

От Pyret к Python

Данил Браун

Весна 2024

Pyret → Python

Зачем?

Зачем?

- ▶ Увидев то, как одни и те же идеи устроены в разных языках, мы лучше понимаем эти идеи (выражения, функции, ...). *То же верно и для естественных языков.*

Зачем?

- ▶ Увидев то, как одни и те же идеи устроены в разных языках, мы лучше понимаем эти идеи (выражения, функции, ...). *То же верно и для естественных языков.*
- ▶ На вашем дальнейшем образовательном пути скорее всего будет какое-то подобие курса по Computer Science, с большой вероятностью там будет программирование, и это, вероятно, будет Python. *На данный момент это, пожалуй, самый популярный язык общего назначения.*

Зачем?

- ▶ Увидев то, как одни и те же идеи устроены в разных языках, мы лучше понимаем эти идеи (выражения, функции, ...). *То же верно и для естественных языков.*
- ▶ На вашем дальнейшем образовательном пути скорее всего будет какое-то подобие курса по Computer Science, с большой вероятностью там будет программирование, и это, вероятно, будет Python. *На данный момент это, пожалуй, самый популярный язык общего назначения.*
- ▶ Если вы в дальнейшем будете профессионально программировать, то вам придётся использовать *разные* языки для *разных* задач.

Зачем?

- ▶ Увидев то, как одни и те же идеи устроены в разных языках, мы лучше понимаем эти идеи (выражения, функции, ...). *То же верно и для естественных языков.*
- ▶ На вашем дальнейшем образовательном пути скорее всего будет какое-то подобие курса по Computer Science, с большой вероятностью там будет программирование, и это, вероятно, будет Python. *На данный момент это, пожалуй, самый популярный язык общего назначения.*
- ▶ Если вы в дальнейшем будете профессионально программировать, то вам придётся использовать *разные* языки для *разных* задач.
- ▶ Pyret создавался с прицелом на то, что в дальнейшем будет произведён переход на Python. *Отсюда схожесть названий и синтаксиса.*

Выражения, функции и типы

Рассмотрим пример функции, вычисляющей стоимость изготовления ручек с принтами. Сравним реализации на Pyret и Python.

Найдите 7 отличий (:

Pyret:

```
fun pen-cost(num-pens :: Number, message ::  
  String) -> Number:  
  doc: ``total cost for pens, each 25 cents  
        plus 2 cents per message character  
        ``  
  
  num-pens * (0.25 + (string-length(message) *  
    0.02))  
  
end
```

Python:

```
def pen_cost(num_pens: int, message: str) -> float:  
    """total cost for pens, each at 25 cents plus  
        2 cents per message character"""  
    return num_pens * (0.25 + (len(message) * 0.02))
```


Мы будем пользоваться средой разработки Thonny. Найдите её на рабочем столе или через меню Пуск и запустите. В ней есть (так же как и в code.pyret.org) два окна: верхнее это текстовый редактор (окно определений), нижнее — интерактивное окно (REPL).

Упражнение 1

Упражнение 1. Перепишите данную функцию на Python. Убедитесь в том, что она работает.

```
fun moon-weight(earth-weight :: Number) ->  
    Number:  
    doc: ```Compute weight on moon from weight  
           on earth```  
    earth-weight * 1/6  
end
```

Возврат значений из функций

- ▶ В Pyret в теле функции есть одно или несколько выражений, и результат функции это всегда результат последнего выражения

Возврат значений из функций

- ▶ В Pyret в теле функции есть одно или несколько выражений, и результат функции это всегда результат последнего выражения
- ▶ В Python нужно специально пометить, что должна вернуть функция, потому что не все функции здесь возвращают результат

Возврат значений из функций

- ▶ В Pyret в теле функции есть одно или несколько выражений, и результат функции это всегда результат последнего выражения
- ▶ В Python нужно специально пометить, что должна вернуть функция, потому что не все функции здесь возвращают результат
- ▶ Результат функции это не обязательно последнее выражение

Возврат значений из функций

- ▶ В Pyret в теле функции есть одно или несколько выражений, и результат функции это всегда результат последнего выражения
- ▶ В Python нужно специально пометить, что должна вернуть функция, потому что не все функции здесь возвращают результат
- ▶ Результат функции это не обязательно последнее выражение
- ▶ Возвращаемый результат помечается словом **return**

Пример двух функций

Запишите определения этих двух функций в редакторе и проверьте, как они работают, в интерактивном окне.

```
def add1v1(x: int) -> int:  
    return x + 1
```

```
def add1v2(x: int) -> int:  
    x + 1
```

Пример двух функций

Запишите определения этих двух функций в редакторе и проверьте, как они работают, в интерактивном окне.

```
def add1v1(x: int) -> int:  
    return x + 1
```

```
def add1v2(x: int) -> int:  
    x + 1
```

Попробуйте выполнить два таких выражения:

```
3 * add1v1(4)
```

```
3 * add1v2(4)
```


Пример двух функций

Запишите определения этих двух функций в редакторе и проверьте, как они работают, в интерактивном окне.

```
def add1v1(x: int) -> int:  
    return x + 1
```

```
def add1v2(x: int) -> int:  
    x + 1
```

Попробуйте выполнить два таких выражения:

```
3 * add1v1(4)
```

```
3 * add1v2(4)
```

Зачем тогда вообще определять функции без **return**? Позже мы это узнаем.

- ▶ В Pyret результаты вычислений выводились (т.е. печатались) автоматически, а ввод мы указывали, передавая значения нужным функциям

Ввод и вывод данных

- ▶ В Pyret результаты вычислений выводились (т.е. печатались) автоматически, а ввод мы указывали, передавая значения нужным функциям
- ▶ Для ввода и вывода (печати) данных в Python существуют специальные функции: `input` и `print`. Примеры.

Ввод и вывод данных

- ▶ В Pyret результаты вычислений выводились (т.е. печатались) автоматически, а ввод мы указывали, передавая значения нужным функциям
- ▶ Для ввода и вывода (печати) данных в Python существуют специальные функции: `input` и `print`. Примеры.
- ▶ Считанное значение всегда сохраняется в виде строки. Если нам нужно получить из строки число, мы можем использовать функцию `int` или `float` для нецелых чисел.

Зайдите на pythontutor.ru, зарегистрируйтесь, и выполните первое задание *Сумма трёх чисел*.