

Введение в программирование на языке Python

Елена Владимировна Ширяева*



Содержание

1	Введение в Python. Первые программы	2
1.1	Простые (примитивные) типы данных	2
1.2	Оператор присваивания	3
1.3	Арифметические операции над данными целого типа	4
1.3.1	Целочисленные операции над отрицательными целыми числами	5
1.3.2	Работа с цифрами целого числа	6
1.4	Вывод данных на экран	6
1.5	Простая программа и ввод данных с клавиатуры	6
1.6	Ввод данных с клавиатуры, вывод данных, f-строки	7
1.7	Функции преобразования целых чисел в строки	7
	Тесты	10
	Компьютерный практикум	11
	Ответы	12

* Тестовый вариант (возможны опечатки). Нулевой курс мехмата ЮФУ. Проект «Отличный код», 2023–2026. Автор: Е. В. Ширяева, <https://t.me/EVShiryaeva>

1 Введение в Python. Первые программы

Чтобы программный код был правильным и эффективным у него должен быть крепкий фундамент. Одним из блоков этого фундамента являются «Базовые элементы языка программирования»:

- **Переменные** — именованные контейнеры для хранения значений разных типов. **Тип данных** определяет возможные значения данных и операции над ними.
- **Литералы** — фиксированные значения, которые напрямую подставлены в коде (числа, строки, булевы значения).

Например,

— числовой литерал 300 относится к целому типу данных и его можно умножить на другое число:

Изменение числового литерала

```
>>> num = 300 # Здесь 300 - это числовой литерал; num - имя переменной
>>> 2*num     # "число 2 умножить на число 300", результат 600
```

— строковый литерал '300' относится к строковому типу данных и его можно «тиражировать», умножая на другое число:

Изменение строкового литерала

```
>>> s = '300' # Здесь '300' - это строковый литерал; s - имя переменной
>>> 2*s       # "строку 300 дублировать дважды", результат '300300'
```

- **Операторы и выражения** — конструкции, которые позволяют выполнять операции над данными и получать результаты.

1.1 Простые (примитивные) типы данных

Обозначение	Тип	Примеры
int	целые числа	3, 40, -100, 0
float	вещественные числа	с фиксированной точкой: 7.0, 3.14, -15.8, 0.001
str	строки	'кот', "ok", 'Подготовка-2025', 'a_ = 25+67', 'True'
bool	логические значения	True, False

1.2 Оператор присваивания

Оператор присваивания — основной оператор любого языка программирования, его синтаксис:

имя_переменной = выражение

Результат выражения (справа) присваивается имени переменной (слева). Для имён переменных (функций и т.п.) используйте описательные имена, чтобы было понятно, что представляет собой объект. Например, `radius`, `name`.

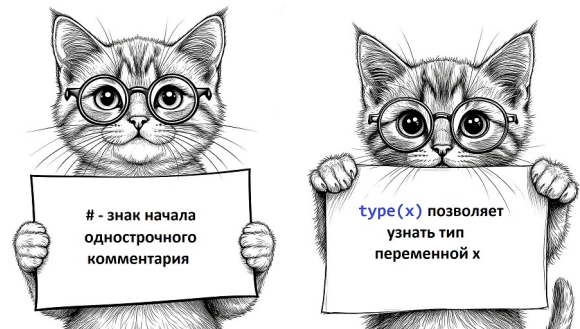
Пример оператора присваивания: `a = 5`, здесь `a` — имя переменной (**идентификатор**).

В момент присваивания автоматически определяется тип переменной (динамическая типизация, которая позволяет также изменять тип переменных во время выполнения программы).

Динамическая типизация

```
a = 5      # целое число
b = 7*a    # целое число
print(b, type(b)) # 35 <class 'int'>

a = '5'    # строка
b = 7*a    # строка
print(b, type(b)) # 5555555 <class 'str'>
```



Правильно написанный идентификатор — это последовательность латинских букв, цифр и знаков подчеркивания, начинающаяся с буквы.

Верные идентификаторы				Неверные идентификаторы		
x	time	t2025	count_	x and y	ok!	25year

Python — регистро-зависимый язык (case sensitivity). Разные имена: `f` и `F`.

Знак подчеркивания в конце имя_ рекомендуется писать, если хочется использовать имя, совпадающее с ключевым словом или стандартным идентификатором. Например, `if_`, `sum_`.

