

Лекция 13

Изменчивые функции

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

```
>>> withdraw(25)
```

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

```
>>> withdraw(25)  
75
```

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое
значение: остаток на
счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое
значение: остаток на
счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

```
>>> withdraw(25)  
50
```


Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое
значение: остаток на
счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

```
>>> withdraw(25)  
50
```

Повторное снятие той
же суммы

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)
```

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)  
'Недостаточно средств'
```

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)  
'Недостаточно средств'
```

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)  
'Недостаточно средств'
```

```
>>> withdraw(15)
```

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)  
'Недостаточно средств'
```

```
>>> withdraw(15)  
35
```

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)  
'Недостаточно средств'
```

```
>>> withdraw(15)  
35
```


Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)  
'Недостаточно средств'
```

```
>>> withdraw(15)  
35
```

Где же хранится баланс?

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)  
'Недостаточно средств'
```

```
>>> withdraw(15)  
35
```

Где же хранится баланс?

```
>>> withdraw = make_withdraw(100)
```

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)  
'Недостаточно средств'
```

```
>>> withdraw(15)  
35
```

Где же хранится баланс?

```
>>> withdraw = make_withdraw(100)
```

В родительском фрейме функции!

Функция, поведение которой меняется с течением времени

Смоделируем банковский счет с начальным балансом в 100 ₺

Возвращаемое значение: остаток на счете

```
>>> withdraw(25)  
75
```

Аргумент:
сумма для снятия

Возвращаемое значение изменилось!

```
>>> withdraw(25)  
50
```

Повторное снятие той же суммы

```
>>> withdraw(60)  
'Недостаточно средств'
```

Где же хранится баланс?

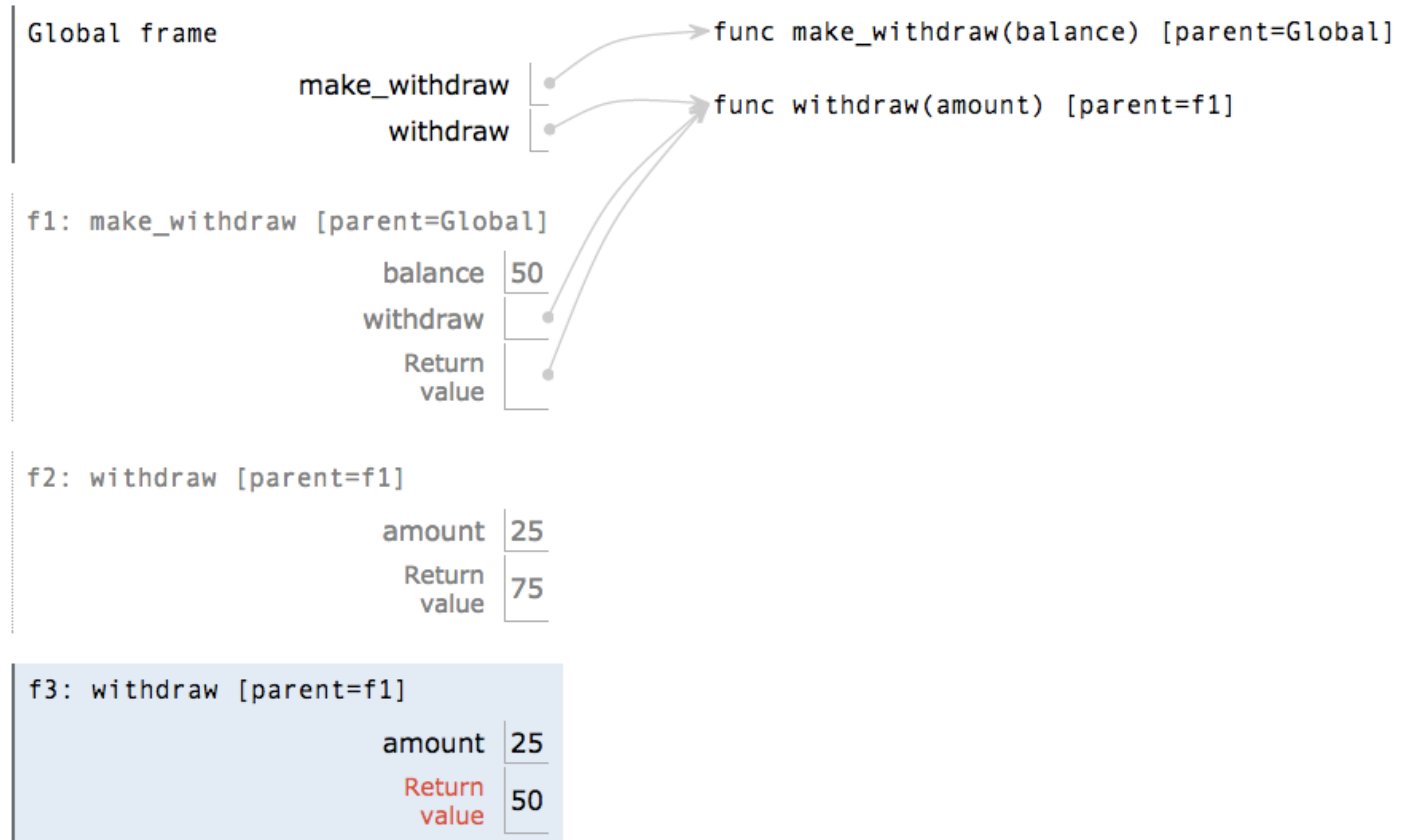
```
>>> withdraw(15)  
35
```

