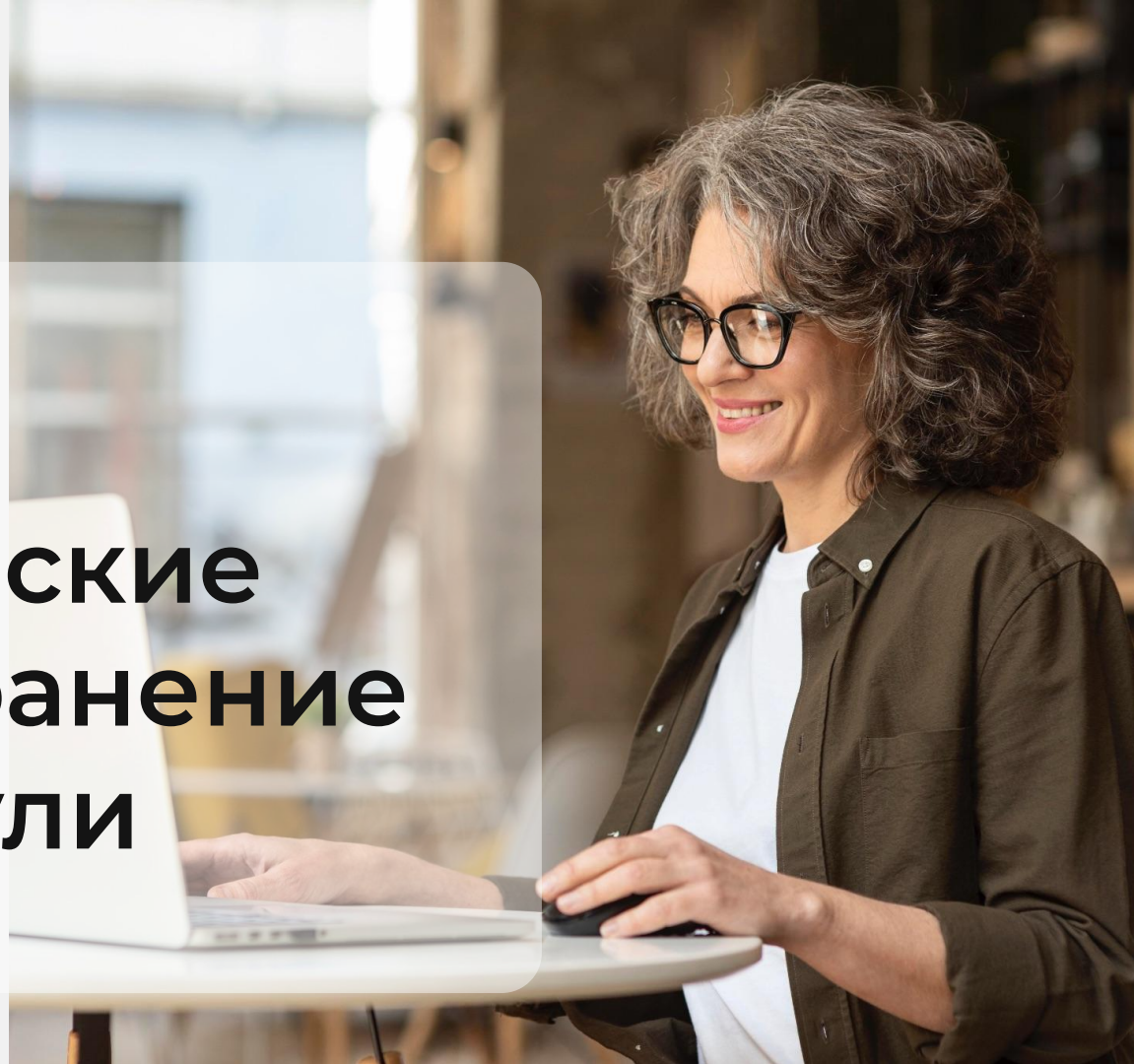
IT CAREER HUB

Python

Математические функции, хранение чисел. Модули



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] Операторы ветвления, ключевые слова
- [x] Оператор if
- [x] Неявное преобразование в bool
- [x] Оператор elif, оператор else
- [x] Использование нескольких операторов if
- [x] Вложенные условия
- [x] Тернарный оператор
- [x] Конструкция match-case

План занятия

- Встроенные математические функции
- Модуль
- Оператор import
- Стандартная библиотека Python
- Модуль math
- Хранение чисел в памяти
- Точность операций с вещественными числами
- Модуль Decimal

ICH IT CAREER HUB

ОСНОВНОЙ БЛОК

ICH IT CAREER HUB

Встроенные математические функции

Полезно знать

ICH
IT CAREER HUB

Встроенные математические функции

Python предоставляет ряд встроенных математических функций, которые упрощают выполнение базовых вычислений и работы с числами.

Эти функции не требуют дополнительного импорта и доступны по умолчанию.

Часто используемые встроенные математические функции

Функция	Описание	Пример кода	Результат
`abs()`	Возвращает абсолютное значение числа	`abs(-7)`	`7`
`max()`	Возвращает максимальное значение	`max(1, 5, 3)`	`5`
`min()`	Возвращает минимальное значение	`min(2, 8, 4, 1)`	`1`
`pow()`	Возводит число в степень	`pow(2, 3)`	`8`
`sum()`	Возвращает сумму элементов	`sum([1, 2, 3, 4])`	`10`
`round()`	Округляет число до ближайшего целого или заданного количества знаков	`round(4.567, 2)`	`4.57`

abs()

Определение

Функция `abs()` возвращает **абсолютное** значение числа (**модуль**), то есть его значение без знака.

Код

```
print(abs(-7))  # Результат: 7

print(abs(3.5))  # Результат: 3.5
```

pow()

Определение

Функция `pow()` принимает два аргумента: число и **степень**, в которую нужно возвести число.

Код

```
# Передаем сначала число, а после степень

print(pow(2, 3))

print(pow(5, 2))
```

max()

Определение

Функция `max()` возвращает **наибольшее** из переданных чисел или элементов коллекции.

Код

```
a = 1
b = 5
c = 3

# Передаем несколько значений
print(max(a, b, c))

# Передаем в виде коллекции
print(max([2, 8, 4, 1]))
```

min()

Определение

Функция `min()` возвращает **наименьшее** из переданных чисел или элементов списка.

Код

```
# Передаем несколько значений

print(min(1, 5, 3))

# Передаем в виде коллекции

print(min([2, 8, 4, 1]))
```

sum()

Определение

Функция sum() возвращает **сумму** всех элементов коллекции.

Коллекции будут изучены позднее.

Код

```
numbers = [1, 2, 3, 4]
# Передаем только в виде коллекции
print(sum(numbers))
```

round()

Определение

Функция `round()` **округляет** число до ближайшего целого или до указанного количества знаков после запятой.

Код

```
# Округление до целого

print(round(4.56))

# Округление до указанного количества знаков)

print(round(4.567, 2))
```

Банковское округление

Пояснение

По умолчанию, `round()` округляет число до **ближайшего целого**.