1.3 Арифметические операции над данными целого типа

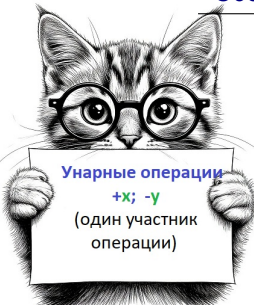
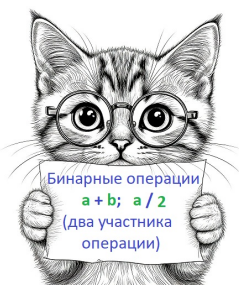
	Обозначение	Описание	Тип результата
	**	возведение в степень	int
	-, +	изменение знака (унарные)	int
	*	умножение	int
	/	деление	float
	+	сложение	int
	-	вычитание	int
			

Таблица 1. Арифметические операции

Унарные операции применяются к одному аргументу; бинарные операции требуют двух аргументов.

В Python три арифметические унарные операции: унарный минус и унарный плюс (см. плакат у котика слева 🐾), а также побитовое отрицание, обозначаемое символом $\sim$ (в рамках данного курса не рассматривается; для любознательных: $\sim x = -(x + 1)$). Остальные арифметические операции бинарные.

Унарные операции

```

a = 10
c = -2
print(+a)      # 10
print(-c)      # 2
print(-100)    # -100

```

Бинарные операции

```

a = 10
b = 3
c = -2
print(a + b)   # 13
print(a - b)   # 7
print(a * b)   # 30
print(a / b)   # 3.3333333333333335
print(4 / 2)   # 2.0
print(a**b)    # 1000

```

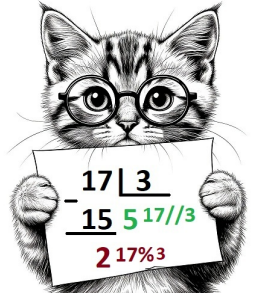
	Обозначение	Описание	Тип результата
	//	целочисленное деление; div	int
	%	остаток от целочисленного деления; mod	int

Таблица 2. Целочисленные операции

Целочисленные операции

```

print(5 // 2)   # 2
print(167 // 10) # 16

print(5 % 2)    # 1
print(4 % 2)    # 0

```

Функция divmod()

```

print(divmod(5, 2)) # (2, 1)

```

Функция `divmod()` позволяет сразу получить и результат целочисленного деления, и остаток от целочисленного деления. Возвращаемое значение — кортеж (неизменяемый список).

Обозначение	Описание
<code>+=</code>	сложение с присваиванием
<code>-=</code>	вычитание с присваиванием
<code>*=</code>	умножение с присваиванием
<code>/=</code>	деление с присваиванием
<code>%=</code>	остаток от деления с присваиванием
<code>**=</code>	возведение в степень с присваиванием
<code>//=</code>	целочисленное деление с присваиванием

Таблица 3. Комбинированные операторы присваивания

Во всех комбинированных операторах присваивания сначала к указанной переменной применяется заданная операция, а затем изменяется результат этой переменной. Например,

Комбинированные операторы присваивания

```
x = 5
x += 2    # аналогично x = x + 2    результат 7
x **= 2   # аналогично x = x**2     результат 49
```

1.3.1 Целочисленные операции над отрицательными целыми числами (★)

Знак результата деления `a // b` совпадает с привычными математическими понятиями, а само значение `a // b` целое и не превосходит значения `a / b`.

Целочисленное деление и отрицательные числа

```
print(7 // 2)      # 3
print(7 // (-2))   # -4
print(-7 // 2)     # -4
print(-7 // (-2))  # 3
```

Знак результата остатка от деления `a % b` всегда совпадает со знаком делителя, а само значение находится из формулы $a = b \cdot (a // b) + (a \% b)$.

