Типизация в Питоне

Как с этим работать?

Типизация

- Первые упоминания о подсказках типов в языке программирования Python появились в базе Python Enhancement Proposals (PEP-483). Такие подсказки нужны для улучшения статического анализа кода и автодополнения редакторами, что помогает снизить риски появления багов в коде.

Типизация

- Для обозначения базовых типов переменных используются сами типы:

1. Str
2. Int
3. Float
4. Bool
5. Complex
6. Bytes
7. etc

Типизация

- Пример использования базовых типов в python-функции:

```python
def func(a: int, b: float) -> str:
    a: str = f"{a}, {b}"
    return a
    # return a * b

example = func(5, 4.5)
print(example)
```

Типизация

- Ну либо такой пример:

main.py

```
1  def func(a: int, b: float):
2      # a: str = f"{a}, {b}"
3      # return a
4      return a * b
5
6  example = func(5, 4.5)
7  print(example)
```

Типизация

- Помимо этого, можно параметризировать более сложные типы, например, List. Такие типы могут принимать значения параметров, которые помогают более точно описать тип функции. Так, например, List[int] указывает на то, что список состоит только из целочисленных значений.

Типизация

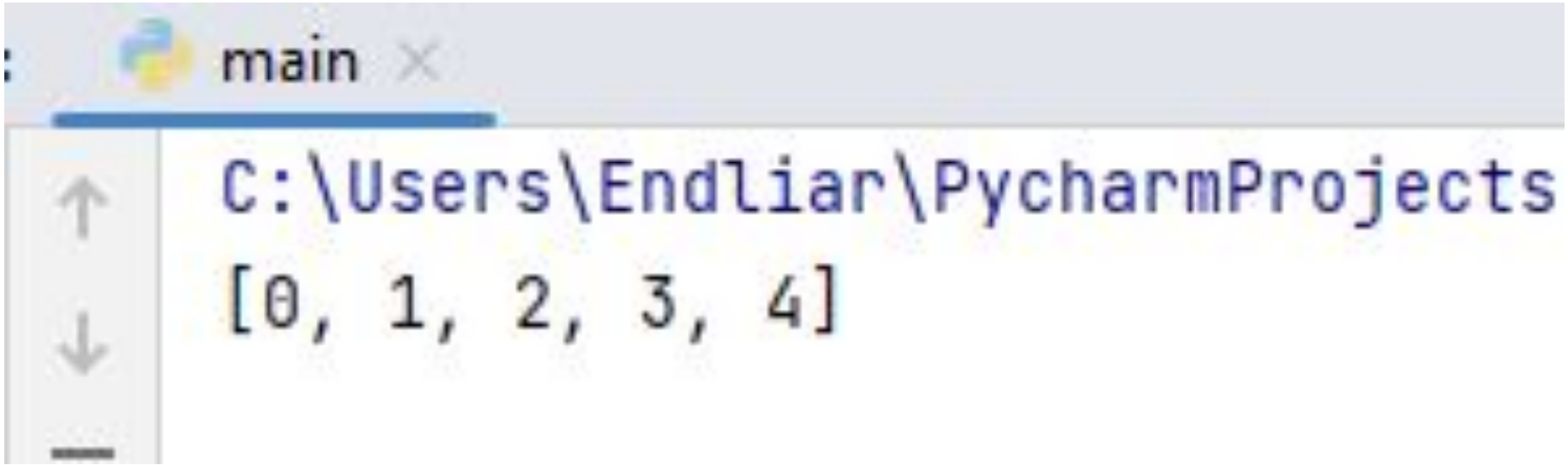
- Пример кода:

main.py

```
1  from typing import List
2
3  def func(n: int) -> List[int]:
4      return list(range(n))
5
6  example = func(5)
7  print(example)
```

Типизация

- Вывод:



Типизация

- Кроме List, существуют и другие типы из модуля typing, которые можно параметризировать. Такие типы называются Generic-типами. Подробнее о них можете погуглить, и посмотреть по официальной (или не очень) документации. Их вполне приемлемое количество на разный цвет и вкус.

