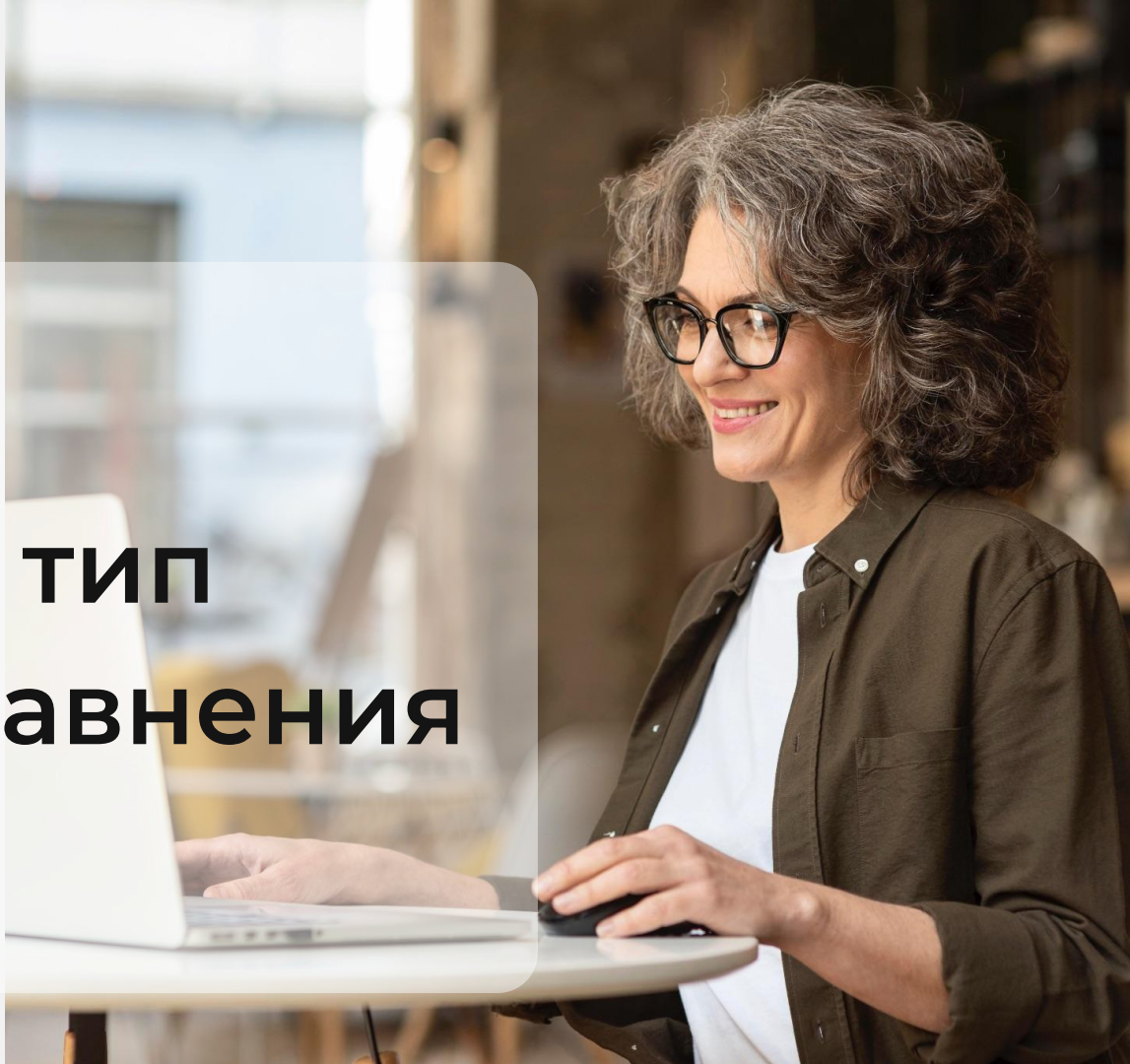
IT CAREER HUB

Python

Логический тип данных и сравнения



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

- [x] Числа
- [x] Арифметические операции с числами
- [x] ZeroDivisionError
- [x] Приоритет математических операций и скобок
- [x] Неизменяемые типы данных
- [x] Операторы приращения
- [x] Множественное присваивание
- [x] Преобразование типов, ValueError

План занятия

- Логический тип данных
- Приведение типов к bool
- Операторы сравнения
- Логические операторы
- Комбинирование логических операторов
- Приоритет логических операторов
- Таблица истинности
- Альтернативная интерпретация логических операторов
- Двойные неравенства

ICH IT CAREER HUB

ОСНОВНОЙ БЛОК

IT CAREER HUB

Логический тип данных

Полезно знать

Логический тип данных

(bool, от англ. boolean)

Этот тип данных используется для представления двух возможных значений: `True` (истина) и `False` (ложь).