Целочисленный остаток и отрицательные числа

```
print(7 % 2)       # 1
print(7 % (-2))    # -1
print(-7 % 2)      # 1
print(-7 % (-2))   # -1
```

Эта формула помогает понять, почему результаты примеров, приведённые выше, именно такие. Вычислим значение `7 % (-2)`:

$$7 \% (-2) = 7 - (-2) \cdot (7 // (-2)) = 7 - (-2) \cdot (-4) = 7 - 8 = -1.$$

Быстрый способ определения остатка при работе с отрицательными числами. Поделим нацело число `7 // (-2)`, результат «-4», так как значения не должно превышать $7/(-2)$. Умножим -4 на делитель -2. Результат 8, а делимое было 7, значит надо вычесть 1. Окончательно `7 % (-2)` равен -1.

Так как обычно нужно просто знать последнюю цифру целого числа, то при вычислении остатка надо избавиться от отрицательности числа с помощью функции `abs()`.

Функция «модуль числа»

```
print(abs(-17))    # 17
print(abs(-17) % 10) # 7
```

1.3.2 Работа с цифрами целого числа

Обозначим цифры натурального трёхзначного числа n , используя номера разрядов: n_2, n_1, n_0 .

Тогда число n можно также записать в виде $n_2 n_1 n_0$.

Общая формула для определения любой цифры целого неотрицательного числа имеет вид:

$$n_d = n // 10^d \% 10, \quad d = 0, 1, \dots$$

Здесь d — номер разряда.

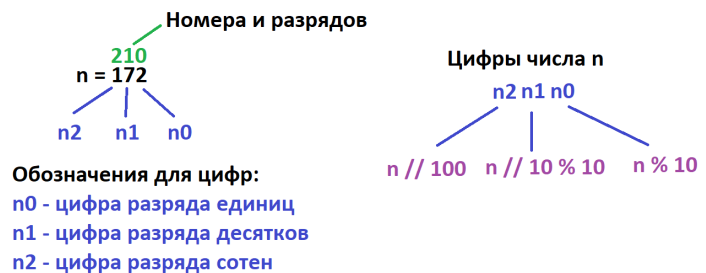


Рис. 1. Цифры натурального трёхзначного числа

Отделение цифр трёхзначного числа

```
n = 732          # для примера

n0 = n % 10      # в полном виде n0 = n // 1 % 10
n1 = n // 10 % 10
n2 = n // 100    # в полном виде n2 = n // 100 % 10

print(n0, n1, n2) # 2 3 7
```

1.4 Вывод данных на экран

Оператор печати print()

```
a = 100
b = a*0.5
print(a, b)      # оператор печати
print(type(a))
print(type(b))
```

Вывод результата

```
100 50.0
<class 'int'>
<class 'float'>
```

1.5 Простая программа и ввод данных с клавиатуры

Функция `input([Текст])` служит для ввода данных с клавиатуры. Эта функция всегда возвращает строку (`str`), поэтому при вводе числовых значений требуется использовать функции преобразования: `int()` для целых чисел и `float()` для вещественных. Аргумент `Текст` может отсутствовать, что является нежелательным при «общении с пользователем».

Ввод с клавиатуры

```
#===== Ввод строки
name = input('Введите свое имя: ')
print('Привет, ' + name + '!')

#===== Ввод целого числа
zarp = int(input('Зарплата: '))
print('Удвоим', 2*zarp)
```

Вывод результата

```
>>> Введите свое имя: Анна
      Привет, Анна!

>>> Зарплата: 20000
      Удвоим 40000
```

1.6 Ввод данных с клавиатуры, вывод данных, f-строки

Ввод с клавиатуры

```
name = input('Введите свое имя: ')
zarp = int(input(f'Какую зарплату желаешь, {name}? '))
print(f'{name}, ты достоин зарплаты {2*zarp}')
```

Вывод результата

```
Введите свое имя: Андрей
Какую зарплату желаешь, Андрей? 50000
Андрей, ты достоин зарплаты 100000
```

1.7 Функции преобразования целых чисел в строки

Функция преобразования числа в строку называется `str()`. Многие действия над числами легче выполнять, преобразовав его в строку, поэтому данная функция достаточно востребованная.

