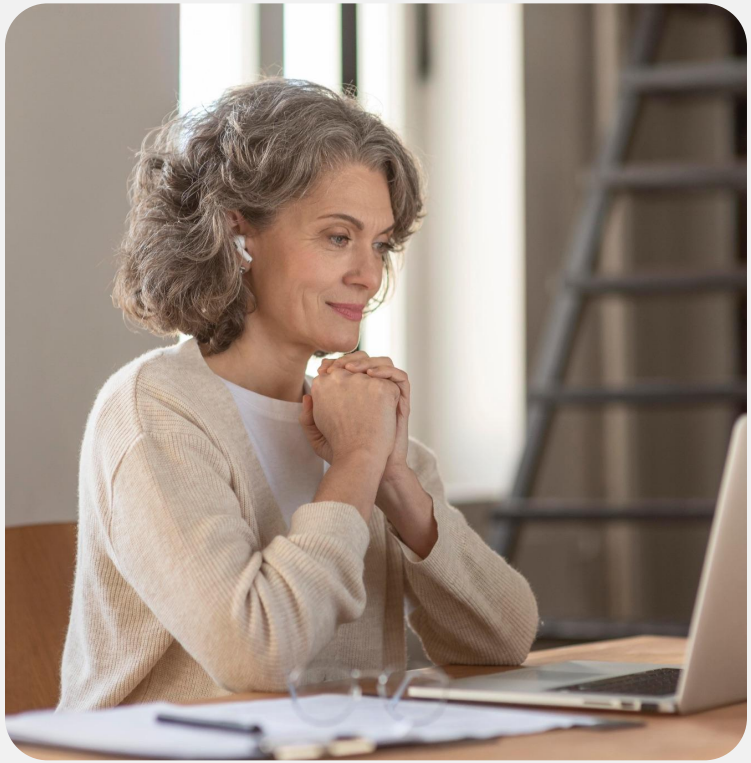
Это один шаг в процессе последовательного перебора элементов итерируемого объекта. На каждой итерации происходит обработка одного элемента из последовательности.

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

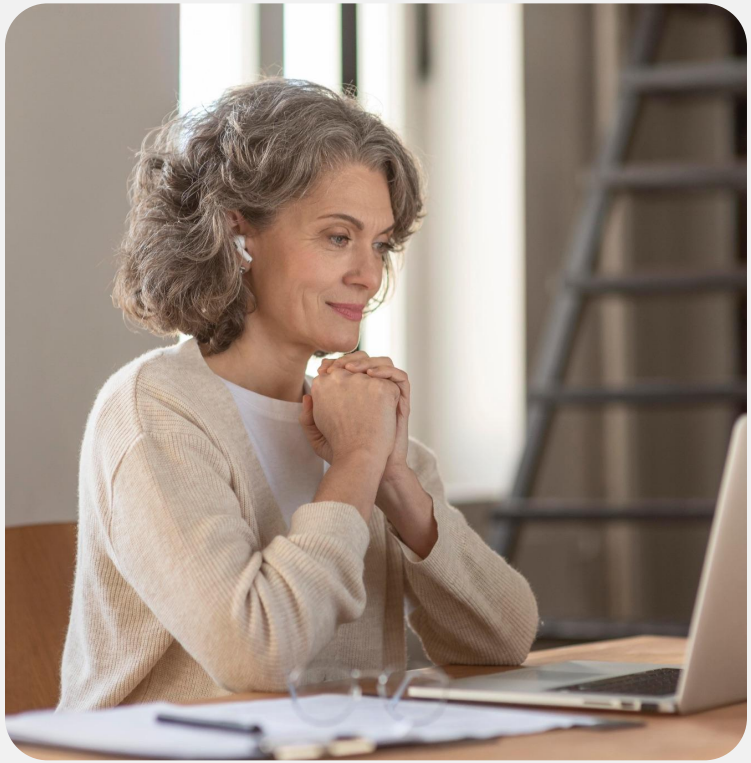


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
text = "Hello"

for letter in text:

    print(letter, end="")
```

a. H e l l o
b. Hello
c. H на новой строке
d. Ошибка



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
text = "Hello"

for letter in text:

    print(letter, end="")
```

a. H e l l o
b. Hello
c. H на новой строке
d. Ошибка

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Функция range

Полезно знать

Функция range

Используется для создания последовательности чисел, которые можно использовать в цикле. Она позволяет задавать диапазоны чисел и управлять шагом между ними.

Синтаксис range

```
range(start, stop, step)
```

Преимущества тернарного оператора:

- **start** (необязательный) — начальное значение (включительно). Если не указано, по умолчанию это 0.
- **stop** (обязательный) — конечное значение (не включается в последовательность).
- **step** (необязательный) — шаг, с которым создаётся последовательность. По умолчанию равен 1.

range с одним аргументом (stop)

Пояснения

Если указать только конечное значение, последовательность начинается с 0 и идёт до указанного числа (не включительно).

Код

```
for i in range(5):

    print(i)
```

range с двумя аргументами (start, stop)

Пояснения

Если указать два значения, последовательность начинается с первого (включительно) и идёт до второго числа (не включительно).

