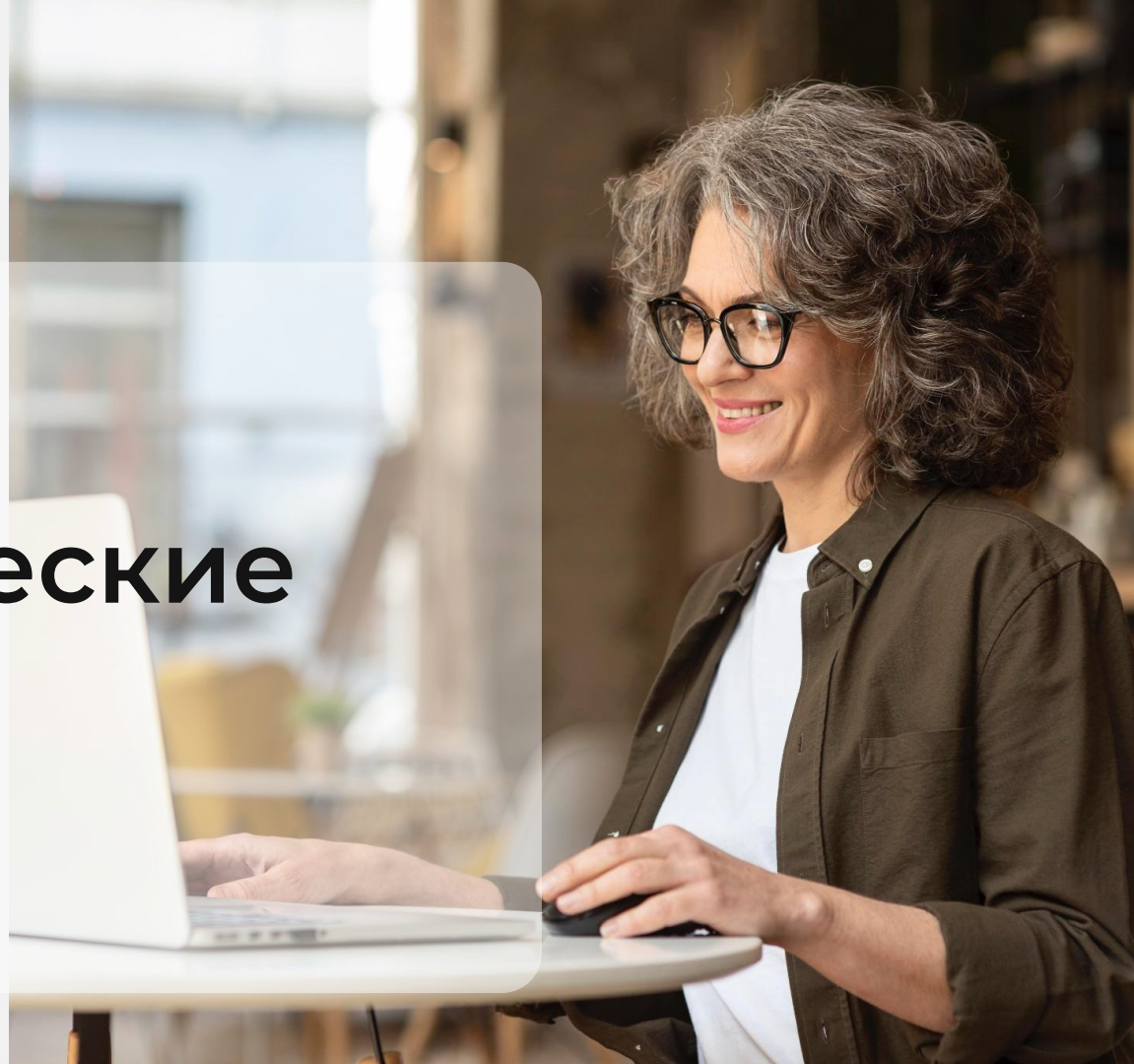
IT CAREER HUB

Python

Арифметические операторы, выражения



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] Тип данных, объект, динамическая типизация
- [x] Примитивные типы данных
- [x] Возвращаемое значение
- [x] Процедуры в Python, передача аргументов в функцию
- [x] Функция type
- [x] Строковые операции
- [x] Работа с кавычками в строках
- [x] Экранирование символов
- [x] Параметры sep и end

План занятия

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

ICH IT CAREER HUB

ОСНОВНОЙ БЛОК

IT CAREER HUB

Числа

В Python числа представлены типами:

- Целые числа (`int`)
- Числа с плавающей точкой или вещественные (`float`)

ICH IT CAREER HUB

Арифметические операции с числами

Полезно знать

Арифметические операции

Python поддерживает стандартные арифметические операции, такие как сложение, вычитание, умножение и деление.

Важно

Арифметические операции могут быть выполнены с числами (целыми или вещественными) и переменными, содержащими числовые значения

Арифметические операции

IT CAREER HUB

Операция	Название	Описание
+	Сложение	Складывает два значения.
-	Вычитание	Вычитает одно значение из другого.
*	Умножение	Умножает два значения.
/	Деление	Делит одно значение на другое.
//	Целочисленное деление	Делит одно значение на другое, возвращая целую часть результата.
%	Остаток от деления	Возвращает остаток от деления двух значений.
**	Возведение в степень	Возводит значение в указанную степень.

Важно

- При обычном делении результат всегда является дробным числом.
- Если в выражении хотя бы одна часть типа `float`, то результат будет тоже `float`.
- Результат целочисленного деления всегда округляется в меньшую сторону.

