Последовательности

№ урока:	5	**Курс:**	Python Essential

Средства обучения:	Python 3; интегрированная среда разработки (PyCharm или Microsoft Visual Studio + Python Tools for Visual Studio)

Обзор, цель и назначение урока

После завершения урока обучающиеся будут иметь представление о последовательностях в Python и основных стандартных последовательностях, их назначении и использовании, смогут реализовывать собственные классы последовательностей.

Изучив материал данного занятия, учащийся сможет:

- Иметь представление о последовательностях
- Определять, какие контейнеры в Python считаются последовательностями и создавать свои последовательности
- Знать основные операции, общие для всех последовательностей и изменяемых последовательностей
- Работать со списками
- Работать с кортежами
- Работать с диапазонами
- Работать со строками
- Лексикографически сравнивать последовательности

Содержание урока

1. Понятие последовательности
2. Операции с последовательностями
3. Списки
4. Кортежи
5. Диапазоны
6. Строки
7. Сравнение последовательностей
8. Функции с произвольным количеством позиционных параметров, распаковка позиционных параметров функций из последовательностей

Резюме

- *Последовательностью* в Python называется итерабельный объект, который поддерживает эффективный доступ к элементам с использованием целочисленных индексов через специальный метод __getitem__() и поддерживает метод __len__(), который возвращает длину последовательности. К основным встроенным типам последовательностей относятся list, tuple, range, str и bytes.
- Последовательности также опционально могут реализовывать методы count(), index(), __contains__() и __reversed__() и другие.
- Операции, которые поддерживаются большинством последовательностей:
    - x in s, x not in s – находится ли элемент x в последовательности s (для строк и последовательностей байтов – является ли x подстрокой s)
    - s + t – конкатенация последовательностей
    - s * n, n * s – конкатенация n нерекурсивных копий последовательности s
    - s[i] – i-й элемент последовательности s
    - s[i:j] – срез последовательности s от i до j
    - s[i:j:k] – срез последовательности s от i до j с шагом k

CyberBionic systematics
CyberBionic Systematics ® 2015
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8473
E-mail: edu@cbsystematics.com
Site:www.edu.cbsystematics.com

    - len(s) – длина последовательности
    - min(s) – минимальный элемент последовательности
    - max(s) – максимальный элемент последовательности
    - s.index(x[, i[, j]]) – индекс первого вхождения x (опционально – начиная с позиции i и до позиции j)
    - s.count(x) – общее количество вхождений x в s
    - sum(s) – сумма элементов последовательности
- Неизменяемые последовательности обычно реализуют операцию hash(s) – хеш-значение объекта.
- Большинство изменяемых последовательностей поддерживают следующие операции:
    - s[i] = x – элемент с индексом i заменяется на x
    - s[i:j] = t, s[i:j:k] = t – элементы с индексами от i до j (с шагом k) заменяются содержимым итерабельного объекта t
    - del s[i:j], del s[i:j:k] – удаление соответствующих элементов из последовательности
    - s.append(x) – добавление x в конец последовательности
    - s.clear() – удаление всех элементов последовательности
    - s.copy() – нерекурсивная копия последовательности
    - s.extend(t) – добавление всех элементов итерабельного объекта в конец последовательности
    - s.insert(i, x) – вставка элемента x по индексу i
    - s.pop(), s.pop(i) – возврат значения по индексу i (по умолчанию – последний элемент) и удаление его из последовательности
    - s.remove(x) – удаление первого вхождения x
    - s.reverse() – разворот последовательности в обратном порядке
- *Списки* – это изменяемые последовательности, обычно используемые для хранения однотипных данных (хотя Python не запрещает хранить в них данные разных типов). Представлены классом list.
- Создание списков:
    - перечисление элементов в квадратных скобках;
    - использование списковых включений (тот же синтаксис, что и у рассмотренных ранее выражений-генераторов, но обрамляется квадратными скобками);
    - использование конструктора типа.
- Поддерживают все общие для всех и изменяемых последовательностей операции, и реализуют один дополнительный метод: sort(*, key=None, reverse=None) -– сортирует список при помощи операции “<”. Опциональный параметр key – функция от одного аргумента, которая извлекает ключ для сортировки, reverse – сортировка в обратном порядке, если он равен True.
- *Кортежи* – это неизменяемые последовательности, обычно используемые, чтобы хранить разнотипные данные. Представлены классом tuple.
- Создание кортежей:
    - () – пустой кортеж;
    - перечисление элементов через запятую (если элемент один, после него всё равно должна стоять запятая);
    - использование конструктора типа.
- Поддерживают все общие для последовательностей операции.
- *Диапазоны* – неизменяемые последовательности чисел, которые задаются началом, концом и шагом. Представлены классом range (в Python 2 – xrange; range в Python 2 – это функция, которая возвращает список).
- Начало по умолчанию равно нулю, шаг – единице. Если задать нулевой шаг, будет выброшено исключение ValueError.
- Параметры конструктора должны быть целыми числами (либо экземпляры класса int, либо любой объект с методом __index__).
- Элементы диапазона r определяются по формуле r[i] = start + step * i, где i >= 0 и r[i] < stop для step > 0 или r[i] > stop для step < 0.
- Поддерживает все общие для последовательностей операции, кроме конкатенации и повторения, а также, в версиях Python до 3.2, срезов и отрицательных индексов.

