# 6.Dictionary_and_set_structure

December 4, 2018

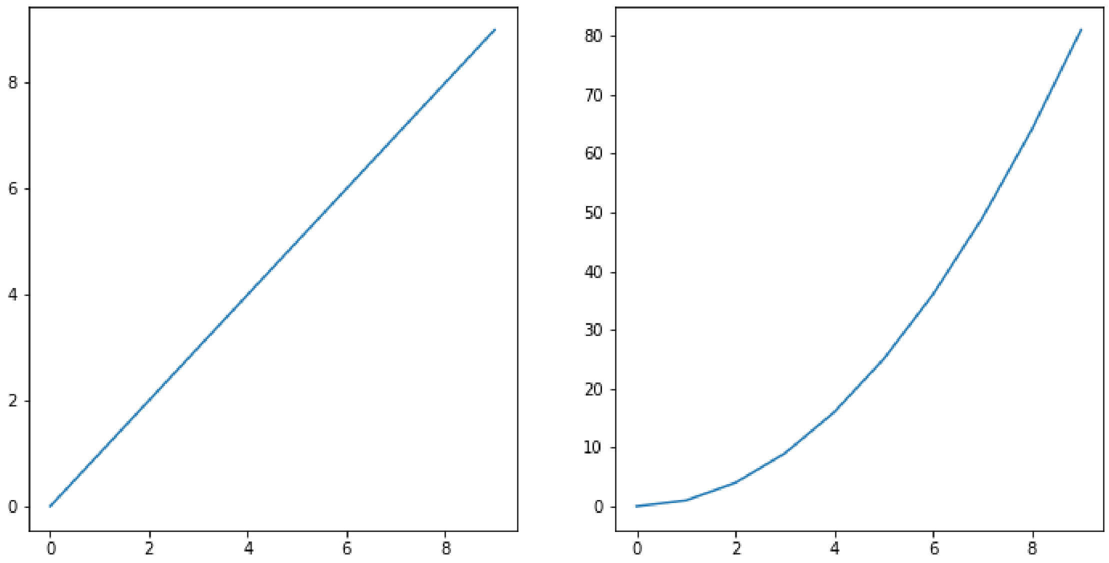
## 1 Dictionaries and deeper look at sets

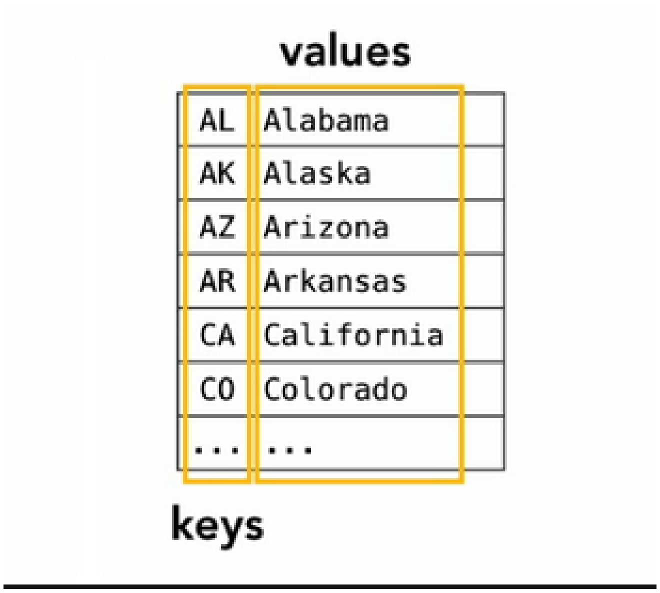
### 1.1 Dict

They are related to socalled associative array and hash maps.
It is a structure which maps some values (keys) to the other values (well... values)

