



Словари Python.

Если вспомнить список (List), то можно увидеть, что обращение к элементам происходит по номерам.

Пример (порядковые номера):

Color [1] или **Name** [2]

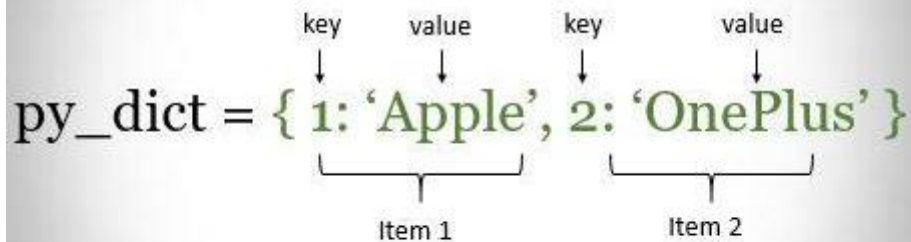
Но бывают ситуации, когда использование чисел не совсем удобно. Например номера рейсов или номера поездов.

Поезд 020У МЕГАПОЛИС

Маршрут: МОСКВА СПб

При численно-буквенном идентификаторе понадобится другой способ хранения и вызова.

Python Dictionary



СТРУКТУРА ДАННЫХ,
обращение к которым происходит
не по числовому индексу, а
произвольно, называется.

СЛОВАРЕМ (DICT)

Создадим пустой словарь:

СТРАНА ↔ СТОЛИЦА

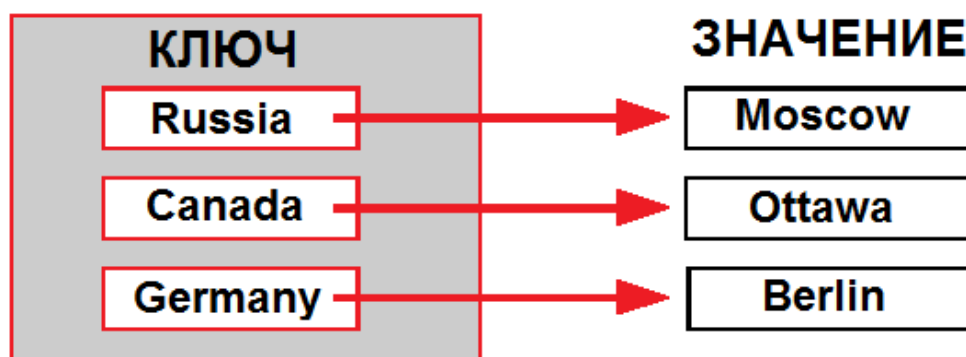
и заполним его:

```
Capitals = dict()
```

```
Capitals['Russia'] = 'Moscow'
```

```
Capitals['Germany'] = 'Berlin'
```

```
Capitals['Canada'] = 'Ottawa'
```



КЛЮЧ (ИНДЕКС)	ЗНАЧЕНИЕ
СТРАНА	СТОЛИЦА

Как видим, **значения** соответствуют уникальному **ключу**. Двух одинаковых ключей в словаре **быть не может**.

Создадим уже знакомый нам список (List):

Countries = ['Russia', 'Germany', 'Canada', 'USA']

и напомним программу опроса словаря:

```
Capitals = dict()
Capitals['Russia'] = 'Moscow'
Capitals['Germany'] = 'Berlin'
Capitals['Canada'] = 'Ottawa'

Countries = ['Russia', 'Germany', 'Canada', 'USA']

for country in Countries:
    if country in Capitals:
        print('Столица страны ' +
              country + ': ' + Capitals[country] + ';')
    else:
        print('Страна с названием ' +
              country + ' отсутствует в словаре.')
```



СЛОВАРЬ РУССКОГО ЯЗЫКА

также показательный пример:

Слово – **ключ**, **пояснение** – **значение**.