Код

```
for i in range(2, 6):

    print(i)
```

range с тремя аргументами (start, stop, step)

Пояснения

Когда используется три аргумента, `range()` создаёт последовательность, начиная с числа `start` (включительно), заканчивая числом `stop` (не включается), с шагом `step`, который указывает, через сколько элементов нужно брать следующее число.

Код

```
for i in range(1, 10, 2):

    print(i)
```

range с отрицательным шагом

Пояснения

Функция `range()` также поддерживает отрицательные значения для шага `step`, что позволяет создавать последовательности чисел в обратном порядке. В этом случае `start` должно быть больше `stop`, чтобы значения уменьшались с каждым шагом.

Код

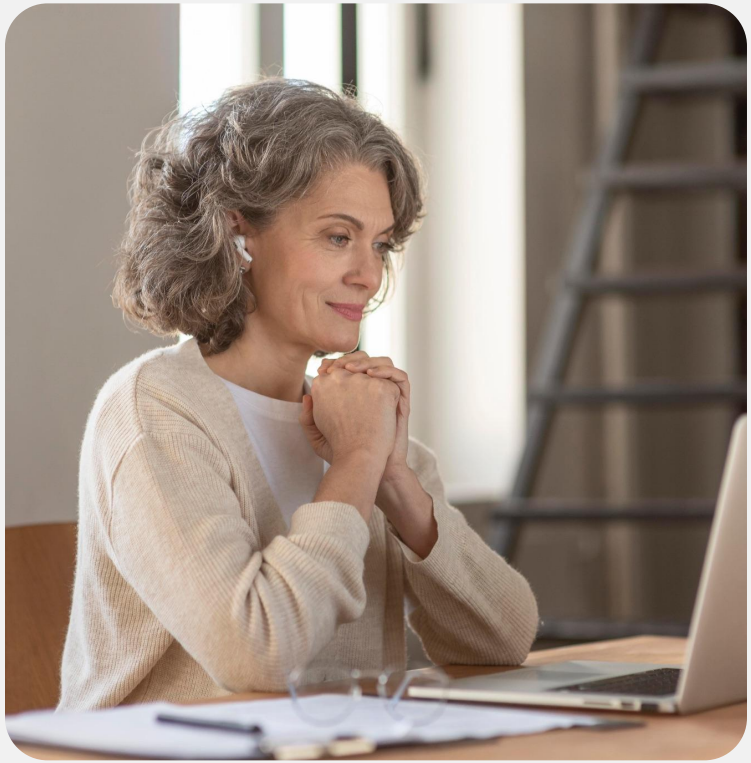
```
for i in range(10, 0, -2):

    print(i)

for i in range(-4, -8, -1):

    print(i)
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

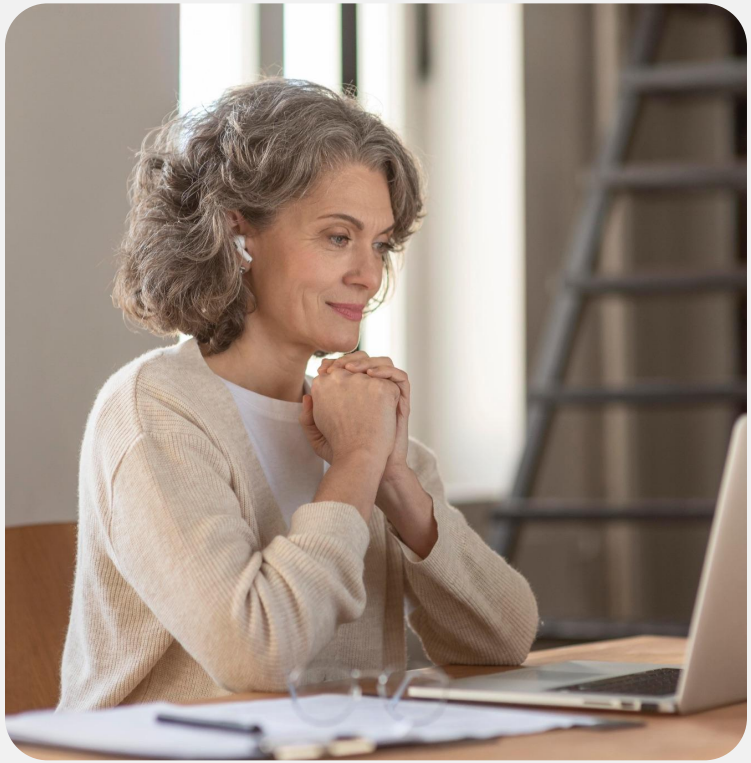


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for i in range(5):

    print(i)
```

a. Числа 1 2 3 4 5 на одной строке
b. Числа 0 1 2 3 4 на разных строках
c. Числа 1 2 3 4 5 на разных строках
d. Ошибка

