

Лекция 2

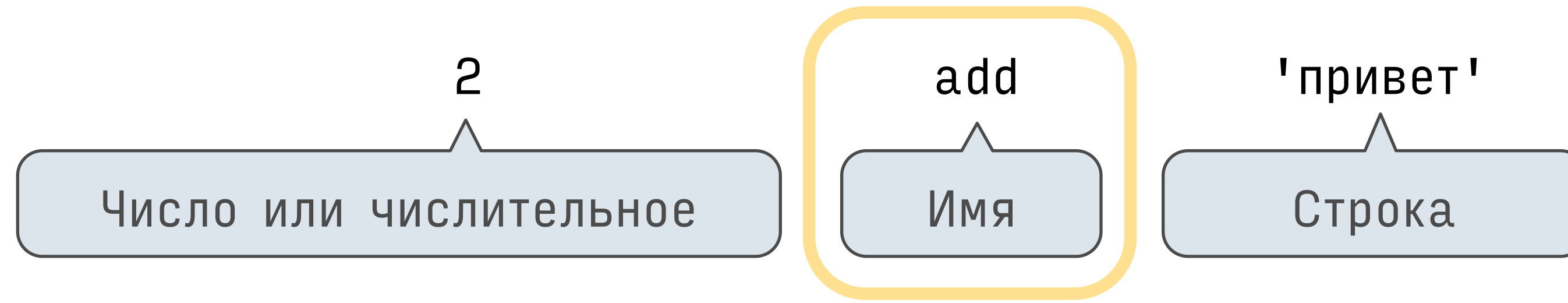
в которой мы поговорим про имена

Имена, присвоения и пользовательские функции

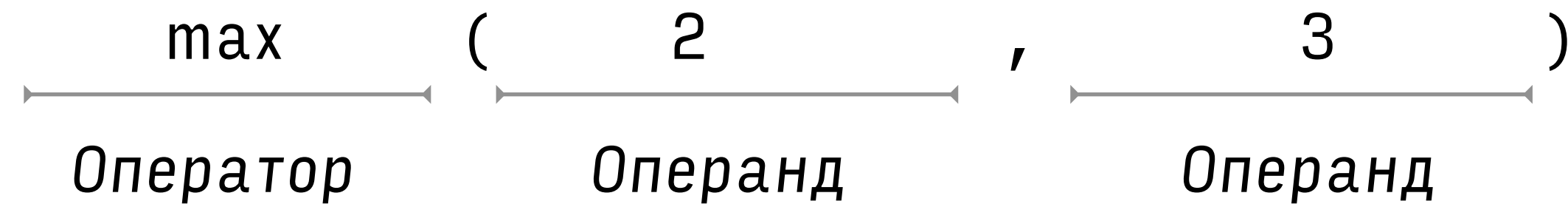
(Пример)

Виды выражений

Простые выражения:



Вызывающие выражения:



Операнд также может
быть вызывающим
выражением

`max(min(pow(3, 5), -4), min(1, -2))`

Вопрос 1

Каков результат последнего выражения в такой последовательности?

```
>>> f = min
```