```
>>> withdraw = make_withdraw(100)
```

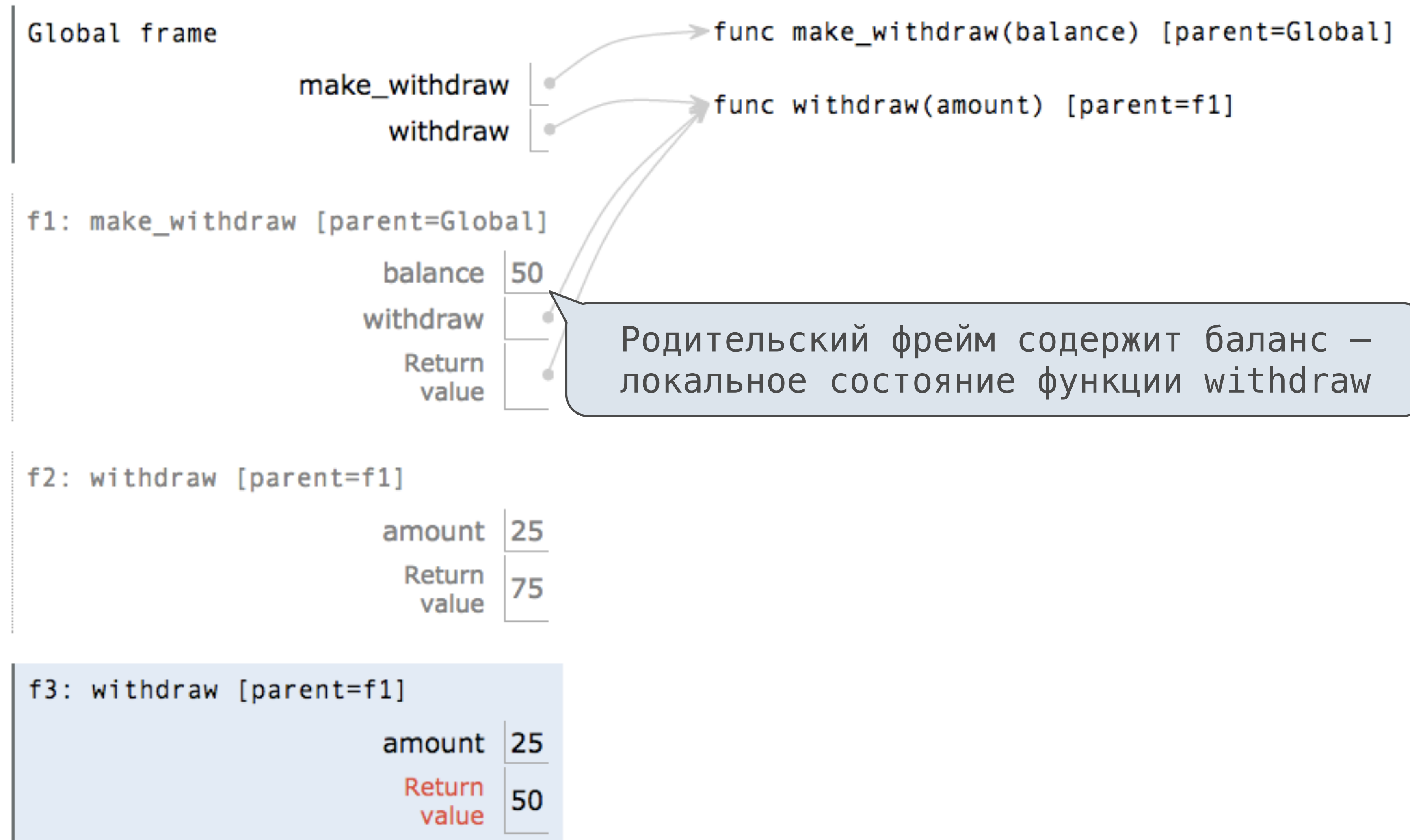
В родительском фрейме функции!

У функции есть тело и родительское окружение

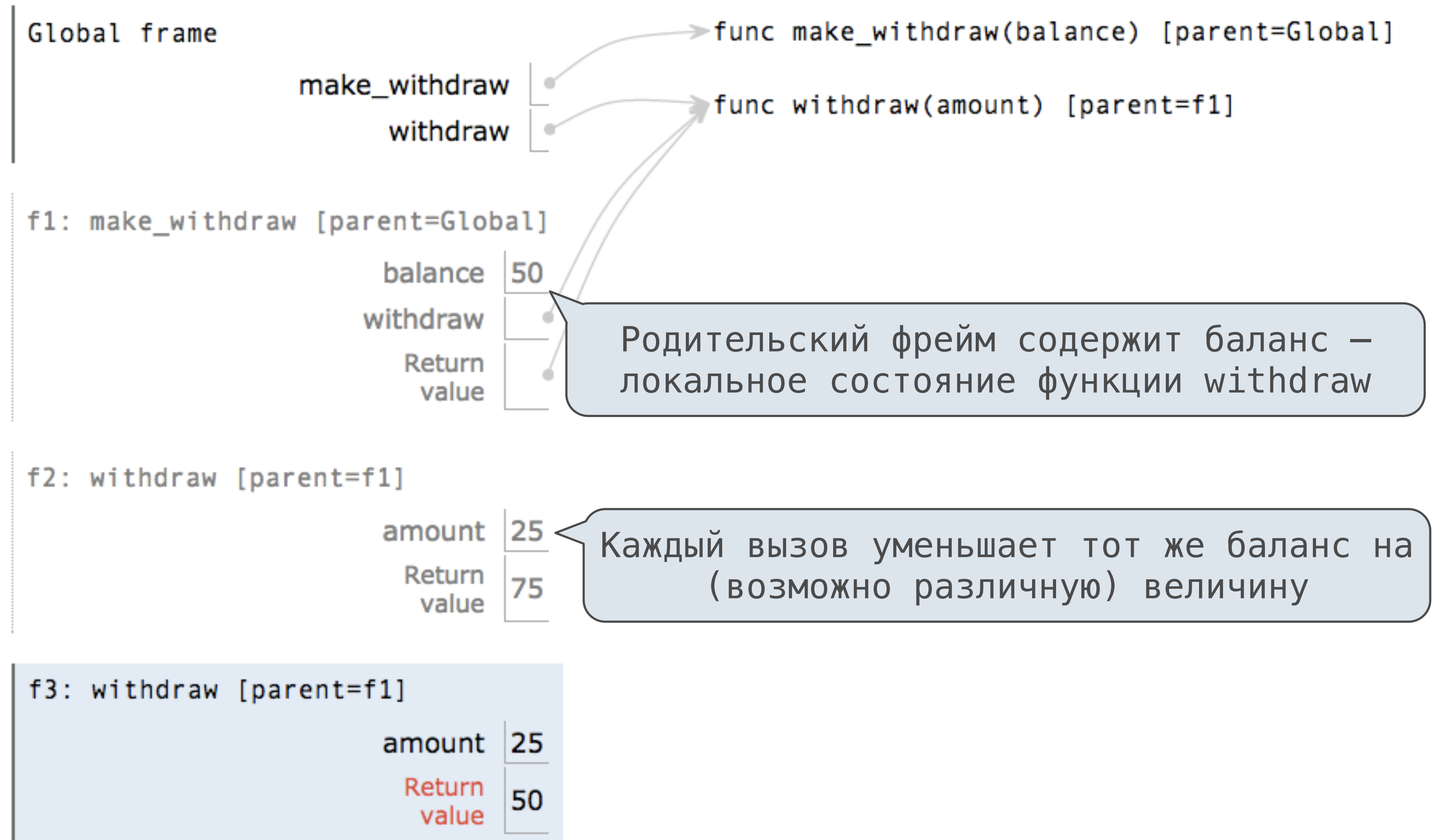
Хранение локального состояния с помощью окружений