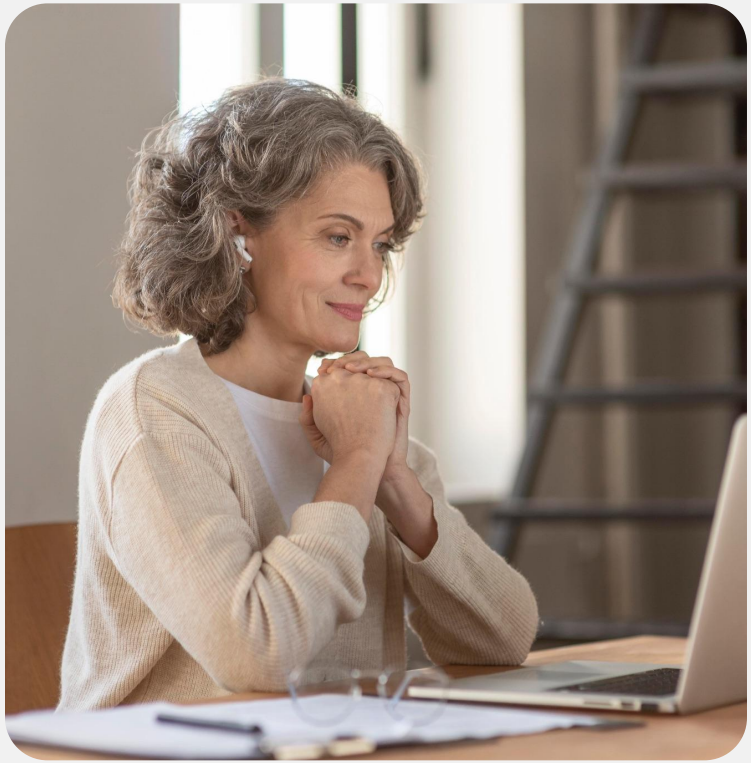


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for i in range(5):

    print(i)
```

a. Числа 1 2 3 4 5 на одной строке
b. Числа 0 1 2 3 4 на разных строках
c. Числа 1 2 3 4 5 на разных строках
d. Ошибка

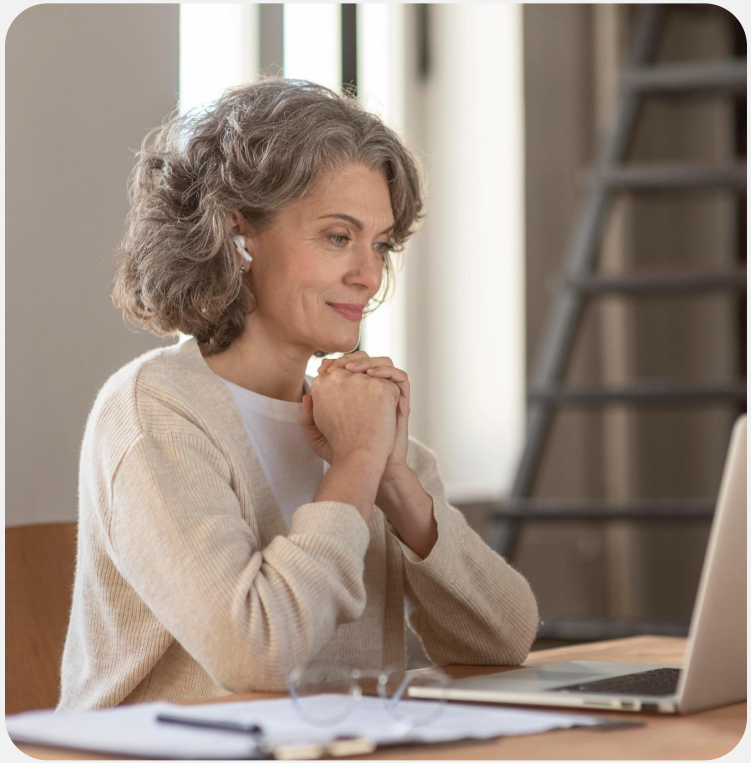


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for i in range(1, 10, 2):

    print(i)
```

a. Числа 1 2 3 4 5 6 7 8 9 на одной строке
b. Числа 1 3 5 7 9 на разных строках
c. Числа 2 4 6 8 10 на разных строках
d. Ошибка

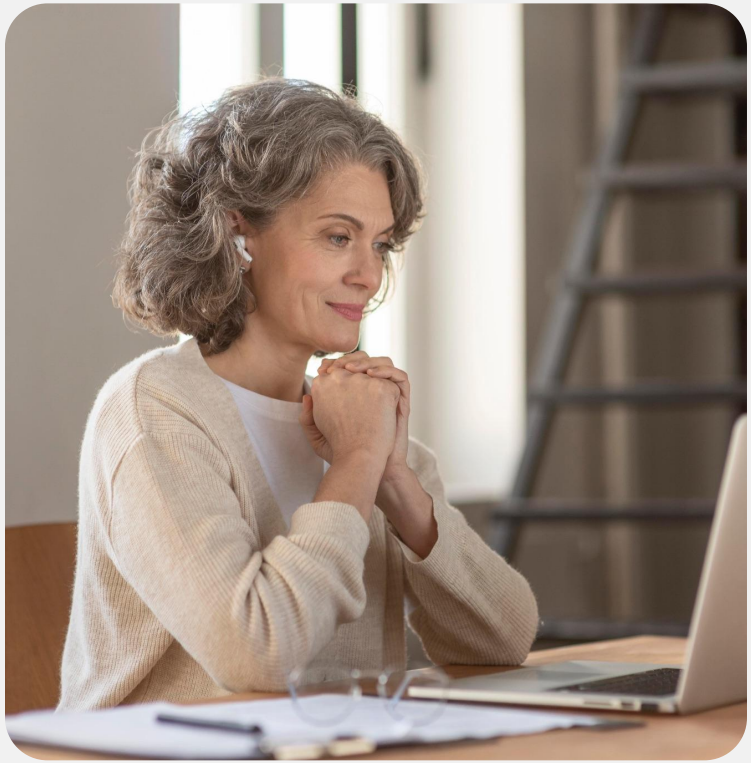


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for i in range(1, 10, 2):