СОЙКА — птица семейства вороновых.
ЧЕРЕПАХА — пресмыкающееся,
покрытое костным панцирем.

```
RusDict = dict()  
RusDict['Сойка'] = 'Птица семейства вороновых'  
RusDict['Черепаха'] = 'пресмыкающееся,  
покрытое костным панцирем'
```

ТЕЛЕФОННЫЙ СПРАВОЧНИК



```
PhoneBook = dict()  
PhoneBook['6930005'] = 'Школа 291'  
PhoneBook['9571829'] = 'Бассейн'
```

В качестве ключа необходимо
использовать **номер телефона**, так
как именно он **уникальный**.



СЛОВАРИ И СПИСКИ

Словари можно создавать не только с помощью метода `dict()`, но и используя фигурные скобки `{}`:

```
Trains = {} # пустой словарь  
Trains = {'001A': "Красная стрела",  
          '019У': "Мегаполис"}
```

В примере выше чего-то не хватает!
Конечно!!! Направления поезда!

На помощь приходят знакомые списки:

```
Trains = {  
    '001A': ["Красная стрела", 'СПб - Москва'],  
    '013У': ["Южный Урал", 'Челябинск - Москва'],  
    '059Г': ["Волга", 'Нижний Новгород - СПб'],  
    '059Е': ["Тюмень", 'Нижневартовск - Москва'],  
    '025В': ["Воронеж", 'Воронеж - Москва']  
}
```

Ключ – номер поезда. А **значение** же уже состоит из двух частей:

`Trains[ключ][0]` – название поезда;
`Trains[ключ][1]` – направление.



Программа «СПРАВОЧНАЯ»

Используя уже знакомые нам приемы программирования, организуем вывод информации о поезде по его номеру (**ключ** словаря).

```
Trains = {
'001A':['"Красная стрела"', 'СПб - Москва'],
'013У':['"Южный Урал"', 'Челябинск - Москва'],
'059Г':['"Волга"', 'Нижний Новгород - СПб'],
'059Е':['"Тюмень"', 'Нижевартовск - Москва'],
'025В':['"Воронеж"', 'Воронеж - Москва']}

num = input('Введите номер поезда: ')

if num in Trains:
    print('Поезд:', num, Trains[num][0], 'следует по
маршруту:', Trains[num][1])
else:
    for item in Trains:
        print('Поезд:', item, Trains[item][0], 'следует
по маршруту: ', Trains[item][1])
    print('Пожалуйста, выберите Ваш номер поезда из
списка:')
```

Задание: добавить время прибытия в словарь и вывести его по запросу.



Программа «СПРАВОЧНАЯ» (полная версия)

```
Trains = {
'001A':["Красная стрела", 'СПб - Москва', '07:55'],
'013У':["Южный Урал", 'Челябинск - Москва', '06:22'],
'059Г':["Волга", 'Нижний Новгород – СПб', '00:29'],
'059Е':["Тюмень", 'Нижевартовск - Москва', '09:30'],
'025В':["Воронеж", 'Воронеж - Москва', '07:50']}

while True:
    num = input('Введите номер поезда: ')

    if num in Trains:
        print('\nПоезд:', num, Trains[num][0],
'следует по маршруту:', Trains[num][1], '\nПрибывает в
Москву в', Trains[num][2])
        break
    else:
        for item in Trains:
            print('\nПоезд:', item, Trains[item][1])
            print('\nПожалуйста выберите Ваш номер
поезда из списка:')
            continue
```

Когда использовать



словари Python

КОГДА ПОЛЕЗНЫ словари PYTHON

- Если необходим подсчет объектов (людей).
Объекты — ключи, а значения — количество.

Легко реализуется ввиду уникальности ключей.

*Например: **класс** — **число учеников**.*

- Хранение каких-либо данных, связанных с объектом. Объекты — ключи, а значения — характеристика.

*Получение **номера дня недели** (**месяца**) по **названию**.*

- Установка соответствия между объектами (например, "родитель — ребенок").

*Ключ — **родитель**, значения — дети (**ребенок**).*

- Если в качестве индекса нельзя использовать число.

*Демонстрировалось на примере **номеров поездов**.*