Примеры

```python
# Сложение
print("Сложение:", "3 + 5 =", 3 + 5)
# Вычитание
print("Вычитание:", "7 - 2 =", 7 - 2)
# Умножение
print("Умножение:", "4 * 6 =", 4 * 6.0)
# Деление
print("Деление:", "8 / 2 =", 8 / 2)
# Целочисленное деление
print("Целочисленное деление:", "7 // 2 =", 7 // 2.0)
# Остаток от деления
print("Остаток от деления:", "7 % 3 =", 7 % 3)
# Возведение в степень
print("Возведение в степень:", "2 ** 3 =", 2 ** 3)
```

Использование переменных

Пояснения

1. Если какой-то результат в последующем нужно использовать повторно, то можно заключить его в переменную.

2. Если какое-то действие нужно совершить один раз, то хранить информацию в переменной бессмысленно

Примеры

1.
```
sum1 = 3 + 5
a = sum1 + 1
b = sum1 * 3
```

2.
```
print(3 + 5)
```

ICH IT CAREER HUB

ZeroDivisionError

ZeroDivisionError

`ZeroDivisionError` — это исключение, которое возникает когда происходит попытка деления на ноль.

ZeroDivisionError

Когда возникает

- При делении целых чисел на ноль
- При делении вещественных чисел на ноль
- При использовании оператора остатка от деления на ноль

Пример

```
a = 10
b = 0
result = a / b
```

ICH IT CAREER HUB

Приоритет математических операций и скобок

Экспресс-опрос

? Какое арифметическое действие выполняется первым?

? В скобках или без?

Важно

В Python, как и в математике, операции выполняются в определённом порядке. Это называется приоритетом операций.

Приоритет операций

Приоритет	Операция	Символы
1	Скобки	()
2	Возведение в степень	**
3	Умножение, Деление, Целочисленное деление, Остаток от деления	*, /, //, %
4	Сложение и Вычитание	+, -

1. Скобки

Пояснения

Операции внутри скобок всегда выполняются первыми, независимо от их приоритета.

Пример

```
result = (2 + 3) * 4  # Скобки заставляют сложить
#2 и 3 перед умножением

print(result)       # Результат: 20
```

2. Возведение в степень (**)

Пояснение

После скобок, возведение в степень имеет наивысший приоритет.

Пример

```
result = 2 * 3 ** 2
# Это интерпретируется как 2 * (3 ** 2)
print(result)       # Результат: 18
```

3. Умножение (*), Деление (/), Целочисленное деление (//), Остаток от деления (%)

Пояснение

Эти операции имеют одинаковый приоритет и выполняются слева направо.

Пример

```
result = 10 / 2 * 3
# Интерпретируется как (10 / 2) * 3
print(result)         # Результат: 15.0
```

4. Сложение (+) и Вычитание (-)

Пояснение

Они имеют самый низкий приоритет и также выполняются слева направо

Пример

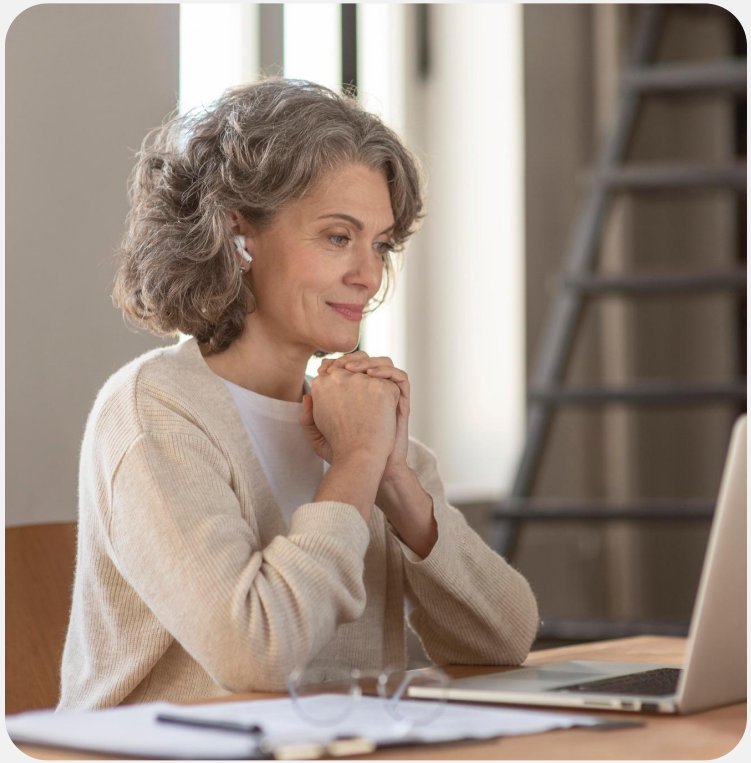
```
result = 10 - 2 + 5
# Интерпретируется как (10 - 2) + 5
print(result)         # Результат: 13
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

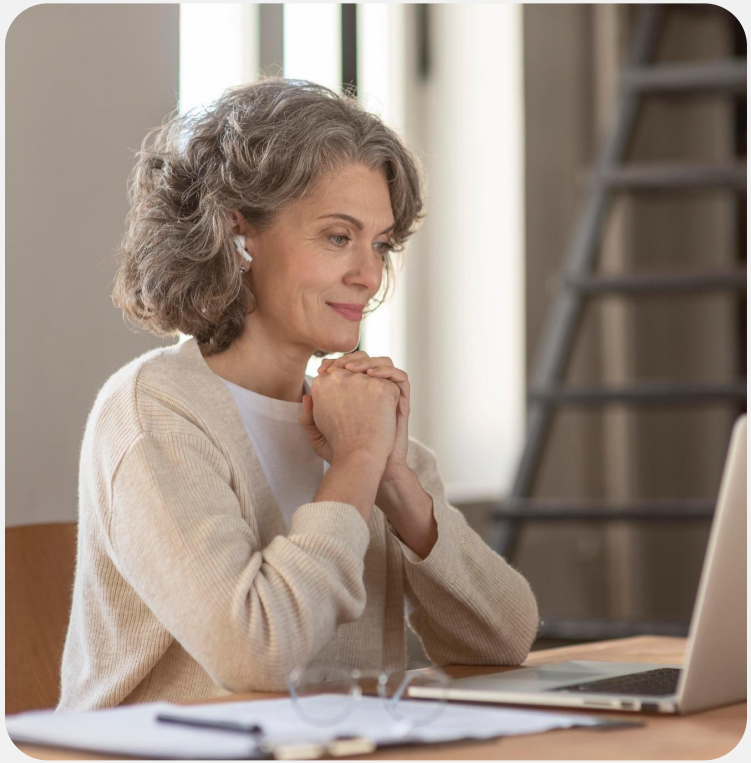


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
print(5 + 3 * 2)
```