Если десятичная часть числа **<** 0.5, округление идёт в **меньшую** сторону, если **>** 0.5 — в **большую**.

Код

```
# Округление в меньшую сторону
print(round(3.4))

# Округление в большую сторону
print(round(3.6))
```

Банковское округление

Пояснение

Для случаев, когда десятичная часть точно **равна** 0.5 Python использует банковское округление.

Это правило заключается в том, что числа округляются в **ближайшее чётное целое**.

Код

```
print(round(1.5))  # Результат: 2

print(round(2.5))  # Результат: 2

print(round(3.5))  # Результат: 4

print(round(4.5))  # Результат: 4
```

Банковское округление

Пояснение

Функция **round**() работает аналогично с отрицательными числами.

Правила те же: числа округляются к ближайшему целому, или к ближайшему чётному числу в случае 0.5.

Код

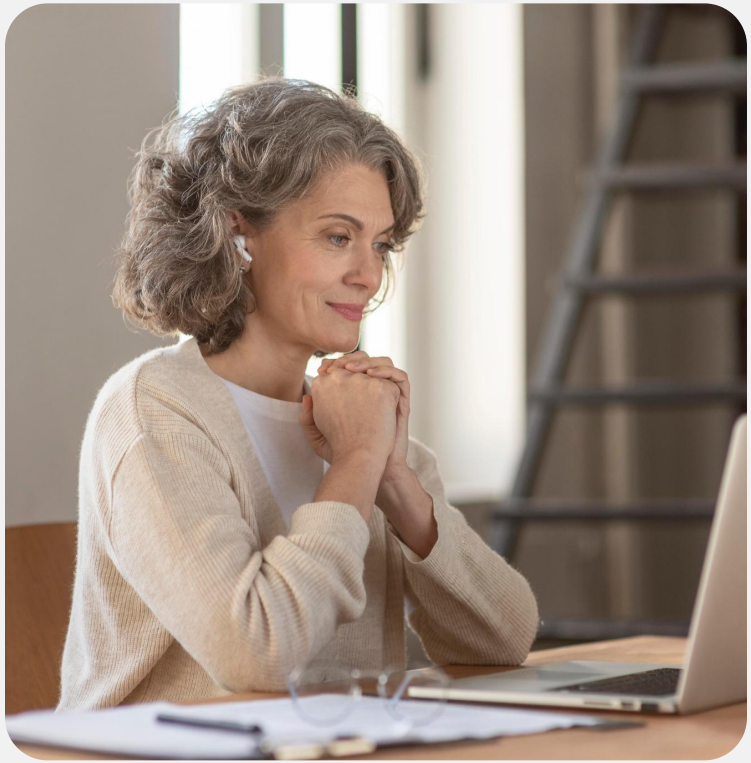
```
print(round(-1.5))  # Результат: -2

print(round(-2.5))  # Результат: -2
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

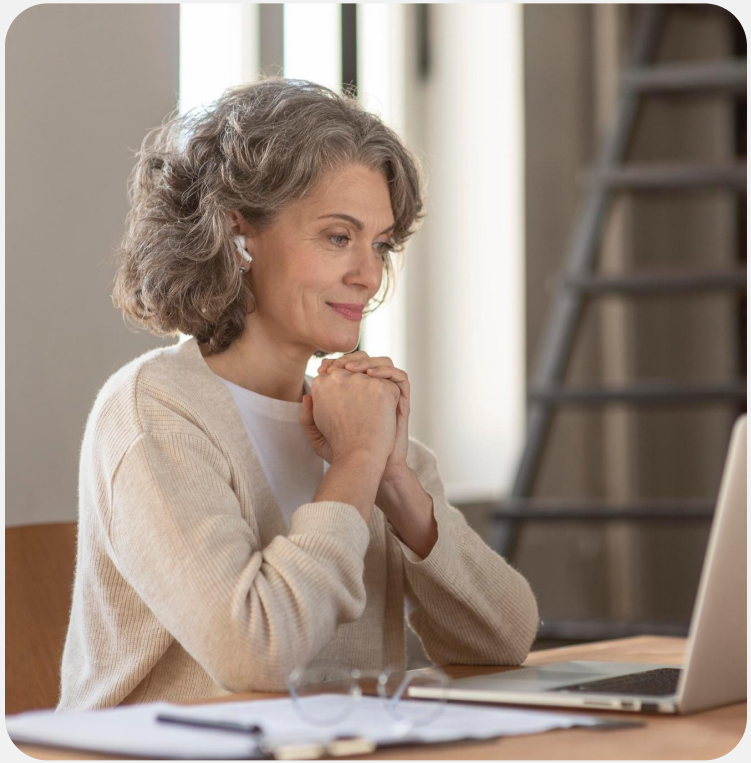


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
print(abs(-10))
```