True и False

Эти значения помогают программе принимать решения. Например, в зависимости от того, истинное значение или ложное, программа может выполнить разные действия.

Логические значения

Пояснения

Они могут быть созданы напрямую или быть результатом логических операций, сравнений или приведения типов к **bool**.

Создание напрямую:

isFinished = True

hasAccess = False

ICH IT CAREER HUB

Приведение типов к bool

Приведение типов к bool

В Python существуют значения, которые интерпретируются как True или False при преобразовании в логический тип

Интерпретация значений

True

Любое ненулевое число, непустая строка, список, кортеж и т.д.

False

None, 0, пустые последовательности (например, [], {}, "").

Преобразования в bool

Автоматическое

Python автоматически преобразует значения в `True` или `False`, когда это необходимо.

При этом Python определяет, является ли значение "истинным" или "ложным", исходя из его типа.

Явное (ручное)

Можно использовать функцию `bool`.

Автоматическое преобразование

- Условия `if`
- Циклы
- Логические операции

Явное (ручное) преобразование

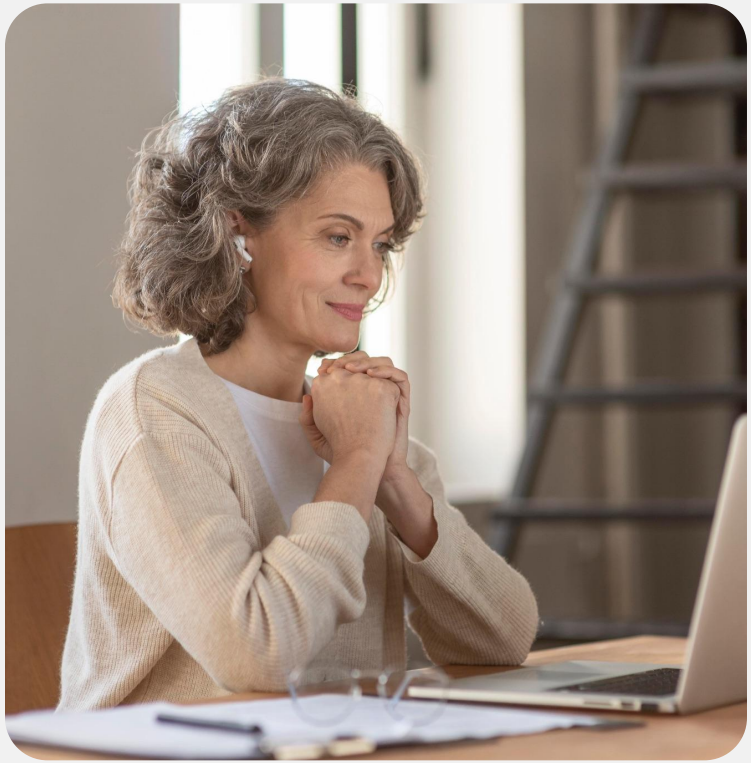
```
print(bool(5))          # True (ненулевое число)

print(bool(0))          # False

print(bool(None))       # False

print(bool([]))         # False (пустой список)

print(bool("Hello"))    # True (непустая строка)
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

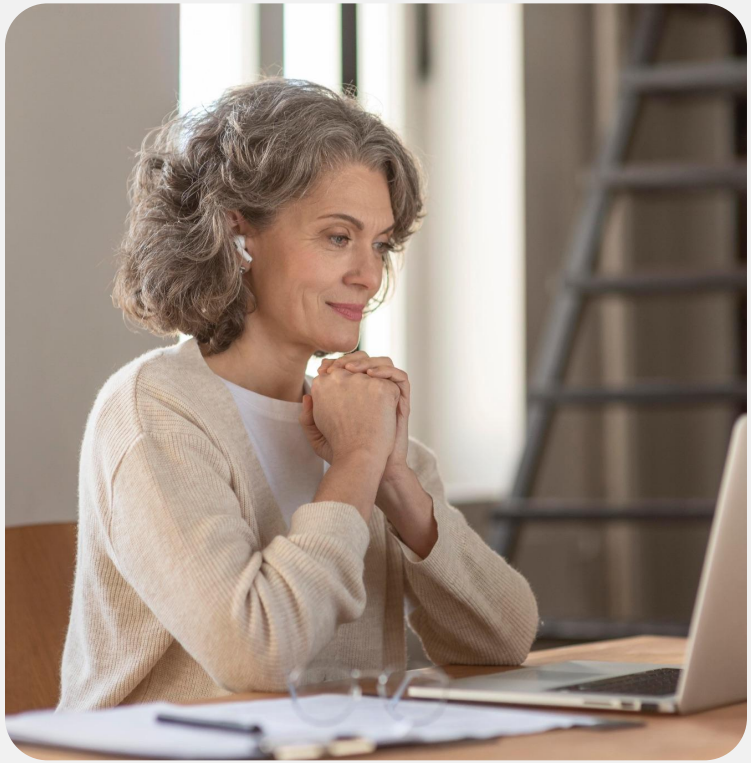
ЗАДАНИЕ



Выберите правильный вариант ответа

Определите, какие из этих значений преобразуются в True при использовании функции bool().

a. `42`
b. `0`
c. `-2`
d. `""`
e. `"0"`
f. `"Python"`
g. `"False"`
h. `[]`
i. `None`
j. `3.14`



Выберите правильный вариант ответа

Определите, какие из этих значений преобразуются в True **при использовании функции** bool().

a. `42`
b. `0`
c. `-2`
d. `""`
e. `"0"`
f. `"Python"`
g. `"False"`
h. `[]`
i. None
j. `3.14`

IT CAREER HUB

Операторы сравнения

Полезно знать

ICH
IT CAREER HUB

Операторы сравнения

Эти операторы применяются для проверки равенства, неравенства, а также для сравнения величин (больше, меньше и т.д.).

Операторы сравнения

IT CAREER HUB

Оператор	Описание	Пример	Результат
==	Равно	5 == 5	True
!=	Не равно	5 != 5	False
>	Больше	3 > 5	False
<	Меньше	4 < 6	True
>=	Больше или равно	5 >= 5	True
<=	Меньше или равно	3 <= 2	False

Операторы сравнения с числами

Операторы

== (равно)
Сравнивает два числа на равенство

!= (не равно)
Проверяет, не равны ли два числа.

> (больше)
Проверяет, больше ли одно число другого.

Примеры