    print(i)
```

a. Числа 1 2 3 4 5 6 7 8 9 на одной строке
b. Числа 1 3 5 7 9 на разных строках
c. Числа 2 4 6 8 10 на разных строках
d. Ошибка

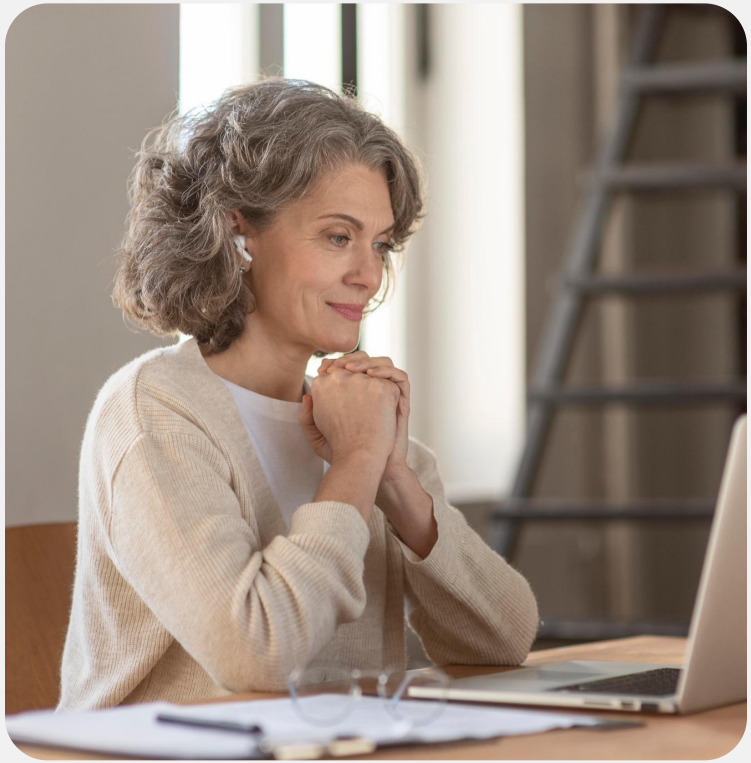


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for i in range(10, 0, -2):

    print(i)
```

a. Числа 10 8 6 4 2 0 на разных строках
b. Числа 10 8 6 4 2 на разных строках
c. Числа 9 7 5 3 1 на разных строках
d. Ошибка



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for i in range(10, 0, -2):

    print(i)
```

a. Числа 10 8 6 4 2 0 на разных строках
b. Числа 10 8 6 4 2 на разных строках
c. Числа 9 7 5 3 1 на разных строках
d. Ошибка

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Операторы break, continue, else в цикле for

Полезно знать

ICH
IT CAREER HUB

break, continue, else в цикле for

В Python цикле `for` можно использовать специальные операторы — `break`, `continue` и `else` — для управления выполнением цикла. Операторы работают также, как и в цикле `while`.

Оператор break

Пояснение

Оператор `break` позволяет прервать выполнение цикла досрочно, как только будет выполнено определённое условие. Цикл завершится, даже если элементы в последовательности ещё остались.

Пример

```
for letter in "Python":

    if letter == "h":

        break  # Останавливаем цикл, если найден символ "h"

    print(letter)
```

Оператор continue

Пояснение

Оператор `continue` позволяет пропустить текущую итерацию цикла и перейти к следующей, не завершая сам цикл. Он используется, когда нужно игнорировать определённые элементы, но продолжить обработку остальных.

Пример

```
for letter in "Python":
    if letter == "h":
# Пропускаем букву "h" и продолжаем
# цикл
        continue
    print(letter)
```

Оператор else

Пояснение

Оператор `else` в цикле for выполняет блок кода, если цикл завершился нормально, без использования оператора `break`. Это полезно, когда нужно выполнить определённые действия, если цикл прошёл через все элементы без прерывания.

Пример