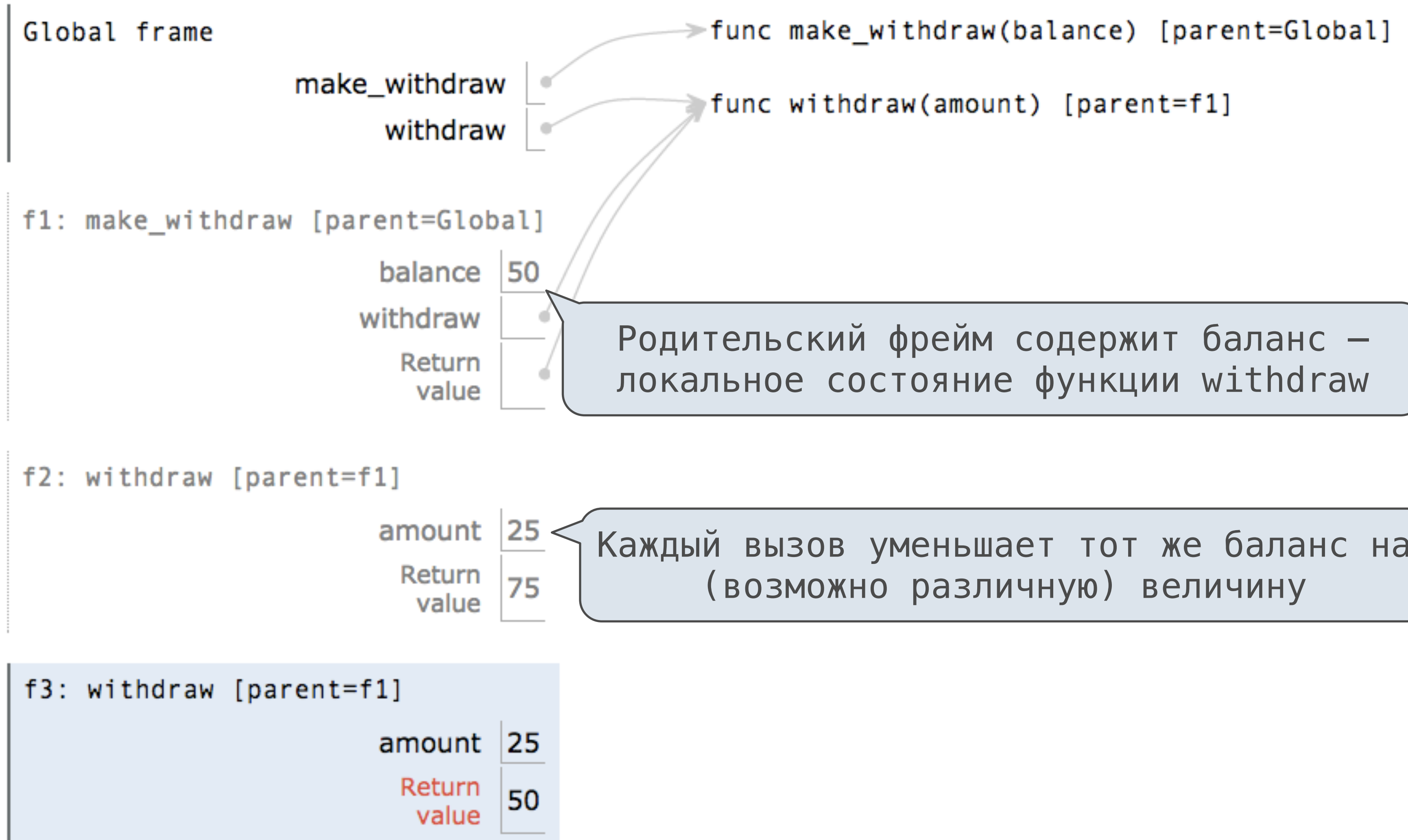
Хранение локального состояния с помощью окружений



Хранение локального состояния с помощью окружений



Хранение локального состояния с помощью окружений



Все вызовы некоторой функции имеют общий родительский фрейм

Повторение: локальное присвоение