a. 16
b. 11
c. 13
d. 10

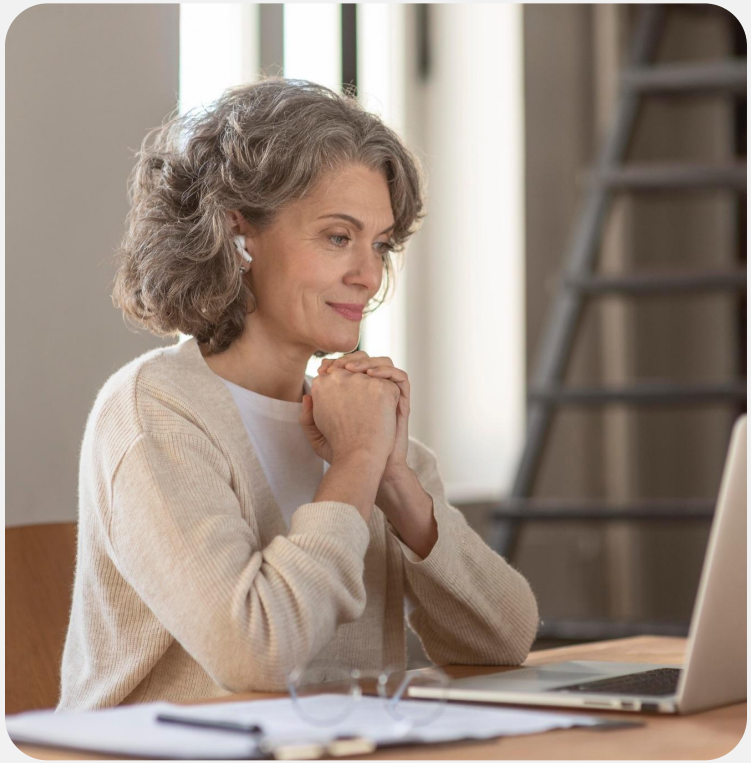


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
print(5 + 3 * 2)
```

a. 16
b. 11
c. 13
d. 10

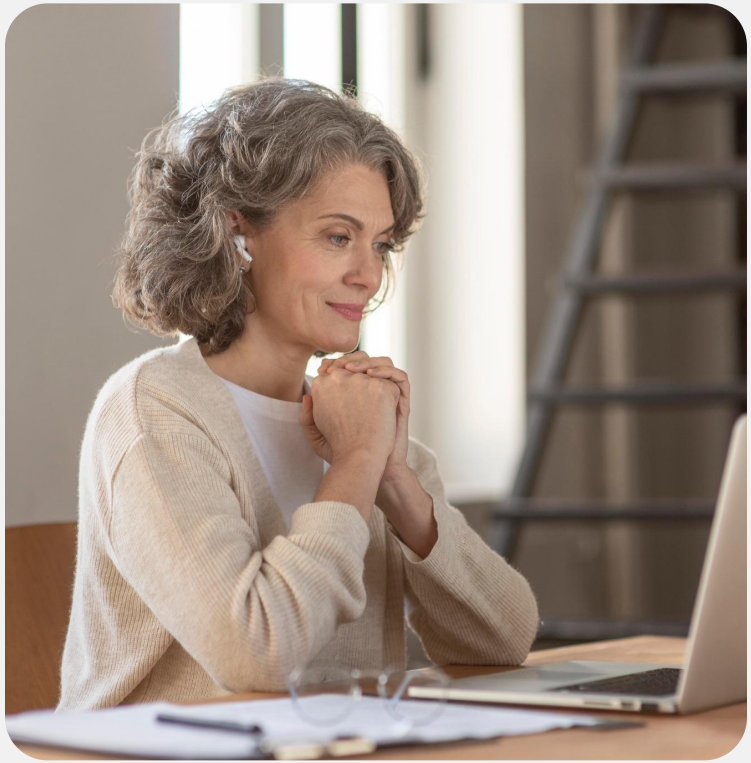


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
print(11 % 4)
```

