Tutorial Exercises Week 2 - Solutions

* Question 1

Assign the strings `'I'`, `'love'` and `'Tilburg'` to the variables `a`, `b` and `c`, respectively. Use these 3 variables to print the output `'I love Tilburg'` on one line.

```
a = 'I'
b = 'love'
c = 'Tilburg'
a + ' ' + b + ' ' + c
```

```
'I love Tilburg'
```

* Question 2

The days of the week in Dutch are maandag, dinsdag, woensdag, donderdag, vrijdag, zaterdag, zondag. Create a dictionary in Python where the *keys* are the days of the week in English and the *values* are the days of the week are in Dutch. Use your dictionary to translate a day of the week from English to Dutch.

```
dow = {
  'Monday'    : 'maandag',
  'Tuesday'   : 'dinsdag',
  'Wednesday' : 'woensdag',
  'Thursday'  : 'donderdag',
  'Friday'    : 'vrijdag',
  'Saturday'  : 'zaterdag',
  'Sunday'    : 'zondag'
}
dow['Thursday']
```

```
'donderdag'
```

* Question 3

Today is Thursday. What day of the week will it be in 200 days' time? *Hint*: make use of the modulus operator.

```
dow = {
  0 : 'Monday',
  1 : 'Tuesday',
  2 : 'Wednesday',
  3 : 'Thursday',
  4 : 'Friday',
  5 : 'Saturday',
  6 : 'Sunday'
}
dow[(3 + 200) % 7]
```

```
'Monday'
```

* Question 4

If `a` and `b` are logical constants (one of `True` or `False`), `not (a or b)` is always the same as which of the following?

- `a or b`
- `a and b`
- `not a or not b`
- `not a and not b`

This is one of De Morgan's laws. NOT (A OR B) is the same as NOT A AND NOT B:

$$\neg (A \vee B) \quad \Leftrightarrow \quad (\neg A) \wedge (\neg B)$$

where $\neg$ means NOT, $\vee$ means OR and $\wedge$ means AND. So the answer is `not a and not b`.

To see this in Python we can create two lists containing every possible combination of `True` and `False`:

```
a = [True, True, False, False]
b = [True, False, True, False]
```

NOT (A OR B) gives for each pair:

```
[not (a[i] or b[i]) for i in range(len(a))]
```

```
[False, False, False, True]
```

NOT A AND B gives for each pair:

```
[not a[i] and not b[i] for i in range(len(a))]
```

```
[False, False, False, True]
```

We can see that these are the same.

The other De Morgan's Law is that NOT (A AND B) is the same as NOT A OR NOT B:

$$\neg (A \wedge B) \quad \Leftrightarrow \quad (\neg A) \vee (\neg B)$$

Try write a set of Python commands to check this law as well!

* Question 5

Consider the following two lists:

```
a = [2, 4, 6, 8]
b = [1, 4, 6]
```

Write a single line of Python code that uses a list comprehension that checks for every element in `a` whether it is contained *somewhere* in `b`. The output should be `[False, True, True, False]` because 4 and 6 are in `b` while 2 and 8 are not.

```
[i in b for i in a]
```

```
[False, True, True, False]
```

* Question 6

From the following list:

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Write a single line of Python code using slices that creates a new list `b` with elements `[1, 2, 3, 8, 9, 10]`.

```
a[:3] + a[7:]
```

```
[1, 2, 3, 8, 9, 10]
```

* Question 7

- Anne is taking the following courses: `'Math'`, `'English'`, `'German'`, `'Physics'`, `'History'`, `'Economics'`.
- Ben is taking the following courses: `'Math'`, `'English'`, `'French'`, `'Chemistry'`, `'History'`, `'Business'`.

Create two Python variables containing the courses Anne and Ben are taking. Write a line of Python code to find the courses they are taking together.

```
anne = set(['Math', 'English', 'German', 'Physics', 'History', 'Economics'])
ben = set(['Math', 'English', 'French', 'Chemistry', 'History', 'Business'])
anne.intersection(ben)
```

```
{'English', 'History', 'Math'}
```

* Question 8

This is continued from Question 7. Write a line of Python code to find what courses Ben is taking that Anne isn't taking.

```
ben.difference(anne)
```

```
{'Business', 'Chemistry', 'French'}
```

* Question 9

Use Python code to find the average of the following list of numbers:

```
a = [2, 4, 3, 8, 5, 3, 5, 9, 1, 7]
```

Write your code in such a way that if you change the list `a` to another one (potentially with a different number of elements) that the code will still work.

```
sum(a) / len(a)
```

```
4.7
```

* Question 10

Write a `for` loop over the following list of numbers that prints `True` if the number is even and `False` if the number is odd:

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
for i in a:
    print(i % 2 == 0)
```

```
False
True
False
True
False
True
False
True
False
True
```

* Question 11

Count how many times the number 10 appears in the following list using 3 different methods:

- Using a `for` loop.
- Using a list comprehension
- Using one of the list's methods.

You may find it useful to use the fact that `True` equals one and `False` equals zero when converted to an integer.

The `for` loop approach:

```
counter = 0
for i in a:
    counter += i == 10
counter
```

```
1
```

The list comprehension approach:

```
sum([i == 10 for i in a])
```

```
1
```

The list method approach:

```
a.count(10)
```

```
1
```

* Bonus Question

In case you manage to complete exercises 1-11 quickly, you can try this question.

Use Python to convert 13,446,566 seconds into days, hours, minutes and seconds. Then get Python to print the message:

`13446566 seconds is A days, B hours, C minutes and D seconds.`

replacing `A`, `B`, `C` and `D` with what you find.

```
secs = 13446566
remaining = secs
days = remaining // (24 * 60 * 60)
remaining %= (24 * 60 * 60)
hours = remaining // (60 * 60)
remaining %= (60 * 60)
minutes = remaining // 60
remaining %= 60
seconds = remaining

str(secs) + " seconds is " + str(days) + " days, " + str(hours) + \
    " hours, " + str(minutes) + " minutes and " + str(seconds) + " seconds."
```

```
'13446566 seconds is 155 days, 15 hours, 9 minutes and 26 seconds.'
```