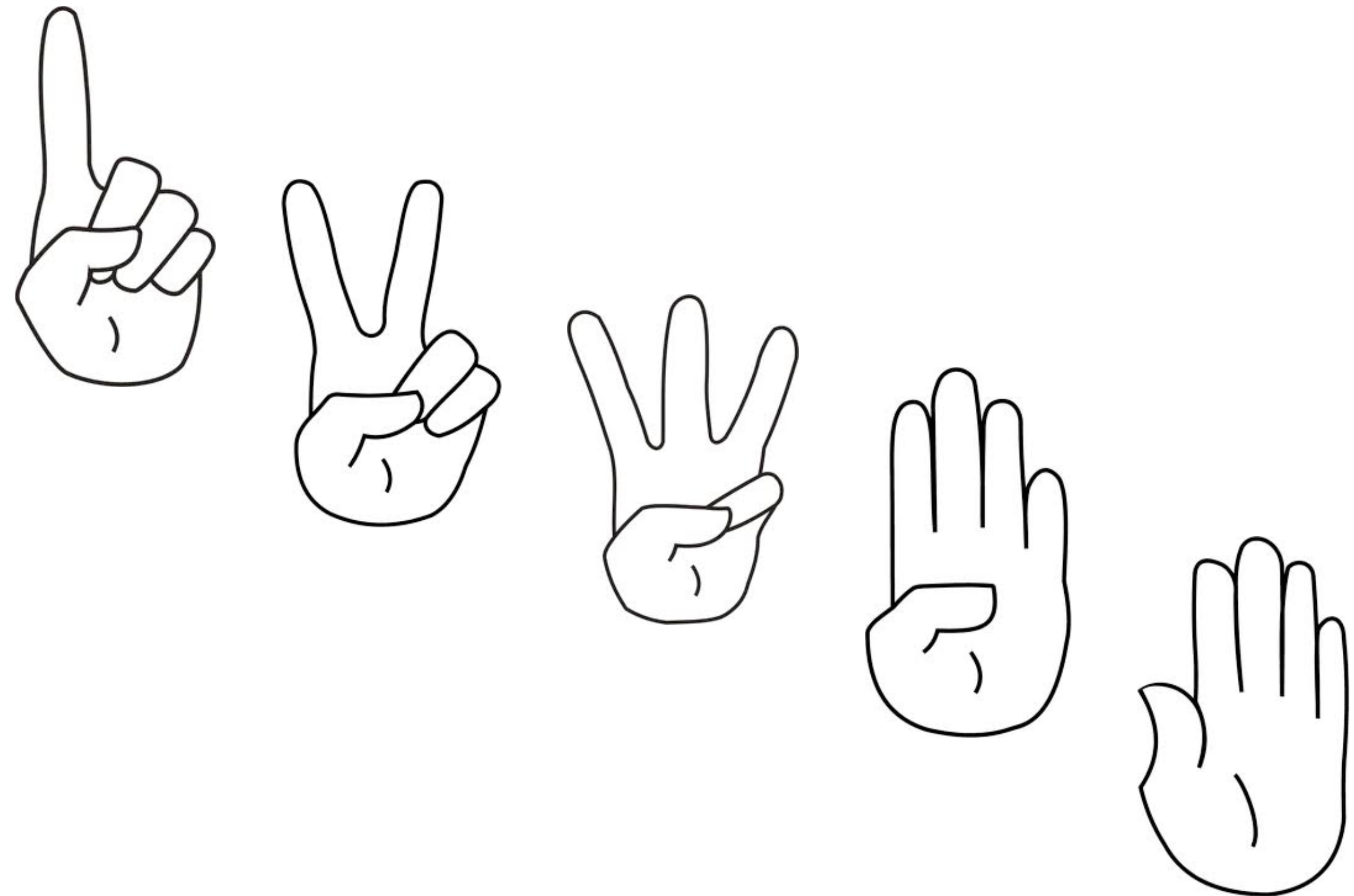
```
>>> f = max
```

```
>>> g, h = min, max
```

```
>>> max = g
```

```
>>> max(f(2, g(h(1, 5), 3)), 4)
```

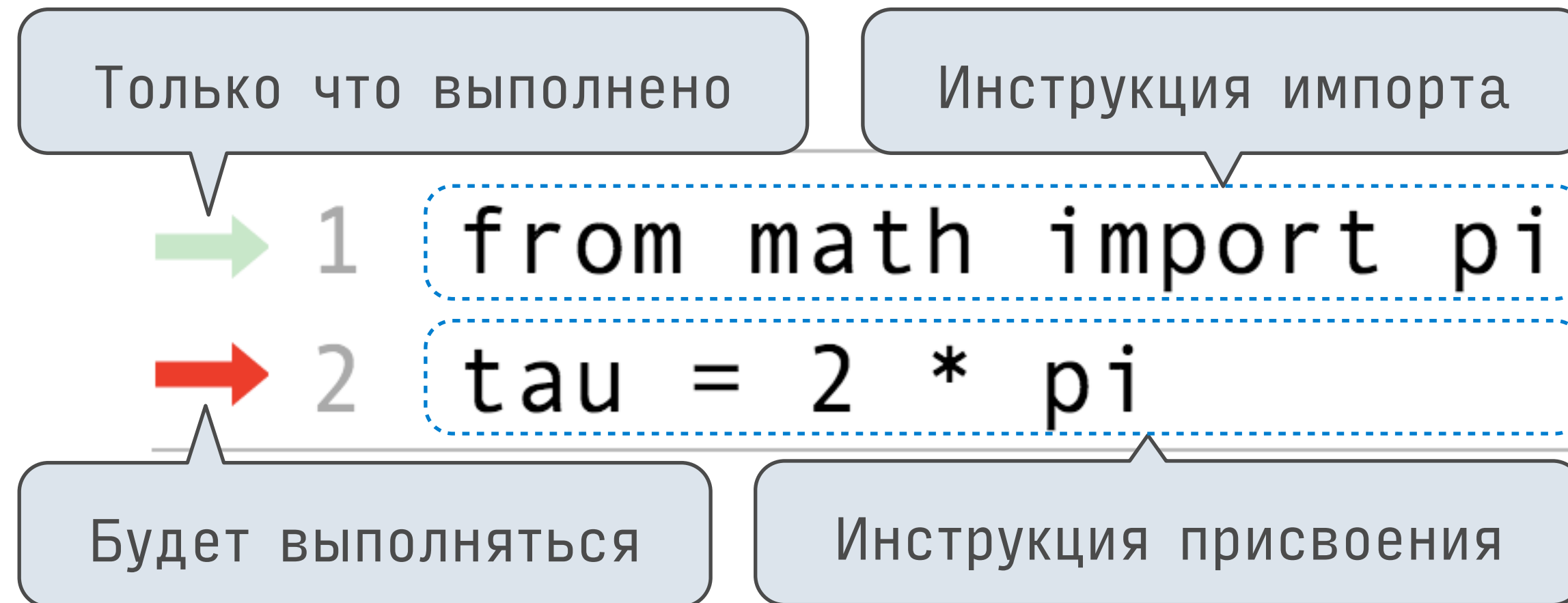
???