a. 1
b. 2
c. 3
d. 4

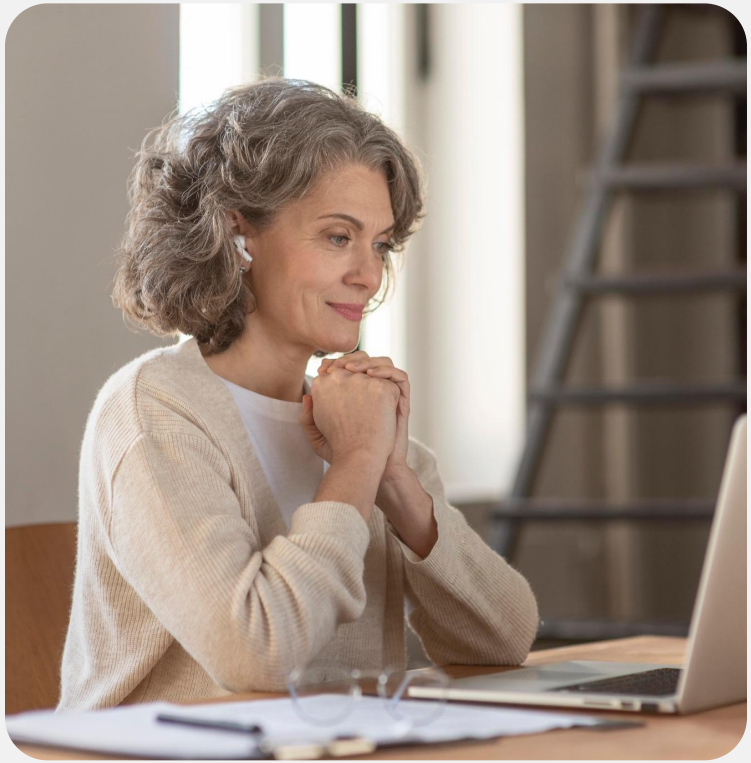


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
print(11 % 4)
```

a. 1
b. 2
c. 3
d. 4

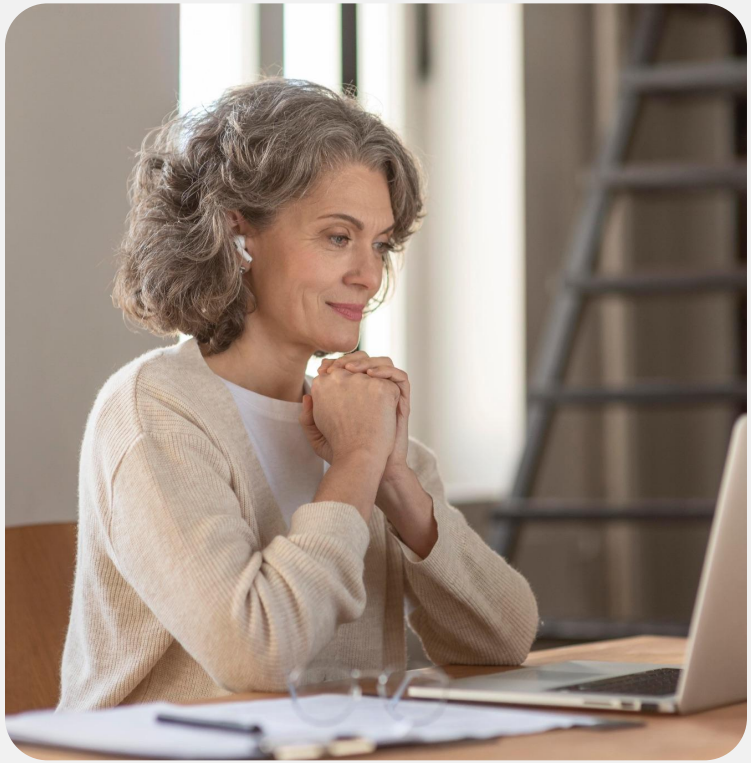


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
print(2 * 3 ** 2)
```

