

Содержание

Вывод информации <code>print()</code>	2
Задания по выводу $\text{print}()$	4
Переменные	4
Задания по переменным a, b, c	5
Типы данных	6
Задания по типам данных a, b, c	8



Вывод информации print()

Для вывода информации на экран используем:

`print()`



Выводимую информацию указываем в скобках

Если хотим вывести текст, то заключаем его в кавычки

(`""` или `' '`)

```
print("Мой текст")
```

или

```
print('Мой текст')
```

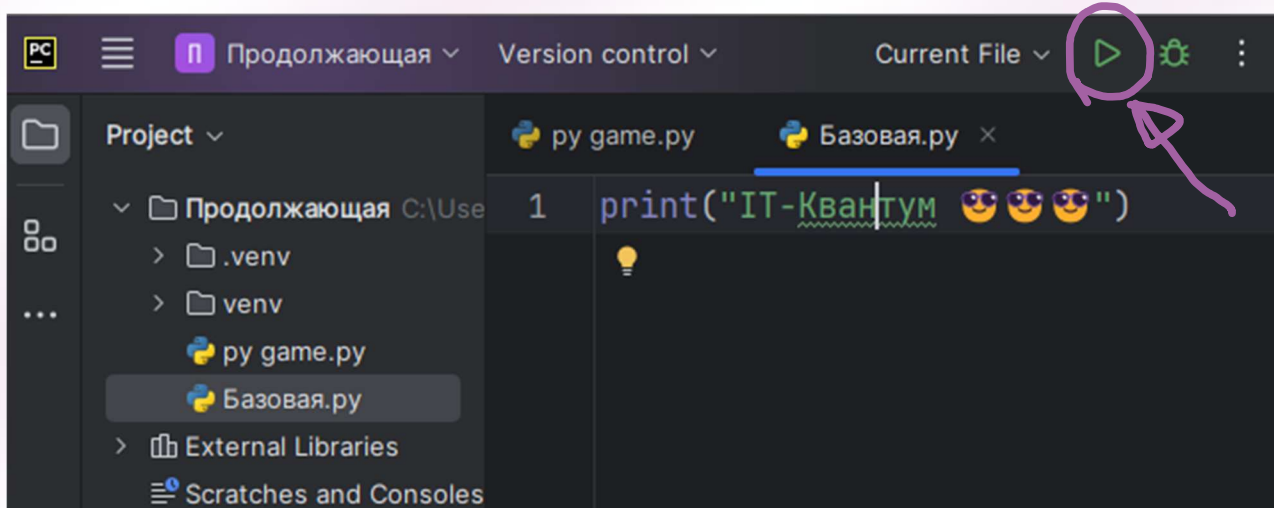
Числа выводим без кавычек

```
print(123)
```