Диаграммы окружения

Диаграммы окружения

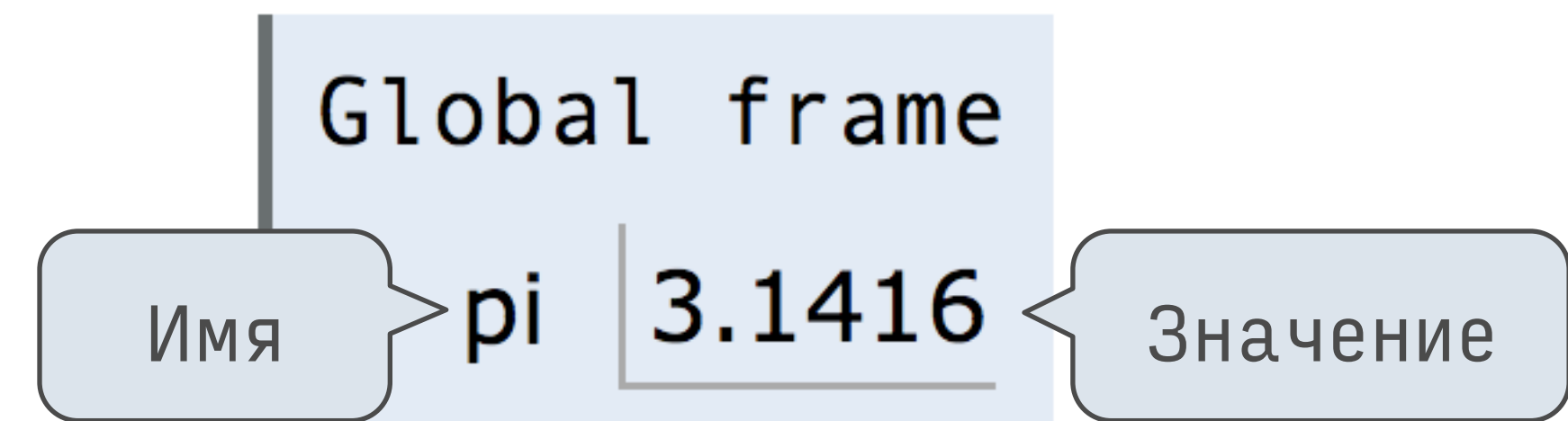
Диаграммы окружения визуализируют процесс интерпретации.



Код (слева):