```
a = 10
b = 10
print(a == b)  # True, так как a равно b
```

```
a = 10
b = 5
print(a != b)  # True, так как a не равно b
```

```
a = 8
b = 10
print(a > b)  # False, так как a меньше b
```

Операторы сравнения с числами

Операторы

< (меньше)
Проверяет, меньше ли одно число другого.

>= (больше или равно)
Проверяет, больше или равно ли одно число другому.

<= (меньше или равно)
Проверяет, меньше или равно ли одно число другому.

Примеры

```
a = 5
b = 7
print(a < b)  # True, так как a меньше b
```

```
a = 5
b = 5
print(a >= b)  # True, так как a равно b
```

```
a = 3
b = 4
print(a <= b)  # True, так как a меньше b
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

Соотнесите оператор сравнения с его результатом

1. 5 != 3	a. False
2. 7 < 6	b. True
3. 8 >= 8	
4. 4 > 2	

Соотнесите оператор сравнения с его результатом

1.	5 != 3	b.	True
2.	7 < 6	a.	False
3.	8 >= 8	b.	True
4.	4 > 2	a.	True

IT CAREER HUB

Логические операторы

Полезно знать

ICH
IT CAREER HUB

Логические операторы

Используются для выполнения логических операций над значениями или выражениями.

Логические операторы

IT CAREER HUB

Оператор	Описание	Пример	Результат
and	Логическое "И": возвращает True, если оба условия истинны	True and False	False
or	Логическое "ИЛИ": возвращает True, если хотя бы одно условие истинно	True or False	True
not	Логическое "НЕ": возвращает противоположное значение	not True	False

Логические операторы

and

Оператор **and** (логическое "И") возвращает **`True`**, если оба условия истинны.

Если хотя бы одно из условий ложно, результатом будет **`False`**.

Пример

```
a = 5
b = 10
# Проверяем что оба числа положительные
print(a > 0 and b > 0)
# True, так как оба условия истинны
```

Пример из жизни для логического оператора and

Вы хотите пойти на прогулку, но только если **солнечно** и у вас **есть свободное время**.
Оба условия должны быть истинными, чтобы вы пошли гулять.

Солнечно	**Есть свободное время**	**Пойду гулять**
True	True	True
True	False	False
False	True	False
False	False	False

"Я пойду гулять, если одновременно и погода хорошая, и у меня есть свободное время."

Логические операторы

or

Оператор **`or`** (логическое "ИЛИ")возвращает **`True`**, если хотя бы одно из условий истинно.

Если оба условия ложны, результатом будет **`False`**.

Пример

```
a = -5
b = 10
# Проверяем что хотя бы одно из чисел
# положительное
print(a > 0 or b > 0)     # True, так как b > 0 истинно
```

Пример из жизни для логического оператора or

Вы хотите заказать пиццу, но вы сделаете заказ, если у вас либо **голод**, либо **желание пиццы**. В этом случае, достаточно, чтобы хотя бы одно из условий было истинным.

Голод	**Желание пиццы**	**Закажу пиццу**
True	True	True
True	False	True
False	True	True
False	False	False

"Я закажу пиццу, если либо я голоден, либо у меня есть настроение её съесть."