Here x are keys and y are values

```
In [20]: import matplotlib.pyplot as plt
         import seaborn as sns
         %matplotlib inline

         _, ax = plt.subplots(ncols=2, figsize=(12, 6))

         sns.lineplot(range(10), range(10), ax=ax[0])
         sns.lineplot(range(10), [i ** 2 for i in range(10)], ax=ax[1]);
```





| | |
|---|---|
| AL | Alabama |
| AK | Alaska |
| AZ | Arizona |
| AR | Arkansas |
| CA | California |
| CO | Colorado |
| ... | ... |

```
In [15]: # Creation
         numbers_to_strings = {1: '1', 2: '2'}
         # Empty ones
         empty_dict = {}
         another_empty_dict = dict()

In [16]: # Retrieve value from the dict
         numbers_to_strings[1]

Out[16]: '1'

In [17]: # Add element to the dict
         empty_dict['my_key'] = 'its_value'
         empty_dict

Out[17]: {'my_key': 'its_value'}

In [18]: # Change value in dict
         numbers_to_strings[1] = 'one'
         numbers_to_strings

Out[18]: {1: 'one', 2: '2'}
```

### 1.2 Some dict methods

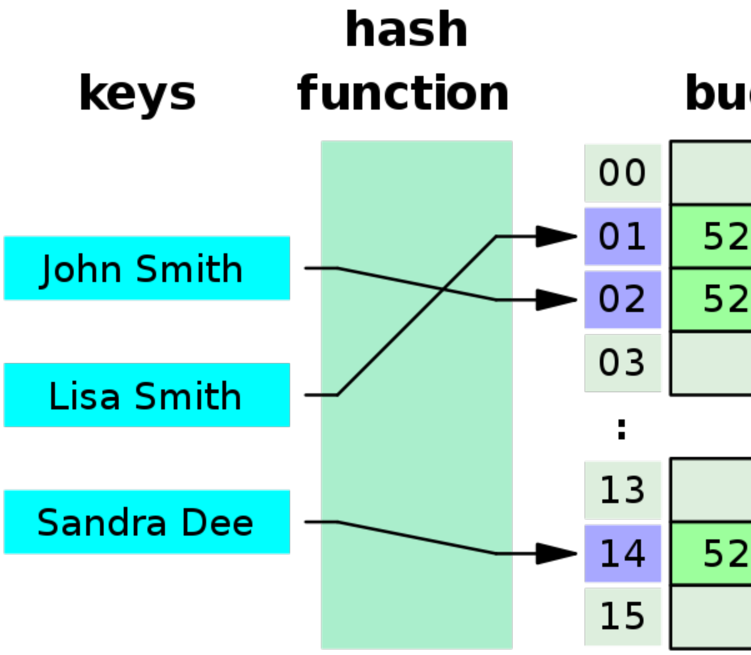
- `keys()` - get iterable with keys of dict
- `values()` - get iterable with values of dict
- `items()` - get iterable with tuples with keys and values of dict
- `get(key, [default])` - try to get value associated with key in dict or default value if there is no such key; default is optional and None by default

- `pop(key, [default])` - try to get value associated with key and delete it from dict; if there is no default it will throw an error if key not in dict, otherwise default will be reterned

```
In [42]: numbers_to_strings

Out[42]: {1: 'one', 2: '2'}

In [29]: for key in numbers_to_strings.keys():
             print("key is", key, "and value is", numbers_to_strings[key])

key is 1 and value is one
key is 2 and value is 2


In [30]: for value in numbers_to_strings.values():
             print("value is", value)

value is one
value is 2


In [31]: for key, value in numbers_to_strings.items():
             print('now we get both key and value -', key, value)

now we get both key and value - 1 one
now we get both key and value - 2 2


In [32]: numbers_to_strings.get(1)

Out[32]: 'one'

In [33]: numbers_to_strings.get(3)

In [34]: numbers_to_strings.get(3, 'Not in dict')

Out[34]: 'Not in dict'

In [43]: numbers_to_strings.pop(5)


        ---------------------------------------------------------------------------

        KeyError                                  Traceback (most recent call last)

        <ipython-input-43-4c8bb606fadd> in <module>()
    ----> 1 numbers_to_strings.pop(5)


        KeyError: 5
```

```
In [44]: numbers_to_strings.pop(5, 'No key')

Out[44]: 'No key'

In [45]: numbers_to_strings.pop(2)

Out[45]: '2'

In [46]: numbers_to_strings

Out[46]: {1: 'one'}
```

### 1.3 Structure of dict and set



Hash function concept is important for both of them

What is a hash function?