Инструкции и выражения

Стрелки указывают порядок исполнения



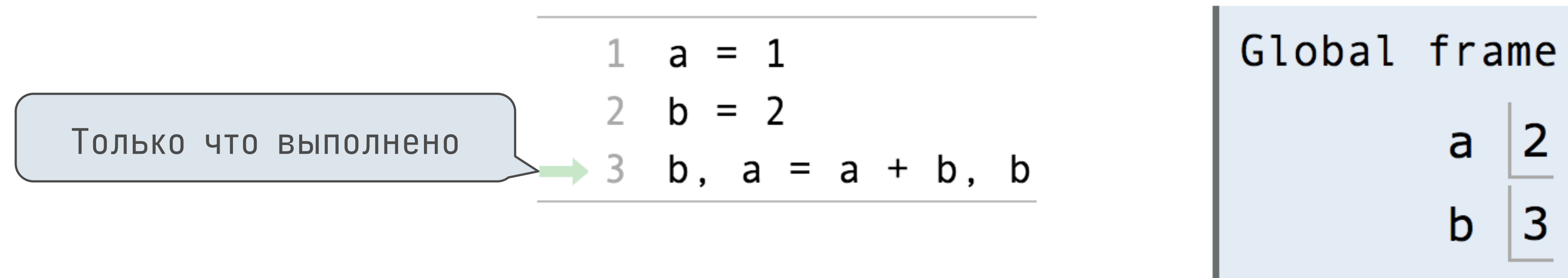
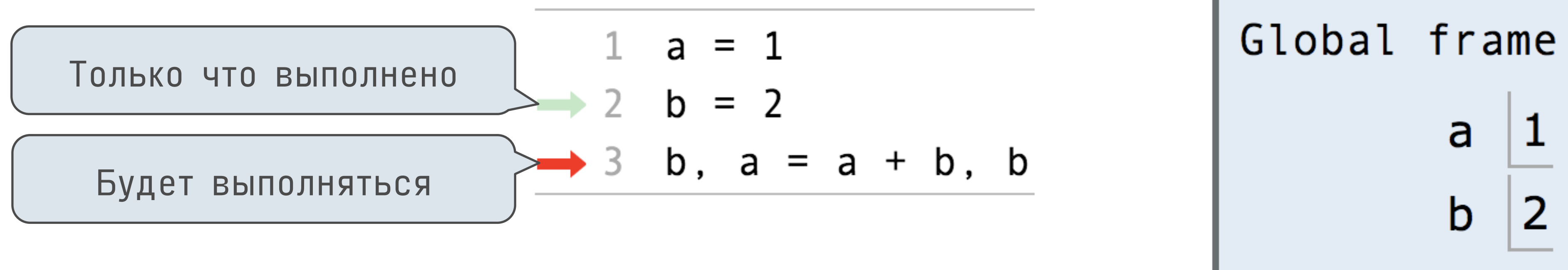
Фреймы (справа):

Каждое имя связано со значением

Внутри фрейма имя не может повторяться

(Пример)

Инструкция присвоения



Правила выполнения инструкций присвоения:

1. Вычислить значения всех выражений справа от = слева направо.
2. Связать в текущем фрейме все имена слева от = с полученными значениями.

Вопрос 1. Решение

(Пример)