a. -10
b. 10
c. Ошибка
d. Программа ничего не выведет

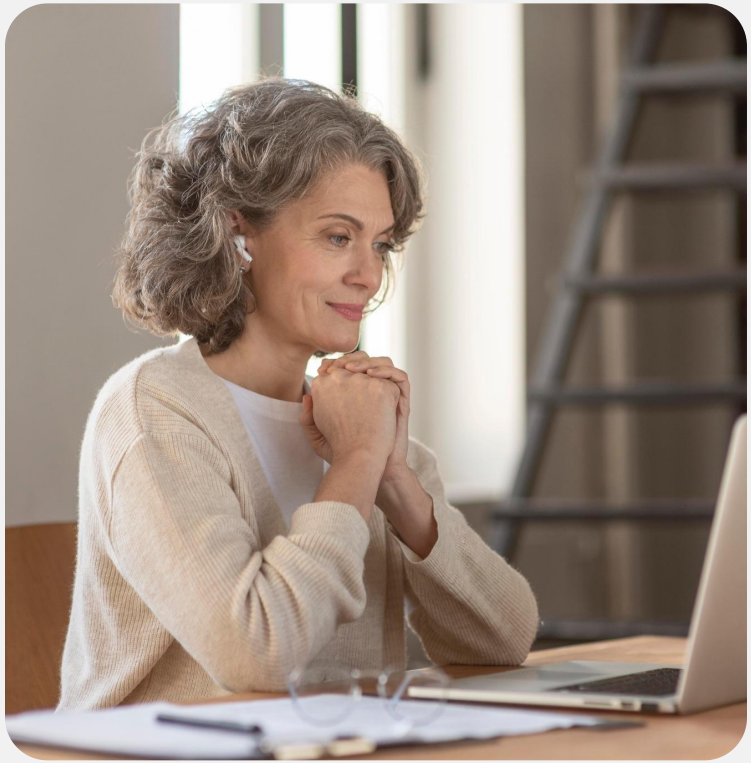


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
print(abs(-10))
```

a. -10
b. 10
c. Ошибка
d. Программа ничего не выведет

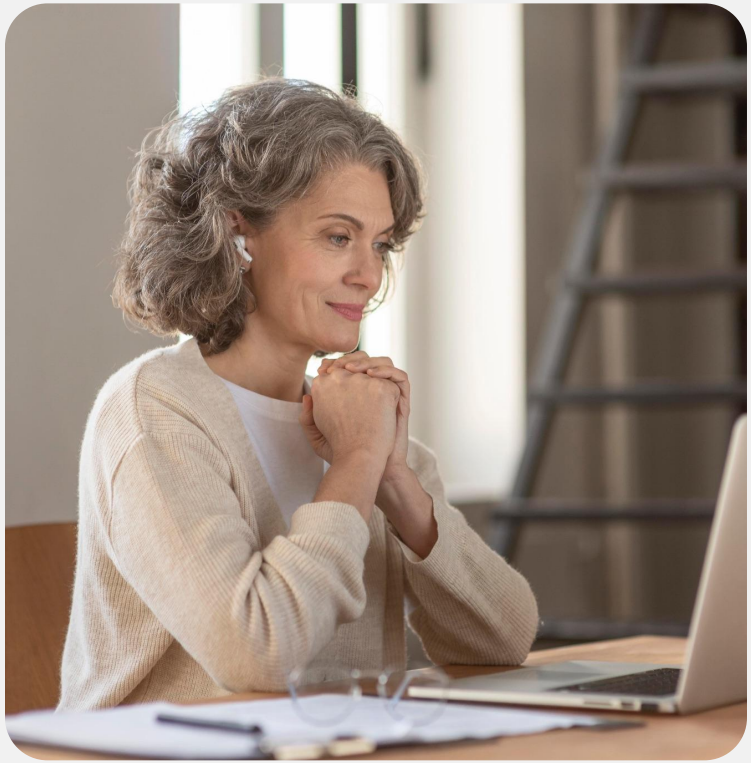


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
print(sum(3, 9, 5))
```

a. 3
b. 17
c. 9
d. Ошибка

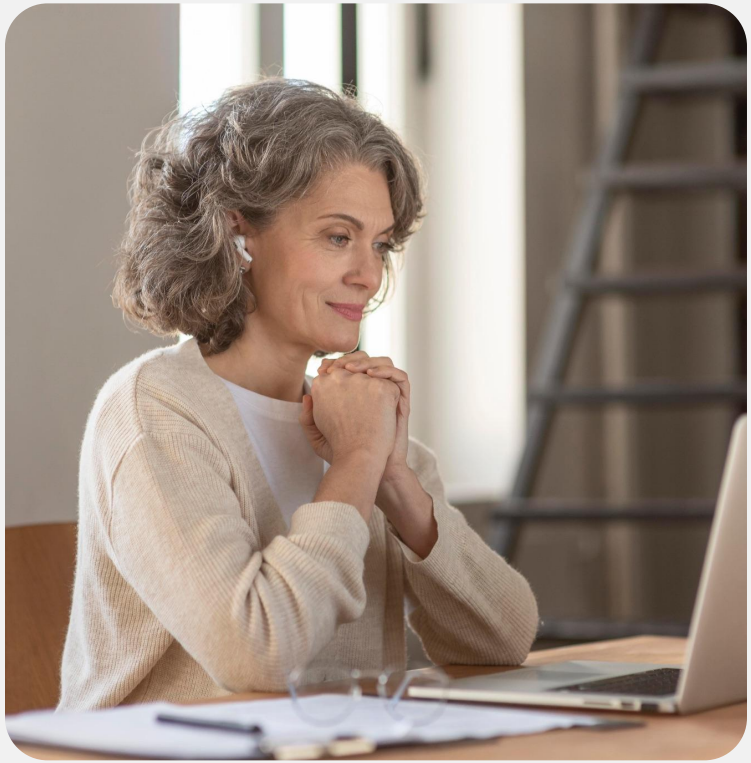


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
print(sum(3, 9, 5))
```

a. 3
b. 17
c. 9
d. Ошибка

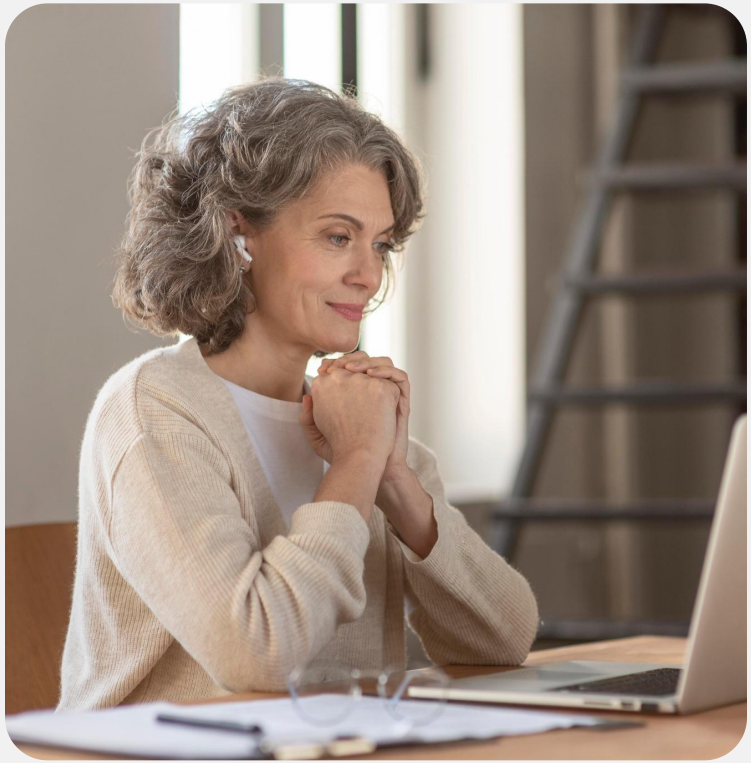


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
print(round(4.678, 2))
```