a. 12
b. 36
c. 18
d. 9

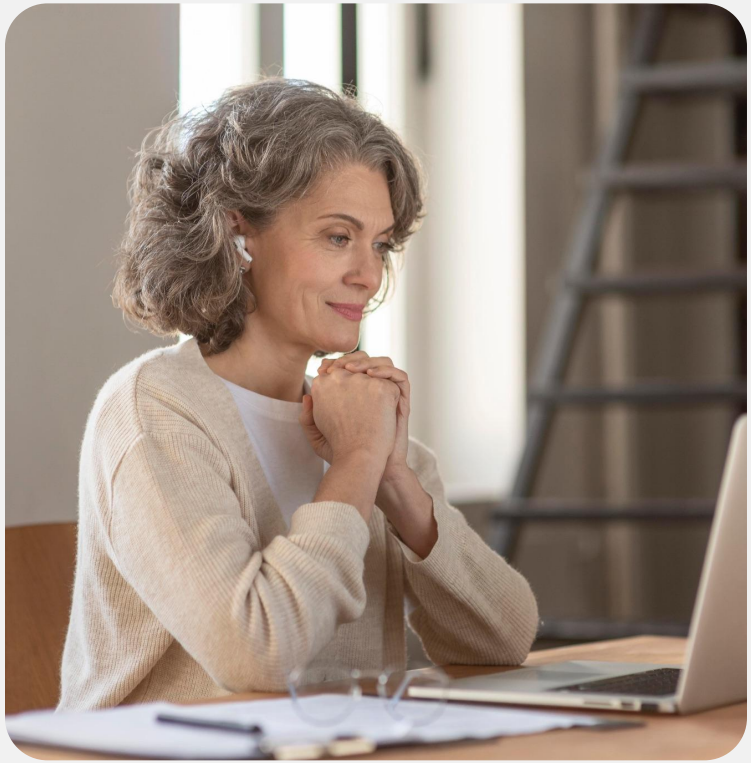


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
print(2 * 3 ** 2)
```

a. 12
b. 36
c. 18
d. 9

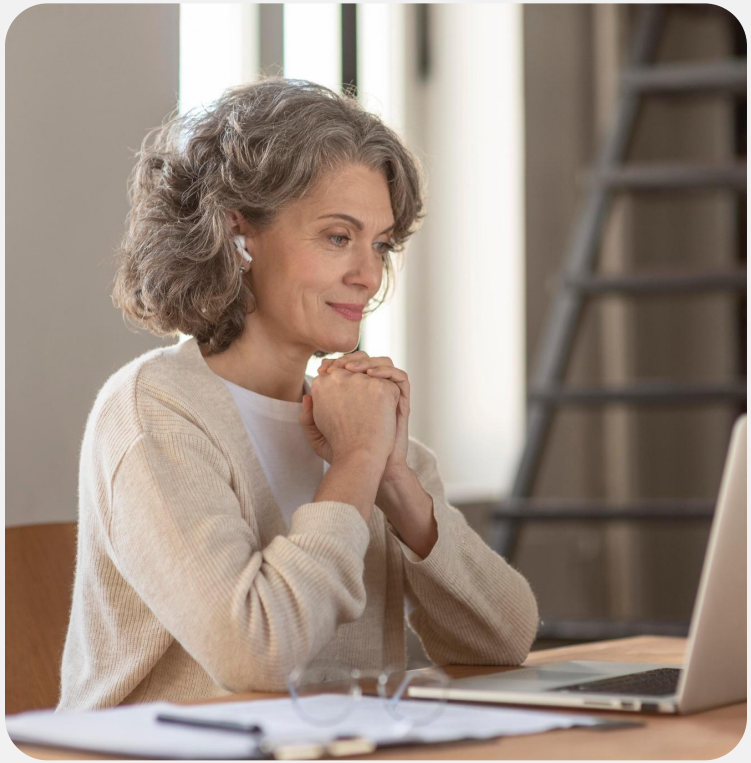


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
print(7 // 0)
```

a. 0
b. 7
c. Ошибка ZeroDivisionError
d. 1



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
print(7 // 0)
```

a. 0
b. 7
c. Ошибка ZeroDivisionError
d. 1

ICH IT CAREER HUB

Неизменяемые типы данных

Полезно знать

ICH
IT CAREER HUB

Неизменяемые типы данных

В Python существуют неизменяемые (`immutable`) и изменяемые (`mutable`) типы данных. Неизменяемые типы данных не могут быть изменены после создания объекта, а изменяемые могут.

Неизменяемые типы данных

Пояснение

То есть если у нас есть переменная num равная 5, то вы не можете изменить значение этой переменной, например с помощью арифметической операции:

Пример

```
num = 5
num + 2          # В переменной num останется
#значение 5
print(num)
```

Для того чтобы изменить значение в num нужно присвоить переменной новое значение.

```
num = 5
num = num + 2      # В переменную num будет
#присвоен результат вычисления 5 + 2
print(num)
```

IT CAREER HUB

Операторы приращения

Полезно знать

ICH
IT CAREER HUB

Операторы приращения

Это операторы, которые изменяют значение переменной на основе её текущего значения. Для этого используются операторы присваивания в сочетании с арифметическими операциями, такими как сложение, вычитание, умножение и другие.

Операторы приращения

IT CAREER HUB

Оператор	Описание	Пример	Эквивалент
+=	Прибавление и присваивание	a += 1	a = a + 1
-=	Вычитание и присваивание	a -= 1	a = a - 1
*=	Умножение и присваивание	a *= 2	a = a * 2
/=	Деление и присваивание	a /= 2	a = a / 2
//=	Целочисленное деление и присваивание	a //= 2	a = a // 2
%=	Остаток от деления и присваивание	a %= 3	a = a % 3
**=	Возведение в степень и присваивание	a **= 2	a = a ** 2

Примеры

Прибавление (+=)

```
a = 5
a += 1  # Эквивалентно: a = a + 1
print(a)  # Результат: 6
```

Умножение (*=)

```
a = 5
a *= 2  # Эквивалентно: a = a * 2
print(a)  # Результат: 10
```

IT CAREER HUB

Множественное присваивание

Полезно знать

Множественное присваивание

Это особенность Python, которая позволяет присваивать значения нескольким переменным одновременно в одной строке. Это делает код более компактным и читабельным, особенно когда требуется инициализировать сразу несколько переменных.

Присваивание нескольких значений

```
a, b, c = 1, 2, 3  # Инициализируем переменные a, b, c значениями 1, 2, 3 соответственно
print(a, b, c)     # Вывод: 1 2 3
```

Присваивание одинакового значения нескольким переменным

```
a = b = c = 0   # Инициализируем каждую из переменных a, b, c значением 0
print(a, b, c)   # Вывод: 0 0 0
```

Обмен значениями между переменными

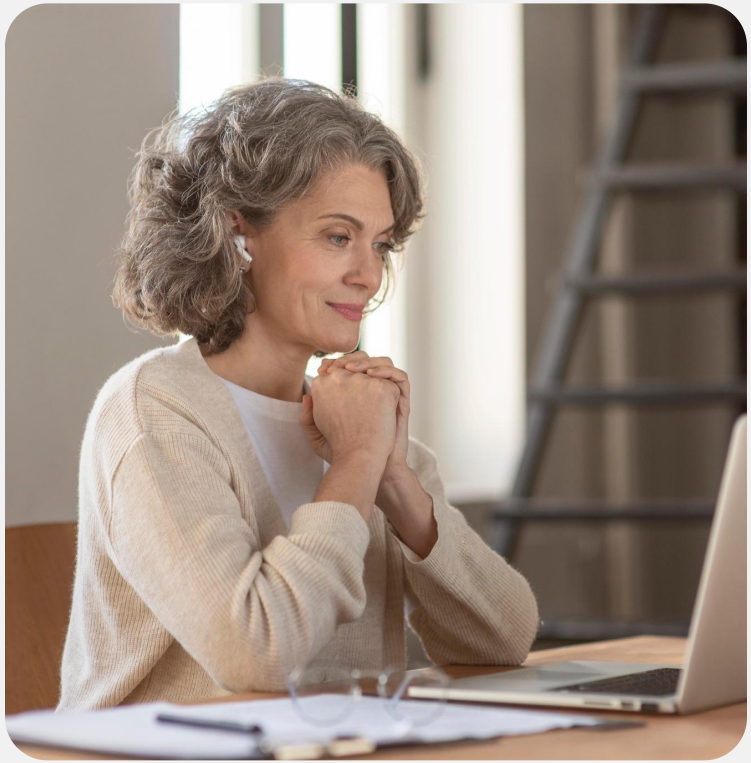
```
a, b = 5, 10
a, b = b, a  # Меняем местами значения a и b
print(a, b)  # Вывод: 10 5
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

