Функция

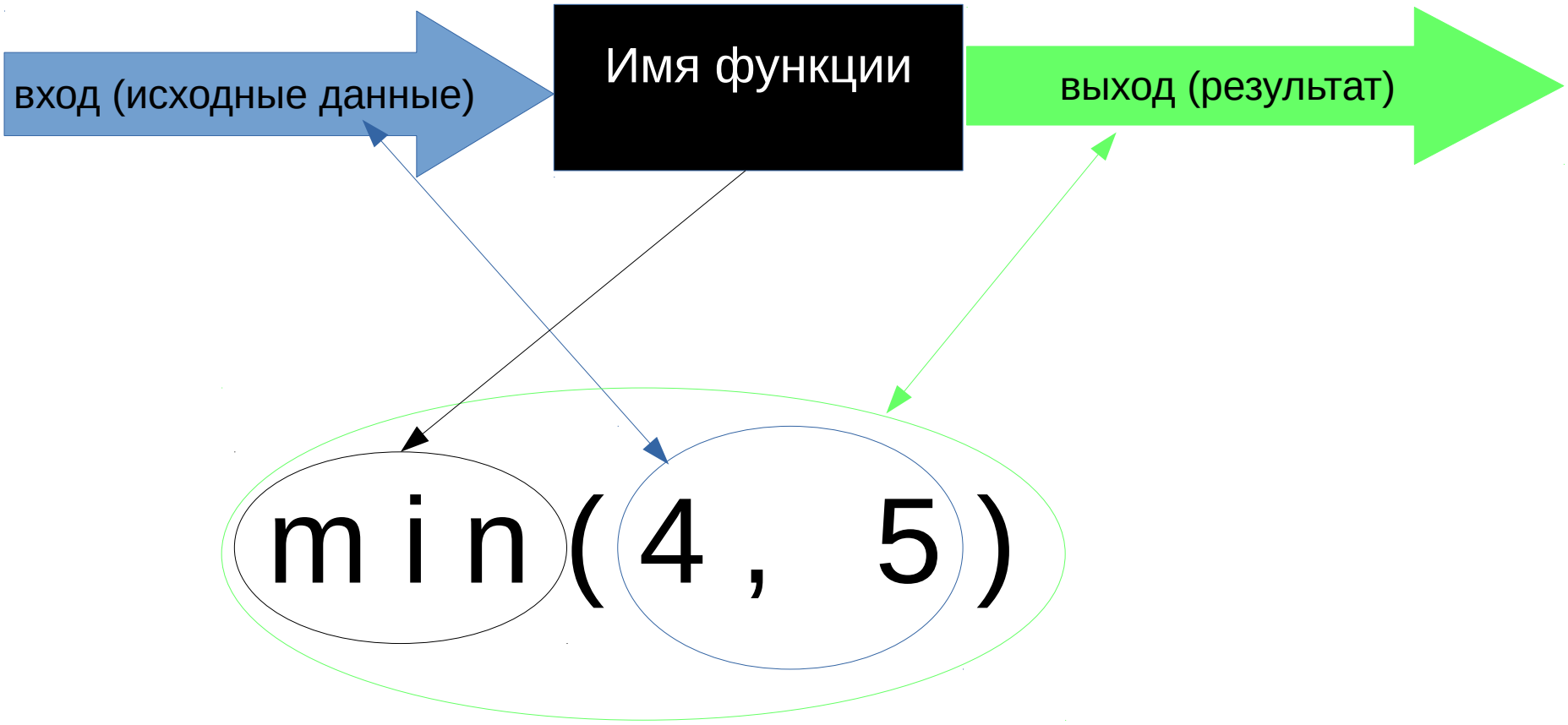
(от лат. functio - исполнение - осуществление)

1. деятельность,обязанность, работа.