```
for letter in "Python":
    if letter == "a":
        break  # Этот код никогда не
#выполнится, так как "a" нет в строке
        print(letter)
else:
    print("Цикл завершён нормально.")
```

Пример с break и else:

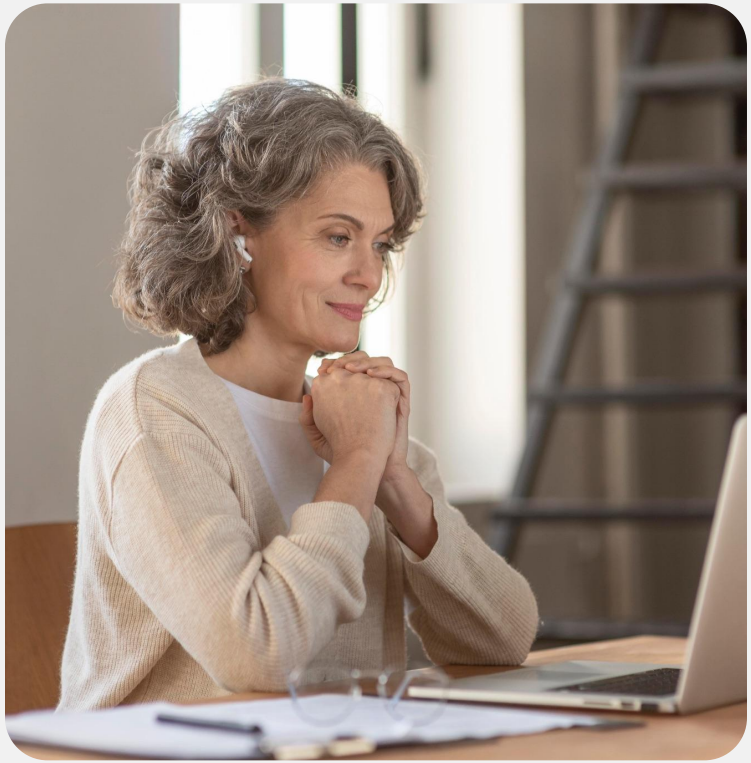
```
for letter in "Python":
    if letter == "h":
        break  # Цикл прерывается на символе "h"
    print(letter)
else:
    print("Цикл завершён нормально.")  # Этот блок не выполнится
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

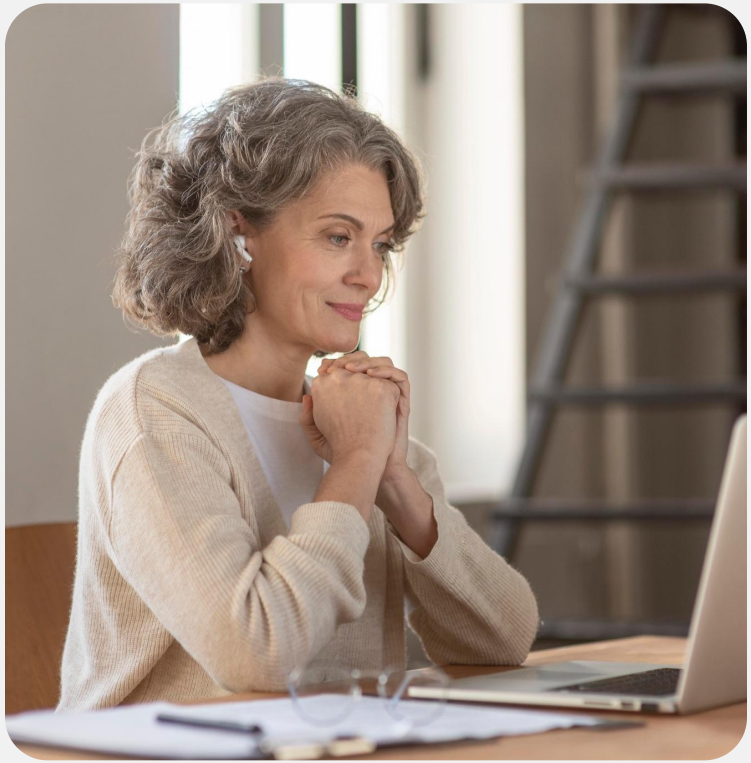


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for letter in "Python":
    if letter == "h":
        break
    print(letter, end=' ')
```

a. P y t
b. P y t h
c. P y t o n
d. Ошибка

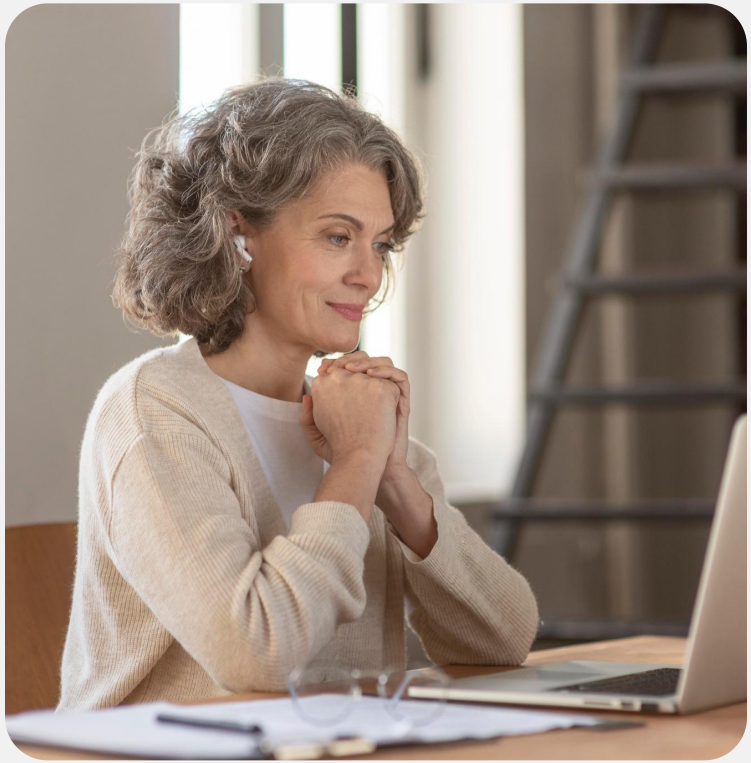


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for letter in "Python":
    if letter == "h":
        break
    print(letter, end=' ')
```