Python имеет **стандартные функции для перевода целого 10-ного числа в с.с. с основаниями 2, 8 и 16:**

Система счисления	Функция <i>n</i> — целое число в 10-ной с.с.	Пример использования	Вид результата (строка)
2-ая	<code>bin(n)</code>	<code>bin(31)</code>	<code>'0b11111'</code>
8-ая	<code>oct(n)</code>	<code>oct(31)</code>	<code>'0o37'</code>
16-ая	<code>hex(n)</code>	<code>hex(31)</code>	<code>'0x1f'</code>

Аргумент любой из этих функций — целое число; результат — строка с префиксом вида «0буква».

```
>>> bin(35) # здесь 35 - аргумент функции bin()
'0b100011' # это результат выполнения bin(35)
```

Рис. 2. Аргумент и результат функции

```
>>> bin(35)[2:] # срез [2:] уберёт префикс 0b
'100011'
```

Рис. 3. Удаление префикса

Функции Python `bin()`, `oct()`, `hex()` не выдают незначащие нули. И записать незначащие нули у целого числа тоже не удастся.

Значащие нули — это нули, являющиеся значимыми для числа. Например, $5 \neq 50 \neq 500$, так как все нули в этих числах значащие. А числа $5 = 05 = 005$, так как все нули в этих числах незначащие.

Преобразование числа в строку

```
a = 13
s = str(a)
s2 = bin(a)
s8 = oct(a)
s16 = hex(a)

print(f'{s = }')    # s = '13'
print(f'{s2 = }')   # s2 = '0b1101'
print(f'{s8 = }')   # s8 = '0o15'
print(f'{s16 = }')  # s16 = '0xd'
```

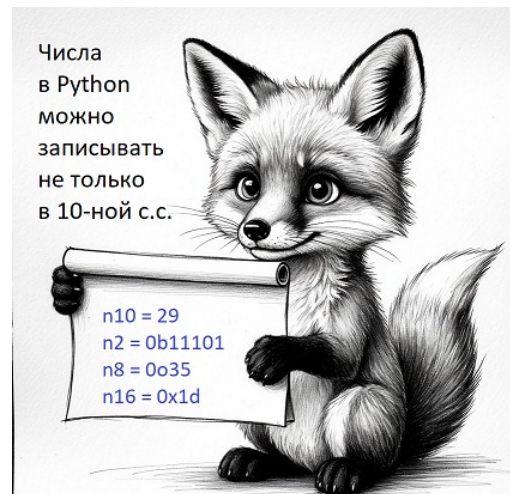
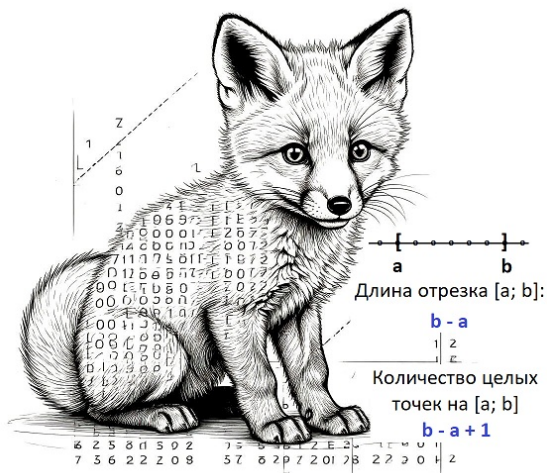


Рис. 4. Полезная информация (длина отрезка и не только)

Важные горячие клавиши IDLE 🔥

File	Edit	Format	Run	Options	Winc
New File	Ctrl+N				
Open...	Ctrl+O				
Open Module...	Alt+M				
Recent Files					
Module Browser	Alt+C				
Path Browser					
Save	Ctrl+S				
Save As...	Ctrl+Shift+S				
Save Copy As...	Alt+Shift+S				
Print Window	Ctrl+P				
Close	Alt+F4				
Exit	Ctrl+Q				

Edit	Format	Run	Options	Window	Help
Undo		Ctrl+Z			
Redo		Ctrl+Shift+Z			
Cut		Ctrl+X			
Copy		Ctrl+C			
Paste		Ctrl+V			
Select All		Ctrl+A			
Find...		Ctrl+F			
Find Again		Ctrl+G			
Find Selection		Ctrl+F3			
Find in Files...		Alt+F3			
Replace...		Ctrl+H			
Go to Line		Alt+G			
Show Completions		Ctrl+space			
Expand Word		Alt+/			
Show Call Tip		Ctrl+backslash			
Show Surrounding Pares		Ctrl+0			

