CS410: Principles and Techniques of Data Science

Python

Python

- **Is this possible?**

```
heterogeneous_list = ["string", 0.1, True]


x = [0,1,2,3,4,5,6,7,8,9]
zero = x[0]
one = x[1]
nine = x[-1]
eight = x[-2]
```

Can you do this?

```
x[0] = -1
```

```
#You can also use square brackets to slice lists.
#The slice i:j means all elements from i (inclusive) to j (not inclusive).
#If you leave off the start of the slice, you'll slice from the beginning of the list, and if you leave of the end of the slice, you'll slice until the end of the list

    x = [-1, 1, 2, 3, 4, 5, 6, 7, 8, 9]

    first_three = x[:3]                         # [-1, 1, 2]
    three_to_end = x[3:]                        # [3, 4, ..., 9]
    one_to_four = x[1:5]                        # [1, 2, 3, 4]
    last_three = x[-3:]                         # [7, 8, 9]
    without_first_and_last = x[1:-1]            # [1, 2, ..., 8]
    copy_of_x = x[:]                            # [-1, 1, 2, ..., 9]
```

List vs Tuple

```
my_list = [1, 2]
my_tuple = (1, 2)
my_list[1] = 3    # my_list is now [1, 3]
```

Dictionaries

```
empty_dict = {}      # Pythonic
empty_dict2 = dict()  # less Pythonic
grades = {"Joel": 80, "Tim": 95}
```

Sets

```
s = set()
s.add(1)  # s is now {1}
s.add(2)  # s is now {1, 2}
```

Why use sets?

Sorting

```
x=[4,1,2,3]
y = sorted(x)  # y is [1, 2, 3, 4], x is unchanged
x.sort()       # now x is [1, 2, 3, 4] in-place
```

Random

```
import random
random.seed(10) # this ensures we get the same results every time

up_to_ten = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
random.shuffle(up_to_ten)

my_best_friend = random.choice(["Alice", "Bob", "Charlie"])

#To randomly choose a sample of elements without replacement (i.e., with no duplicates), you can use random.sample:
lottery_numbers = range(60)
winning_numbers = random.sample(lottery_numbers, 6) # [16, 36, 10, 6, 25, 9]
```

Zip function

The zip function transforms multiple iterables into a single iterable of tuples of corresponding function:

```
list1 = ['a', 'b', 'c']
list2 = [1, 2, 3]
zip(list1, list2) # is [('a', 1), ('b', 2), ('c', 3)]
```

Summary

https://colab.research.google.com/drive/1B_hTKK_hMBDdUpDjt89gfQNFFV9F45pn?usp=sharing

THANK YOU!