2. В математике: зависимость одной переменной величины от другой: $y = f(x)$

или определенное действие над переменной

Функция - как черный ящик



Использование результата функции

1. Результат функции можно сохранить в переменной:

```
res = min(4, 5)
```

2. Результат функции можно передать другой функции:

```
print(min(4, 5))
```

Пример использования функции

```
x = -2
y = abs(x)
```

Функция abs находит модуль числа (абсолютное значение)

Аргумент (параметр) функции. Если несколько аргументов, то они перечисляются через запятую

Имя функции

При x = -2 этой строке программы будет равнозначна:

```
y = 2
```

Функция в программировании

- это фрагмент кода, который выполняет какие-либо действия
- функция может использоваться много раз в одной или разных программах
- функция **может** получать набор исходных данных для их обработки или уточнения того, что она должна делать
- функция **может** возвращать значение - результат своей работы, (а может просто работать и не возвращать)
- В программе функцию узнаем по скобке "(" сразу после имени.

Ввод значений

- Для ввода значений с клавиатуры нужно использовать функцию **input()**. Эта функция принимает все введенные символы с клавиатуры до нажатия кнопки Enter, а возвращает **строку**.
- Введенное значение можно сохранить в переменной, чтобы использовать его далее в программе.

```
>>> st = input()
Привет
>>> st
'Привет'
```

Важно: даже если в функцию не передаются параметры, нужно использовать круглые скобки для запуска функции

Задание

Напишите программу вводящую два числа и выводящую их сумму.

Типы данных

- Данные могут быть разных типов. От типа зависит что можно делать с данными и какой получится результат.
- Один или несколько символов (букв, цифр, знаков), заключенные в кавычки имеют строковый тип - **str**. Примеры: "abcd" , '7' , 'Привет, мир!'
- Целые числа, например 4, 2, 20 имеют тип **int**.
- Числа с дробной частью (5.0, 1.6), даже если она равна нулю, имеют тип **float**. В качестве разделителя целой и дробной части всегда используется точка.

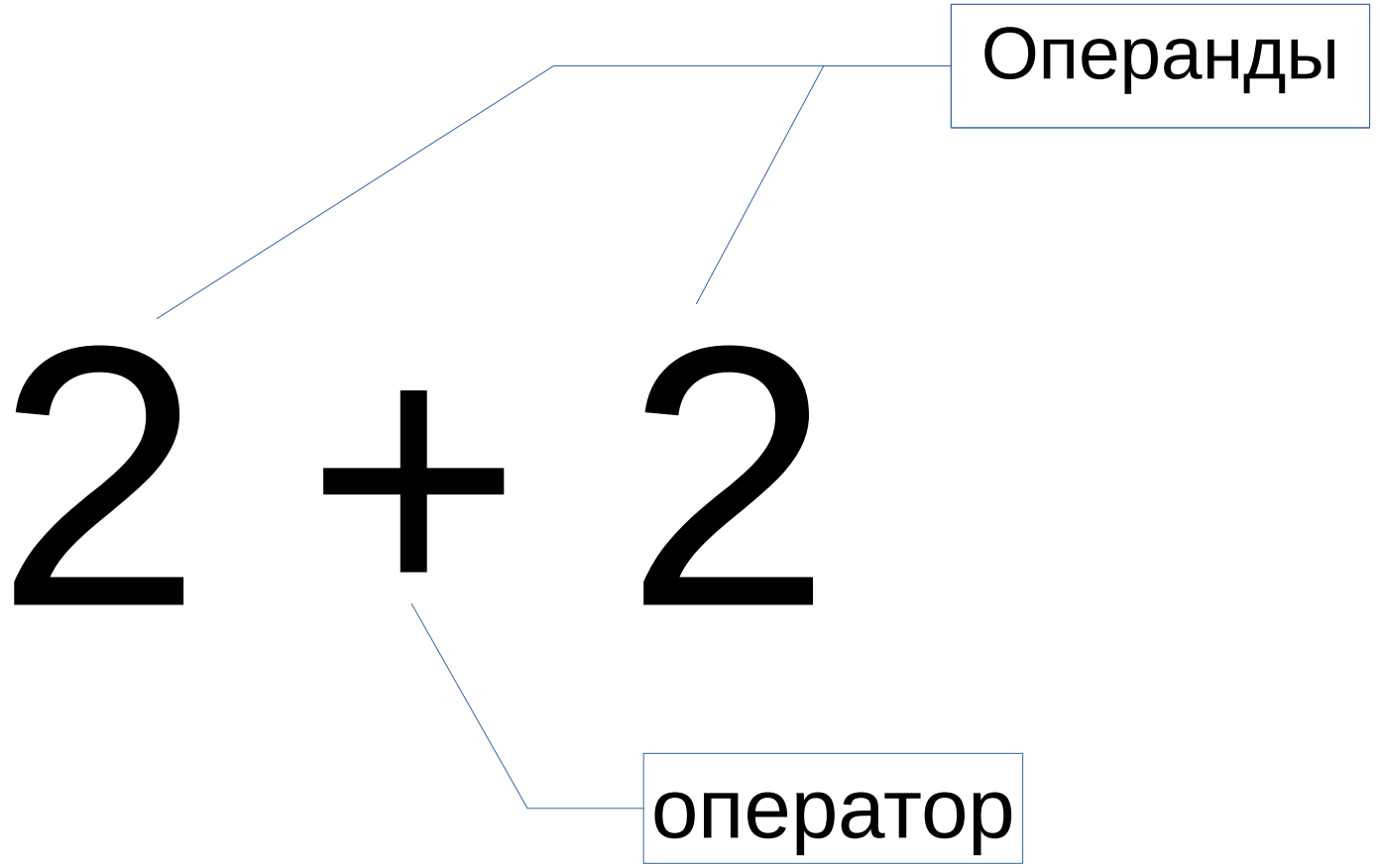
- Если необходимо ввести не строковое значение, то нужно использовать преобразование строки к нужному типу.
- i = int(input('Введите целое значение')) # Вводим целое число
- fl = float(input('Введите число с точкой')) #Вводим вещественное число.
- Если при вводе будут введены не ожидаемые символы, например буквы, то выполнение программы прервется.
- Обратите внимание, что в данных примерах результат работы одной функции (input) передается другой функции (int или float)