Format	Run	Options	Window	Help
Format Paragraph			Alt+Q	
Indent Region			Ctrl+]	
Dedent Region			Ctrl+[	
Comment Out Region			Alt+3	
Uncomment Region			Alt+4	
Tabify Region			Alt+5	
Untabify Region			Alt+6	
Toggle Tabs			Alt+T	
New Indent Width			Alt+U	
Strip Trailing Whitespace				

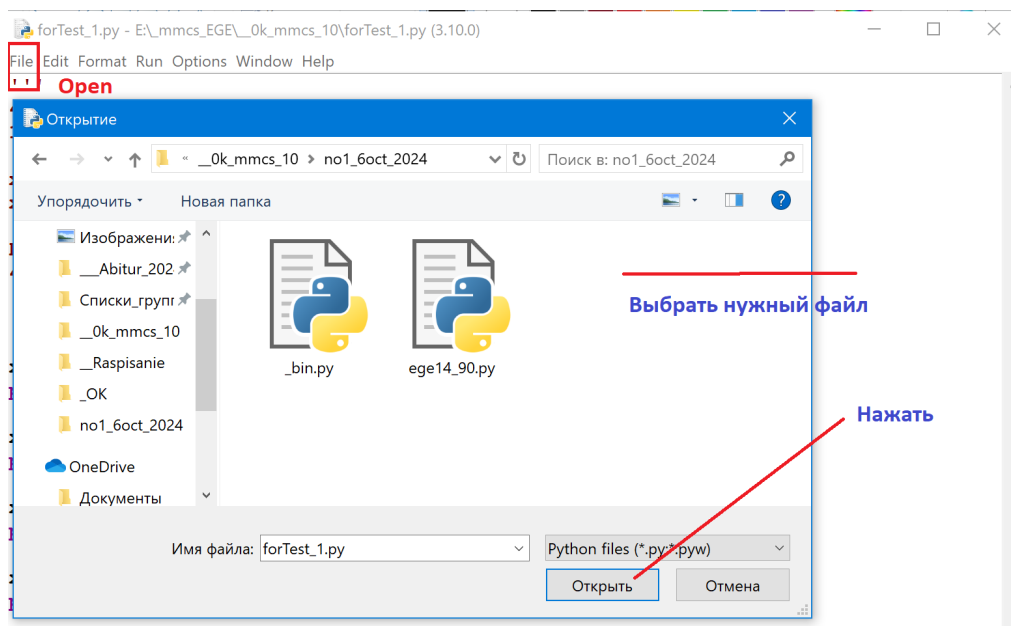
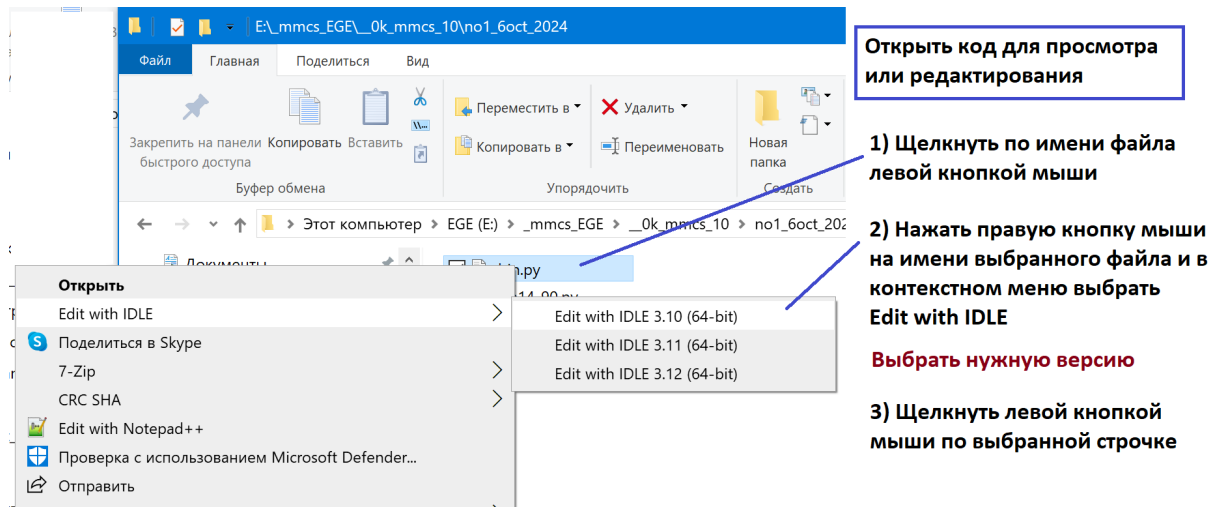
Run	Options	Window	Help
Run Module	F5		
Run... Customized	Shift+F5		
Check Module	Alt+X		
Python Shell			