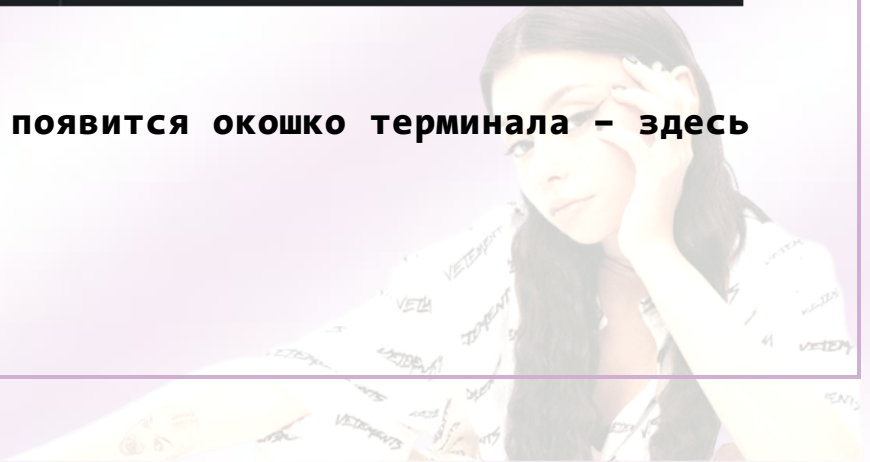
или

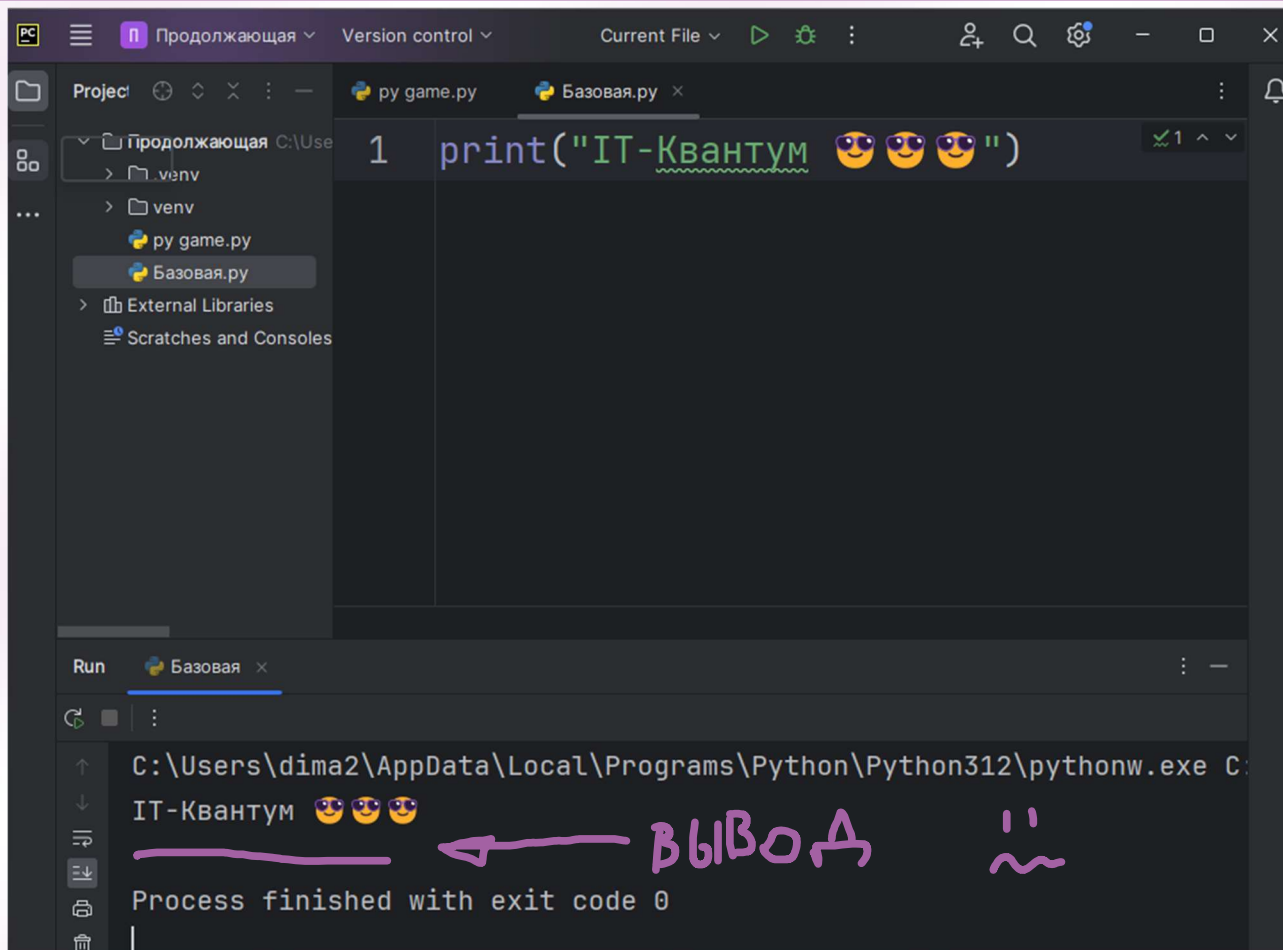
```
print(3.22)
```

Для выполнения кода нажать на стрелочку



После этого внизу экрана появится окошко терминала – здесь выводится наш текст





The screenshot shows an IDE window with a project named "Продолжающая". The file explorer on the left shows a folder "venv" and two files: "py game.py" and "Базовая.py". The main editor displays the code in "Базовая.py":

```
1 print("IT-Квантум 🤓🤓🤓")
```

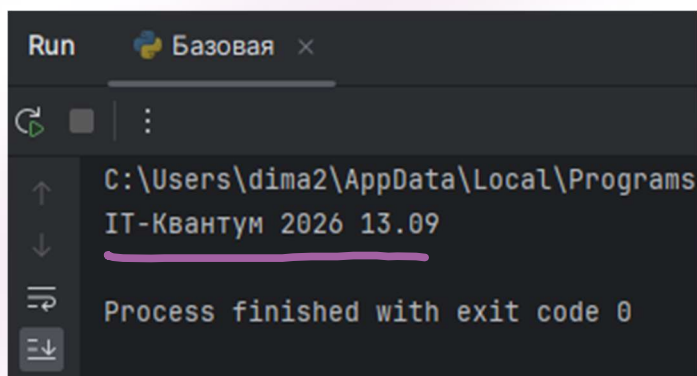
The output console at the bottom shows the command prompt path and the output:

```
C:\Users\dima2\AppData\Local\Programs\Python\Python312\pythonw.exe C:  
IT-Квантум 🤓🤓🤓  
Process finished with exit code 0
```

Handwritten purple annotations include a line under "IT-Квантум", an arrow pointing to the word "ВЫВОД" (Output), and a squiggly line.

Можно вывести все сразу одной строкой – если написать через запятую

```
print("IT-Квантум", 2026, 13.09)
```



The screenshot shows the "Run" console for the "Базовая" file. The output is:

```
C:\Users\dima2\AppData\Local\Programs  
IT-Квантум 2026 13.09  
Process finished with exit code 0
```

The output "IT-Квантум 2026 13.09" is underlined with a purple line.



1. Выведи название своего любимого фильма/сериала/аниме с помощью `print()`
2. Выведи любое целое и любое дробное число
3. Выведи одной командой свое имя, возраст и любимый цвет одной строкой

Переменные

Чтобы где-то хранить данные существуют переменные

Их можно обзывать почти как угодно (хранить в них тоже можно что угодно)

Объявляются переменные так:

ИМЯ ПЕРЕМЕННОЙ = ЗНАЧЕНИЕ ПЕРЕМЕННОЙ

Примеры объявления переменных:

```
money = 150
деньги = 100
мой_счет = 500
```

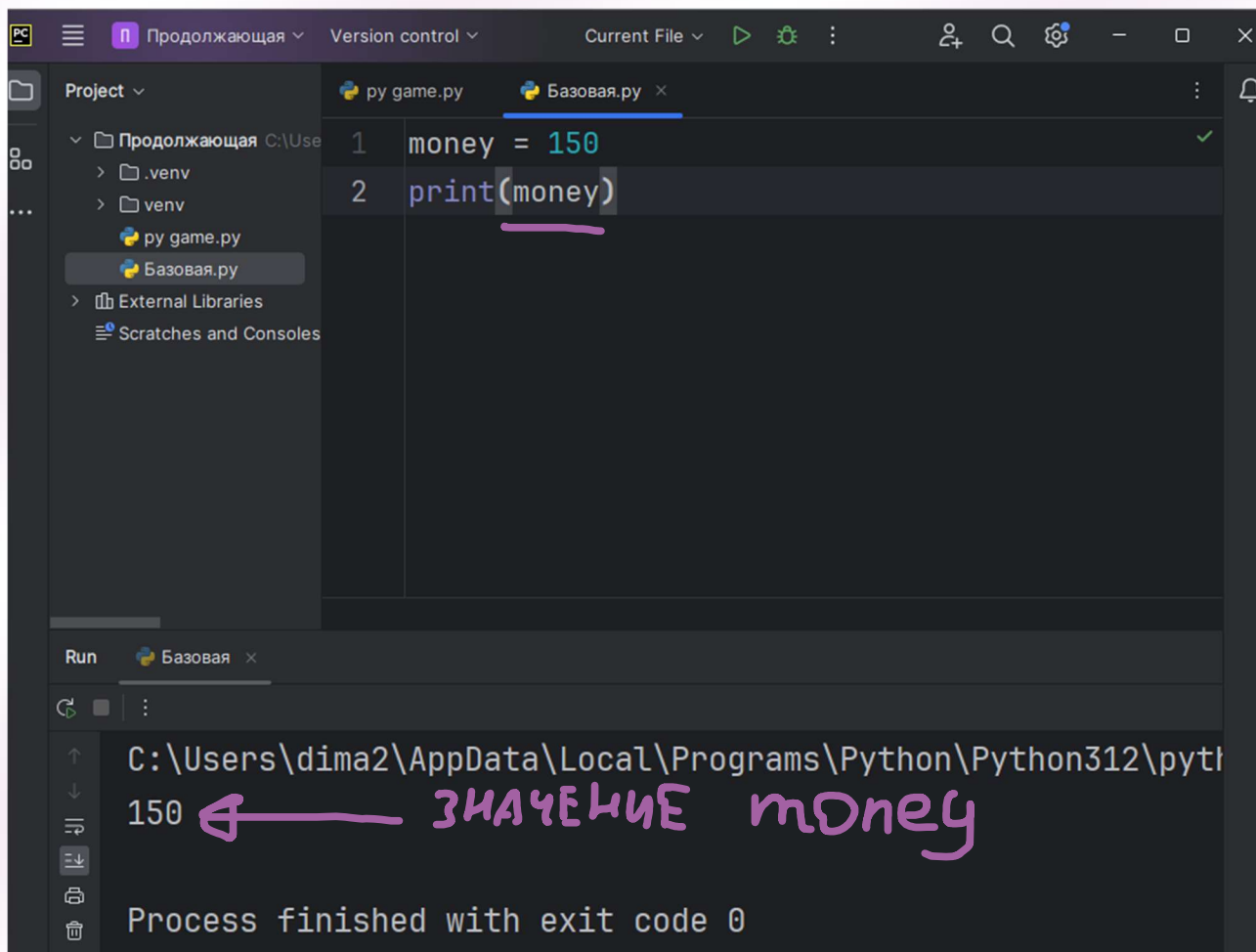
Чтобы никого не запутать (а главное не запутаться самому!) обзываем переменные понятными именами

Имена переменных не могут:

- Начинаться с цифры
- Содержать пробелы – вместо них используем `_`
- Совпадать с зарезервированными именами `python`

Переменные выводятся по имени:

`print(ИМЯ_ПЕРЕМЕННОЙ)`



The screenshot shows a Python IDE with a project named 'Продолжающая'. The file explorer on the left shows a folder structure with '.venv', 'venv', 'py game.py', and 'Базовая.py'. The editor window shows the following code:

```
1 money = 150
2 print(money)
```

The variable 'money' is highlighted with a pink box, and the value '150' is underlined with a pink line. The Run console at the bottom shows the output:

```
C:\Users\dima2\AppData\Local\Programs\Python\Python312\python.exe
150
Process finished with exit code 0
```

A pink arrow points from the text 'ЗНАЧЕНИЕ money' to the output '150'.