a. P y t
b. P y t h
c. P y t o n
d. Ошибка

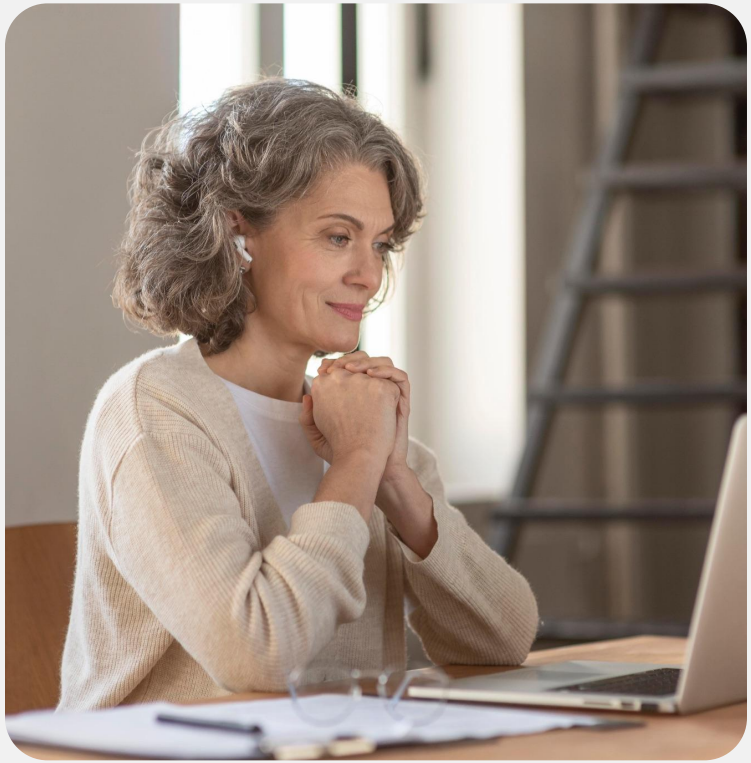


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for letter in "Python":
    if letter == "h":
        continue
    print(letter, end=' ')
```

a. P y t h o n
b. P y t o n
c. P y h o n
d. Ошибка

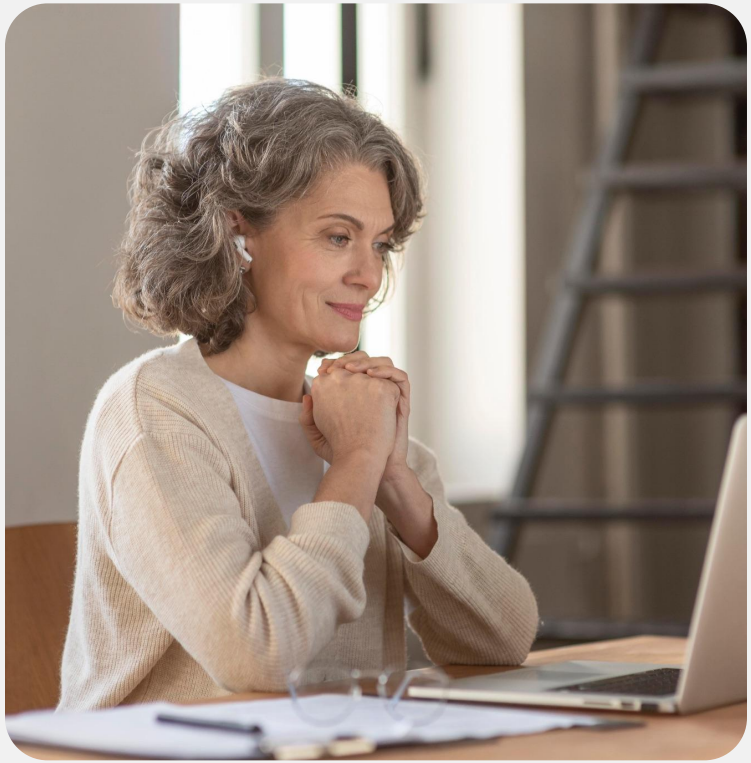


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for letter in "Python":
    if letter == "h":
        continue
    print(letter, end=' ')
```