Логические операторы

not

Оператор **`not`** (логическое "НЕ") возвращает противоположное значение.

Если выражение истинно, **`not`** превращает его в ложное, и наоборот.

Пример

```
a = False

print(not a)

# True, так как это обратное значение от False
```

Пример из жизни для логического оператора not

Вы на работе и планируете уйти домой. Вы пойдёте домой только в том случае, если у вас нет задач. Здесь оператор `not` работает как отрицание: если задач нет, то вы идёте домой.

Наличие задач	**Пойду домой**
True	False
False	True

"Я иду домой если у меня нет задач"

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

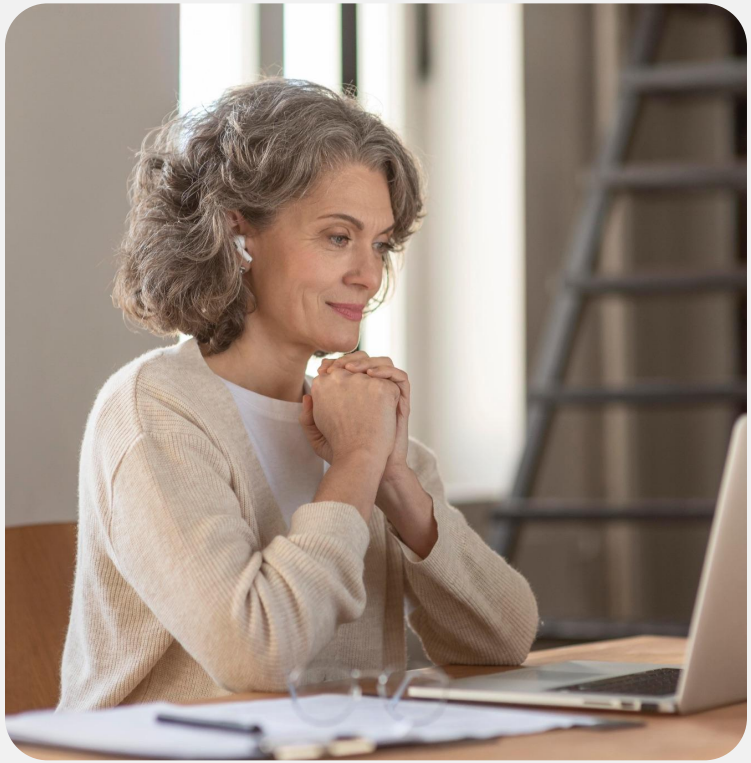
ЗАДАНИЕ

Соотнесите оператор сравнения с его результатом

1. True and False	a. True
2. True or False	b. False
3. not True	

Соотнесите оператор сравнения с его результатом

1.	True and False	b.	False
2.	True or False	a.	True
3.	not True	b.	False

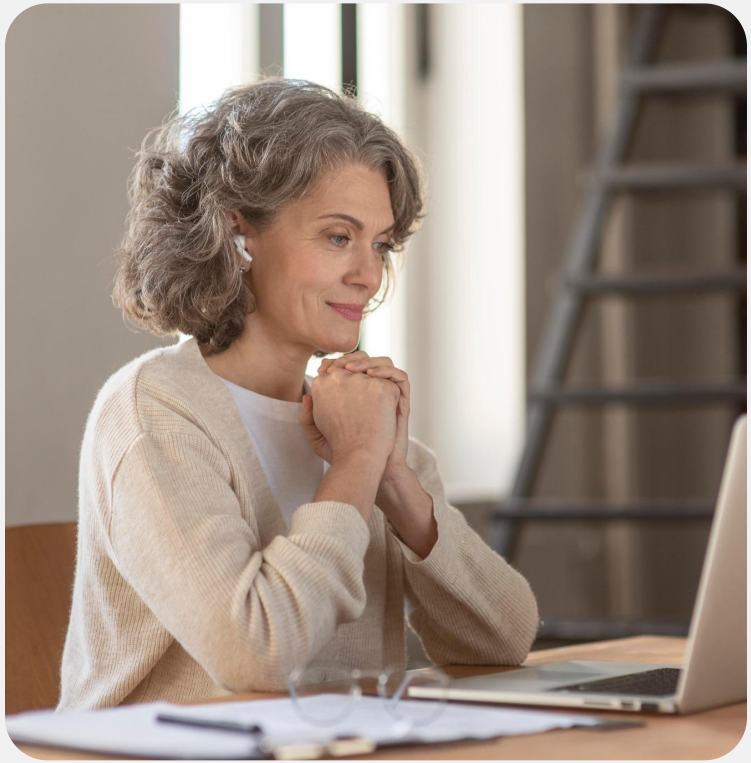


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
x = 5

print(not x > 3)
```

a. True
b. False
c. Ошибка
d. None

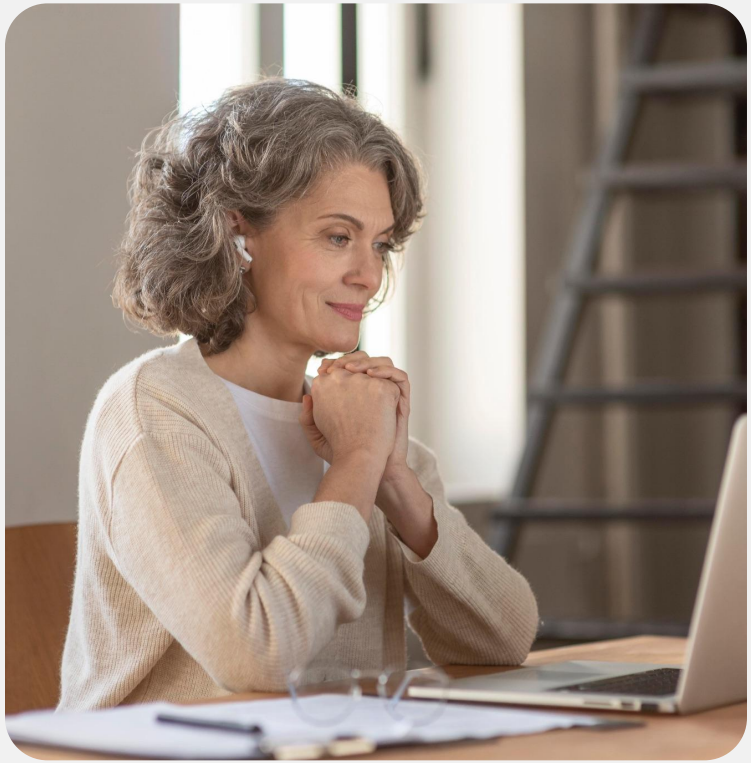