```
def percent_difference(x, y):  
    difference = abs(x-y)  
    return 100 * difference / x  
diff = percent_difference(40, 50)
```

Global frame

percent_difference

func percent_difference(x, y) [parent=Global]

```
f1: percent_difference [parent=Global]
```

x	40
y	50
difference	10

Повторение: локальное присвоение

```
def percent_difference(x, y):  
    difference = abs(x-y)  
    return 100 * difference / x  
diff = percent_difference(40, 50)
```

Присвоение связывает имя со значением в первом фрейме текущего окружения

Global frame

percent_difference

func percent_difference(x, y) [parent=Global]

f1: percent_difference [parent=Global]

x	40
y	50
difference	10

Повторение: локальное присвоение

```
def percent_difference(x, y):  
    difference = abs(x-y)  
    return 100 * difference / x  
diff = percent_difference(40, 50)
```

Присвоение связывает имя со значением в первом фрейме текущего окружения

Global frame

percent_difference

func percent_difference(x, y) [parent=Global]

f1: percent_difference [parent=Global]

x 40

y 50

→ difference 10

Повторение: локальное присвоение

```
def percent_difference(x, y):  
    difference = abs(x-y)  
    return 100 * difference / x  
diff = percent_difference(40, 50)
```

Присвоение связывает имя со значением в первом фрейме текущего окружения

Global frame

percent_difference

func percent_difference(x, y) [parent=Global]

f1: percent_difference [parent=Global]

x 40

y 50

→ difference 10

Правила выполнения инструкции присвоения:

Повторение: локальное присвоение

```
def percent_difference(x, y):  
    difference = abs(x-y)  
    return 100 * difference / x  
diff = percent_difference(40, 50)
```

Присвоение связывает имя со значением в первом фрейме текущего окружения

Global frame

percent_difference

func percent_difference(x, y) [parent=Global]

f1: percent_difference [parent=Global]

x 40

y 50

→ difference 10

Правила выполнения инструкции присвоения:

1. Вычислить все выражения справа от =, слева направо.
2. Связать имена слева от = с полученными значениями в **текущем фрейме**

Нелокальное присвоение и хранение локального состояния

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):
```

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):  
    """Возвращает функцию withdraw с начальным балансом."""
```


Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):  
    """Возвращает функцию withdraw с начальным балансом."""  
    def withdraw(amount):
```

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):  
    """Возвращает функцию withdraw с начальным балансом."""  
  
    def withdraw(amount):  
        nonlocal balance
```

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):  
    """Возвращает функцию withdraw с начальным балансом."""  
  
    def withdraw(amount):  
        nonlocal balance  
  
        if amount > balance:
```

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):  
    """Возвращает функцию withdraw с начальным балансом."""  
  
    def withdraw(amount):  
        nonlocal balance  
  
        if amount > balance:  
            return 'Недостаточно средств'
```

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):  
    """Возвращает функцию withdraw с начальным балансом."""  
  
    def withdraw(amount):  
        nonlocal balance  
  
        if amount > balance:  
            return 'Недостаточно средств'  
  
        balance = balance - amount
```

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):  
    """Возвращает функцию withdraw с начальным балансом."""  
  
    def withdraw(amount):  
        nonlocal balance  
  
        if amount > balance:  
            return 'Недостаточно средств'  
  
        balance = balance - amount  
  
        return balance
```

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):  
    """Возвращает функцию withdraw с начальным балансом."""  
    def withdraw(amount):  
        nonlocal balance  
        if amount > balance:  
            return 'Недостаточно средств'  
        balance = balance - amount  
        return balance  
    return withdraw
```

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):
```

```
    """Возвращает функцию withdraw с начальным балансом."""
```

```
    def withdraw(amount):
```

```
        nonlocal balance
```

Имя «balance» объявлено нелокальным в начале тела функции, в которой произойдет пересвязывание

```
        if amount > balance:
```

```
            return 'Недостаточно средств'
```

```
        balance = balance - amount
```

```
        return balance
```

```
    return withdraw
```


Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):
```

```
    """Возвращает функцию withdraw с начальным балансом."""
```

```
    def withdraw(amount):
```

```
        nonlocal balance
```

```
        if amount > balance:
```

```
            return 'Недостаточно средств'
```

```
        balance = balance - amount
```

```
        return balance
```

```
    return withdraw
```

Имя «balance» объявлено нелокальным в начале тела функции, в которой произойдет пересвязывание

Пересвязывание баланса в первом нелокальном фрейме, в котором это имя уже существует

Нелокальное присвоение и хранение локального состояния

```
def make_withdraw(balance):
```

```
    """Возвращает функцию withdraw с начальным балансом."""
```

```
    def withdraw(amount):
```

```
        nonlocal balance
```

Имя «balance» объявлено нелокальным в начале тела функции, в которой произойдет пересвязывание

```
        if amount > balance:
```

```
            return 'Недостаточно средств'
```

```
        balance = balance - amount
```

Пересвязывание баланса в первом нелокальном фрейме, в котором это имя уже существует

```
        return balance
```

```
    return withdraw
```

(Пример)

Нелокальное присвоение

Действие нелокальных присвоений

`nonlocal <имя>`

Действие нелокальных присвоений

`nonlocal <имя>`

Результат: Последующие присвоения этому имени влияют только на уже существующую связь в **первом нелокальном фрейме** текущего окружения, в котором это имя связано.

Действие нелокальных присвоений

`nonlocal <имя>`

Результат: Последующие присвоения этому имени влияют только на уже существующую связь в **первом нелокальном фрейме** текущего окружения, в котором это имя связано.

Документация Python: в «области видимости»

Действие нелокальных присвоений

```
nonlocal <имя>, <имя>, ...
```

Результат: Последующие присвоения этому имени влияют только на уже существующую связь в **первом нелокальном фрейме** текущего окружения, в котором это имя связано.

Документация Python: в
«области видимости»

Действие нелокальных присвоений

```
nonlocal <имя>, <имя>, ...
```

Результат: Последующие присвоения этому имени влияют только на уже существующую связь в **первом нелокальном фрейме** текущего окружения, в котором это имя связано.

Документация Python: в
«области видимости»

Из руководства по Python 3:

Действие нелокальных присвоений

```
nonlocal <имя>, <имя>, ...
```

Результат: Последующие присвоения этому имени влияют только на уже существующую связь в **первом нелокальном фрейме** текущего окружения, в котором это имя связано.

