**Словарь (dictionary)**

**Вспомним списки (list)**

В списке (list) индекс (номер) - целое число. Номера начинаются с 0. Элементы в списке идут по порядку. Порядок сам не меняется. В список можно добавить элемент или удалить. Можно изменить элемент. Можно отсортировать список или перемешать элементы.

Использование списка:

>>> Days = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']

>>> Days[0]

'Sunday'

>>> Days[1]

'Monday'

Хочется получить обратное соответствие. По названию дня недели найти его номер. Можно функцией index найти число, но это долго. Хочется получать *быстро* так:

>>> Days['Sunday']

0

>>> Days['Monday']

1

Для этого используют словарь (dictionary).

**Словарь**

Пара - это 2. Пара носков. Пара ботинок. Пара луж. Пара рук. Много пар обуви.

Запишем для людей как зовут человека -> какой у него рост. Ключ - как зовут человека. Все люди разные. Какой рост - значение. Рост разных людей может быть одинаковый.

Словарь (dict) - это много пар ключ (key) и значение (value).

* Ключ и значение вместе называют **item** (пара, элемент словаря)
* **Ключ должен быть уникальным**. Не может быть 2 одинаковых ключей в одном словаре.
* **Ключ должен быть неизменяем** (число, строка, turple - неизменяемы, могут быть ключом; list, set - изменяемы, не могут быть ключом).
* **Значением** может быть все, что угодно: None, число, строка, список, еще один словарь.

Создадим словарь день недели (ключ) - его порядковый номер (значение). Получать по ключу его значение - быстро:

Days = {

'Sunday': 0,

'Monday': 1,

'Tuesday': 2,

'Wednessday': 3,

'Thursday': 4,

'Friday': 5,

'Saturday': 6

}

>>> Days['Sunday'] # взять в словаре Days значение по ключу 'Sunday'

0

>>> Days['Monday']

1

>>> Days['Yesterday'] # такого ключа нет в словаре, получили ошибку KeyError

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

KeyError: 'Yesterday'

В словарь можно добавлять элементы и удалять их.

>>> Days['Yesterday'] = -1 # добавили в словарь Days пару ключ 'Yesterday' -> значение -1

>>> Days['Yesterday'] # взять в словаре значение по ключу 'Yesterday' (теперь оно есть)

-1

Ключ должен быть уникальным, значения могут совпадать.

>>> Days['Tomorrow'] = -1

>>> Days['Yesterday'] == Days['Tomorrow']

True

Значение в паре можно изменить.

>>> Days['Tomorrow']

-1

>>> Days['Tomorrow'] = -2

>>> Days['Tomorrow']

-2

**Создание словаря**

Пустой словарь:

d1 = dict()

d2 = {} # вот почему {} нельзя использовать для создания пустого множества, только set()

Создадим словарь страна - столица:

capitals = {'Russia': 'Moscow', 'Ukraine': 'Kiev', 'USA': 'Washington', 'Myanmar':'Naypyidaw', 'Mongolia':'Ulaanbaatar', 'China':'Beijing'}

capitals = dict(Russia = 'Moscow', Ukraine = 'Kiev', USA = 'Washington', )

capitals = dict([("Russia", "Moscow"), ("Ukraine", "Kiev"), ("USA", "Washington")])

capitals = dict(zip(["Russia", "Ukraine", "USA"], ["Moscow", "Kiev", "Washington"]))

Пишем красиво:

capitals = {

'Russia': 'Moscow',

'Ukraine': 'Kiev',

'USA': 'Washington',

'Myanmar':'Naypyidaw',

'Mongolia':'Ulaanbaatar',

'China':'Beijing'

}

**dict comprehensions**

Можно использовать dict comprehensions:

cities = ["Moscow", "Kiev", "Washington"]

states = ["Russia", "Ukraine", "USA"]

capitalsOfState = {state: city for city, state in zip(cties, states)}

Или с числами:

square1 = {x : x\*x for x in range(10) } # dict comprehensions

square2 = {0=0, 1=1, 2=4, 3=9, 4=16, 5=25, 6=36, 7=49, 8=64, 9=81} # задаем явно пары ключ=значение