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
x = 5

print(not x > 3)
```

a. True
b. False
c. Ошибка
d. None

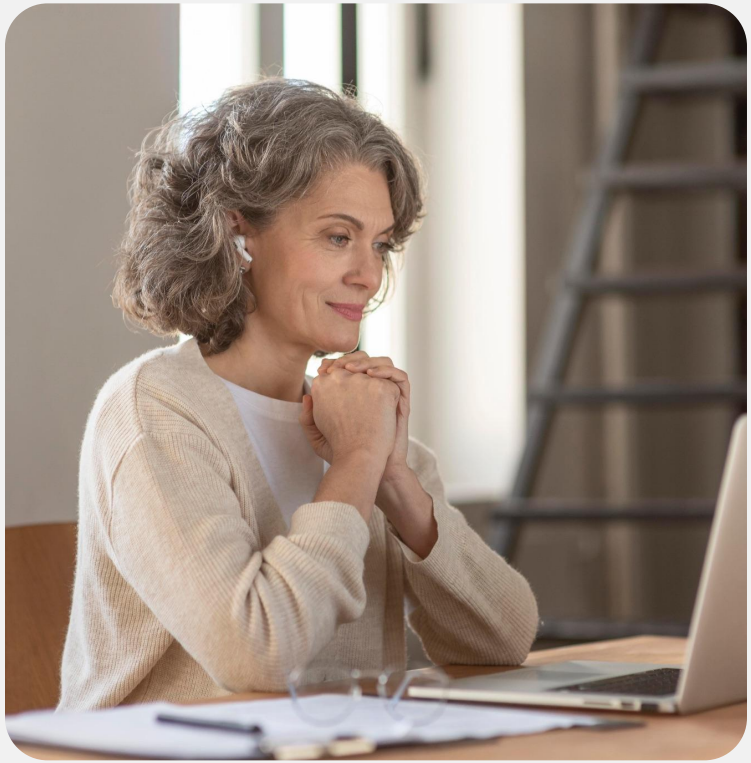


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
a = 5

b = 10

print(a < b and b < 10)
```

a. True
b. False
c. Ошибка
d. None



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
a = 5

b = 10

print(a < b and b < 10)
```

a. True
b. False
c. Ошибка
d. None

ICH IT CAREER HUB

Комбинирование логических операторов

Комбинирование

Возможности

Вы можете комбинировать несколько логических операторов в одном выражении для создания более сложных условий.

В этом примере результат будет `True`, потому что оба числа больше 0 (условие с `and` истинно), и также `a == 5` (условие с `or` истинно).

Пример

```
a = 5

b = 10

print(a > 0 and b > 0 or a == 5)
```

IT CAREER HUB

Приоритет логических операторов

Приоритет логических операторов

В Python логические операторы имеют определённый порядок выполнения (приоритет).

Порядок приоритета

1. **not** — выполняется первым.
2. **and** — выполняется вторым.
3. **or** — выполняется последним.