Преобразование типов

Для преобразования данных из одного типа в другой нужно использовать функцию с именем, совпадающим с именем нужного нам типа. Примеры:

```
>>> print(int('7') + int('8'))
15
>>>print(str(7) + str(8))
78
>>>print(int(3.1415926))
3
>>>print(7 + str(8))
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Операция



Выражения

Выражение, это то, что можно вычислить и получить результат какого-то типа.

Выражение может состоять из:

- значений
- переменных
- функций
- операторов
- скобок для задания порядка выполнения операций

Использование выражений

Результат, полученный при вычислении выражения может быть:

- сохранен в переменной для дальнейшего использования:

  res = ((t1*v1) + (t2*v2)) / (t1 + t2)

- передан как параметр в функцию:

  print(((t1*v1) + (t2*v2)) / (t1 + t2))

Операторы (кратко)

- +, -,* , /, ( )
- 3**2 - возведение в степень — три в квадрате.
- / - деление всегда возвращает число типа float.
- // - целочисленное деление (неполное частное). При таком делении получается целое число без остатка от деления. Например, 7//3 даст 2.
- % - остаток от деления (взятие по модулю). 7%3 даст 1.
- Если в операции одно число вещественное, а другое целое, то результат будет приведен к float.

```
>>> 3*2.0
6.0
```

Решите эти примеры в тетради

9 // 4 =

8 // 4 =

25 // 10 =

3 // 10 =

4 // 8 =

90 // 60 =

9 % 4 =

8 % 4 =

25 % 10 =

3 % 10 =

4 % 8 =

90 % 60 =

Ввод нескольких значений

Если мы хотим ввести несколько чисел из одной строки, разделенные пробелом, например:

15 20

то используем магическое заклинание №1:

```
a, b = map(int, input().split())
```

Если числа каждое с новой строки, то для каждой свой input(). **Правило: "Сколько входных строк, столько input'ов в программе"**

```
15
20
```

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

Комментарии

- Комментарии позволяют разбираться с назначением программы не только другим людям, но и автору программы.
- Однострочный комментарий начинается с #
- Многострочный комментарий начинается с тройных кавычек ''' и ими же заканчивается '''
- Комментирование можно использовать для временного отключения каких либо блоков программы во время ее отладки.

Пример комментариев

```
# Это первый комментарий,
spam = 1  # а это второй,
          # ... теперь третий!
text = "# А это не комментарий, потому что он внутри кавычек."
```