ex39\_ 더해보기

1. Apply this to the case of other countries’ province and city mapping.

Well rather than repeating basically same stuffs. I thought about chain of dicts within a dict.

A screen shot of a monitor

Description automatically generated

I thought about dictionary within a dictionary, and found that abc\_mart stock database management could get a lot of help from using the datatype of ‘Dictionary’. It is composed of the number of stock mapped to sizes and sizes mapped to a certain type of a shoe model and shoe models mapped to a brand and there being a lot of brands in ABC\_Mart database.

2. search pydoc to get to know more about dictionary, or dict.

A close up of text on a white background

Description automatically generated

A close up of text on a white background

Description automatically generated

studied pydoc dict.

I could get to know more about, generating new dictionary object in 4 different ways and the methods to edit, refer to, or check on a dictionary object.

A few methods are left ununderstood. I will add it to honest question list and get to know later.

3. Are there some things that can’t be properly dealt with with the data structure dictionary? unsolved.

Think about <index-like>ness ( order-considered) of dictionary.

1) Basically dictionary functions rather like a stack



it’s hard to change order in which stack objects are ordered.

which comes last pops first / and the first one is destined to be served to the iterater.

what if I need control over the order in which keyword – value is saved?

2). python ordered dict module

1 definition collections.OrderedDict ( ) => blank ordereddict made.

d = collections.OrderedDict()

d['a'] = 'A'

d['b'] = 'B'

d['c'] = 'C'

d['d'] = 'D'

d['e'] = 'E'

common with python3.6 dict in :

1). iterate from the firstly input factor

differs from python3.6 in:

1). equality test considers even the order in which the key-value has been put.

2). can refer to factors by order in which it was put

A screenshot of a cell phone

Description automatically generated

OrderedDict는 사전내 페어들의 배열 순서를 리오더링 하기위해 디자인되었다. /

알고리즘적으로, 빈번한 사전 요소 재배열을 잘 처리할 수 있다.

popitem()시에 꼭 스택처럼 나중에 들어온 요소가 먼저 나가지 않는다. 어떤 아이템을 popitem()할지 구체화 할 수 있다.

move\_to\_end()라는 메소드가 있어서, 효과적으로 어떤 요소를 스택에서 맨위로 올릴 수 있다.