Options	Window	Help
Configure IDLE		
Show Code Context		
Show Line Numbers		
Zoom Height	Alt+2	

Help
About IDLE
IDLE Help
Python Docs F1
Turtle Demo

Проект "Отличный Код"

Открытие файла с расширением ru



Тесты

T1. Даны типы данных: str, int, bool, float. Для каждого оператора присваивания запишите тип значения переменной x (см. пример ответа для задания 7):

№ п.п.	1)	2)	3)	4)	5)	6)
Оператор присваивания	$x = 10$	$x = 10.0$	$x = 10//2$	$x = '10'$	$x = 4**0.5$	$x = 10/2$
Ответ						

№ п.п.	7)	8)	9)	10)	11)
Оператор присваивания	$x = 5 > 3$	$x = 'True'$	$x = False$	$x = 15 \% 4$	$x = bin(5)$
Ответ	bool				

T2. Выберите верный результат выражения $7 // 2$:

а) 1 б) 3.5 в) 2 г) 3 д) Ошибка

T3. В результате выполнения выражения $156 \square 9 \square 10$ был получен ответ 7. Восстановите пропущенные знаки операций, стоящие на месте $\square$.

T4. Значение выражения $135 - 78*2 + 16 // 10$ станет равно 118, если в нём расставить скобки нужным образом. Восстановите скобки в данном выражении.

T5. Выберите верный ответ на вопрос «Какая функция в Python обозначает модуль числа x ?»:

а) $\text{mod}(x)$ б) $\text{module}(x)$ в) $\text{abs}(x)$ г) $|x|$.

T6. Не используя компьютер, запишите результаты работы следующих операторов:

$\text{divmod}(7, 3)$	
$\text{divmod}(-7, 3)$	
$\text{divmod}(7, -3)$	
$\text{divmod}(-7, -3)$	

T7. Дано число $n = 7896$. Установите соответствие между именами переменных $n89$, $n7$, $n96$, $n6$, $n789$ и выражениями для выделения из числа n чисел 89, 7, 96, 6 и 789, соответственно. Пример ответа см. для строки D:

Буква	Имя	Цифра	Выражение	Буква+Цифра	Оператор присваивания
A	$n89$	1	$n \% 10$		
B	$n7$	2	$n // 10$		
C	$n96$	3	$n \% 100$		
D	$n6$	4	$n // 1000$	D1	$n6 = n \% 10$
E	$n789$	5	$n // 10 \% 100$		

T8. Выберите неверные записи целых чисел в Python:

а) 067 б) 0 в) 0xA г) '300' д) -45 е) 0o93 ё) 0b101 ж) 0b12 з) 12 / 6

T9. Выберите верный результат вычисления выражения $3**2**2 - 2**2**3$:

а) -175 б) 17 в) 456533 г) 0

Компьютерный практикум

Темы: ввод целых чисел, печать, работа с числами, вычисления по формулам, системы счисления

Задание: написать программы для решения задач 1–10.

Ограничения на программы: нельзя использовать оператор if.

Примечания: Не забывайте, что целое число может быть отрицательным. Натуральные числа принадлежат интервалу: $[1, +\infty)$.

№ 1 (→)]. С клавиатуры вводятся два натуральных числа. Найти и вывести на экран их среднее арифметическое.

№ 2 (→)]. С клавиатуры вводятся два натуральных числа. Найти и вывести на экран их среднее геометрическое.

Помощь: квадратный корень из числа n можно вычислить с помощью выражения: $n^{**0.5}$.

№ 3 (→)]. С клавиатуры вводится натуральное число. Напечатать это число в 2-ой, 8-ой, 10-ой и 16-ой системах счисления. Каждый ответ выводите в новой строке.

№ 4 (→)]. С клавиатуры вводится двузначное целое число. Вывести на экран цифры этого числа: сначала цифру разряда единиц, затем цифру разряда десятков.