Порядок приоритета

Пример

a = True
b = True
c = True

```
print(not a or b and c)
```

Пояснения

В этом выражении сначала выполнится оператор `not`, затем `and`, и последним — `or`:

```
 1.   not a or b and c
# Подставим значения переменных
 2.   not True or True and True
# not True -> False
 3.   False or True and True
# True and True -> True
 4.   False or True
# False or True -> True
 5.   True
```

Скобки для изменения приоритета

Если вы хотите изменить порядок выполнения логических операторов, можно использовать скобки, чтобы явно указать, какие операции должны быть выполнены в первую очередь, так как скобки имеют наивысший приоритет.

Порядок приоритета со скобками

Пример

```
result = not (a or b) and c
```

Пояснения

Теперь оператор or выполняется первым (из-за скобок), затем not, и в последнюю очередь — and:

1. `not (a or b) and c # Подставим значения переменных`
2. `not (False or True) and True # False or True -> True`
3. `not True and True # not True -> False`
4. `False and True # False and True -> False`
5. `False`

Математические операции и операторы сравнения

Пояснения

Математические операции имеют более высокий приоритет, чем операторы сравнения. Это значит, что сначала Python выполнит все вычисления, а затем сравнить результат.

Пример

```
result = 3 + 2 > 4

# Сначала выполняется сложение (3 + 2), затем
# результат сравнивается с 4

print(result)      # True, так как 5 > 4
```

Пример

Задача

Вы хотите узнать, можно ли пойти гулять.

Условия: (погода хорошая **или** день выходной), **и** у вас есть свободное время.

Код

```
good_weather = True
is_weekend = False
free_time = True

can_go_out = (good_weather or is_weekend) and free_time
print(can_go_out)  # Результат: True
```

Пример

Задача

Вы планируете поехать в отпуск, если у вас **нет** работы **и** достаточно денег.

Код

```
have_work = True
enough_money = True

can_go_vacation = not have_work and enough_money
print(can_go_vacation)  # Результат: False
```

Пояснения

- `not have_work` → превращает `True` в `False` (у вас есть работа, значит, вы не можете поехать).
- `and enough_money` → проверяет, достаточно ли денег (результат: `False`, так как денег недостаточно).
- Окончательный результат: `False`, вы не можете поехать в отпуск.

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Таблица истинности

Таблица истинности

Показывает результат работы логических операторов для всех возможных комбинаций значений **`True`** и **`False`**.

Таблица истинности логических операторов

ICH
IT CAREER HUB

A	B	A and B	A or B	not A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

IT CAREER HUB

Альтернативная интерпретация логических операторов

Альтернативная интерпретация логических операторов

В Python логические операторы **and**, **or** и **not** можно сопоставить с арифметическими операциями.

Альтернативная интерпретация логических операторов

ICH
IT CAREER HUB

A	B	A and B (A * B)	A or B (A + B)	not A (инверсия A)
True	True	True (1 * 1)	True (1 + 1)	False (1 → 0)
True	False	False (1 * 0)	True (1 + 0)	False (1 → 0)
False	True	False (0 * 1)	True (0 + 1)	True (0 → 1)
False	False	False (0 * 0)	False (0 + 0)	True (0 → 1)

IT CAREER HUB

Двойные неравенства

Двойные неравенства

В Python можно использовать **двойные неравенства**, что позволяет проверять, попадает ли число в определённый диапазон значений. Это делает код более читабельным и удобным, поскольку не требует явного использования логических операторов and.

Синтаксис двойных неравенств:

```
min_value < variable < max_value
```

Двойные неравенства

Пример

```
x = 5
print("1 < x <= 10")
# x находится между 1 и 10 (включительно)
```

Эквивалент

```
x = 5
print(1 < x and x <= 10)
# x находится между 1 и 10 (включительно)
```

Пояснения

В этом примере Python проверяет сразу два условия:

1. 1 < x — проверка, что x больше 1.
2. x <= 10 — проверка, что x меньше или равно 10.