a. 4.67
b. 4.68
c. 4.7
d. Ошибка

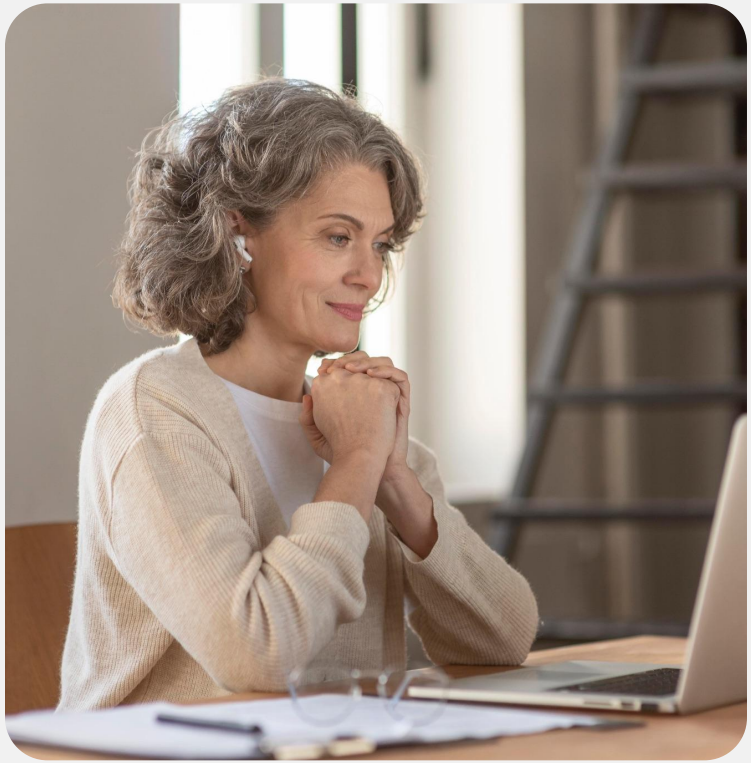


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
print(round(4.678, 2))
```

a. 4.67
b. 4.68
c. 4.7
d. Ошибка

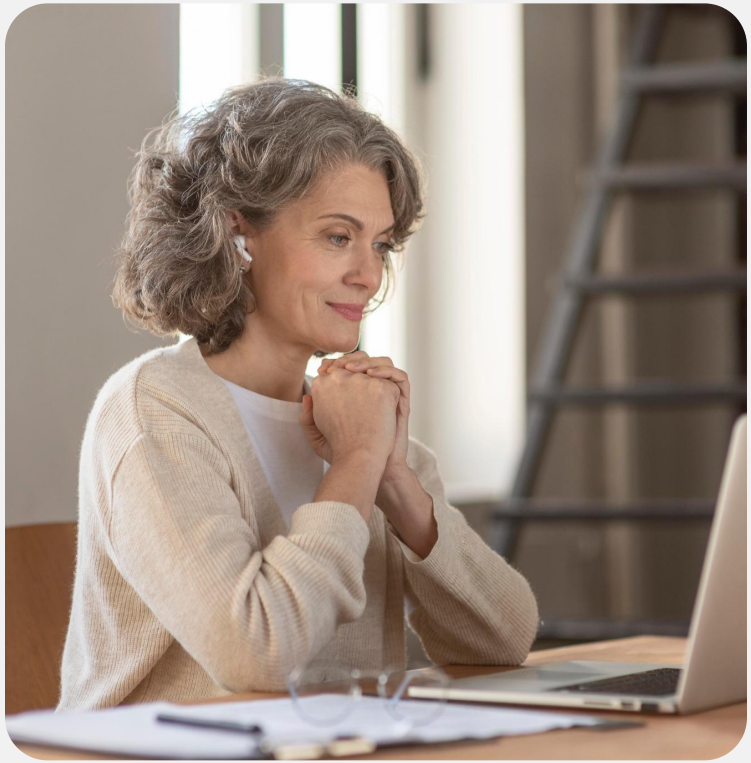


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
print(round(2.5))
```

a. 3
b. 4
c. 2
d. Ошибка



Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
print(round(2.5))
```

a. 3
b. 4
c. 2
d. Ошибка

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Модули

Полезно знать

ICH
IT CAREER HUB

Модуль

Это файл с расширением .py, который содержит код на языке Python: функции, классы, переменные, и даже исполняемые инструкции.

Зачем нужны модули?

- **Организация кода:** Модули помогают разбить большие программы на более мелкие, логически связанные части, что упрощает поддержку и понимание кода.
- **Повторное использование:** Модуль можно импортировать в несколько разных программ, не переписывая код каждый раз заново.
- **Изоляция кода:** Модули помогают изолировать код, что позволяет избежать конфликтов имён функций, классов и переменных.

Полезно знать

ICH
IT CAREER HUB

Использование модулей

Модули в Python можно **импортировать** в программу, используя ключевое слово import. Это позволяет использовать необходимый функционал в других частях программы.

IT CAREER HUB

Оператор import

Полезно знать

ICH
IT CAREER HUB

Оператор import

`import` — это ключевое слово в Python, которое используется для загрузки **модулей** (или пакетов) в программу.

Способы использования import

Способ использования	Пример кода
Импорт всего модуля	`import math`
Импорт конкретных функций, переменных	`from math import sqrt, pi`
Импорт всего модуля с указанием псевдонима	`import math as m`
Импорт всего содержимого модуля	`from math import *`

Импорт всего модуля

Пояснение

Самый простой способ использования оператора `import` — это импортировать весь модуль.

Для использования функций нужно обращаться к ним через имя модуля, например, **math.sqrt()**.

Код

```
import math

# Используем функцию sqrt() из модуля
#math

print(math.sqrt(16))
```

Импорт конкретных элементов

Пояснение

Можно импортировать конкретные функции, классы или переменные из модуля с помощью ключевого слова `from`.

Нужно использовать их напрямую, например, **sqrt()** или **pi**

Код

```
from math import sqrt, pi

# Используем функцию sqrt()
# без указания имени модуля
print(sqrt(16))
# Используем константу pi
print(pi)
```

Импорт модуля с псевдонимом

Пояснение

Если название модуля длинное или если вы хотите использовать более короткое имя, можно задать **псевдоним** для модуля с помощью ключевого слова as.

Для использования функций нужно обращаться обращаемся через псевдоним, например, **m.sqrt()**.

Код

```
import math as m

# Используем функцию sqrt() из модуля
#math, но с псевдонимом "m"
print(m.sqrt(16))
```

Импорт всего содержимого модуля

Пояснение

Можно импортировать всё содержимое модуля в текущую область видимости с помощью оператора *.

Теперь функции и переменные можно использовать напрямую без указания имени модуля.

Однако это не рекомендуется, так как может привести к конфликтам имён и затруднить чтение и сопровождение кода.

Код

```
from math import *

# Используем функцию sqrt() напрямую
print(sqrt(16))
# Используем константу pi напрямую
print(pi)
```

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Стандартная библиотека Python

Полезно знать

ICH
IT CAREER HUB

Стандартная библиотека Python

Это набор **модулей** и **пакетов**, которые включены в Python по умолчанию и предоставляют широкий спектр функциональности для выполнения различных задач.

Модули из стандартной библиотеки

- **os** — взаимодействие с операционной системой
- **sys** — доступ к переменным и функциям, связанным с интерпретатором Python
- **math** — математические функции
- **datetime** — работа с датами и временем
- **json** — работа с JSON-форматом
- **re** — работа с регулярными выражениями
- **random** — генерация случайных чисел

IT CAREER HUB

Модуль math

Полезно знать

Модуль math

Это стандартный модуль, предоставляющий функции для выполнения различных математических операций, таких как вычисление тригонометрических функций, логарифмов, возведение в степень, и многое другое.

Как использовать модуль math?

Чтобы использовать функции из модуля math, необходимо сначала импортировать его в свой код

```
import math
```

Основные функции модуля math

Функция/переменная	Описание
`math.sqrt(x)`	Возвращает квадратный корень числа x
`math.pow(x, y)`	Возводит число x в степень y
`math.factorial(x)`	Возвращает факториал числа x (произведение всех чисел от 1 до x)
`math.ceil(x)`	Округляет число x в большую сторону (до ближайшего целого числа)
`math.floor(x)`	Округляет число x в меньшую сторону (до ближайшего целого числа)
`math.pi`	Константа, представляющая число Пи (π ≈ 3.14159)

Примеры для функций модуля math

Квадратный корень