Вид диалога программы с пользователем
Введите двузначное число: -67
Цифры: 7 и 6

№ 5 (→)]. С клавиатуры вводится двузначное натуральное число. Найти и вывести на экран сумму цифр этого числа.

Вид диалога программы с пользователем
Введите двузначное число: 32
Сумма 3 + 2 = 5

№ 6 (→)]. Даны числа в 16-ной и 8-ой с.с.: $x = \text{DED}_{16}$ и $y = 76_8$. Вычислить значение выражения $x + y$.

№ 7 (→)]. С клавиатуры вводятся два целых числа a и b , являющиеся концами отрезка $[a; b]$. Найти длину этого отрезка. (Подсказка на рис. 4).

№ 8 (→)]. С клавиатуры вводятся два целых числа a и b , являющиеся концами отрезка $[a; b]$. Найти количество целых точек, расположенных на отрезке $[a; b]$. (Подсказка на рис. 4).

№ 9 (→)]. С клавиатуры вводится целое число — площадь квадрата. Найти длину стороны и длину диагонали квадрата.

№ 10 (→)]. С клавиатуры вводятся четыре целых числа: координаты двух точек на плоскости $A(x_1, y_1)$, $B(x_2, y_2)$. Вычислить расстояние между точками A и B .

Помощь: используйте формулу расстояния между двумя точками на плоскости:

$$|AB| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Ответы

Тест-Т1.

№ п.п.	1)	2)	3)	4)	5)	6)
Оператор присваивания	x = 10	x = 10.0	x = 10//2	x = '10'	x = 4**0.5	x = 10/2
Ответ	int	float	int	str	float	float

№ п.п.	7)	8)	9)	10)	11)
Оператор присваивания	x = 5 > 3	x = 'True'	x = False	x = 15 % 4	x = bin(5)
Ответ	bool	str	bool	int	str

Тест-Т2. г) **Тест-Т3.** 156 // 9 % 10 **Тест-Т4.** 135 - (78*2 + 16) // 10 **Тест-Т5.** в)

Тест-Т6.

divmod(7, 3)	(2, 1)
divmod(-7, 3)	(-3, 2)
divmod(7, -3)	(-3, -2)
divmod(-7, -3)	(2, -1)

Тест-Т7. Дано число $n = 7896$.

Буква	Имя	Цифра	Выражение	Буква+Цифра	Оператор присваивания
A	n89	1	n % 10	A5	n89 = n // 10 % 100
B	n7	2	n // 10	B4	n7 = n // 1000
C	n96	3	n % 100	C3	n96 = n % 100
D	n6	4	n // 1000	D1	n6 = n % 10
E	n789	5	n // 10 % 100	E2	n789 = n // 10

Тест-Т8. а г е ж з

а)	067	число не может начинаться с 0
г)	'300'	это строка
е)	0o93	в записи восьмеричного числа нет цифры 9
ж)	0b12	в записи двоичного числа нет цифры 2
з)	12 / 6	это выражение, а не число

Тест-Т9. а. **Комментарий к ответу:** операция ** ассоциативна справа, то есть вычисляется справа-налево. $3**2**2 - 2**2**3 = 3^4 - 2^8 = 81 - 256 = -175$.

Решение задачи № 1:

```
n = int(input('n >> '))
m = int(input('m >> '))

mean = (n + m) / 2

print(mean)
```

Примеры работы программы

```
n >> 6
m >> 8
7.0
===== Restart =====
n >> 10
m >> 3
6.5
```

Решение задачи № 2:

```
n = int(input('n >> '))
m = int(input('m >> '))

meanG = (n * m)**0.5

print(meanG)
```

Примеры работы программы

```
n >> 6
m >> 2
3.4641016151377544
===== Restart =====
n >> 12
m >> 3
6.0
```

Решение задачи № 3:

```
n = int(input('n >> '))

print(bin(n))
print(oct(n))
print(str(n))
print(hex(n))
```

Примеры работы программы

```
n >> 13
0b1101
0o15
13
0xd
===== Restart =====
n >> 100
0b1100100
0o144
100
0x64
```

Решение задачи № 4:

```
n = int(input('Введите двузначное число: '))
nn = abs(n)

n0 = nn % 10 # цифра разряда единиц
n1 = nn // 10 # цифра разряда десятков

print(f'Цифры: {n0} и {n1}')
# или print('Цифры:', n0, 'и', n1)
```