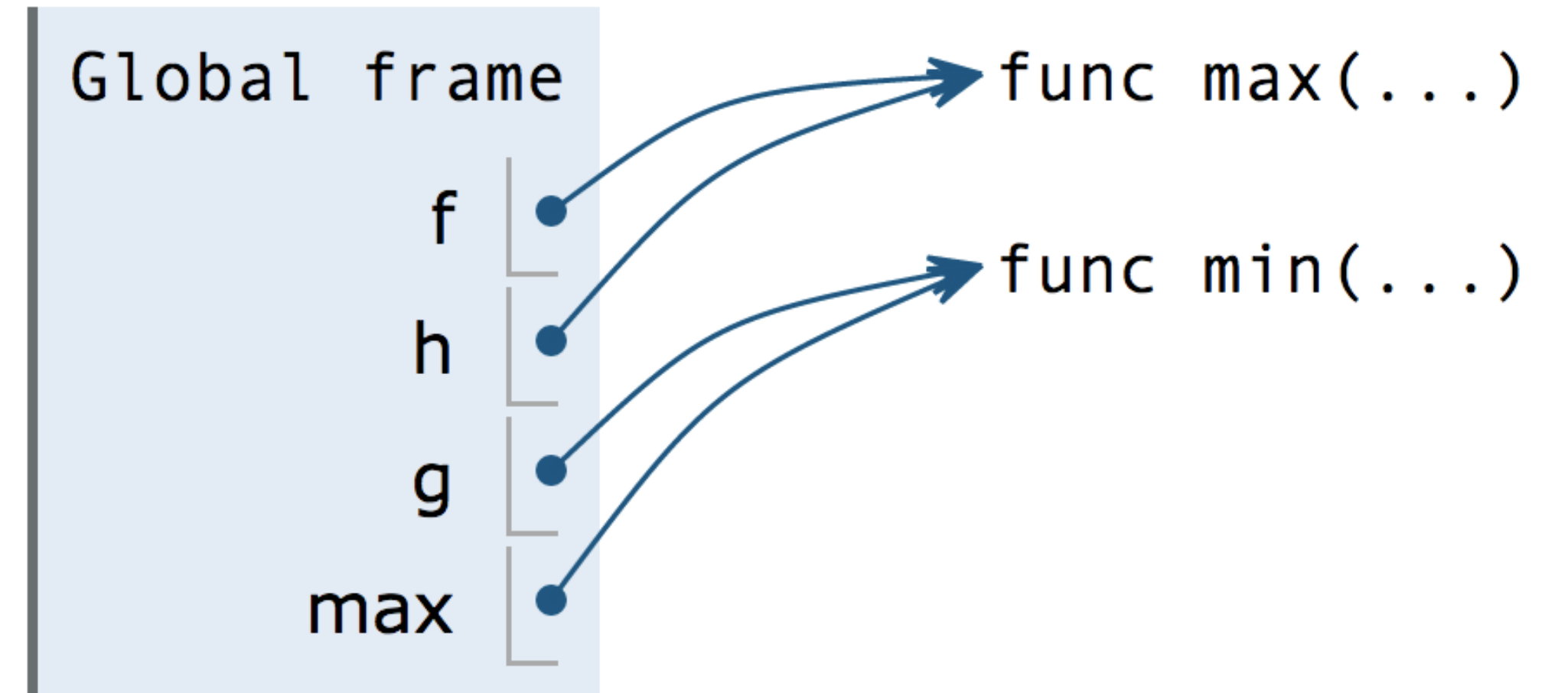
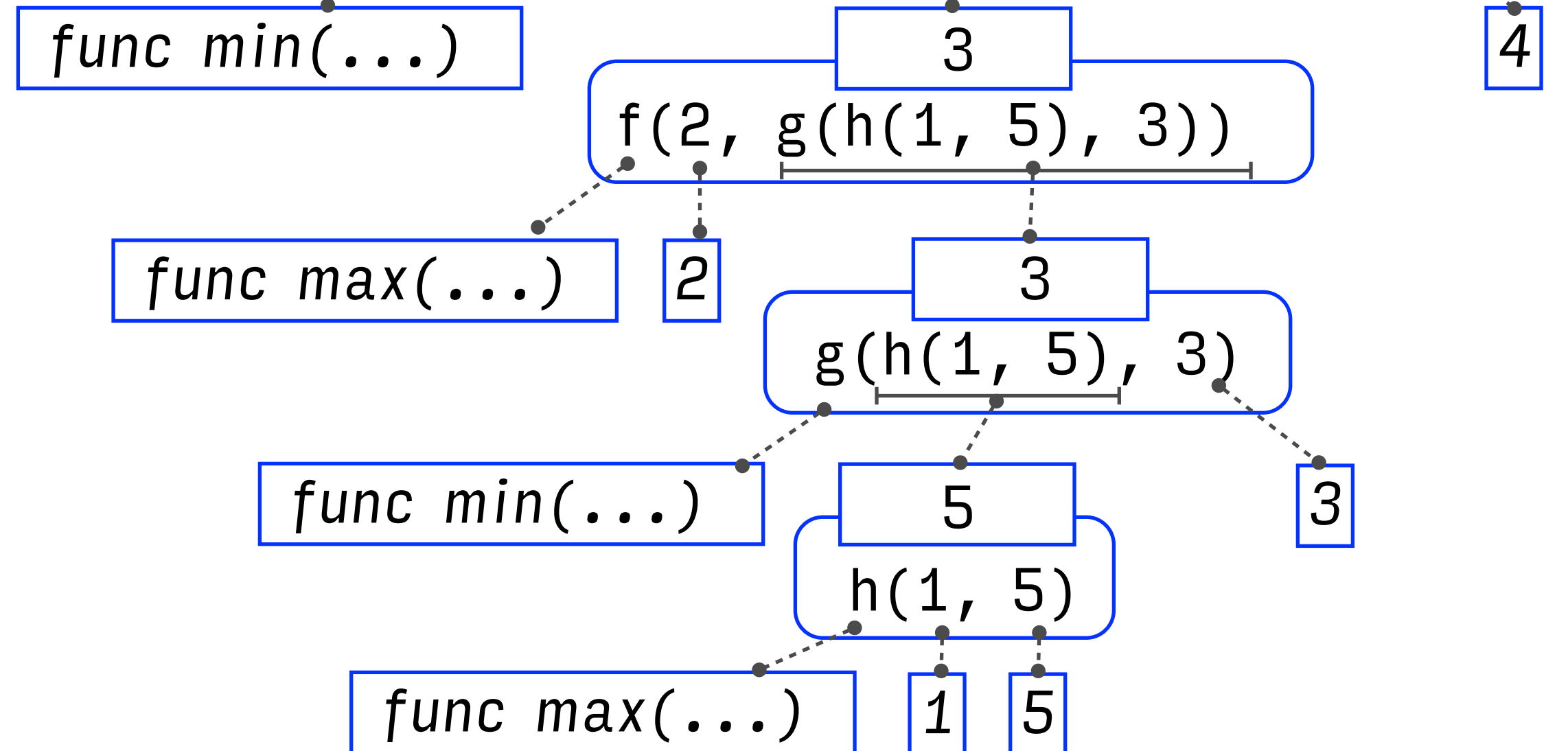
```
1 f = min
2 f = max
3 g, h = min, max
```



```
4 max = g
```



```
5 max(f(2, g(h(1, 5), 3)), 4)
```



3

Объявление функций

Объявление функций

Присвоение является простым средством абстракции: связывает имена со значениями.