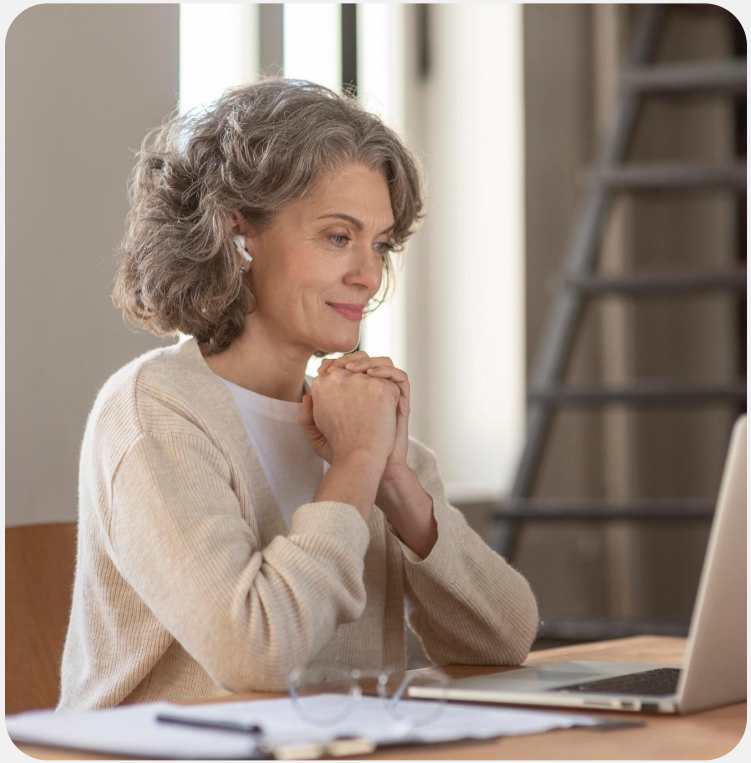


Выберите правильный вариант ответа

Какое значение будет у переменной num после выполнения следующего кода?

```
num = 5
num + 3
print(num)
```

a. 8
b. 5
c. Ошибка
d. 3

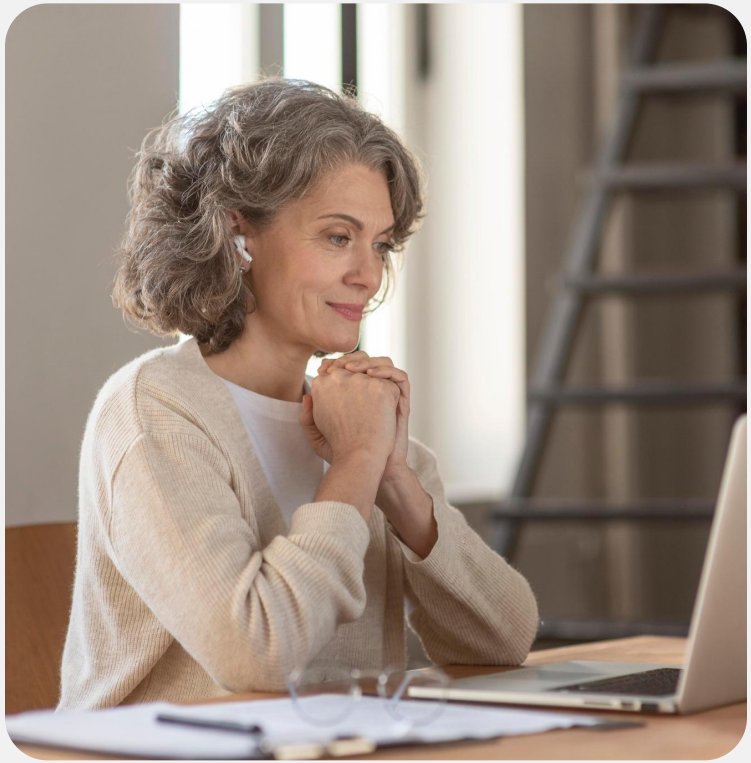


Выберите правильный вариант ответа

Какое значение будет у переменной num после выполнения следующего кода?

```
num = 5
num + 3
print(num)
```