It is a function which map from hashable (containing immutable parts) object to a number. For example

$H$ - hash funcion
$point$ - a point object with coordinates, let's say tuple (5, 9)

$H(point) = 2$

How to compute a hash?

Hash function convert everything into a number and get a remainder to limit a number to some range - size of array

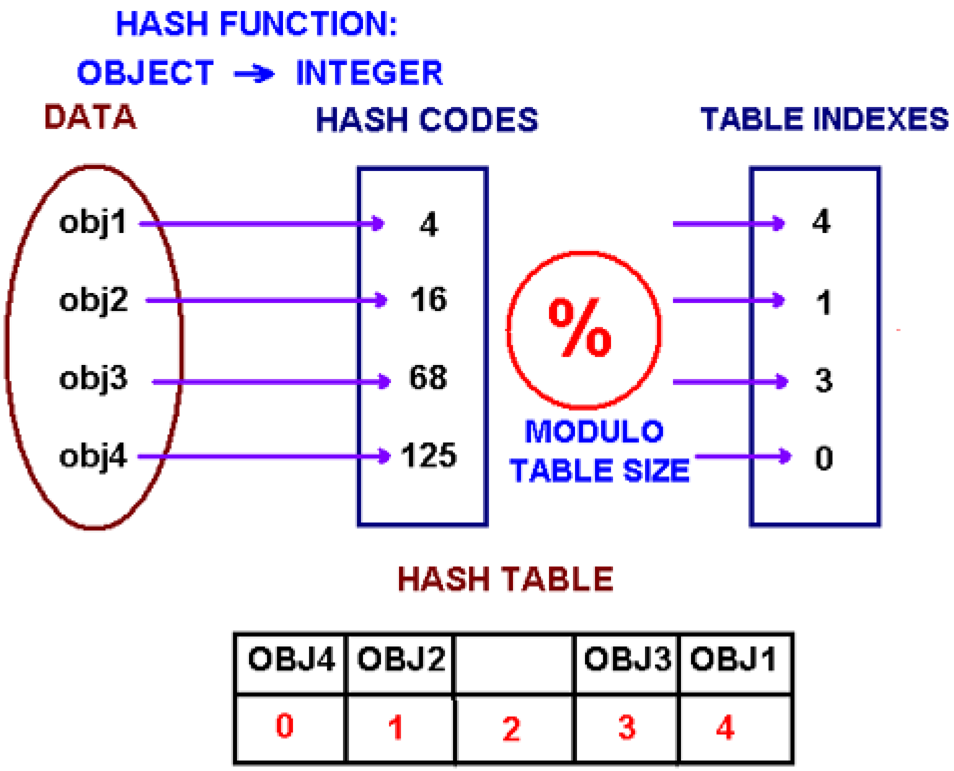
$5 \cdot 2 + 9 \cdot 3 = 37$

Let's say that size of our array is 35. Therefore

$37 \bmod 35 = 2$

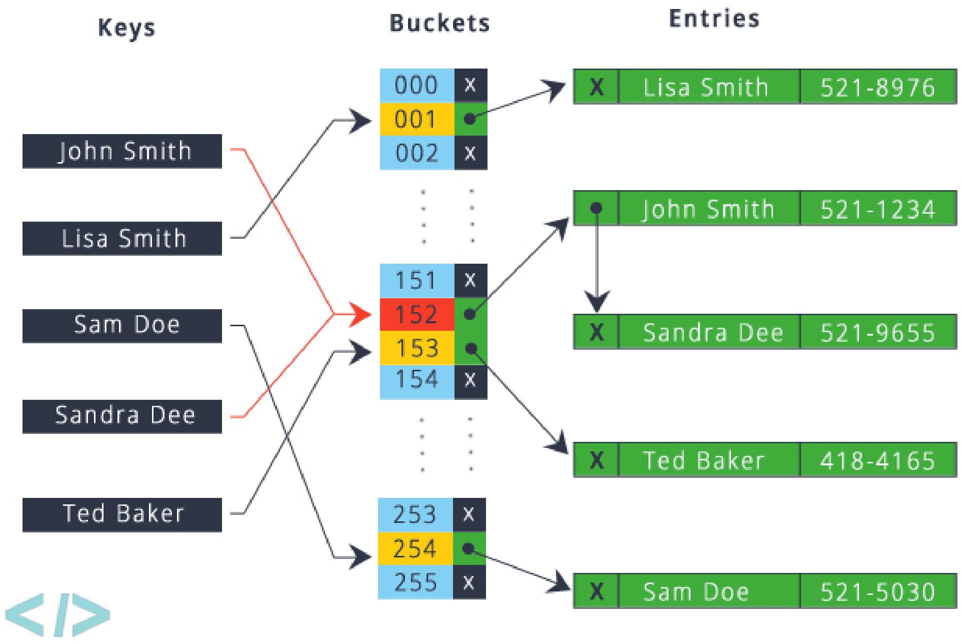
Why we need a hash?

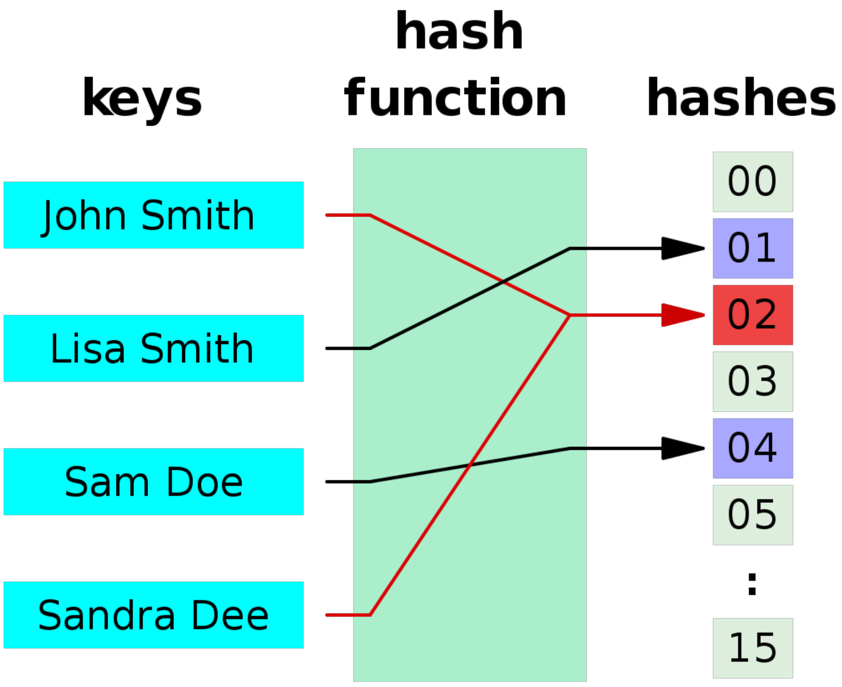
Dictionary is similar to an array with hash function. Hashing is a way to convert nonnumeric indices to numeric indices with which we can refer to appropriate cell in array



| OBJ4 | OBJ2 |  | OBJ3 | OBJ1 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |

Hash functions which give unique values are of great favour - to have unique indices for dict values. It's not always the case and sometimes hash function give 1 velue for several keys. We call





it collision

- We can just make a list in these cases
- Sets have the same structure, yet they have dummy variables as values