Удобно, если нужно вывернуть словарь "наоборот":

>>> StateByCapital = {CapitalsOfState[state]: state for state in CapitalsOfState}

>>> stateByCapital

{'Kiev': 'Ukraine', 'Moscow': 'Russia', 'Washington': 'USA'}

И с числами:

sqrts = {square1[x]:x for x in square1}

Этот код будет работать чуть-чуть быстрее:

sqrts2 = {v:k for k,v in square1.items()}

qqq

**Операции над словарем**

|  |  |
| --- | --- |
| [**Операция над словарем А**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonDict?sortcol=0&table=1&up=0#sorted_table) | [**Значение**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonDict?sortcol=1&table=1&up=0#sorted_table) |
| value = A[key] | Получение элемента по ключу. Если элемента с заданным ключом в словаре нет, то возникает исключение KeyError[?](http://acm.mipt.ru/twiki/bin/edit/Cintro/KeyError?topicparent=Cintro.PythonDict). |
| value = A.get(key) | Получение элемента по ключу. Если элемента в словаре нет, то get возвращает None. |
| value = A.get(key, default\_value) | То же, но вместо None метод get возвращает default\_value. |
| key in A | Проверить принадлежность ключа словарю. |
| key not in A | То же, что not key in A. |
| A[key] = value | Добавление нового элемента в словарь или изменяет старое значение на value |
| del A[key] | Удаление пары ключ-значение с ключом key. Возбуждает исключение KeyError[?](http://acm.mipt.ru/twiki/bin/edit/Cintro/KeyError?topicparent=Cintro.PythonDict), если такого ключа нет. |
| if key in A:   del A[key] | Удаление пары ключ-значение с предварительной проверкой наличия ключа. |
| try:   del A[key] except KeyError[?](http://acm.mipt.ru/twiki/bin/edit/Cintro/KeyError?topicparent=Cintro.PythonDict):   pass | Удаление пары ключ-значение с перехватыванием и обработкой исключения. |
| value = A.pop(key) | Удаление пары ключ-значение с ключом key и возврат значения удаляемого элемента. Если такого ключа нет, то возбуждается KeyError[?](http://acm.mipt.ru/twiki/bin/edit/Cintro/KeyError?topicparent=Cintro.PythonDict). |
| value = A.pop(key, default\_value) | То же, но вместо генерации исключения возвращается default\_value. |
| A.pop(key, None) | Это позволяет проще всего организовать безопасное удаление элемента из словаря. |
| len(A) | Возвращает количество пар ключ-значение, хранящихся в словаре. |
| A.keys() | Возвращает список ключей |
| A.values() | Возвращает список значений (порядок в нем такой же, как для списка ключей) |
| A.items() | Возвращает список пар ключ, значение |

**Перебор словаря**

DEA! При переборе словаря порядок ключей и пар может быть любой. Порядок может измениться со временем (а может остаться прежним).

In [12]: capital = {'Russia': 'Moscow', 'Ukraine': 'Kiev', 'USA': 'Washington',

...: 'Myanmar':'Naypyidaw', 'Mongolia':'Ulaanbaatar', 'China':'Beijing'}

In [13]: capital

Out[13]:

{'China': 'Beijing',

'Mongolia': 'Ulaanbaatar',

'Myanmar': 'Naypyidaw',

'Russia': 'Moscow',

'USA': 'Washington',

'Ukraine': 'Kiev'}

По ключам:

In [14]: for s in capital:

...: print(s, capital[s])

...:

China Beijing

Mongolia Ulaanbaatar

Ukraine Kiev

Russia Moscow

USA Washington

Myanmar Naypyidaw

Можно явно написать, что по списку ключей

In [15]: for s in capital.keys():

...: print(s, capital[s])

China Beijing

Mongolia Ulaanbaatar

Ukraine Kiev

Russia Moscow

USA Washington

Myanmar Naypyidaw

и отсортировать ключи:

In [18]: for s in sorted(capital.keys()):

...: print(s, capital[s])

China Beijing

Mongolia Ulaanbaatar

Myanmar Naypyidaw

Russia Moscow

USA Washington

Ukraine Kiev

Только значения:

In [19]: for c in capital.values():

...: print(c)

...:

Beijing

Ulaanbaatar

Kiev

Moscow

Washington

Naypyidaw

(да, capital.values() тоже можно отсортировать!)