a. 8
b. 5
c. Ошибка
d. 3

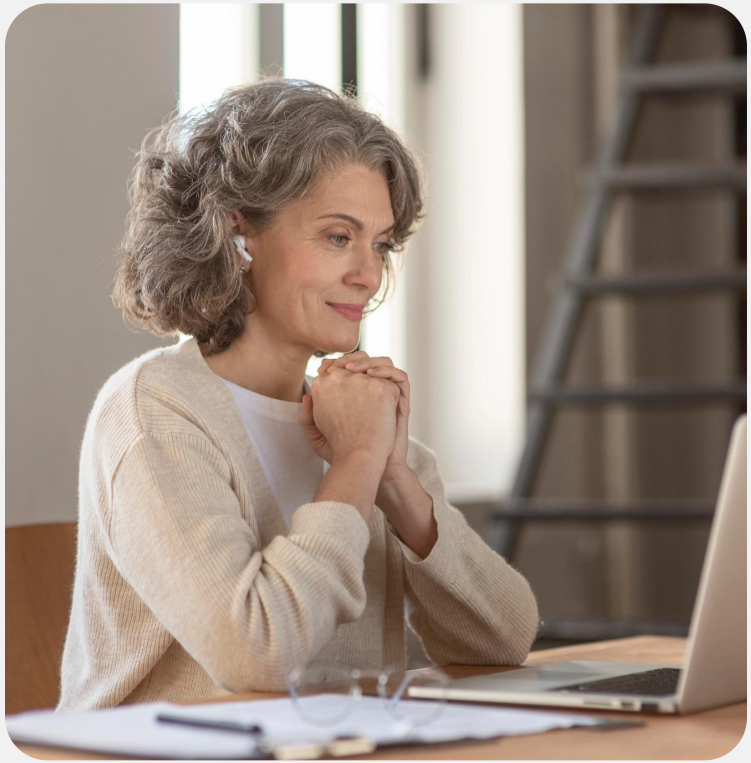


Выберите правильный вариант ответа

Какое значение будет у переменной num после выполнения следующего кода?

```
a, b = 1, 2
a, b = b, a
print(a, b)
```

a. 1, 2
b. 2, 1
c. 3, 3
d. Произойдет ошибка



Выберите правильный вариант ответа

Какое значение будет у переменной num после выполнения следующего кода?

```
a, b = 1, 2
a, b = b, a
print(a, b)
```

a. 1, 2
b. 2, 1
c. 3, 3
d. Произойдет ошибка

ICH IT CAREER HUB

Преобразование типов

Экспресс-опрос

? Какие типы данных бывают?

Полезно знать

ICH
IT CAREER HUB

Преобразование типов

Это процесс преобразования значения одного типа данных в значение другого типа. Это может быть необходимо, когда нужно выполнить определенные операции со значением или значениями разных типов.

Преобразование типов

Пример без преобразования

```
num1 = "10"
num2 = "5.0"

print(num1 / num2)
# Попытка выполнить деление строк
```

Пример с преобразованием

```
num1 = "10"
num2 = "5.0"

result = int(num1) / float(num2)
# Приведение строк к числовым типам и
# выполнение деления
print(result)
# Вывод: 2.0
```

Полезно знать

Неявное преобразование

Python может автоматически преобразовывать некоторые типы данных в других контекстах, например, при выполнении арифметических операций между числами разных типов.

Пример неявного преобразования

```
result = 5
print(type(result))   # <class 'int'>
result += 3.14        # Неявное преобразование int в float
print(type(result))   # <class 'float'>
```

Полезно знать

ICH
IT CAREER HUB

Явное преобразование

Явное преобразование происходит, когда разработчик вручную преобразует один тип данных в другой с помощью встроенных функций.

IT CAREER HUB

Функции для явного преобразования в примитивные типы данных

Функции для явного преобразования в примитивные типы данных

Функция	Описание	Пример использования
`int()`	Преобразует в целое число	`int("10")`
`float()`	Преобразует в вещественное число	`float("10.5")`
`str()`	Преобразует в строку	`str(10)`
`bool()`	Преобразует в логический тип	`bool(1)`

Пример: целое число в строку

```
a = 5
b = str(a)      # Преобразуем целое число в строку
print(b)        # Вывод: '5'
print(type(b))  # Вывод: <class 'str'>
```

Пример: строка в число

```
a = "10"
b = int(a)      # Преобразуем строку в целое число
print(b)        # Вывод: 10
print(type(b))  # Вывод: <class 'int'>
```

Пример: вещественное число в целое

```
a = 10.7
b = int(a)  # Преобразуем вещественное число в целое
print(b)  # Вывод: 10 (дробная часть отбрасывается)
```

Особенности и ошибки преобразования

Не все типы могут быть преобразованы друг в друга

```
a = "abc"
b = int(a)  # Ошибка: ValueError
```

При преобразовании вещественных чисел в целые дробная часть отбрасывается

```
a = 5.99
b = int(a)
print(b)  # Вывод: 5
```

Полезно знать

IT CAREER HUB

ValueError

Это встроенное исключение в Python, которое возникает, когда функция получает аргумент правильного типа, но с некорректным значением.

ValueError

Пояснение

Преобразование строки, которая не может быть интерпретирована как число, в целое или вещественное число, вызовет исключение ValueError.