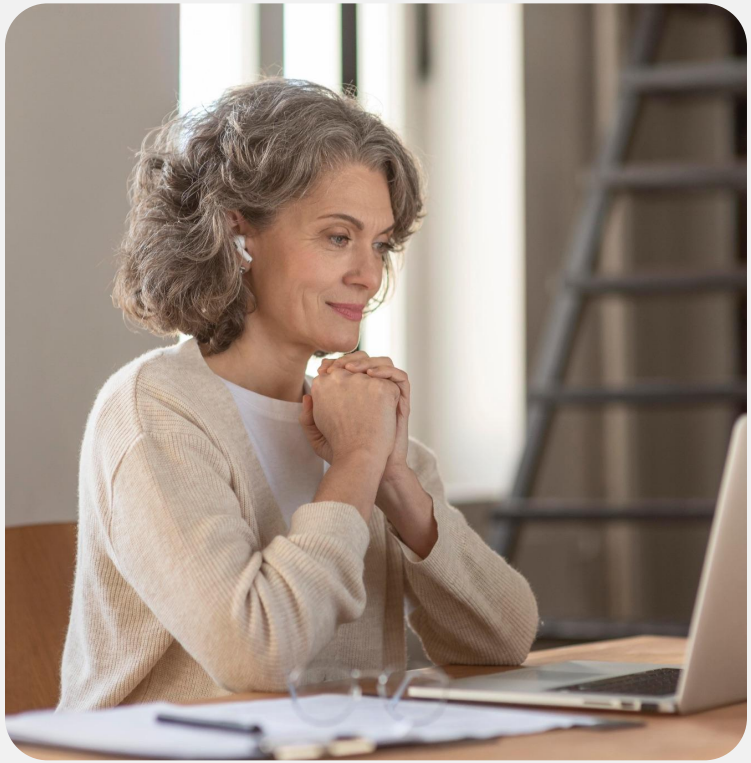
Оба подхода работают одинаково, но двойное неравенство делает код более кратким и читаемым.