Типизация

- При этом функции тоже имеют свои типы. Например, для описания функции можно использовать тип Callable, где указываются типы входных параметров и возвращаемых значений. Пример использования представлен ниже:

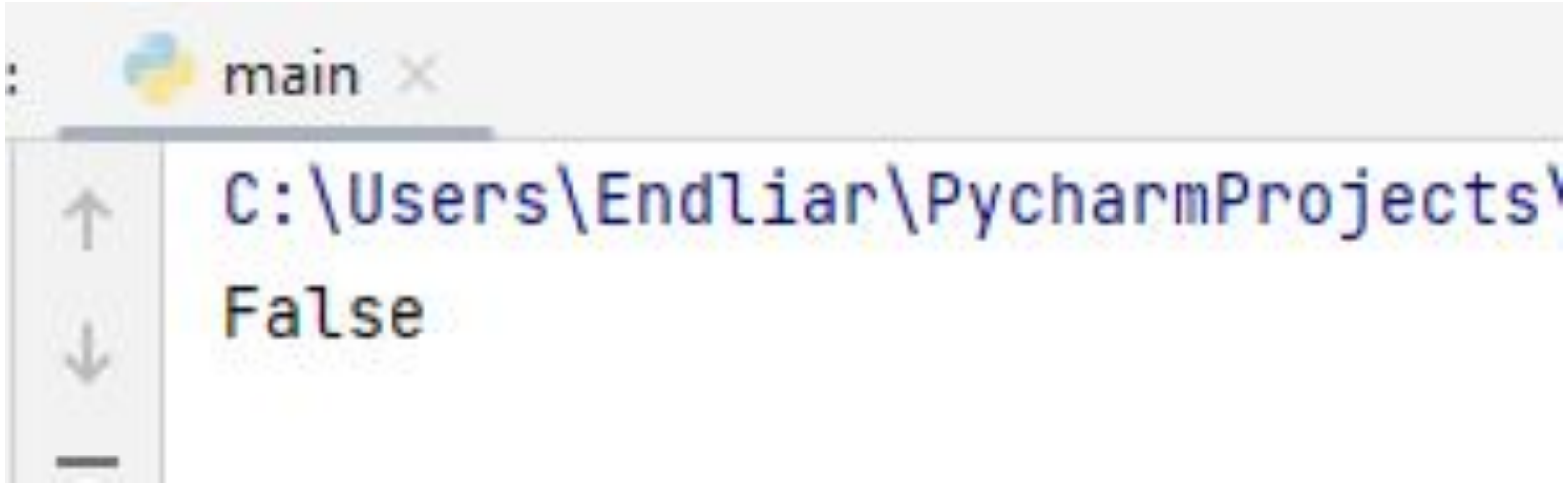
Типизация

- Пример:

```
3
4   from typing import Callable
5
6   def func(f: Callable[[int, int], bool]) -> bool:
7       return f(5, 6)
8
9   example = func(lambda x, y: x == y)
10  print(example)
```

Типизация

- Вывод:



Lambda выражение в Python

- lambda оператор или lambda функция в Python это способ создать анонимную функцию, то есть функцию без имени. Такие функции можно назвать одноразовыми, они используются только при создании. Как правило, lambda функции используются в комбинации с функциями filter, map, reduce.

Типизация

- Тип Callable говорит о том, что:

1. У объекта реализован метод __call__.
2. Описывает типы параметров этому методу.

- На первом месте стоит массив типов входных параметров, на втором — тип возвращаемого значения.
- Про остальные абстрактные типы контейнеров можно прочитать в документации Python.

Основной смысл

- Цель — указать разработчику на ожидаемый тип данных при получении или возврате данных из функции или метода. В свою очередь, это позволяет сократить количество багов, ускорить написание кода и улучшить его качество.

Основной смысл