Множества и отображения

№ урока:	6	**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 множествах и отображениях в Python и основных стандартных классах множеств и отображений, их назначении и использовании.

Изучив материал данного занятия, учащийся сможет:

- Иметь представление о множествах
- Использовать классы set и frozenset
- Иметь представление об отображениях и словарях
- Использовать класс dict и другие классы из модуля collections
- Использовать представления словарей
- Создавать функции с произвольным количеством именованных параметров
- Распаковывать словари и другие отображения в именованные параметры функции

Содержание урока

1. Множества
2. Изменяемые и неизменяемые множества
3. Создание множеств
4. Операции с множествами
5. Отображения
6. Произвольное количество именованных параметров функции, распаковка именованных аргументов из отображения
7. Словари (ассоциативные массивы)
8. Создание словарей
9. Операции со словарями
10. Представления словарей

Резюме

- Объект называется *хешируемым*, если он имеет хеш-значение (целое число), которое никогда не изменяется на протяжении его жизненного цикла и возвращается методом __hash__(), и может сравниваться с другими объектами (реализует метод __eq__()). Равные хешируемые объекты должны иметь равные хеш-значения.
- Хешируемые объекты могут быть использованы как ключи словарей и члены множеств.
- Все стандартные неизменяемые объекты хешируемые. Все стандартные изменяемые объекты не хешируемые.
- *Множество* – это неупорядоченная коллекция хешируемых объектов, которые не повторяются.
- Обычно используются для проверки элемента на вхождение в множество и удаление повторений элементов и выполнения таких операций, как объединение, пересечение, разница и симметрическая разница.
- В множествах нет понятия позиции элемента. Соответственно, они не поддерживают индексацию и срезы.
- Встроенные классы множеств: set (изменяемое множество), frozenset (неизменяемое множество).
- Создание множества:
    - перечисление элементов в фигурных скобках (только set);

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

    - использование конструктора типа;
    - включение множеств (аналогично списковым включениям, только set).
- Для создания пустого множества нужно использовать конструктор типа без параметров, так как пустые фигурные скобки создают пустой словарь.
- Операции с множествами:
    - set([iterable])
      frozenset([iterable]) – создание множества (пустого или из элементов итерабельного объекта)
    - len(s) – количество элементов множества
    - x in s
      x not in s – проверка нахождения элемента в множестве
    - s.isdisjoint(t) – проверка того, что данное множество не имеет общих элементов с заданным
    - s.issubset(t)
      s <= t – проверка того, что все элементы множества s являются элементами множества t
    - s < t – проверка того, что s <= t и s != t
    - s.isuperset(t)
      s >= t – проверка того, что все элементы множества t являются элементами множества s
    - s > t – проверка того, что s >= t и s != t
    - s.union(t, …)
      s | t | … – создание нового множества, которое является объединением данных множеств
    - s.intersection(t, …)
      s & t & … – создание нового множества, которое является пересечением данных множеств
    - s.difference(t, …)
      s - t - … – создание нового множества, которое является разницей данных множеств
    - s.symmetric_difference(t)
      s ^ t – создание нового множества, которое является симметрической разницей данных множеств (то есть, разница объединения и пересечения множеств)
    - s.copy() – неполная копия множества s
- Операции над множествами, которые являются методами, принимают в качестве аргументов любые итерабельные объекты. Операции над множествами, записанные в виде бинарных операций, требуют, чтобы второй операнд операции тоже был множеством, и возвращают множество того типа, которым было первое множество.
- Операции над изменяемыми множествами:
    - s.update(t, …)
      s |= t | … – добавить в данное множество элементы из других множеств
    - s.intersection_update(t, …)
      s &= t & … – оставить в данном множестве только те элементы, которые есть и в других множествах
    - s.difference_update(t, …)
      s -= t | … – удалить из данного множества те элементы, которые есть в других множествах
    - s.symmetric_difference_update(t)
      s ^= t – оставить или добавить в s элементы, которые есть либо в s, либо в t, но не в обоих множествах
    - s.add(element) – добавить новый элемент в множество
    - s.remove(element) – удалить элемент из множества; если такого элемента нет, возникает исключение KeyError
    - s.discard(element) – удалить элемент из множества, если он в нём находится
    - s.pop() – удалить из множества и вернуть произвольный элемент; если множество пустое, возникает исключение KeyError
    - s.clear() – удалить все элементы множества.
- Проверка множеств на равенство происходит поэлементно, независимо от типов множеств.