Документация Python: в
«области видимости»

Из руководства по Python 3:

Указанные в инструкции `nonlocal` имена должны существовать в области видимости.

Действие нелокальных присвоений

```
nonlocal <имя>, <имя>, ...
```

Результат: Последующие присвоения этому имени влияют только на уже существующую связь в **первом нелокальном фрейме** текущего окружения, в котором это имя связано.

Документация Python: в
«области видимости»

Из руководства по Python 3:

Указанные в инструкции `nonlocal` имена должны существовать в области видимости.

Указанные в инструкции `nonlocal` имена не должны совпадать с существующими в локальной области видимости.

Действие нелокальных присвоений

```
nonlocal <имя>, <имя>, ...
```

Результат: Последующие присвоения этому имени влияют только на уже существующую связь в **первом нелокальном фрейме** текущего окружения, в котором это имя связано.

Документация Python: в «области видимости»

Из руководства по Python 3:

Указанные в инструкции `nonlocal` имена должны существовать в области видимости.

Указанные в инструкции `nonlocal` имена не должны совпадать с существующими в локальной области видимости.

Текущем фрейме

Действие нелокальных присвоений

```
nonlocal <имя>, <имя>, ...
```

Результат: Последующие присвоения этому имени влияют только на уже существующую связь в **первом нелокальном фрейме** текущего окружения, в котором это имя связано.

Документация Python: в «области видимости»

Из руководства по Python 3:

Указанные в инструкции `nonlocal` имена должны существовать в области видимости.

Указанные в инструкции `nonlocal` имена не должны совпадать с существующими в локальной области видимости.

Текущем фрейме

http://docs.python.org/release/3.1.3/reference/simple_stmts.html#the-nonlocal-statement

Действие нелокальных присвоений

```
nonlocal <имя>, <имя>, ...
```

Результат: Последующие присвоения этому имени влияют только на уже существующую связь в **первом нелокальном фрейме** текущего окружения, в котором это имя связано.

Документация Python: в «области видимости»

Из руководства по Python 3:

Указанные в инструкции `nonlocal` имена должны существовать в области видимости.

Указанные в инструкции `nonlocal` имена не должны совпадать с существующими в локальной области видимости.

Текущем фрейме

http://docs.python.org/release/3.1.3/reference/simple_stmts.html#the-nonlocal-statement

<http://www.python.org/dev/peps/pep-3104/>

Множественные толкования инструкции присвоения

$x = 2$

Множественные толкования инструкции присвоения

$$x = 2$$

Состояние

Результат

Множественные толкования инструкции присвоения

`x = 2`

Состояние

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Результат

Множественные толкования инструкции присвоения

`x = 2`

Состояние

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Результат

Создается связь имени «x» с объектом 2 в первом фрейме текущего окружения

Множественные толкования инструкции присвоения

`x = 2`

Состояние

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Результат

Создается связь имени «x» с объектом 2 в первом фрейме текущего окружения

-
- Нет инструкции `nonlocal`
 - «x» **связано** локально
-
-
-

Множественные толкования инструкции присвоения

`x = 2`

Состояние

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Результат

Создается связь имени «x» с объектом 2 в первом фрейме текущего окружения

- Нет инструкции `nonlocal`
- «x» **связано** локально

Пересвязывание имени «x» с объектом 2 в первом фрейме текущего окружения

Множественные толкования инструкции присвоения

`x = 2`

Состояние

Результат

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Создается связь имени «x» с объектом 2 в первом фрейме текущего окружения

- Нет инструкции `nonlocal`
- «x» **связано** локально

Пересвязывание имени «x» с объектом 2 в первом фрейме текущего окружения

- `nonlocal x`
- «x» **связано** в нелокальном фрейме

Множественные толкования инструкции присвоения

$x = 2$

Состояние

Результат

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Создается связь имени «x» с объектом 2 в первом фрейме текущего окружения

- Нет инструкции `nonlocal`
- «x» **связано** локально

Пересвязывание имени «x» с объектом 2 в первом фрейме текущего окружения

- `nonlocal x`
- «x» **связано** в нелокальном фрейме

Пересвязывание «x» с 2 в первом нелокальном фрейме текущего окружения, в котором присутствует «x»

Множественные толкования инструкции присвоения

$x = 2$

Состояние

Результат

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Создается связь имени «x» с объектом 2 в первом фрейме текущего окружения

- Нет инструкции `nonlocal`
- «x» **связано** локально

Пересвязывание имени «x» с объектом 2 в первом фрейме текущего окружения

- `nonlocal x`
- «x» **связано** в нелокальном фрейме

Пересвязывание «x» с 2 в первом нелокальном фрейме текущего окружения, в котором присутствует «x»

- `nonlocal x`
- «x» **не** связано в нелокальном фрейме

Множественные толкования инструкции присвоения

`x = 2`

Состояние

Результат

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Создается связь имени «x» с объектом 2 в первом фрейме текущего окружения

- Нет инструкции `nonlocal`
- «x» **связано** локально

Пересвязывание имени «x» с объектом 2 в первом фрейме текущего окружения

- `nonlocal x`
- «x» **связано** в нелокальном фрейме

Пересвязывание «x» с 2 в первом нелокальном фрейме текущего окружения, в котором присутствует «x»

- `nonlocal x`
- «x» **не** связано в нелокальном фрейме

`SyntaxError: no binding for nonlocal 'x' found`

Множественные толкования инструкции присвоения

`x = 2`

Состояние

Результат

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Создается связь имени «x» с объектом 2 в первом фрейме текущего окружения

- Нет инструкции `nonlocal`
- «x» **связано** локально

Пересвязывание имени «x» с объектом 2 в первом фрейме текущего окружения

- `nonlocal x`
- «x» **связано** в нелокальном фрейме

Пересвязывание «x» с 2 в первом нелокальном фрейме текущего окружения, в котором присутствует «x»

