

Python

Курс “Python Backend”

16.02.2023

Стас Клоков

Базовые типы

- Числовые типы: (bool, int, float, complex)
- Строки: (str)
- Базовые коллекции: (list, tuple, set, dict)

Числовой тип

- `bool` - числовой подтип, которые является по своей сути адаптером обычного числового типа для ограничения скоупа значений набором из двух возможных: `True` и `False` (с большой буквы)
- `int` - целочисленный тип с реализованной длинной арифметикой
- `float` - тип с плавающей точкой, имеющий ограничения, о которых можно узнать, с помощью метода `sys.float_info`
- `complex` - числовой тип для комплексного исчисления

Строковый тип

- Неизменяемый (immutable)
- Для работы со строками есть очень удобный инструмент - срезы

Встроенные коллекции

- list - список элементов (изменяемый)
- tuple - кортеж элементов (неизменяемый)
- set - множество элементов (неупорядоченное)
- dict - словарь или ассоциативный контейнер для работы с объектами вида {[key]: value}
- set и dict поддерживают операции вставки, поиска удаления над элементами, которые выполняются за константную асимптотическую оценку

Модули и пакеты

Самое важное, что стоит знать здесь - это, что пакет - упакованная внешняя сущность, готовая к переиспользованию и управлению пакетным менеджером. А модуль - внутренняя композиция функционала на этапе работы над структурой проекта.

Итераторы и генераторы

Iterator - объект, поддерживающий функцию `next()` для взятия следующего.

Generator - iterator, элементы которого можно пройти только однажды.

Итерируемый объект - объект элементы которого можно обойти и его можно преобразовать к итератору.

Декораторы

Декоратор - вспомогательная сущность, являющаяся оберткой над исполняемым разделом кода.

Контекстный менеджер

Контекстный менеджер - менеджер работы над разделом скоупа, в котором создается контекст работы над результатов исполнения какой-либо “зависимой” операции.

Лямбда функции

Лямбды - функции переменные. Для нас это по своей сути встроенное средство для сокращения записи работы с функциями высших порядков.

Классы и функции

- Важно знать, что нет явных модификаторов доступа у классов
- Важно уметь работать

Магические методы

Магические методы позволяют работать с удобными внешними интерфейсами объектов классов.

Аннотации типов

С недавнего времени в питон приехала типизация. Типизация в питоне как была динамической, так и остается. Это означает, что мы все еще не даем никаких сильных гарантий, добавляя аннотацию, но при этом мы пытаемся сообщить, что мы ожидаем в той или иной переменной.

Библиотека funcy

Просто очень удобная штука))

Ссылочка, где можно про это почитать

<https://pypi.org/project/funcy/>

Асинхронные воркеры

- Celery (внешний инструмент для исполнения асинхронных задач)
- Cron (внешний инструмент для исполнения регулярных задач)

Немного инфры

- venv
- Makefile
- requirements.txt
- modules