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

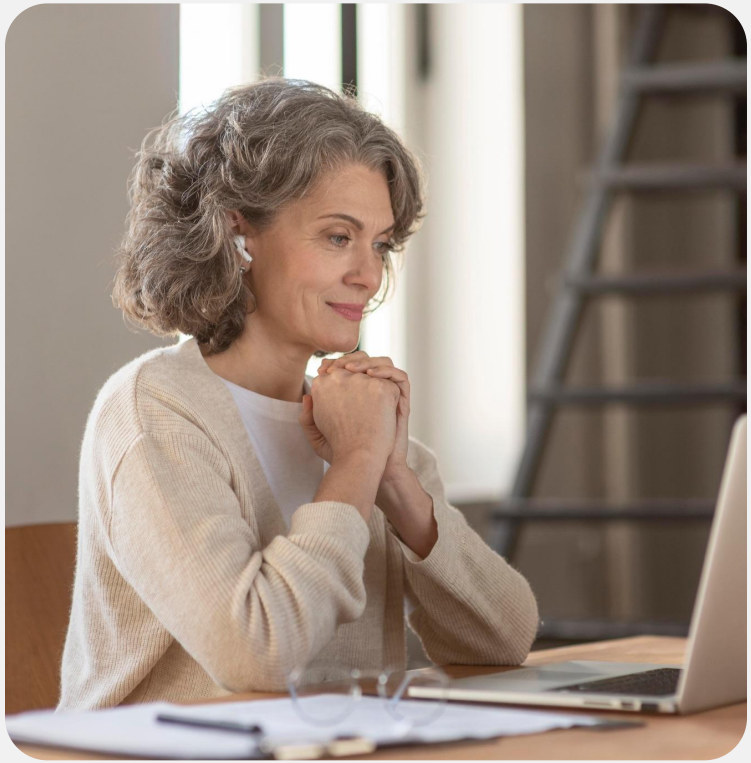


Выберите правильный вариант ответа

Что произойдёт при выполнении следующего кода?

```
x = 3
print(1 < x < 5)
```

a. True
b. False
c. Ошибка
d. None

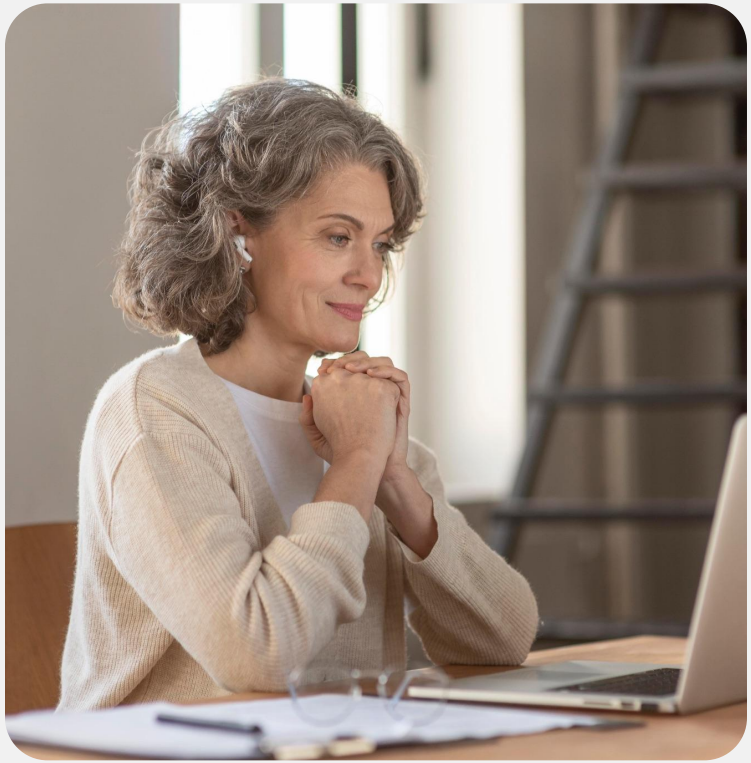


Выберите правильный вариант ответа

Что произойдёт при выполнении следующего кода?

```
x = 3
print(1 < x < 5)
```

a. True
b. False
c. Ошибка
d. None

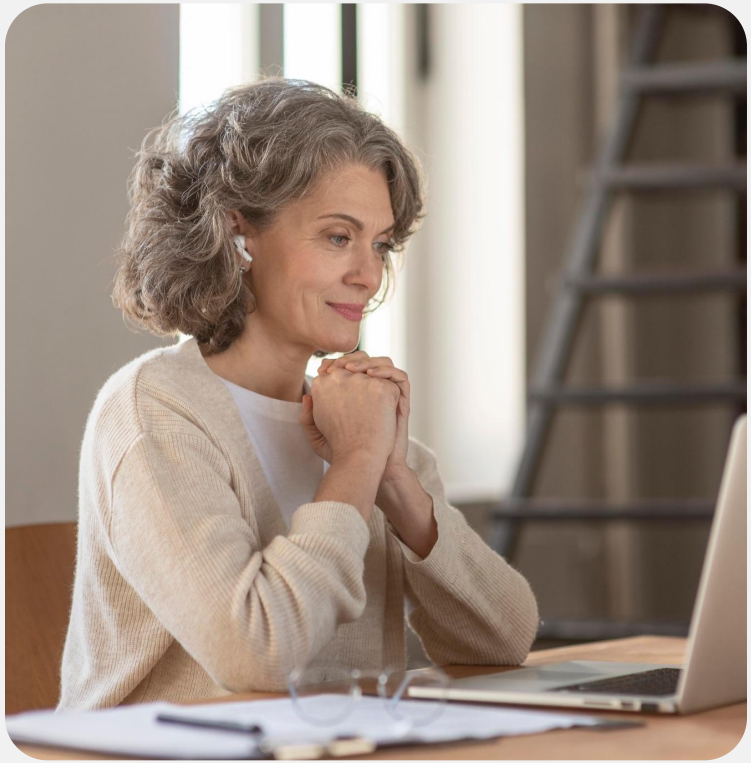


Выберите правильный вариант ответа

Что произойдёт при выполнении следующего кода?

```
a = False
b = True
c = True
print(not a or b and c)
```

a. True
b. False
c. Ошибка
d. None



Выберите правильный вариант ответа

Что произойдёт при выполнении следующего кода?

```
a = False
b = True
c = True
print(not a or b and c)
```

a. True
b. False
c. Ошибка
d. None

IT CAREER HUB

ПРАКТИЧЕСКАЯ РАБОТА

1. Число в диапазоне

Напишите программу, которая получит два числа от пользователя и выведет, попадает ли первое число в диапазон от 1 до второго числа (включая границы).

Пример вывода:

```
Введите первое число: 3
Введите второе число: 5
True
```

2. Сравнение чисел

Напишите программу, которая получит от пользователя три числа и выведет результат сравнения первого числа с остальными двумя, используя операторы `больше`, `меньше` и `равны`.

Пример вывода:

```
Введите первое число: 7
Введите второе число: 5
Введите третье число: 7
Первое больше второго: True
Первое меньше третьего: False
Первое равно третьему: True
```

IT CAREER HUB

Домашнее задание

Домашнее задание

1. **Логические операции**

Напишите программу, которая получит два логических значения от пользователя и выведет результат логических операций and, or, not для этих значений, а также сравнение на равенство и неравенство. Для операции not используйте первое число. Продумайте в каком виде получать ввод от пользователя для логического значения.

Пример вывода:

```
Enter first value: <value1>
Enter second value: <value1>
and: True
or: True
not: False
equal: False
not equal: True
```

Домашнее задание

2. **Проверка условий**

Напишите программу, которая принимает на вход логические значения двух переменных (свет включён и окно открыто) и проверяет:

- Оба ли условия выполнены.
- Хотя бы одно из условий выполнено.

Пример вывода:

```
Свет включён? True
Окно открыто? False
Оба условия выполнены? False
Хотя бы одно условие выполнено? True
```

Домашнее задание

3. *** Стоимость мобильного тарифа**

Напишите программу для расчёта стоимости использования мобильного тарифа:

- Базовая стоимость: 30 евро.
- Каждый мегабайт интернета сверх 500 МБ стоит 0.2 евро.

Программа должна запросить у пользователя количество использованных мегабайтов и вывести сколько всего он заплатил (базовый и переплата).

Пример вывода:

```
Введите использованные мегабайты: 510

Общая стоимость: 32.0
```

Заключение