a. P y t h o n
b. P y t o n
c. P y h o n
d. Ошибка

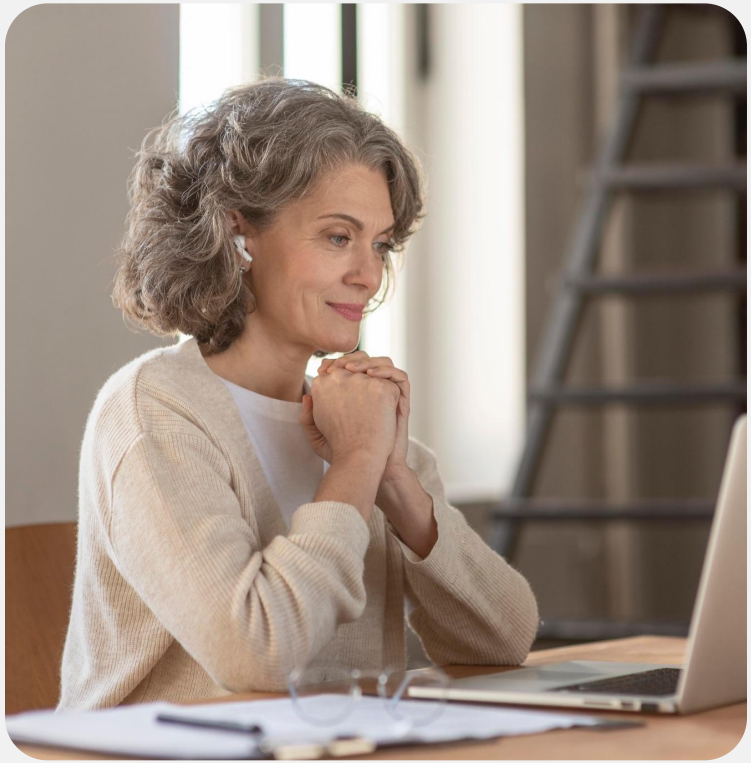


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for letter in "Python":
    if letter == "a":
        break
    print(letter, end=' ')
else:
    print("Цикл завершён нормально.")
```

a. Вывод: P y t h o n
b. Вывод: P y t h o n Цикл завершён нормально.
c. Вывод: Цикл завершён нормально.
d. Ошибка



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for letter in "Python":
    if letter == "a":
        break
    print(letter, end=' ')
else:
    print("Цикл завершён нормально.")
```

a. Вывод: P y t h o n
b. Вывод: P y t h o n Цикл завершён нормально.
c. Вывод: Цикл завершён нормально.
d. Ошибка

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Вложенные циклы

Полезно знать

Вложенные циклы

Это конструкции, в которых один цикл находится внутри другого. Вложенный цикл выполняется полностью для каждого прохода внешнего цикла.

Синтаксис вложенных циклов:

```
for внешняя_переменная in внешняя_последовательность:
    # код, выполняемый внутри внешнего цикла
    for внутренняя_переменная in внутренняя_последовательность:
        # код, выполняемый внутри обоих циклов
```

Особенности вложенных циклов:

- **Внешний цикл** выполняет итерации по своей последовательности
- Для каждой итерации внешнего цикла, **внутренний цикл** проходит через все свои элементы
- Когда внутренний цикл заканчивает выполнение всех итераций, внешний цикл переходит к следующей итерации
- При необходимости, можно использовать переменную внешнего цикла внутри внутреннего

Примеры вложенных циклов

```
# Пример 1
for i in "AB":
    for j in "12":
        print(i, j)

# Пример с выводом времени
for hour in range(24):
    for minute in range(60):
       if minute < 10:
                print("Время (часов:минут): ", hour, ':0', minute, sep='')
        else:
            print("Время (часов:минут): ", hour, ':', minute, sep='')
```

ICH IT CAREER HUB

Вложенные циклы с использованием while и for

Полезно знать

ICH
IT CAREER HUB

Вложенные циклы

Это не только комбинация двух `for` циклов, но также можно использовать комбинации `for` и `while` циклов для решения различных задач.