```
import math

x = 16
result = math.sqrt(x)
print("Корень из", x, "равен", result)
```

Округление числа

```
import math
# Округление вверх: 5
print(math.ceil(4.3))
# Округление вниз: 4
print(math.floor(4.8))
```

Примеры для функций модуля math

Вычисление факториала

```
import math

n = 5
factorial_result = math.factorial(n)
print("Факториал числа", n, ":",
factorial_result)
```

Константы в модуле math

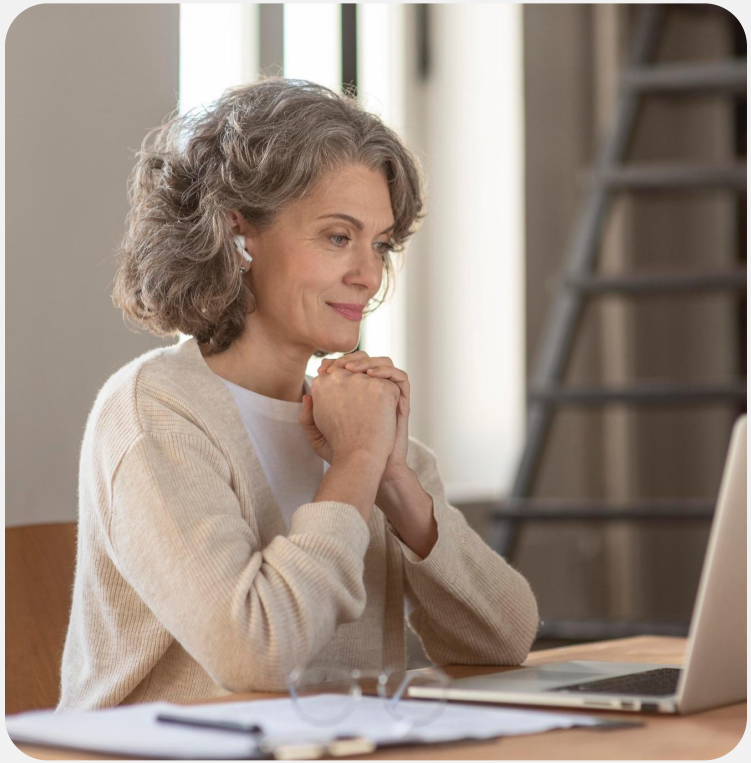
- `math.pi:` Число Пи (π ≈ 3.14159)
- `math.e:` Основание натурального логарифма (e ≈ 2.71828)

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

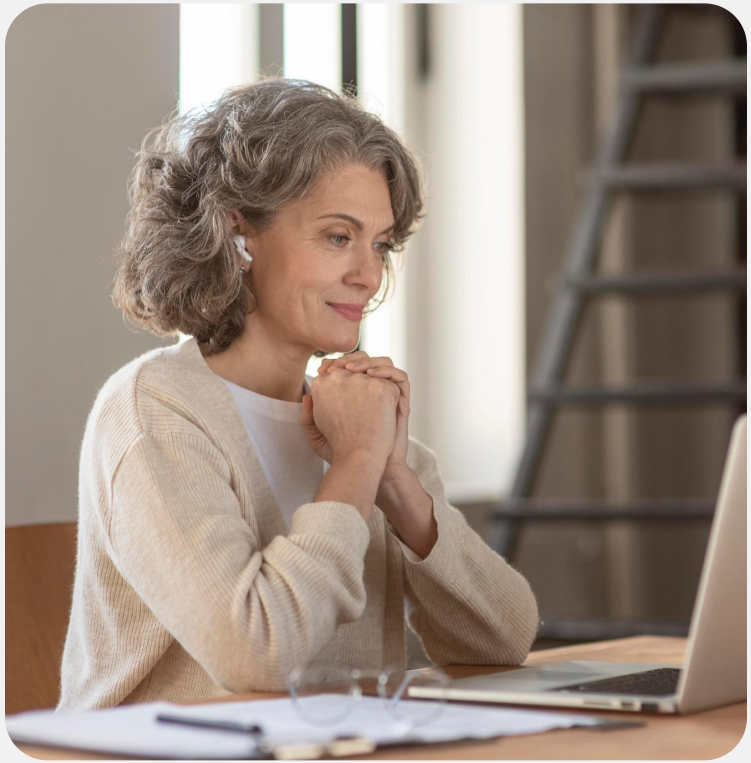


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
import math

print(math.sqrt(25))
```