Задания по переменным



1. Создай переменные name (имя), surname (фамилия), age (возраст) и задай им значения
2. Выведи эти переменные на экран с помощью print()
3. Попробуй обозвать переменную с цифры (1name) – что будет? Объясни, почему так.
4. Попробуй обозвать переменную с пробелом (my name) – что будет? Объясни, почему так.

Типы данных

Что можно хранить в переменных:

Тип данных	Описание	Пример
int	Целые числа	157 0 -45
float	Дробные числа (дробная часть пишется через точку)	1.45 -0.5 13.33333
str	Строки (текст или буквы)	"Мой текст" 'Еще один текст' "A" ""
bool	Логическое значение: • True (да) • False (нет) Обязательно с заглавной!	True False
list	Список значений (куча переменных в одной)	[1, 2, 3] ["яблоко", "апельсин", "вишня"]
dict	Словарь (ключ = значение)	{ "name": "Dimasik", "kvantum": "IT", "age": 23 }
None	Буквально ничего	None
И еще много-много всего...		

Примеры непосредственно в коде:

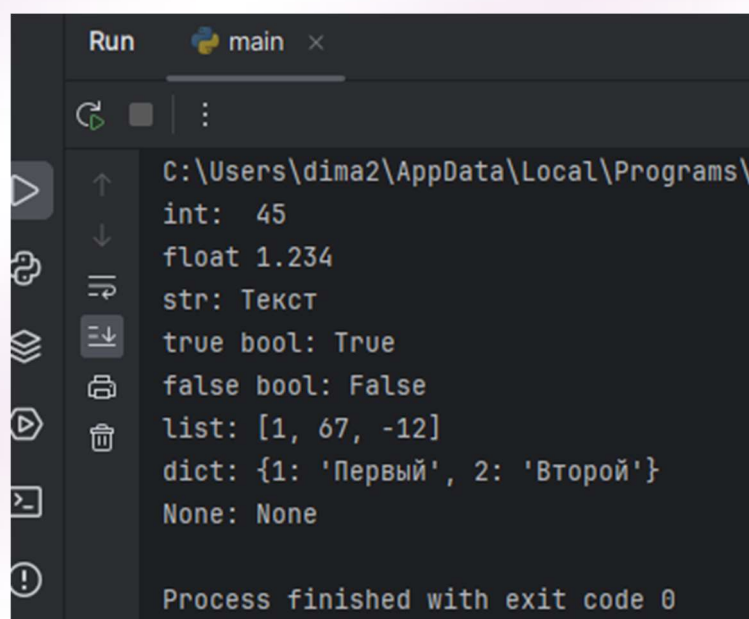
```
main.py x
1 int_number = 45      ← int
2
3 float_number = 1.234 ← float
4
5 my_text = "Текст"    ← str
6
7 true_bool = True     ← bool
8 false_bool = False
9
10 spisok = [1, 67, -12] ← list
11
12 slovar = {1: "Первый", 2: "Второй"} ← dict
13
14 NOTHING = None      ← None
```

Выведем переменные и посмотрим, что получится:

Команды:

```
print("int: ", int_number)
print("float", float_number)
print("str:" my_text)
print("true bool:", true_bool)
print("false bool:", false_bool)
print("list:", spisok)
print("dict:", slovar)
print("None:", NOTHING)]
```

Результат вывода:



The screenshot shows a terminal window with the following output:

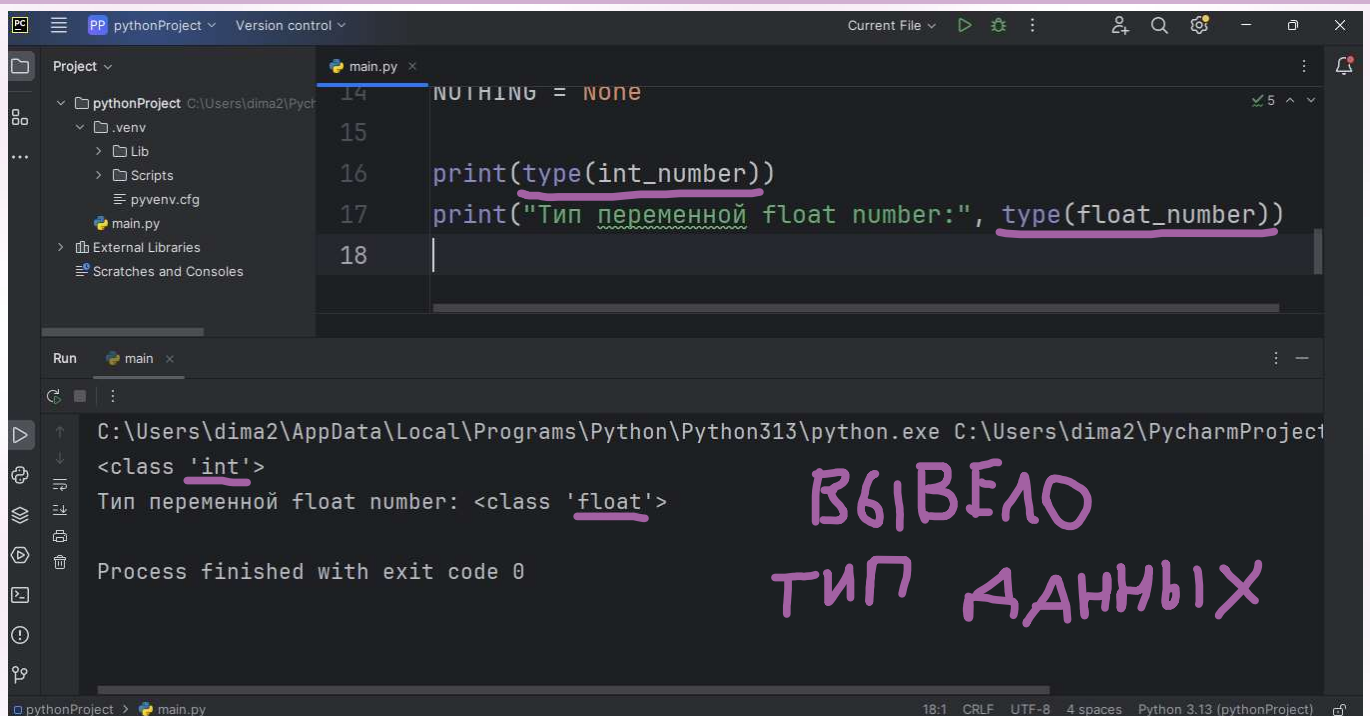
```
C:\Users\dima2\AppData\Local\Programs\
int: 45
float 1.234
str: Текст
true bool: True
false bool: False
list: [1, 67, -12]
dict: {1: 'Первый', 2: 'Второй'}
None: None

Process finished with exit code 0
```

Чтобы узнать, какого типа переменная, можно использовать команду **type:**

type(ИМЯ_ПЕРЕМЕННОЙ)





The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, running, and debugging. The left sidebar shows the project structure with folders like .venv, Lib, Scripts, and pyvenv.cfg, and files like main.py. The main editor window displays a Python script in main.py with the following code:

```
14 NOTHING = None
15
16 print(type(int_number))
17 print("Тип переменной float number:", type(float_number))
18
```

The Run window at the bottom shows the execution output:

```
C:\Users\dima2\AppData\Local\Programs\Python\Python313\python.exe C:\Users\dima2\PycharmProject
<class 'int'>
Тип переменной float number: <class 'float'>
Process finished with exit code 0
```

Handwritten purple text is overlaid on the Run window output, reading "ВЫВЕЛО" and "ТИП ДАННЫХ".

Задания по типам данных

≧^π^π^≦

1. Создай переменные разных типов – целое число, дробное, строку, логическое значение.
2. Выведи их с помощью print().
3. С помощью type() выведи тип каждой переменной

