zip_dict

December 20, 2017

1 The peanut-butter and jelly of paired data: zip() and dict()

Let's say we have two lists of names and want to store them as paired data in a dictionary. All we need to do to get a nice, tidy, dictionary out of it is the following (I'll show the running code and then go through it afterwards with a more realistic use case).

```
In [54]: # here are our two lists
         first = ["A","B","C"]
         second = ["1","2","3"]

         # make a dict from the output of zipping the two iterables
         third = dict(zip(first, second))

         # iterate thorough and show we got what we were looking for
         for item in z:
             print(item, third[item])
```

```
A 1
B 2
C 3
```

That's it. What's going on here? Let's take a closer look.

Python has a "zip" function that will return a zip object of tuples (don't worry if that sounds weird; it just means it is a zip class and has our data stored in a way such that it can't be modified). It takes iteratable objects like lists as arguments, in our case, just two. It goes through the lists in order, and "zips" them together, first item to first item, second to second, and so on.

```
In [50]: # here are our to lists
         first = ["Christopher","Jo-Vaughn","Sean","Sir Robert Bryson","Jermaine","Alexander",'
         last = ["Wallace","Scott","Carter","Hall II","Cole","Hamilton","Trotter","Duckworth",'

         # call the zip function that takes iteratable structure
         names = zip(first, last)

         # if we print it out, we see we have a zip object and memory address
         print(names)
         print(type(names))
```

```
<zip object at 0x7efc545fb608>
<class 'zip'>
```

While they are in the object, we can't use them as paired data directly. Fortunately, the dict() function will take a list of tuples and return a dictionary.

```
In [51]: # use the dict function on the zip object
         names = dict(names)

         # print out the evidence
         for item in names:
             print(item, names[item])
```

```
Christopher Wallace
Jo-Vaughn Scott
Sean Carter
Sir Robert Bryson Hall II
Jermaine Cole
Alexander Hamilton
Tariq Trotter
Kendrick Duckworth
Tupac Shakur
```

Note that this will also work with ints and floats. That's all there is to it. Thanks for reading.