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

- *Отображение (mapping)* – это объект-контейнер, который поддерживает произвольный доступ к элементам по ключам и описывает все методы, описанные в абстрактном базовом классе collections.Mapping (get(), items(), keys(), values()) или collections.MutableMapping (clear(), get(), items(), keys(), pop(), popitem(), setdefault(), update(), values()). К отображениям относятся классы dict, collections.defaultdict, collections.OrderedDict и collections.Counter.
- Встроенным классом отображения является dict, который реализует такую структуру данных, как словарь, или ассоциативный массив, то есть неупорядоченная изменяемая коллекция пар (ключ, значение), которая поддерживает произвольный доступ к её элементам по их ключам.
- Ключи словарей должны быть хешируемыми значениями.
- Числовые ключи в словарях подчиняются правилам сравнения чисел. Таким образом, int(1) и float(1.0) считаются одинаковым ключом. Однако из-за того, что значения типа float сохраняются приближенно, не рекомендуется использовать их в качестве ключей.
- Подобно тому, как можно передавать в функции произвольное количество позиционных аргументов, которые сохраняются в кортеже, можно передавать произвольное количество именованных аргументов, которые сохраняются в словаре. Для этого перед именем данного словаря в списке формальных параметров ставится два символа **. Если используются оба способа передачи произвольного количества аргументов, параметр в форме **kwargs в сигнатуре функции должен идти после параметра в форме *args.
- Аналогично можно и распаковывать любые отображения в именованные параметры при вызове функции.
- Создание словарей:
  - перечисление пар ключ-значение, разделённых символом двоеточия, через запятые в фигурных скобках;
  - включения словарей (аналогично списковым включениям);
  - использование конструктора класса dict.
- Вызов конструктора:
  - dict(**kwargs)
  - dict(mapping, **kwargs)
  - dict(iterable, **kwargs)
- Если не передан никакой позиционный аргумент, создаётся пустой словарь. Если позиционный аргумент является отображением, то создаётся словарь с теми же самыми парами ключей и значений. Иначе позиционный аргумент должен быть итерабельным объектом, элементами которого являются итерабельные объекты, состоящие из двух элементов, где первый соответствует ключу, а второй – значению. Если одному ключу соответствует несколько значений, то последнее замещает все предыдущие.
- Если переданы именованные аргументы, то эти аргументы с их значениями добавляются в созданный из позиционного аргумента словарь. Если один из ключей уже существует, то соответствующее значение именованного аргумента замещает то, что было получено из позиционного.
- Для того, чтобы передавать ключи как именованные аргументы, они должны быть допустимыми идентификаторами.
- Операции со словарями и другими отображениями:
  - len(d) – количество элементов.
  - d[key] – получение значения с ключом key. Если такой ключ не существует и отображение реализует специальный метод __missing__(self, key), то он вызывается. Если ключ не существует и метод __missing__ не определён, выбрасывается исключение KeyError.
  - d[key] = value – изменить значение или создать новую пару ключ-значение, если ключ не существует.
  - key in d
    key not in d – проверка наличия ключа в отображении.
  - iter(d) – то же самое, что iter(d.keys()).
  - clear() – удалить все элементы словаря.
  - copy() – создать неполную копию словаря.
  - (метод класса) dict.fromkeys(sequence[, value]) – создаёт новый словарь с ключами из последовательности sequence и заданным значением (по умолчанию – None).