- `nonlocal x`
- «x» **не** связано в нелокальном фрейме

`SyntaxError: no binding for nonlocal 'x' found`

- `nonlocal x`
- «x» **связано** в нелокальном фрейме
- «x» также связано локально

Множественные толкования инструкции присвоения

`x = 2`

Состояние

Результат

- Нет инструкции `nonlocal`
- «x» **не** связано локально

Создается связь имени «x» с объектом 2 в первом фрейме текущего окружения

- Нет инструкции `nonlocal`
- «x» **связано** локально

Пересвязывание имени «x» с объектом 2 в первом фрейме текущего окружения

- `nonlocal x`
- «x» **связано** в нелокальном фрейме

Пересвязывание «x» с 2 в первом нелокальном фрейме текущего окружения, в котором присутствует «x»

- `nonlocal x`
- «x» **не** связано в нелокальном фрейме

`SyntaxError: no binding for nonlocal 'x' found`

- `nonlocal x`
- «x» **связано** в нелокальном фрейме
- «x» также связано локально

`SyntaxError: name 'x' is parameter and nonlocal`

Особенность Python

```
def make_withdraw(balance):  
    def withdraw(amount):  
        if amount > balance:  
            return 'Недостаточно средств'  
        balance = balance - amount  
        return balance  
    return withdraw
```

```
wd = make_withdraw(20)  
wd(5)
```

Особенность Python

Python предвычисляет, какой фрейм содержит какое имя до выполнения тела функции.

```
def make_withdraw(balance):  
    def withdraw(amount):  
        if amount > balance:  
            return 'Недостаточно средств'  
        balance = balance - amount  
        return balance  
    return withdraw
```

```
wd = make_withdraw(20)  
wd(5)
```

Особенность Python

Python предвычисляет, какой фрейм содержит какое имя до выполнения тела функции.

Внутри тела функции все упоминания имени должны относиться к одному фрейму.

```
def make_withdraw(balance):  
    def withdraw(amount):  
        if amount > balance:  
            return 'Недостаточно средств'  
        balance = balance - amount  
        return balance  
    return withdraw
```

```
wd = make_withdraw(20)  
wd(5)
```

Особенность Python

Python предвычисляет, какой фрейм содержит какое имя до выполнения тела функции.

Внутри тела функции все упоминания имени должны относиться к одному фрейму.

```
def make_withdraw(balance):  
    def withdraw(amount):  
        if amount > balance:  
            return 'Недостаточно средств'  
        balance = balance - amount  
        return balance  
    return withdraw
```

```
wd = make_withdraw(20)  
wd(5)
```

Локальное присвоение

Особенность Python

Python предвычисляет, какой фрейм содержит какое имя до выполнения тела функции.

Внутри тела функции все упоминания имени должны относиться к одному фрейму.

```
def make_withdraw(balance):  
    def withdraw(amount):  
        if amount > balance:  
            return 'Недостаточно средств'  
        balance = balance - amount  
        return balance  
    return withdraw
```

```
wd = make_withdraw(20)  
wd(5)
```

Локальное присвоение

UnboundLocalError: local variable 'balance' referenced before assignment

Изменяющиеся значения и хранение локального состояния

Изменяющиеся значения могут быть изменены без инструкции «nonlocal».

Изменяющиеся значения и хранение локального состояния

Изменяющиеся значения могут быть изменены без инструкции «nonlocal».

```
def make_withdraw_list(balance):
    b = [balance]
    def withdraw(amount):
        if amount > b[0]:
            return 'Insufficient funds'
        b[0] = b[0] - amount
        return b[0]
    return withdraw

withdraw = make_withdraw_list(100)
withdraw(25)
```


Изменяющиеся значения и хранение локального состояния

Изменяющиеся значения могут быть изменены без инструкции «nonlocal».

Имя связано за пределами тела withdraw

```
def make_withdraw_list(balance):  
    b = [balance]  
    def withdraw(amount):  
        if amount > b[0]:  
            return 'Insufficient funds'  
        b[0] = b[0] - amount  
        return b[0]  
    return withdraw  
  
withdraw = make_withdraw_list(100)  
withdraw(25)
```

Изменяющиеся значения и хранение локального состояния

Изменяющиеся значения могут быть изменены без инструкции «nonlocal».