CyberBionic systematics
CyberBionic Systematics ® 2015
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8473
E-mail: edu@cbsystematics.com
Site:www.edu.cbsystematics.com

- *Строки* – неизменяемые последовательности кодов символов (в Python 3 – в кодировке Unicode, в Python 2 – в ASCII). Представлены классом str. В Python 2 также есть класс unicode, который представляет Unicode-строки подобно str в Python 3.
- Строковые литералы выделяются одинарными или двойными кавычками. Можно использовать утроенные кавычки для создания многострочных строк. Если перед строковым литералом стоит префикс r, то большинство escape-последовательностей игнорируются. В Python 2 префикс u задаёт Unicode-литерал.
- Если между двумя строковыми литералами нет ничего, кроме пробелов и переносов строк, они интерпретируются как один литерал.
- Строки поддерживают все общие для последовательностей операции, а также реализуют огромное количество собственных методов, которые описаны в документации: https://docs.python.org/3/library/stdtypes.html#string-methods
- Есть также две встроенные функции, которые работают со строками, состоящими из одного символа. Функция ord(char) возвращает код символа char, а функция chr(code) возвращает символ с кодом code.
- Две последовательности равны, если они имеют одинаковый тип, равную длину и соответствующие элементы обоих последовательностей равны.
- Последовательности одинаковых типов можно сравнивать. Сравнения происходят в лексикографическом порядке: последовательность меньшей длины меньше, чем последовательность большей длины, если же их длины равны, то результат сравнения равен результату сравнения первых отличающихся элементов.
- Функция может иметь произвольное количество аргументов. После всех позиционных параметров функции или вместо них (но перед теми, которые предполагается использовать как именованные) в её сигнатуре можно указать специальный аргумент с символом * перед именем. Тогда оставшиеся фактические параметры сохраняются в кортеже с этим именем.
- Также существует и обратная возможность. Если при вызове функции перед именем итерабельного объекта поставить символ *, то его элементы распаковываются в позиционные аргументы.

Закрепление материала

- Что такое последовательность?
- Какие методы обязана реализовать любая последовательность?
- Какие методы реализуют большинство последовательностей?
- Какие методы реализуют большинство изменяемых последовательностей?
- Какие вы знаете встроенные неизменяемые последовательности в Python?
- Какие вы знаете встроенные изменяемые последовательности в Python?
- Каким образом можно сравнивать последовательности?
- Как реализуется передача произвольного количества аргументов функции в Python?

Дополнительное задание

Задание
Напишите программу, которая вводит с клавиатуры последовательность чисел и выводит её отсортированной в порядке возрастания.

Самостоятельная деятельность учащегося

Задание 1
Создайте функцию от произвольного количества аргументов, которая вычисляет среднее арифметическое данных чисел. Вычислите при помощи неё среднее арифметическое двух заданных чисел и среднее арифметическое чисел из заданного диапазона.

Задание 2
Используя документацию, ознакомьтесь с методами класса str.

CyberBionic systematics
CyberBionic Systematics ® 2015
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8473
E-mail: edu@cbsystematics.com
Site:www.edu.cbsystematics.com

Задание 3
Напишите программу, которая вводит с клавиатуры текст и выводит отсортированные по алфавиту слова данного текста.
Задание 4
Ознакомьтесь при помощи документации с классами namedtuple и deque модуля collections.

Рекомендуемые ресурсы

Документация Python
https://docs.python.org/3/glossary.html#term-sequence
https://docs.python.org/3/tutorial/datastructures.html
https://docs.python.org/3/library/stdtypes.html#sequence-types-list-tuple-range
https://docs.python.org/3/library/stdtypes.html#common-sequence-operations
https://docs.python.org/3/library/stdtypes.html#immutable-sequence-types
https://docs.python.org/3/library/stdtypes.html#mutable-sequence-types
https://docs.python.org/3/library/stdtypes.html#lists
https://docs.python.org/3/library/stdtypes.html#tuples
https://docs.python.org/3/library/stdtypes.html#ranges
https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str
https://docs.python.org/3/library/stdtypes.html#string-methods
https://docs.python.org/3/tutorial/controlflow.html#arbitrary-argument-lists
https://docs.python.org/3/tutorial/controlflow.html#unpacking-argument-lists
https://docs.python.org/3/library/collections.html

Статьи в Википедии о ключевых понятиях, рассмотренных на этом уроке
https://ru.wikipedia.org/wiki/Список_(информатика)
https://ru.wikipedia.org/wiki/Строковый_тип
https://ru.wikipedia.org/wiki/Списковое_включение

CyberBionic systematics
CyberBionic Systematics ® 2015
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8473
E-mail: edu@cbsystematics.com
Site:www.edu.cbsystematics.com