a. 25
b. 5
c. Ошибка
d. Программа ничего не выведет

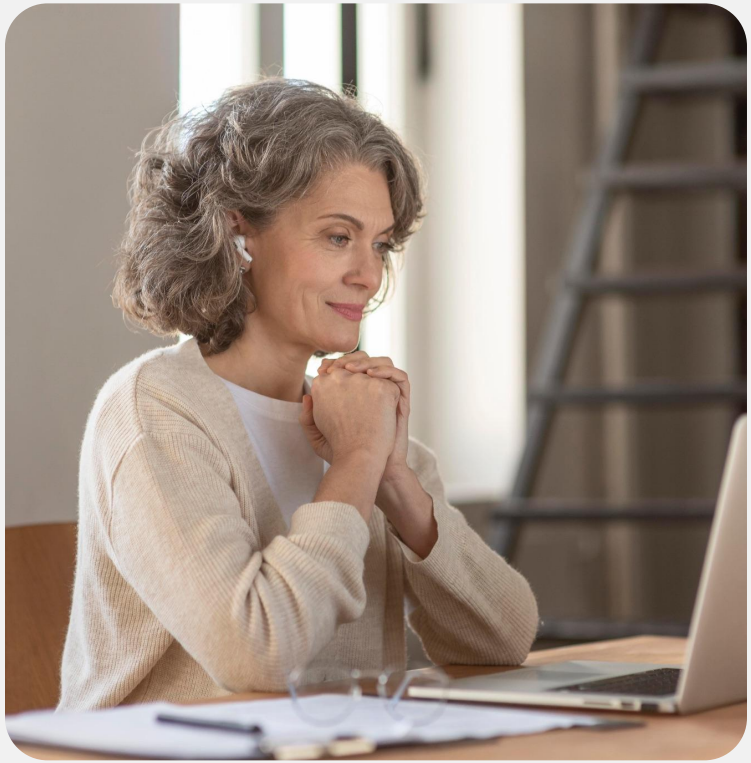


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
import math

print(math.sqrt(25))
```

a. 25
b. 5
c. Ошибка
d. Программа ничего не выведет

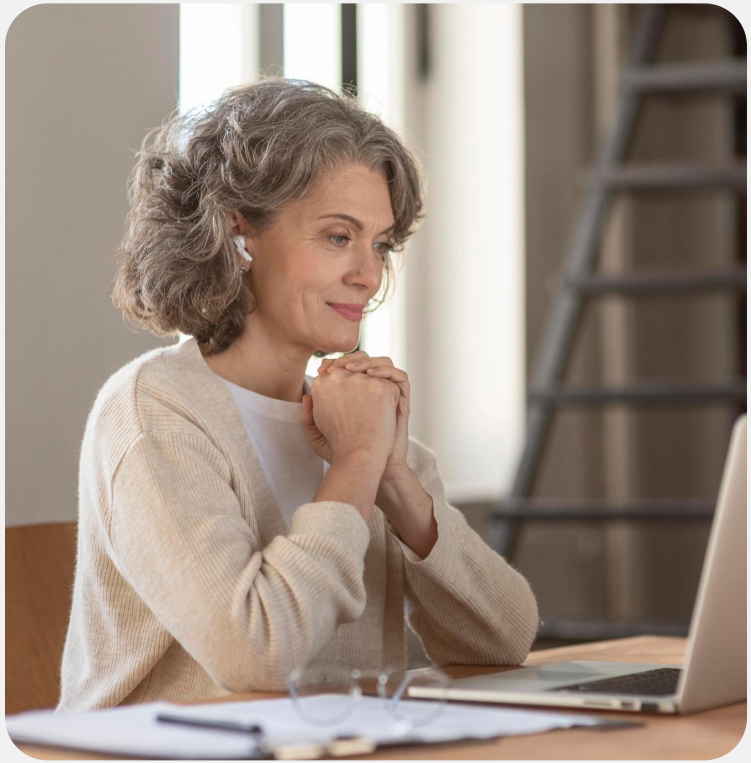


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
import math

print(pi)
```

a. 3.14
b. 3.141592653589793
c. Ошибка
d. Программа ничего не выведет



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
import math

print(pi)
```

a. 3.14
b. 3.141592653589793
c. Ошибка
d. Программа ничего не выведет

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Биты и байты

Полезно знать

ICH
IT CAREER HUB

Биты и байты

Компьютеры работают с данными в виде битов и байтов. Эти единицы являются основой для представления информации в памяти.

Полезно знать

ICH
IT CAREER HUB

Бит

Это минимальная единица информации, которая может иметь одно из двух значений: 0 или 1.

Биты представляют данные в двоичной системе счисления, которая лежит в основе всех вычислительных процессов.

Полезно знать

ICH
IT CAREER HUB

Байт

Это набор из **8 битов**. Он используется как базовая единица для хранения данных в памяти компьютера.

Например, 1 байт может хранить одно целое число в пределах от 0 до 255 (или от -128 до 127 в случае знаковых чисел).

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Хранение чисел в памяти

Полезно знать

ICH
IT CAREER HUB

Хранение чисел в памяти

Числа хранятся в памяти в двоичной системе счисления.

Целые числа (int)

32-битные целые числа:

- Занимают 4 байта памяти (32 бита).
- Могут хранить значения от -2^31 до 2^31 - 1 (от -2 147 483 648 до 2 147 483 647).

64-битные целые числа:

- Занимают 8 байт памяти (64 бита).
- Могут хранить значения от -2^63 до 2^63 - 1 (от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807).

Полезно знать

Хранение чисел в памяти

Python — это высокоуровневый язык программирования, и одна из его особенностей заключается в том, что целые числа в Python не имеют фиксированного размера.

Вещественные числа (float)

В Python вещественные числа обычно представлены в формате 64-битного числа с плавающей запятой.

- Занимают 8 байт памяти (64 бита).
- В таком формате числа хранятся как комбинация мантиссы и экспоненты.

Пример диапазона для 64-битных чисел с плавающей запятой:

- Примерно от ±10^−308 до ±10^308.

Полезно знать

ICH
IT CAREER HUB

Пример целого числа

Число **42** в двоичной системе записывается как:
00101010 (в 8-битном формате)

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Точность операций с вещественными числами

Полезно знать

ICH
IT CAREER HUB

Вещественные числа

Числа с плавающей запятой (вещественные числа) в Python хранятся по стандарту IEEE 754, который используется большинством современных систем.

Вещественные числа

- **1 бит** — знак числа (положительное или отрицательное).
- **11 битов** — экспонента, которая указывает на степень числа.
- **52 бита** — мантисса, представляющая саму дробную часть числа.

Вещественные числа

Пример

$0.0001011 = 1.011 * 10^{-4}$

Здесь

Мантисса: 1.011 (сохраняет значимые цифры числа, избавившись от нолей)

Экспонента: -4 (указывает на степень 10, то есть количество потерянных нолей)

Полезно знать

Ограниченная точность

Компьютер может хранить только определённое количество цифр в мантиссе.

В Python это обычно 52 бита для мантиссы (в 64-битной системе).

Погрешность в вычислениях

Пример

Когда мы складываем такие числа, могут возникнуть неточности:

```
print(0.1 + 0.2)  # Ожидаем 0.3
```

Результат

0.30000000000000004

Экспресс-опрос

Почему возникают такие ошибки?

Полезно знать

ICH
IT CAREER HUB

Почему возникают ошибки?

Компьютеры используют двоичную систему, а многие десятичные дроби, такие как 0.1 или 0.2, не могут быть точно представлены в двоичной системе.

Погрешность в вычислениях

Пример

Представление числа 0.1 в двоичной системе

0.0001100110011001100110011...
(бесконечная последовательность)

Пояснения

Компьютер вынужден обрезать эту последовательность, потому что у него есть ограниченное количество битов для хранения числа. В итоге, точное значение не может быть сохранено, и при вычислениях возникают ошибки.

Сравнение вещественных чисел

Операции сравнения

Из-за этих погрешностей сравнение вещественных чисел может работать не так, как ожидается.

Например, сравнение 0.1 + 0.2 == 0.3 вернет False из-за малой ошибки в представлении.

Пример

```
print(0.1 + 0.2 == 0.3)  # Вывод: False
```

Сравнение вещественных чисел

Округление результата

Для устранения небольших ошибок при выводе или сравнении результатов можно использовать функцию round(), которая округляет вещественные числа до указанного количества знаков после запятой.

Пример