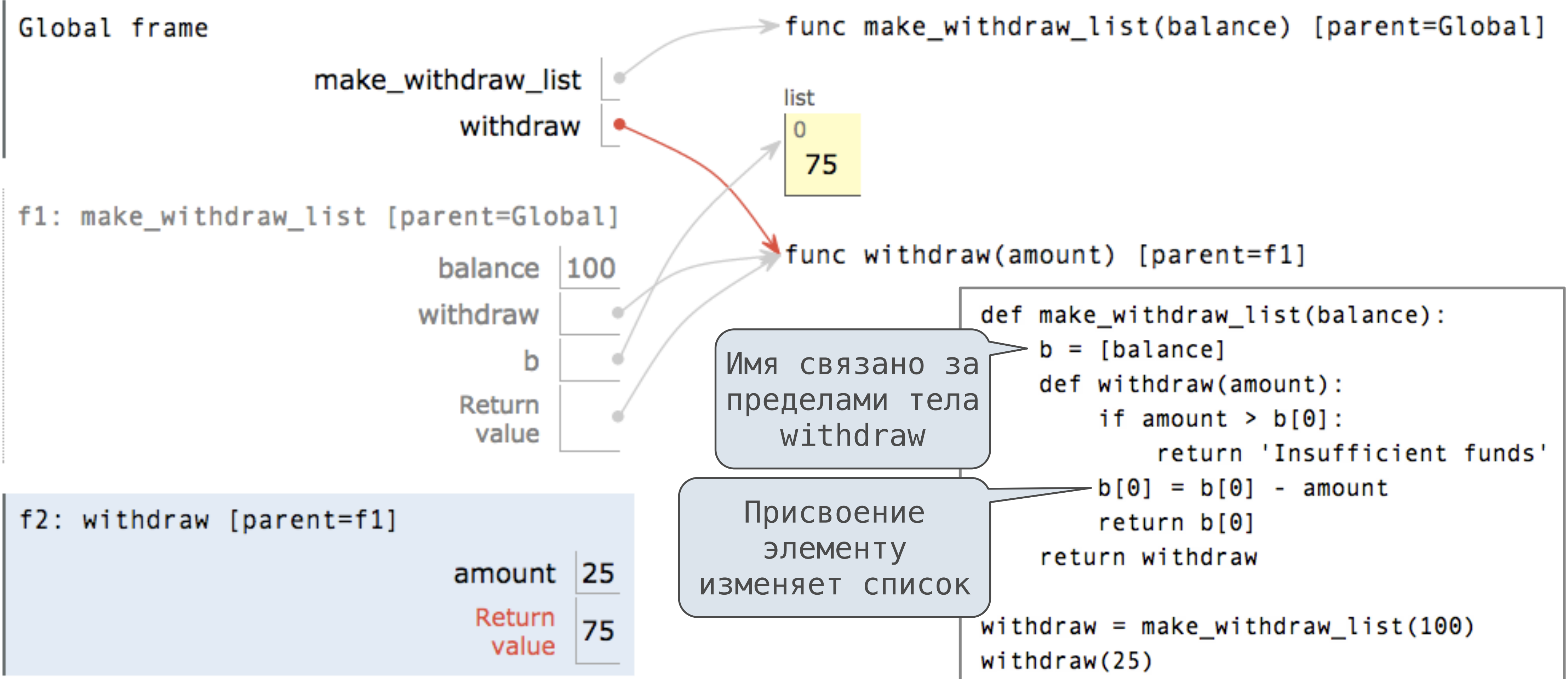
Имя связано за пределами тела withdraw

Присвоение элементу изменяет список

```
def make_withdraw_list(balance):  
    b = [balance]  
    def withdraw(amount):  
        if amount > b[0]:  
            return 'Insufficient funds'  
        b[0] = b[0] - amount  
        return b[0]  
    return withdraw  
  
withdraw = make_withdraw_list(100)  
withdraw(25)
```

Изменяющиеся значения и хранение локального состояния

Изменяющиеся значения могут быть изменены без инструкции «nonlocal».



Имя связано за пределами тела withdraw

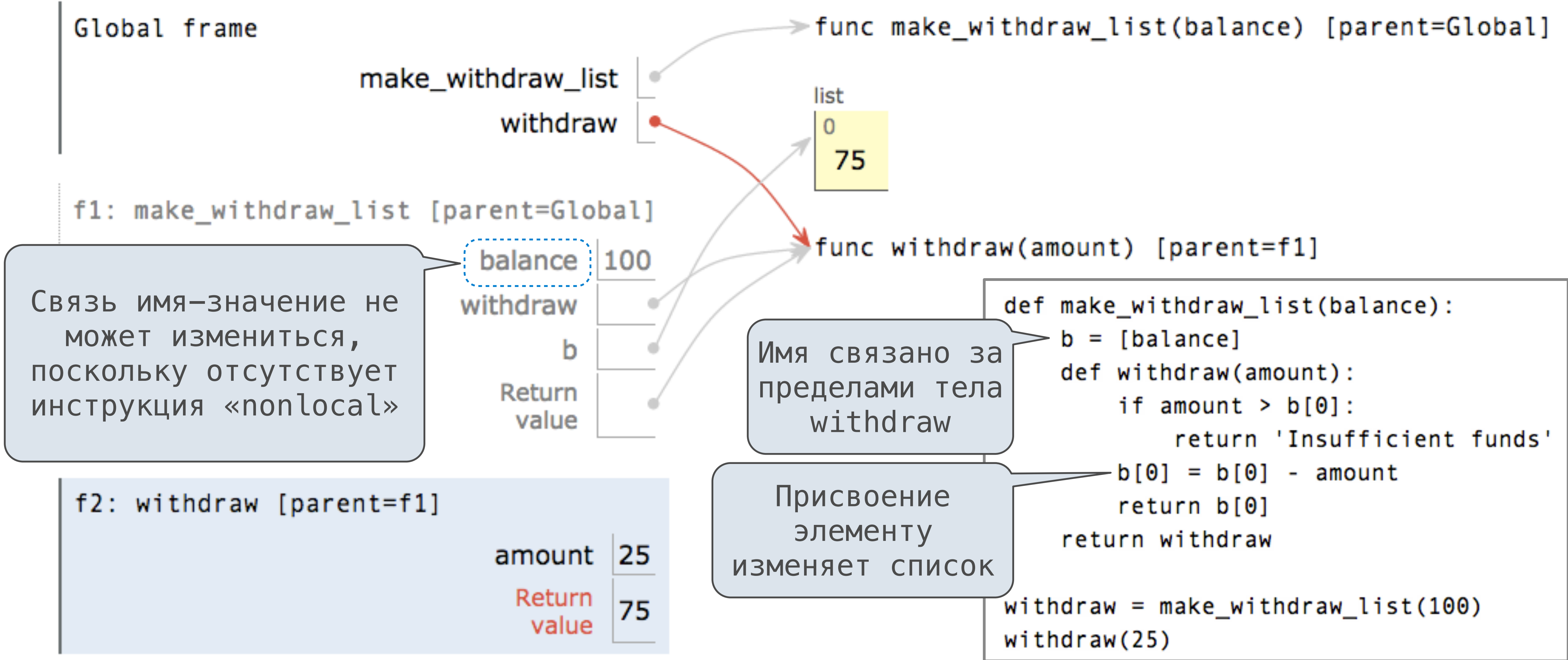
Присвоение элементу изменяет список

def make_withdraw_list(balance):
 b = [balance]
 def withdraw(amount):
 if amount > b[0]:
 return 'Insufficient funds'
 b[0] = b[0] - amount
 return b[0]
 return withdraw

withdraw = make_withdraw_list(100)
withdraw(25)