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

    - d.get(key[, default]) – безопасное получение значения по ключу (никогда не выбрасывает KeyError). Если ключ не найден, возвращается значение default (по-умолчанию – None).
    - d.items() – в Python 3 возвращает объект представления словаря, соответствующий парам (двухэлементным кортежам) вида (ключ, значение). В Python 2 возвращает соответствующий список, а метод iteritems() возвращает итератор. Аналогичный метод в Python 2.7 – viewitems().
    - d.keys() – в Python 3 возвращает объект представления словаря, соответствующий ключам словаря. В Python 2 возвращает соответствующий список, а метод iterkeys() возвращает итератор. Аналогичный метод в Python 2.7 – viewkeys().
    - d.pop(key[, default]) – если ключ key существует, удаляет элемент из словаря и возвращает его значение. Если ключ не существует и задано значение default, возвращается данное значение, иначе выбрасывается исключение KeyError.
    - d.popitem() – удаляет произвольную пару ключ-значение и возвращает её. Если словарь пустой, возникает исключение KeyError. Метод полезен для алгоритмов, которые обходят словарь, удаляя уже обработанные значения (например, определённые алгоритмы, связанные с теорией графов).
    - d.setdefault(key[, default]) – если ключ key существует, возвращает соответствующее значение. Иначе создаёт элемент с ключом key и значением default. default по умолчанию равен None.
    - d.update(mapping) – принимает либо другой словарь или отображение, либо итерабельный объект, состоящий из итерабельных объектов – пар ключ-значение, либо именованные аргументы. Добавляет соответствующие элементы в словарь, перезаписывая элементы с существующими ключами.
    - d.values() – в Python 3 возвращает объект представления словаря, соответствующий значениям. В Python 2 возвращает соответствующий список, а метод itervalues() возвращает итератор. Аналогичный метод в Python 2.7 – viewvalues().
- Объекты, возвращаемые методами items(), keys() и values() (viewitems(), viewkeys(), viewvalues() в Python 2.7) – это объекты *представления словаря*. Они предоставляют динамическое представление элементов словаря, то есть изменения данного словаря автоматически отображаются и на этих объектах.
- Операции с представлениями словарей:
    - iter(dictview) – получение итератора по ключам, значениям или парам ключей и значений. Все представления словарей при итерировании возвращают элементы словаря в одинаковом порядке. При попытке изменить словарь во время итерирования может возникнуть исключение RuntimeError.
    - len(dictview) – количество элементов в словаре.
    - x in dictview – проверка существования ключа, значения или пары ключ-значение в словаре.
- В модуле collections также определены другие классы отображений, одним из которых является Counter, реализующий понятие мультимножества – множества, каждый из элементов которого может повторяться несколько раз. Эта структура данных часто используется для эффективного подсчёта количества определённых значений (что отображено в названии класса: «counter» – «счётчик»).

Закрепление материала

- Что такое хешируемый объект?
- Может ли изменяемый объект быть хешируемым?
- Что такое множество?
- Назовите основные способы создания множеств (set и frozenset).
- Учитываются ли типы множеств при их сравнении?
- Чем отличаются операции над множествами, реализованные в виде отдельных методов и путём перегрузки бинарных операций?
- Что такое отображение?
- Что такое словарь?

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

- Назовите основные способы создания словарей.
- Что такое объект представления словаря? Чем отличаются методы items() в Python 3, items() в Python 2, iteritems() и viewitems()?

Дополнительное задание

Задание
Создайте словарь с ключами-строками и значениями-числами. Создайте функцию, которая принимает произвольное количество именованных параметров. Вызовите её с созданным словарём и явно указывая параметры.

Самостоятельная деятельность учащегося

Задание 1
Даны две строки. Выведите на экран символы, которые есть в обоих строках.

Задание 2
Создайте программу, которая эмулирует работу сервиса по сокращению ссылок. Должна быть реализована возможность ввода изначальной ссылки и короткого названия и получения изначальной ссылки по её названию.

Задание 3
Ознакомьтесь при помощи документации с классами OrderedDict, defaultdict и ChainMap модуля collections.

Рекомендуемые ресурсы

Документация Python
https://docs.python.org/3/tutorial/datastructures.html#sets
https://docs.python.org/3/tutorial/datastructures.html#dictionaries
https://docs.python.org/3/library/stdtypes.html#set-types-set-frozenset
https://docs.python.org/3/library/stdtypes.html#mapping-types-dict
https://docs.python.org/3/library/stdtypes.html#dictionary-view-objects
https://docs.python.org/3/library/collections.html

Статьи в Википедии о ключевых понятиях, рассмотренных на этом уроке
https://ru.wikipedia.org/wiki/Множество
https://ru.wikipedia.org/wiki/Множество_(тип_данных)
https://en.wikipedia.org/wiki/Map_(mathematics)
https://ru.wikipedia.org/wiki/Ассоциативный_массив
https://ru.wikipedia.org/wiki/Мультимножество

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