```
a = 0.1 + 0.2

print(round(a, 2))          # Вывод: 0.3

print(round(a, 2) == 0.3)   # Вывод: True
```

Полезно знать

ICH
IT CAREER HUB

Использование модуля decimal

Если требуется высокая точность при работе с вещественными числами, в Python можно использовать модуль decimal, который позволяет задавать точность операций и избегать ошибок округления.

IT CAREER HUB

Модуль Decimal

Полезно знать

ICH
IT CAREER HUB

Модуль Decimal

Позволяет представлять числа с плавающей запятой в десятичной системе без потери точности.

Модуль Decimal

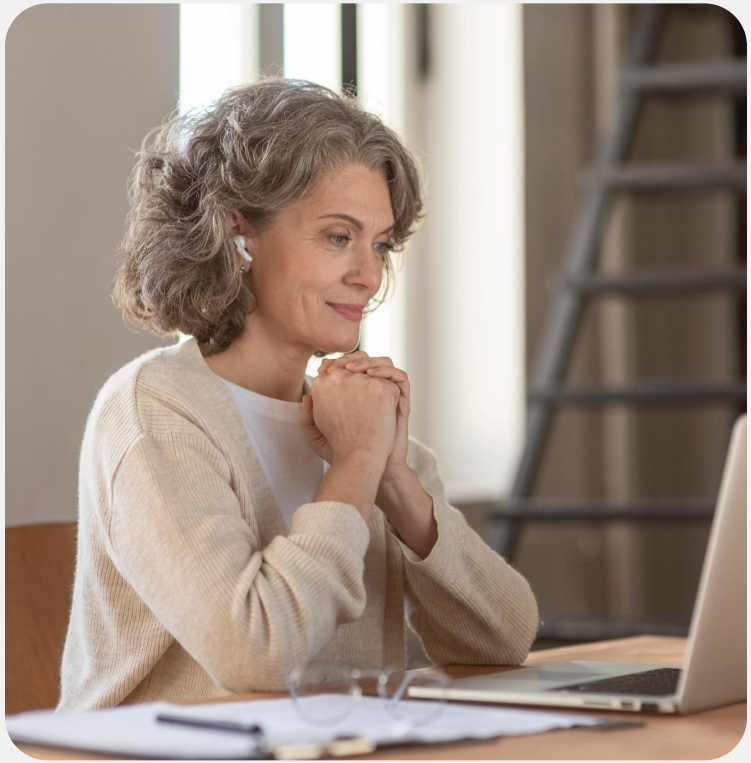
```python
# Импортируем класс Decimal из модуля decimal
from decimal import Decimal

# Создаём два объекта класса Decimal с помощью строк '0.1' и '0.2'
# Это гарантирует, что числа будут точно представлены в десятичном формате
a = Decimal('0.1')
b = Decimal('0.2')
# Сложение без ошибки округления
c = a + b
print(c)  # Вывод: 0.3
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

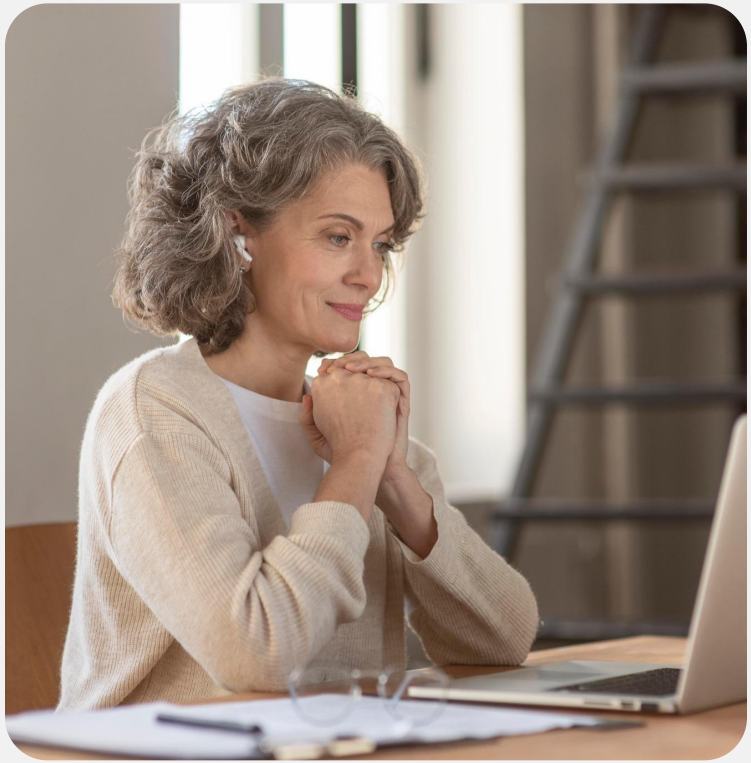
ЗАДАНИЕ



Выберите правильный вариант ответа

Сколько битов в одном байте?

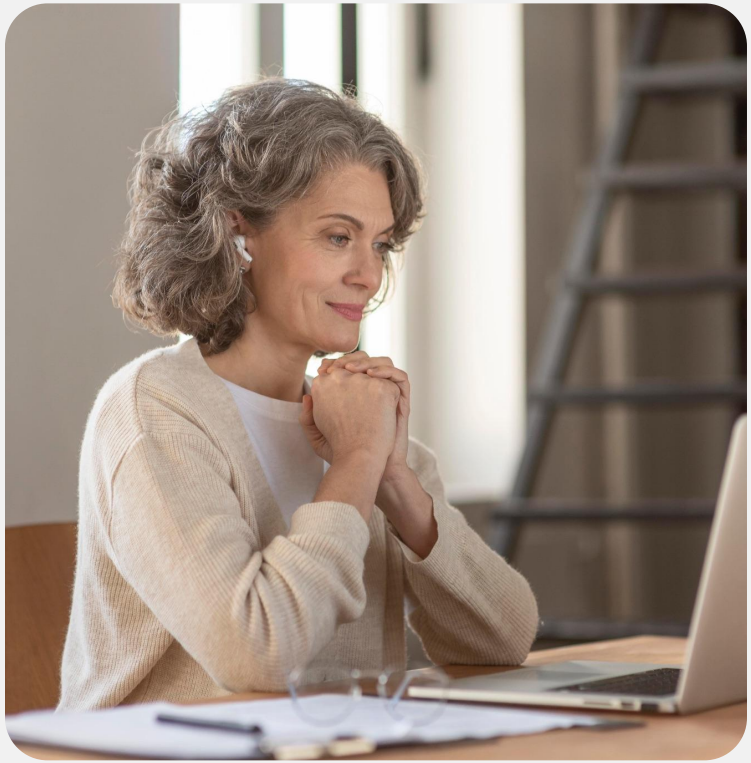
a. 4
b. 8
c. 16
d. 32



Выберите правильный вариант ответа

Сколько битов в одном байте?

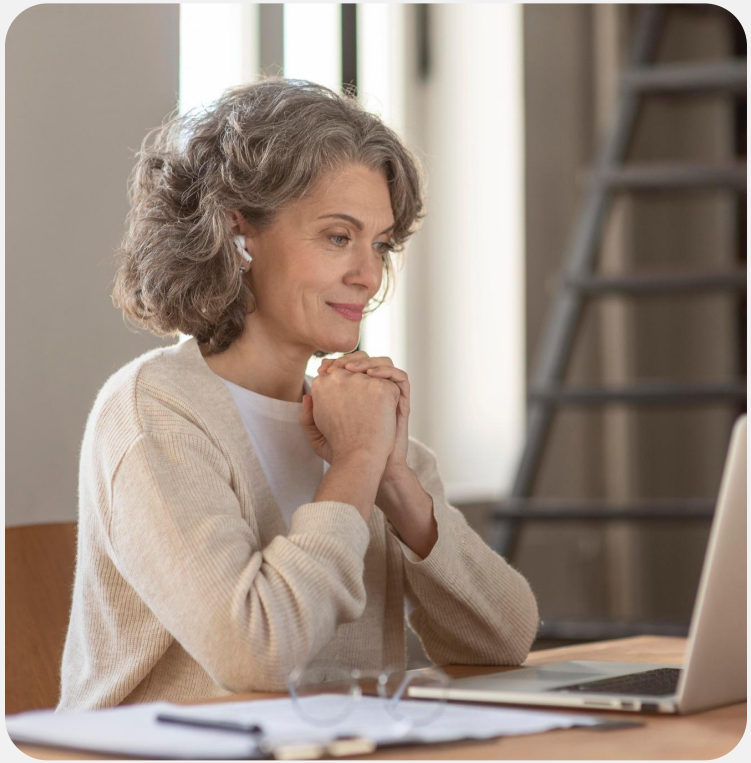
a. 4
b. 8
c. 16
d. 32



Выберите правильные варианты ответа

Какое значение может хранить 1 байт?