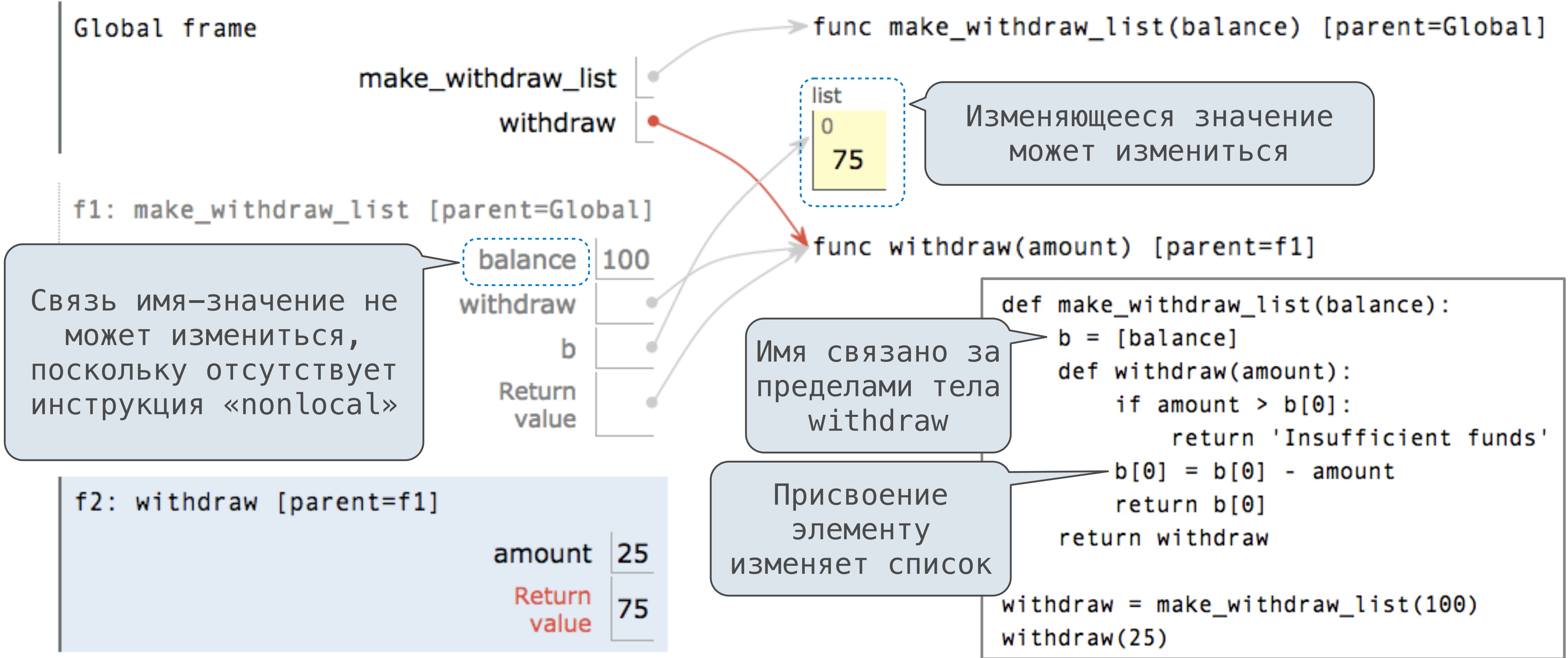
Изменяющиеся значения и хранение локального состояния

Изменяющиеся значения могут быть изменены без инструкции «nonlocal».



Изменяющиеся значения и хранение локального состояния

Изменяющиеся значения могут быть изменены без инструкции «nonlocal».



Множественные изменчивые функции

(Пример)

Ссылочная прозрачность

Ссылочная прозрачность

- Выражение называют **прозрачным для ссылок**, если при замене любого подвыражения его результатом не происходит изменение смысла программы.

Ссылочная прозрачность

- Выражение называют **прозрачным для ссылок**, если при замене любого подвыражения его результатом не происходит изменение смысла программы.

```
mul(add(2, mul(4, 6)), add(3, 5))
```

Ссылочная прозрачность

- Выражение называют **прозрачным для ссылок**, если при замене любого подвыражения его результатом не происходит изменение смысла программы.

```
mul(add(2, mul(4, 6)), add(3, 5))
```

```
mul(add(2, 24), add(3, 5))
```

Ссылочная прозрачность

- Выражение называют **прозрачным для ссылок**, если при замене любого подвыражения его результатом не происходит изменение смысла программы.

```
mul(add(2, mul(4, 6)), add(3, 5))
```

```
mul(add(2, 24), add(3, 5))
```

```
mul(26, add(3, 5))
```

Ссылочная прозрачность

- Выражение называют **прозрачным для ссылок**, если при замене любого подвыражения его результатом не происходит изменение смысла программы.

```
mul(add(2, mul(4, 6)), add(3, 5))
```

```
mul(add(2, 24), add(3, 5))
```

```
mul(26, add(3, 5))
```

- Изменяющиеся операции нарушают условие *прозрачности для ссылок* поскольку они не только возвращают результат; **они изменяют окружение.**

Ссылочная прозрачность

- Выражение называют **прозрачным для ссылок**, если при замене любого подвыражения его результатом не происходит изменение смысла программы.



```
mul(add(2, mul(4, 6)), add(3, 5))
```

```
mul(add(2, 24), add(3, 5))
```

```
mul(26, add(3, 5))
```

- Изменяющиеся операции нарушают условие *прозрачности для ссылок* поскольку они не только возвращают результат; **они изменяют окружение.**

Ссылочная прозрачность

- Выражение называют **прозрачным для ссылок**, если при замене любого подвыражения его результатом не происходит изменение смысла программы.



```
mul(add(2, mul(4, 6)), add(3, 5))
```

```
mul(add(2, 24), add(3, 5))
```

```
mul(26, add(3, 5))
```



- Изменяющиеся операции нарушают условие *прозрачности для ссылок* поскольку они не только возвращают результат; **они изменяют окружение.**