Типизированный Python (модуль typing)

№ урока: 6 **Курс:** Python Advanced

Средства обучения: Python3.6, PyCharm

Обзор, цель и назначение урока

Изучить возможности использования типизации в Python. Получить навыки использования модуля typing. Использовать библиотеку «mypy» для проверки программ, использующих типизацию.

Изучив материал данного занятия, учащийся сможет:

- Создавать типизированные переменные и функции
- Использовать типизацию для подсказок IDE PyCharm
- Использовать модуль typing
- Использовать модуль mypy для проверки созданных типизированных модулей

Содержание урока

1. Статическая и динамическая типизация.
2. Плюсы и минусы статической типизации.
3. Типизация в Python 2 и 3.
4. Модуль typing.
5. Установка и использование модуля mypy.

Резюме

- Язык Python является языком с динамической типизацией. Мы можем создавать любое количество переменных, без указания типов. А определение типа переменной происходит во время выполнения программы. Данный вид типизации позволяет очень быстро разрабатывать программы и модули на языке Python, не заботясь о многих особенностях статической типизации.
- Начиная с версии Python 3, появляется возможность использовать аннотацию типов для переменных и параметров функций. А также создавать достаточно сложные типы данных. Для этих целей в стандартной библиотеке языка существует библиотека typing, позволяющая комбинировать примитивы в достаточно сложные типы данных (массивы, словари, последовательности, итераторы с указанием типов содержимого, объединения типов и много другое).
- Язык Python не будет уведомлять нас о возможных несовпадениях типов. Такого рода функционал реализован сторонними библиотеками, например, mypy. Данная библиотека устанавливается средствами пакетного менеджера pip и используется как обычный интепретатор Python. Мы запускаем python скрипт используя команду mypy. В случае возникновения ошибок или несоответствий, mypy сгенерирует отчет и выведет в консоль сообщения с указанием номеров строк с ошибками.
- Используя аннотацию типов, вы получите дополнительную помощь при использовании IDE. В случае явного указания типов, IDE будет автоматически показывать возможные ошибки или недочеты при использовании переменных. Например, использование неизвестных методов для типа str/int/dict/etc.

CyberBionic systematics

CyberBionic Systematics ® 2019
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Закрепление материала

- Что такое статическая и динамическая типизация?
- Какой вид типизации использует язык Python?
- Начиная с какой версии язык Python поддерживает аннотацию типов, как часть синтаксиса?
- Какой модуль из стандартной библиотеки Python позволяет комбинировать различные типы?
- Как создать тип списка с элементами типа int используя модуль typing?
- Как установить библиотеку mypy?
- В случае возникновения несоответствий типов, как будет вести себя интерпретатор Python и в чем отличие поведения от mypy?

Дополнительное задание

Задание
Создайте два класса Directory (папка) и File (файл).

Класс Directory должен иметь следующие поля:
- название (name типа str);
- родительская папка (root типа Directory);
- список файлов (files типа список, состоящий из экземпляров File)
- список подпапок (sub_directories типа список, состоящий из экземпляров Directory).

Класс Directory должен иметь следующие поля:
- добавление папки в список подпапок (**add_sub_directory** принимающий экземпляр **Directory** и присваивающий поле **root** для принимаемого экземпляра);
- удаление папки из списка подпапок (**remove_sub_directory**, принимающий экземпляр Directory и обнуляющий у него поле **root**. Метод также удаляет папку из списка **sub_directories**).
- добавление файла в папку (**add_file**, принимающий экземпляр **File** и присваивающий ему поле **directory** - см. класс File ниже);
- удаление файла из папки (**remove_file**, принимающий экземпляр **File** и обнуляющий у него поле **directory**. Метод удаляет файл из списка **files**).

Класс File должен иметь следующие поля:
- название (name типа str);
- папка (directory типа Directory);

Самостоятельная деятельность учащегося

Задание 1
Создать функцию, которая принимает список из элементов типа int, а возвращает новый список из строковых значений исходного массива. Добавить аннотацию типов для входных и результирующих значений функции.

Рекомендуемые ресурсы

Официальный сайт Python (3.6) - typing
https://docs.python.org/3.6/library/typing.html
Официальный сайт пакета mypy
https://mypy.readthedocs.io/en/latest/

CyberBionic systematics
CyberBionic Systematics ® 2019
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com