Пример: вывод времени за три часа, но только до конца дня

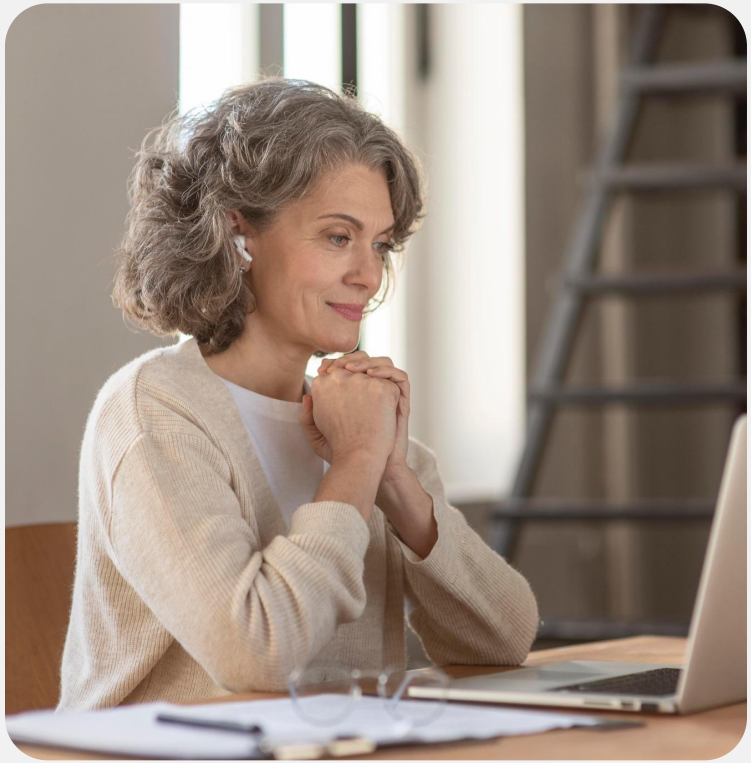
```
current_hours = int(input("Введите текущий час: "))  # Текущее время

hours = 3

end_time = current_hours + 3


while current_hours < 24 and current_hours < end_time:  # Внешний цикл с использованием while

    for minutes in range(60):  # Внутренний цикл с использованием for

        print("Время (часов:минут): ", current_hours, ':', minutes, sep='')

    current_hours += 1  # Увеличение значения часов на 1
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

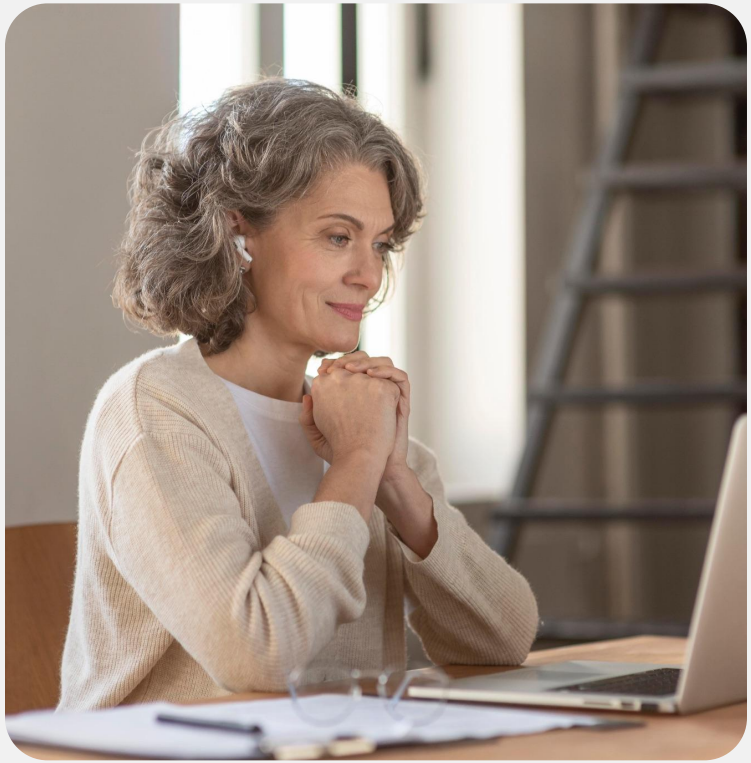


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for i in range(3):
    for j in range(3):
        print(i + j, end=" ")
```

a. 1 2 3 2 3 4 3 4 5
b. 1 2 3 3 4 5 5 6 7
c. 0 1 2 1 2 3 2 3 4
d. Ошибка



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
for i in range(3):
    for j in range(3):
        print(i + j, end=" ")
```

a. 1 2 3 2 3 4 3 4 5
b. 1 2 3 3 4 5 5 6 7
c. 0 1 2 1 2 3 2 3 4
d. Ошибка

IT CAREER HUB

Практическая работа

1. Факториал

Напишите программу, которая находит факториал числа, введённого пользователем и выводит его на экран. Не используйте модуль math для решения.

Факториал числа — это произведение всех натуральных чисел от 1 до самого этого числа включительно.

5! = 1 × 2 × 3 × 4 × 5 = 120

Пример вывода:

```
Введите число: 5
Факториал числа 5 равен 120
```

2. Звездный прямоугольник

Напишите программу, которая рисует прямоугольник из символов *, где ширина и высота вводятся пользователем. Используйте вложенные циклы для решения задачи.

Пример вывода:

```
Введите ширину: 5
Введите высоту: 3
*****
*****
*****
```

3. Простое число

Напишите программу, которая проверяет, является ли введённое пользователем число простым. Простое число — это число, которое делится только на себя и на 1.

Пример вывода:
Введите число: 11
Является простым

Введите число: 12
Не является простым

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ДОМАШНЕЕ ЗАДАНИЕ

Домашнее задание

1. **Таблица умножения**

Напишите программу, которая выводит таблицу умножения для чисел от 1 до n. Где n это число, которое введет пользователь. Оформите вывод так, чтобы столбцы были ровные (число ровно под числом)

Пример вывода:

```
Введите число: 5

1    2    3    4    5
2    4    6    8    10
3    6    9    12   15
4    8    12   16   20
5    10   15   20   25
```

Домашнее задание

2. Лестница чисел

Напишите программу, которая принимает число n и выводит "лестницу" из чисел. Каждая строка лестницы содержит числа от 1 до номера строки.

Пример вывода:

```
Введите число: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Домашнее задание

3. Удаление всех повторяющихся символов

Напишите программу, которая принимает строку и удаляет из неё все повторяющиеся символы, сохраняя только первые их вхождения.

Пример вывода:

```
Введите строку: Python programming

Результат: Python prgami
```

IT CAREER HUB

ВОПРОСЫ

Заключение