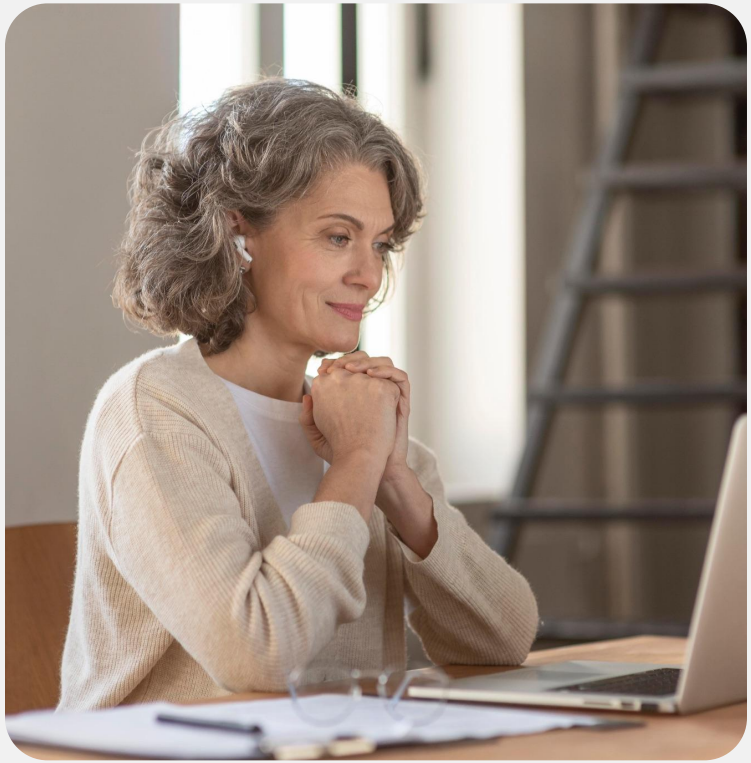
a. от 0 до 255
b. от -255 до 255
c. от -128 до 127
d. от 0 до 1024



Выберите правильные варианты ответа

Какое значение может хранить 1 байт?

a. от 0 до 255
b. от -255 до 255
c. от -128 до 127
d. от 0 до 1024

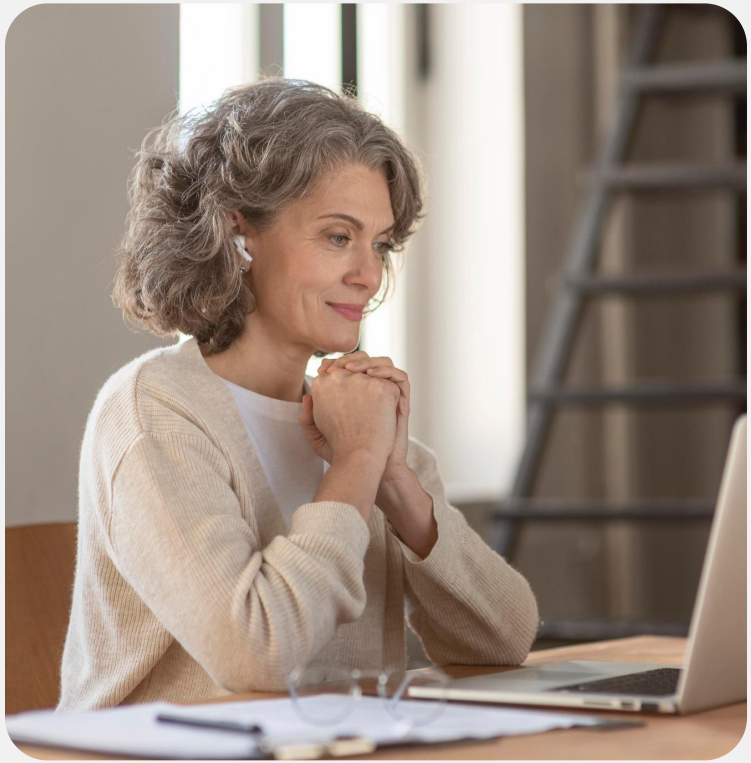


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
from decimal import Decimal

a = Decimal('1.1')

b = Decimal('2.2')

print(a + b)
```

a. 3.3000000000000003
b. 3.3
c. Ошибка
d. Программа ничего не выведет



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
from decimal import Decimal

a = Decimal('1.1')

b = Decimal('2.2')

print(a + b)
```

a. 3.3000000000000003
b. 3.3
c. Ошибка
d. Программа ничего не выведет

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Практическая работа

1. Скидки для оптовых покупателей

Напишите программу, которая получает от пользователя цену на товар и количество товара. Программа должна рассчитать итоговую сумму заказа со скидкой, исходя из количества товара. Цену нужно округлить до сотых.

Условия скидки:

- Если количество товара от 10 до 19 — скидка 5%.
- Если количество товара от 20 до 49 — скидка 10%.
- Если количество товара от 50 до 99 — скидка 15%.
- Если количество товара 100 и более — скидка 20%.

Пример вывода:

```
Введите цену товара: 100
Введите количество товара: 25
Сумма заказа с учётом скидки: 2250.00
```

2. Площадь и длина круга

Напишите программу, которая определяет площадь круга и длину окружности по радиусу, полученному от пользователя, и выводит их на экран, округлив до сотых.

Формулы:

- Площадь круга: S = π * r^2
- Длина окружности: C = 2 * π * r

Пример вывода:

```
Введите радиус: 5
Площадь круга: 78.54
Длина окружности: 31.42
```

IT CAREER HUB

ВОПРОСЫ

Домашнее задание

1. Математическое округление

Напишите программу, которая принимает число с плавающей точкой и округляет его до целого числа в соответствии с правилами школьной математики, а не банковского округления. Программа должна корректно работать как с положительными, так и с отрицательными числами.

Пример вывода:

```
Введите вещественное число: 2.5
Округленное значение: 3
```

Пример вывода:

```
Введите вещественное число: -2.5
Округленное значение: -3
```

Домашнее задание

2. Гипотенуза треугольника

Напишите программу, которая запрашивает у пользователя длины двух катетов прямоугольного треугольника и вычисляет длину гипотенузы. Гипотенуза равна квадратному корню из суммы квадратов катетов.

Пример вывода:

```
Введите длину первого катета: 3
Введите длину второго катета: 4
Длина гипотенузы: 5.0
```

Заключение