Если нам в цикле будет нужен и ключ, и значение, лучше перебирать по парам:

In [20]: for s, c in capital.items():

...: print(s, c)

China Beijing

Mongolia Ulaanbaatar

Ukraine Kiev

Russia Moscow

USA Washington

Myanmar Naypyidaw

Пары тоже можно отсортировать:

In [21]: for s, c in sorted(capital.items()):

...: print(s, c)

China Beijing

Mongolia Ulaanbaatar

Myanmar Naypyidaw

Russia Moscow

USA Washington

Ukraine Kiev

В обратном порядке reverse=True:

In [22]: for s, c in sorted(capital.items(), reverse=True):

...: print(s, c)

Ukraine Kiev

USA Washington

Russia Moscow

Myanmar Naypyidaw

Mongolia Ulaanbaatar

China Beijing

**Задачи на работу с набором объектов**

**Задача 0.**

Даны числа. Напечатать сколько всего чисел и сколько *разных* чисел.

**Задача 1.**

Даны числа, напечатать число (порядок - любой) и сколько раз оно встречалось.

**Задача 2.**

Даны числа, напечатать число (по возрастанию) и сколько раз оно встречалось.

**Задача 3.**

Даны числа, напечатать число и сколько раз оно встречалось (по убыванию).

**Задача 4 Страны и города**

Дан список стран и городов каждой страны. Затем даны названия городов. Для каждого города укажите, в какой стране он находится.

Программа получает на вход количество стран N. Далее идет N строк, каждая строка начинается с названия страны, затем идут названия городов этой страны. В следующей строке записано число M, далее идут M запросов — названия каких-то M городов, перечисленных выше.

Для каждого из запроса выведите название страны, в котором находится данный город.

Пример:

|  |
| --- |
| **Входные данные** |
| 2  Russia Moscow Petersburg Novgorod Kaluga  Ukraine Kiev Donetsk Odessa  3  Odessa  Moscow  Novgorod |
| **Выходные данные** |
| Ukraine  Russia  Russia |

**Задача 4а.**

Постройте словарь страна -> список ее городов. Напечатайте словарь.

**Задача 4b.**

По этому словарю постройте другой словарь: город -> страна. Напечатайте словарь.

**Задача 5. Сколько было сделано покупок?**

Дан файл [bet.log](http://acm.mipt.ru/twiki/pub/Cintro/PythonListTask/bet.log)

В нем записаны данные работы интернет-магазина.

* секунд с 1 января 1970
* номер клиента (1, 2, 3, ....)
* номер чека

Сколько всего покупок было сделано в магазине?

**Задача 6. Сколько работал магазин?**

По этому же файлу определите, сколько времени в часах, минутах и секундах работал магазин.

**Задача 7. Время между первой и последней покупкой клиента N**

По этому же файлу определите, время между первой и последней покупкой клиента N. Номер клиента = номер варианта.

**Задача 8. Сколько разных клиентов?**

По этому же файлу определите, сколько разных клиентов сделали покупки в магазине.

**Задача 9. СПИСОК сколько клиенты сделали покупок?**

По этому же файлу напечатайте список: на каждой строке номер\_клиента сколько\_клиент\_сделал\_покупок

**Задача 10. Клиенты, которые сделали больше N покупок (клиент - количество покупок)**

Напечатайте только номера клиентов, которые сделали больше N покупок. N задать в программе. Например, N=10