Задание функций является более мощным средством абстракции: связывает имена с выражениями.

Сигнатура определяет количество аргументов, передаваемых в функцию

```
>>> def <имя> (<формальные параметры>):  
    return <выражение возврата>
```

Тело описывает вычисление, выполняемое при вызове функции

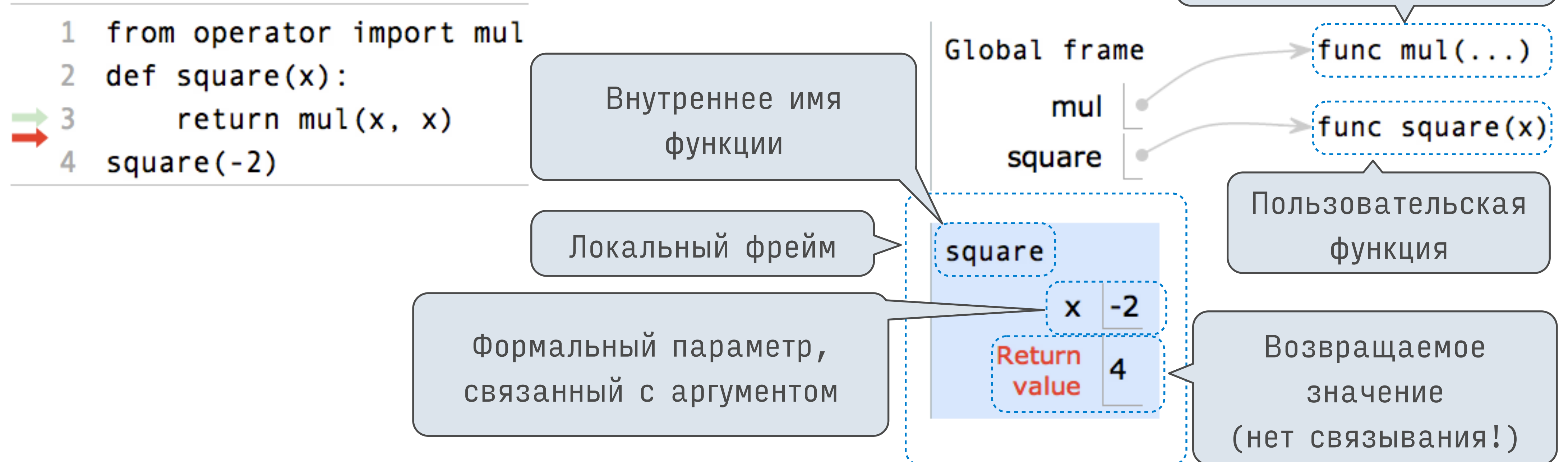
Выполнение инструкции `def`:

1. Создать функцию с сигнатурой `<имя> (<формальные параметры>)`.
2. Определить тело функции по отступам строк.
3. В текущем фрейме связать `<имя>` с созданной функцией.

Вызов пользовательских функций

Процедура вызова/применения пользовательской функции (версия 1):

1. Создать новый локальный фрейм, сформировав новое окружение.
2. Связать в этом фрейме формальные параметры функции с аргументами.
3. Выполнить тело функции в новом окружении.



