## Tuples

- Sequence of elements, similar to lists
- Unlike lists Once you initialize a tuple, you cannot edit it anymore aka tuples are **immutable** and lists are **mutable**

https://docs.python.org/2/library/stdtypes.html#typesseq
(https://docs.python.org/2/library/stdtypes.html#typesseq)

```
In [1]:
```

```
tup = ('check',2,3,4)
tup
```

```
Out[1]:

('check', 2, 3, 4)
```

```
In [2]:
```

```
type(tup)
```

```
Out[2]:

tuple
```

```
In [3]:
```

```
tup[0]
```

```
Out[3]:

'check'
```

```
In [4]:
```

```
len(tup)
```

```
Out[4]:

4
```

```
In [5]:
```

```
for element in tup:
    print(element)
```

```
check
2
3
4
```

In [6]:

```
tup[0] = 1
```

```
---------------------------------------------------------------------
-------
TypeError                                 Traceback (most recent cal
l last)
<ipython-input-6-7e408c05585b> in <module>()
----> 1 tup[0] = 1

TypeError: 'tuple' object does not support item assignment
```

## Sets

Similar to lists but elements inside the set are **distinct (all different)**

Common uses:

```
- membership testing
- removing duplicates from a sequence
- computing mathematical operations such as intersection, union, differenc
e, and symmetric difference
```

In [7]:

```
s = {1,2,3,1,1,1,1}
s
```

Out[7]:

```
{1, 2, 3}
```

In [25]:

```
type(s)
```

Out[25]:

```
set
```

In [26]:

```
s = {1,2,3,1,1,1,1}
s
```

Out[26]:

```
{1, 2, 3}
```

In [31]:

```
set1 = {1,2,3,4}
set2 = {3,4,5,6,7}
set1 & set2
```

Out[31]:

```
{3, 4}
```

In [32]:

```
set1 | set2
```

Out[32]:

```
{1, 2, 3, 4, 5, 6, 7}
```

In [33]:

```
set1^set2
```

Out[33]:

```
{1, 2, 5, 6, 7}
```

In [34]:

```
set1 - set2
```

Out[34]:

```
{1, 2}
```

## Example1:

- We asked the students in each REDI class to state their mother tongue languages
  - Python class: russian, french, arabic, arabic, arabic, arabic, farsi
  - Java class: english, kurdish, arabic, arabic, french

**We would like the answer the below questions:**

- How many different languages do we have in each class?
- How many different languages in both classes?
- How many common languages between the two classes?
- What are the languages that are found in only one class (not in both)

In [15]:

```
languages1 = {'russian', 'french', 'arabic', 'arabic','arabic', 'farsi'}
languages1
```

Out[15]:

```
{'arabic', 'farsi', 'french', 'russian'}
```

In [11]:

```
#languages1 = ['russian', 'french', 'arabic', 'arabic','arabic', 'farsi']
#distinct_languages = set(languages1)
```

In [12]:

```
len(distinct_languages)
```

Out[12]:

```
4
```

In [13]:

```
languages2 = {'english', 'kurdish', 'arabic', 'arabic', 'french'}
```

In [14]:

```
len(languages2)
```

Out[14]:

```
4
```

In [16]:

```
common_languages = languages1 & languages2
common_languages
```

Out[16]:

```
{'arabic', 'french'}
```

In [18]:

```
one_class_languages = languages1 ^ languages2
one_class_languages
```

Out[18]:

```
{'english', 'farsi', 'kurdish', 'russian'}
```

## Dictionaries

### Motivating Exercise:

- Suppose we have grades for a specific student, the student has different subjects and each subject has a grade

- Math => 5
- Science => 3
- English => 10
- German => 6

- How do we represent this information in our program?
- What will we use it for?
  - Write a function that takes as input subject name and prints out the student's grade
  - Write a function that takes as input subject name and a grade and do one of two things:

    ```
      - if we already have a previous grade for the subject, upd
    ate the grade
      - if we did not have this subject before, add it to the st
    udent's grades
    ```

In [20]:

```
grades = [['Math', 5], ['Science', 3], ['English', 10], ['German', 6]]
```

```
In [23]:
```

```
grades[0][0]
```

```
Out[23]:
```

```
'Math'
```

```
In [27]:
```

```
##Solution 1 using lists
grades = [['Math', 5], ['Science', 3], ['English', 10], ['German', 6]]

subject_name = 'blah'

for grade in grades:
    if grade[0] == subject_name:
        print(grade[1])
```

### How to define a dictionary

```
In [33]:
```

```
grades_dict = {'Math': 5,'Science': 3, 'English': 10,'German': 6}
```

### Accessing an element in a list vs in a dictionary

```
In [34]:
```

```
grades_list[0]
```

```
Out[34]:
```

```
['Math', 5]
```

```
In [39]:
```

```
grades_dict['Science']
```

```
Out[39]:
```

```
3
```

### Going over the values of the dict

```
In [44]:
```

```
# Way 1
for key in grades_dict:
    print(grades_dict[key])
```

```
6
3
5
10
```

In [45]:

```
# Way 2
for value in grades_dict.values():
    print(value)
```

```
6
3
5
10
```

**Getting Dictionary keys**

In [42]:

```
grades_dict.keys()
```

Out[42]:

```
['German', 'Science', 'Math', 'English']
```

In [43]:

```
grades_dict.values()
```

Out[43]:

```
[6, 3, 5, 10]
```

In [ ]:

```
### Solution 2 using dicts
grades_dict = {'Math': 5,'Science': 3, 'English': 10,'German': 6}
subject_name = raw_input('Enter subject ')

if subject_name in grades_dict:
    print(grades_dict[subject_name])
else:
    print('subject not found')
```

## Syntax

- General form

```
{key1:value1, key2:value2, key3:value3....}
```

- Each key cannot occur more than once in the dictionary

## Exercise 2

```
d = {'a':1, 2:'b'}
```

1. What are the keys in the above dictionary?
2. what is the value of d[2]?

source: https://www.coursera.org/learn/learn-to-program (https://www.coursera.org/learn/learn-to-program)

In [46]:

```
d = {'a':1, 2:'b'}
```

In [47]:

```
d.keys()
```

Out[47]:

```
['a', 2]
```

In [48]:

```
d[2]
```

Out[48]:

```
'b'
```

## Exercise 3

Given a list of elements in an array, we want to know how many times each element occurs in the list

example : list_words = ['english', 'german', 'spanish', 'italian', 'english', 'arabic', 'farsi', 'german', 'english']

output:

- english: 3
- german: 2
- spanish: 1
- italian: 1
- arabic: 1
- farsi: 1

## Exercise 4 (revisiting example)

Write a program that takes as input a list of student grades. A student grade can be either A/B/C/D/F. output the number of students having each grade.

Example:

- input_list = ['A', 'A', 'B', 'C', 'C', 'C', 'D','F','F']

output:

- A: 2 Students
- B: 1 Student
- C: 3 Students
- D: 1 Student
- F: 2 students