Пример

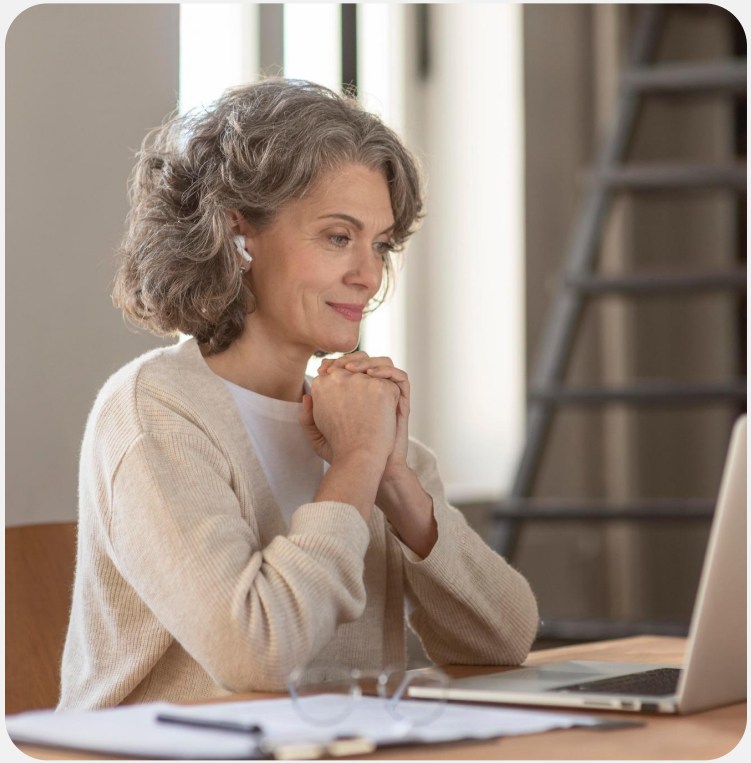
```
a = "abc"
b = float(a)  # Ошибка: ValueError: invalid literal for
#float() with base 10: 'abc'
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

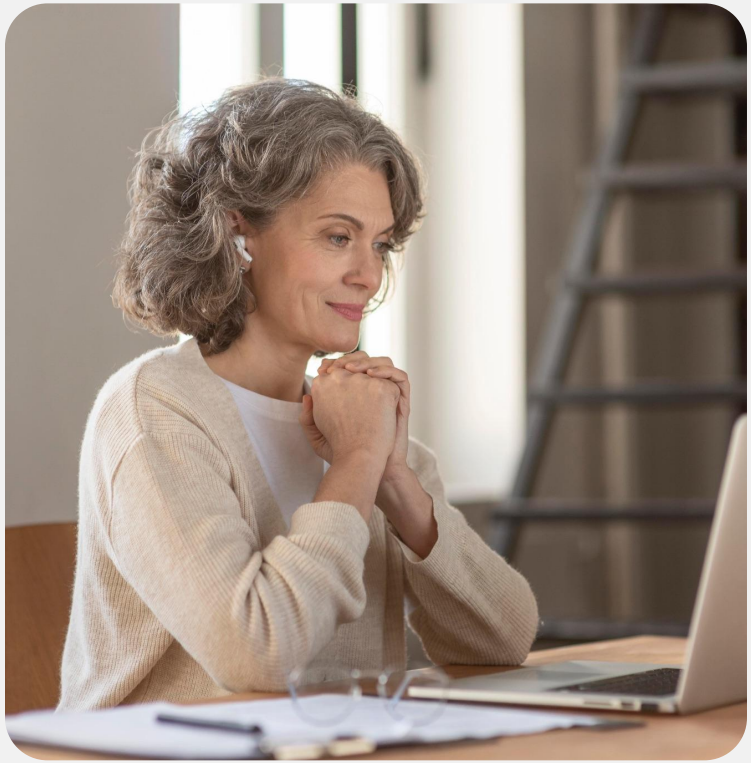


Выберите правильный вариант ответа

Что произойдёт при выполнении следующего кода?

```
result = 5
result += 3.14
print(type(result))
```

a. <class 'int'>
b. <class 'float'>
c. Ошибка
d. <class 'str'>



Выберите правильный вариант ответа

Что произойдёт при выполнении следующего кода?

```
result = 5
result += 3.14
print(type(result))
```

a. <class 'int'>
b. <class 'float'>
c. Ошибка
d. <class 'str'>

Соотнесите функцию для преобразования типа с её результатом:

1.	int("10.0")	a.	"10"
2.	float("3.14")	b.	3.14
3.	str(10)	c.	False
4.	bool(0)	d.	Ошибка

Соотнесите функцию для преобразования типа с её результатом:

1.	int("10.0")	d.	Ошибка
2.	float("3.14")	b.	3.14
3.	str(10)	a.	"10"
4.	bool(0)	c.	False

IT CAREER HUB

Практическая работа

1. Определите математические операции

Напишите программу, которая имеет две переменные a = 7, b = 3. Выполните математические операции, которые в результате дадут числа 4, 1, 2 (по одной операции для значения). Выведите на экран результат.

Пример вывода:

```
Результат операции <название>: 4

Результат операции <название>: 1

Результат операции <название>: 2
```

2. Расчёт сдачи

Напишите программу, которая принимает от пользователя стоимость товара, количество и сумму, которую он заплатил. Программа должна рассчитать и вывести сдачу.

Пример вывода:

```
Введите стоимость товара: 60
Введите количество товара: 3
Введите сумму оплаты: 500
Сдача: 320 рублей
```

IT CAREER HUB

Домашнее задание

Домашнее задание

1. **Сумма цифр числа**

Напишите программу, которая получит четырехзначное число от пользователя и выведет на экран сумму цифр этого числа.

Пример вывода:

```
Введите четырехзначное число: 1634

Сумма цифр числа: 14
```

Домашнее задание

2. Умножение чисел

Напишите программу, которая получает два числа от пользователя, умножает одно число на второе и выводит результат и оба числа через дефис. Не сохраняете результат умножения в переменной.

Пример вывода:

Введите первое число: 4
Введите второе число: 5
Результат: 20-4-5

Домашнее задание

3. Вычисление без операторов % и //

Напишите программу, которая получает два числа от пользователя и вычисляет:

- Остаток от деления
- Результат целочисленного деления

Решить без использования операторов `%` и `//`.

Пример ввода:

```
Введите первое число: 10

Введите второе число: 3
```

Пример вывода:

```
Остаток от деления: 1

Целочисленное деление: 3
```

IT CAREER HUB

ВОПРОСЫ

Заключение