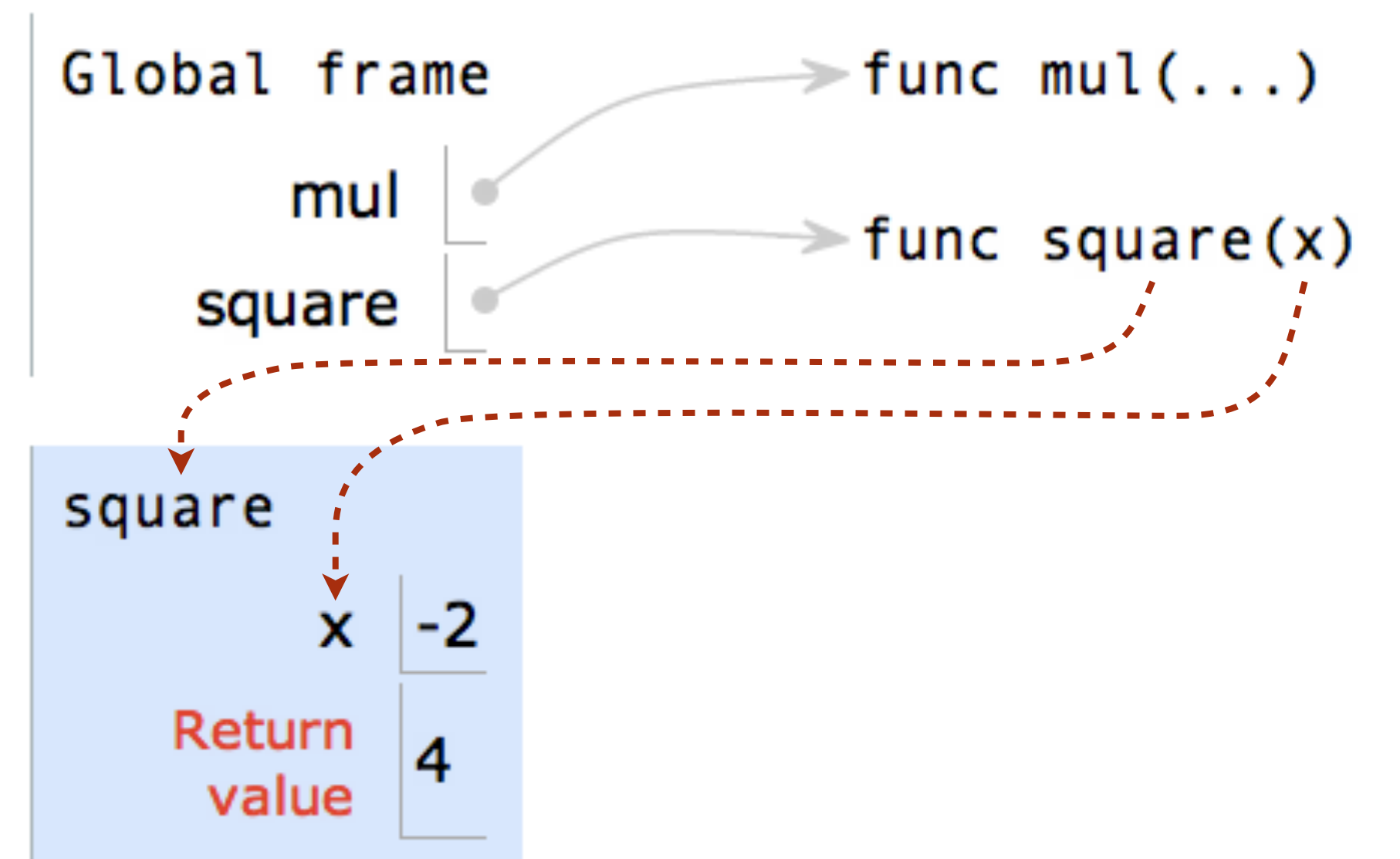
Вызов пользовательских функций

Процедура вызова/применения пользовательской функции (версия 1):

1. Создать новый локальный фрейм, сформировав новое окружение.
2. Связать в этом фрейме формальные параметры функции с аргументами.
3. Выполнить тело функции в новом окружении.

```
1 from operator import mul
2 def square(x):
3     return mul(x, x)
4 square(-2)
```

Сигнатура функции содержит всю необходимую информацию для создания фрейма



Поиск имен в окружении

Всякое выражение вычисляется в контексте окружения.

Пока что, текущее окружение может состоять из:

- Глобального фрейма, или
- Локального фрейма и, далее, глобального фрейма.

Две самые важные мысли в этой лекции:

- Окружение — это последовательность фреймов.
- Значение имени в текущем окружении берётся из первого фрейма, в котором это имя встречается.

То есть для поиска некоторого имени внутри тела функции **square**:

- ищем это имя в локальном фрейме;
- если не находим, то ищем это имя в глобальном фрейме
(встроенные имена вроде «`max`» также находятся в глобальном фрейме, однако они не отображаются на диаграмме окружения).

(Пример)

Print и None

(Пример)

None указывает, что ничего не возвращается

Специальное значение **None** означает в Python «ничто».

Функция без **return** неявно возвращает **None**.

Внимание: **None** не отображается интерпретатором в качестве значения выражения.

```
>>> def does_not_square(x):
```

```
...     x * x
```

```
...     
```

Нет **return**

```
>>> does_not_square(4)
```

Значение **None** не показывается

```
>>> sixteen = does_not_square(4)
```

```
>>> sixteen + 4
```

Имя **sixteen**
теперь связано со
значением **None**

```
Traceback (most recent call last):
```

