Developers Institute

Python Course

<u>Exercise 1</u>: List Comprehension

1) Create a list of random numbers from 10 to 20
    1. Create another list that will contain only the numbers divisible by 3
    2. Create another list that will contain only the numbers divisible by the length of the original list
    3. Create another list that will contain only the numbers bigger than 12
    4. Create another list that will contain all the numbers squared
    5. Create another list that will contain all the numbers that are not duplicates

<u>Exercise 2</u>: Dictionary Comprehension

Here are two lists

→ users = [**"Mickey"**, **"Minnie"**, **"Donald"**,**"Ariel"**,**"Pluto"**]

→ french_words= [**"Bonjour"**, **"Au revoir"**, **"Bienvenue"**, **"A bientôt"**]

1) Look at these results

```
{'mickey': 0, 'minnie': 1, 'donald': 2, 'ariel': 3, 'pluto': 4}

{0: 'mickey', 1: 'minnie', 2: 'donald', 3: 'ariel', 4: 'pluto'}

{'ariel': 0, 'donald': 1, 'mickey': 2, 'minnie': 3, 'pluto': 4}

{'bonjour': 'Hello', 'au revoir': 'Goodbye', 'bienvenue': 'Welcome', 'a bientôt': 'See you soon'}
```

2) Create the code that is needed to make this code run as expected.
   Some more explanations:
   - The 3rd result is in the alphabetical order
   - For the 4th result, find a way to get the translation of the French words, without hardcoding it.
3) Recreate the code of the 1st result, only if:
    1. The names contain the letter **"i"**
    2. The names start with the letter **"m"** or **"p"**
    3. The names don't contain letter duplicates
4) Recreate the code so that it renders this result
   {'pluto': 4, 'ariel': 3, 'donald' : 2, 'minnie' : 1, 'mickey' : 0}

<u>Exercice 3</u>: Exceptions

<u>Explaination of tutorialpoint</u> → What is Exception?

An exception is an event, which occurs during the execution of a program that disrupts the normal flow of the program's instructions. In general, when a Python script encounters a situation that it cannot cope with, it raises an exception. An exception is a <u>Python object that represents an error</u>.

When a Python script raises an exception, it must either handle the exception immediately otherwise it terminates and quits.

Syntax

Here is simple syntax of *try....except...else* blocks –

```
try:
   You do your operations here;
   ......................
except ExceptionI:
   If there is ExceptionI, then execute this block.
except ExceptionII:
   If there is ExceptionII, then execute this block.
   ......................
else:
   If there is no exception then execute this block.
```

The *except* Clause with No Exceptions

You can also use the except statement with no exceptions defined as follows –

```
try:
   You do your operations here;
   ......................
except:
   If there is any exception, then execute this block.
   ......................
else:
   If there is no exception then execute this block.
```

The try-finally Clause

You can use a **finally:** block along with a **try:** block. The finally block is a place to put any code that must execute, whether the try-block raised an exception or not. The syntax of the try-finally statement is this –

```
try:
   You do your operations here;
   ......................
   Due to any exception, this may be skipped.
finally:
   This would always be executed.
   ......................
```

1) Create a program that divides two numbers, if the division is by zero, raise an error
2) Create a program that divides two variables, if one of them is a string, raise an error
3) Create a program that handles the two previous error, but this time, use Python Exceptions