Примеры работы программы

```
Введите двузначное число: -67
Цифры: 7 и 6
===== Restart =====
Введите двузначное число: 73
Цифры: 3 и 7
```

Решение задачи № 5:

```
n = int(input('Введите двузначное число: '))

n0 = n % 10 # цифра разряда единиц
n1 = n // 10 # цифра разряда десятков
sm = n0 + n1

print(f'Сумма {n1} + {n0} = {sm}')
```

Пример работы программы

```
Введите двузначное число: 32
Сумма 3 + 2 = 5
```

Решение задачи № 6:

```
x = 0xded
y = 0o76

print(x + y)
```

Результат работы программы
3627

Решение задачи № 7:

```
a = int(input('a >> '))
b = int(input('b >> '))

dist = abs(b - a)

print(dist)
```

Примеры работы программы

```
a >> 5
b >> 12
7
===== Restart =====
a >> -10
b >> -3
7
```

Решение задачи № 8:

```
a = int(input('a >> '))
b = int(input('b >> '))

k = b - a + 1

print(k)
```

Пример работы программы

```
a >> 3
b >> 6
4
```

Решение задачи № 9:

```
s = int(input('s >> '))

a = s**0.5          # сторона квадрата
d1 = a * 2**0.5     # длина диагонали: поиск через сторону квадрата
d2 = (2*s)**0.5     # длина диагонали: поиск через площадь квадрата

print(a, d1, d2)
```

Примеры работы программы

```
s >> 2
1.4142135623730951 2.0000000000000004 2.0
===== Restart =====
s >> 9
3.0 4.242640687119286 4.242640687119285
```

Решение задачи № 10:

```
x1 = int(input('x1 >> '))
y1 = int(input('y1 >> '))

x2 = int(input('x2 >> '))
y2 = int(input('y2 >> '))

dist = ((x2 - x1)**2 + (y2 - y1)**2)**0.5

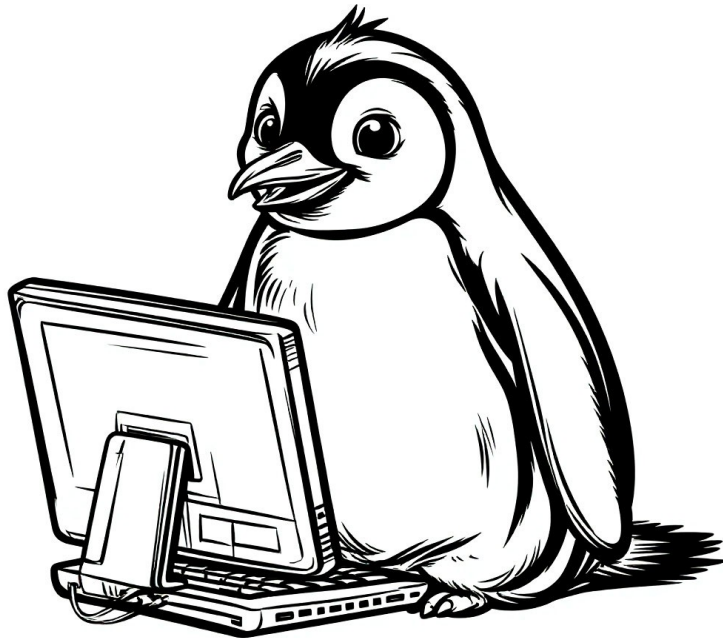
print(dist)
```

Примеры работы программы

```
# Тест 2: A(0, 1), B(0, 7) ответ 6
x1 >> 0
y1 >> 1
x2 >> 0
y2 >> 7
6.0
===== Restart =====

# Тест 2: A(0, 0), B(1, 1) ответ 2**0.5
x1 >> 0
y1 >> 0
x2 >> 1
y2 >> 1
1.4142135623730951
===== Restart =====

# Просто пример
x1 >> 1
y1 >> 3
x2 >> 7
y2 >> 2
6.082762530298219
```



Все чувствуют себя пингвинами при первом знакомстве с программированием. Главное — не останавливаться и продолжать кодить, даже если у вас лапки (или крылышки).