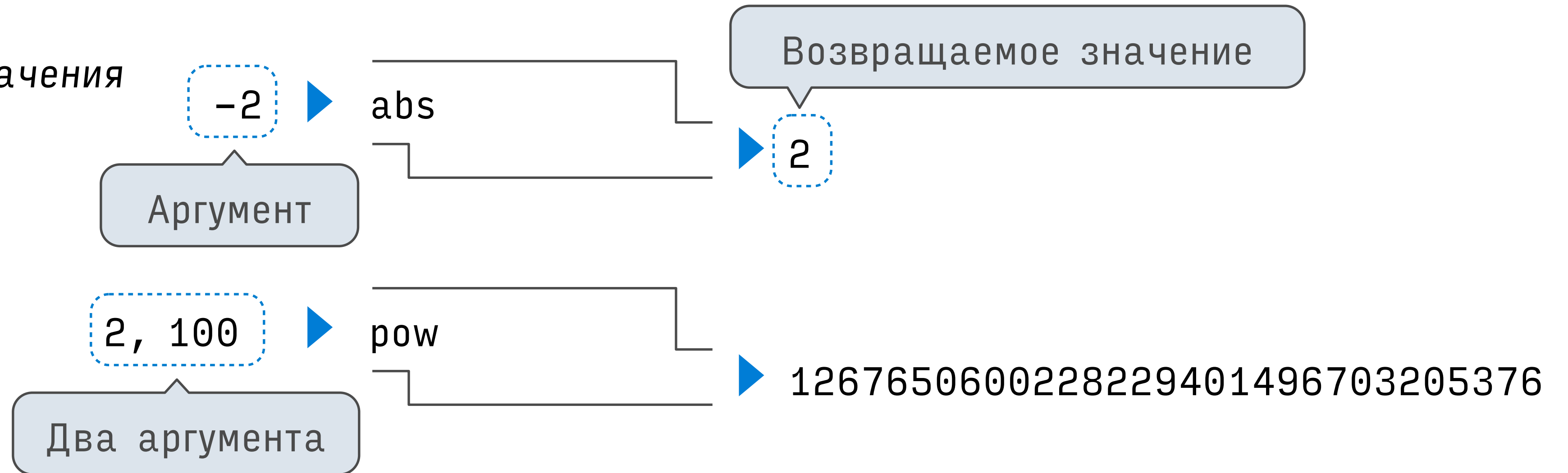
```
  File "<stdin>", line 1, in <module>
```

```
TypeError: unsupported operand type(s) for +: 'NoneType' and 'int'
```

Чистые и нечистые функции

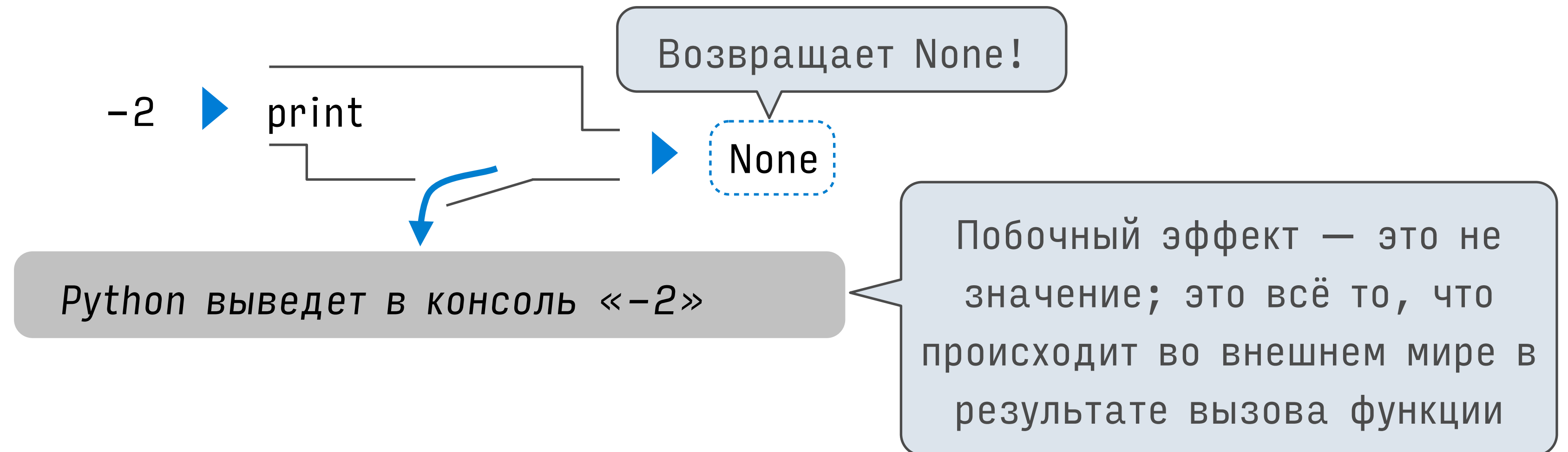
Чистые функции

просто возвращают значения



Нечистые функции

имеют побочные эффекты



Вложенные выражения с print

