14. Zip and unzip

Zip

Zip is a useful function that allows you to combine two lists easily.

After calling zip, an iterator is returned. In order to see the content wrapped inside, we need to first convert it to a list.

Example:

```
first_name = ['Joe','Earnst','Thomas','Martin','Charles']

last_name = ['Schmoe','Ehlmann','Fischer','Walter','Rogan','Green']

age = [23, 65, 11, 36, 83]

print(list(zip(first_name,last_name, age)))

# Output
#
# [('Joe', 'Schmoe', 23), ('Earnst', 'Ehlmann', 65), ('Thomas', 'Fischer', 11), ('Martin',
'Walter', 36), ('Charles', 'Rogan', 83)]
```

One advantage of zip is that it improves readability of for loops.

For example, instead of needing multiple inputs, you only need one zipped list for the following for loop:

```
first_name = ['Joe','Earnst','Thomas','Martin','Charles']
last_name = ['Schmoe','Ehlmann','Fischer','Walter','Rogan','Green']
age = [23, 65, 11, 36, 83]

for first_name, last_name, age in zip(first_name, last_name, age):
    print(f"{first_name} {last_name} is {age} years old")

# Output
#
# Joe Schmoe is 23 years old
# Earnst Ehlmann is 65 years old
# Thomas Fischer is 11 years old
# Martin Walter is 36 years old
# Charles Rogan is 83 years old
```

Unzip

We can use the *zip* function to unzip a list as well. This time, we need an input of a list with an asterisk before it.

The outputs are the separated lists.

Example:

```
full_name_list = [('Joe', 'Schmoe', 23),
                  ('Earnst', 'Ehlmann', 65),
                  ('Thomas', 'Fischer', 11),
                  ('Martin', 'Walter', 36),
                  ('Charles', 'Rogan', 83)]

first_name, last_name, age = list(zip(*full_name_list))
print(f"first name: {first_name}\nlast name: {last_name} \nage: {age}")

# Output

# first name: ('Joe', 'Earnst', 'Thomas', 'Martin', 'Charles')
# last name: ('Schmoe', 'Ehlmann', 'Fischer', 'Walter', 'Rogan')
# age: (23, 65, 11, 36, 83)
```