# Coding exercise

## Summary

You are given the task to determine, in a simplistic way for now, the state of play in the property market. The complexity is not an issue at this stage as you have at your disposal quite powerful servers, storage space, and the results are not required to be real-time.

The only requirement is that it must be built in Python (using classes whenever possible) and MSSQL (Microsoft SQL).

## Data

1. The url https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads contains the yearly price paid data for all properties in the UK since 1997. We will only focus from 2013 onwards.
2. The column names are specified here: https://www.gov.uk/guidance/about-the-price-paid-data. These refer to the data you would download from the previous link.
3. The house price index data is found here: https://www.gov.uk/government/statistical-data-sets/uk-house-price-index-data-downloads-march-2021
4. The average price per region is found here: http://publicdata.landregistry.gov.uk/market-trend-data/house-price-index-data/Average-prices-2021-03.csv?utm_medium=GOV.UK&utm_source=datadownload&utm_campaign=average_price&utm_term=9.30_19_05_21
5. The CPI index is found here: https://www.ons.gov.uk/economy/inflationandpriceindices/datasets/consumerpriceindices
6. The coordinates (the centres) of the postcodes can be downloaded from here: https://opendata.camden.gov.uk/Maps/National-Statistics-Postcode-Lookup-UK-Coordinates/77ra-mbbn

## Goal of the analysis

The goal is to build a module that loads this data and tries to answer the questions that are listed below. The module should have classes for a transaction, a postcode, and a property. While this might look over-complicating, we want to enrich this module with more functionality in the future, so we are willing to take a hit in complexity up front. We would expect that for now there is a property class, a postcode class that contains a number of properties, and a transaction class that refers to a property.

Please note that you do not need to worry about adding the downloading process in your solution. Assume that these files would be already downloaded and available to the script. Also, do not be concerned about how the results look aesthetically. So, download the files from the list above, put them in a directory, and write your scripts. Try to add some comments whenever possible, though this is not needed. For the questions below, describe your SQL schema but for the coding exercise you can use the CSV files you have downloaded. There is no need to populate a database. In addition to creating a Jupiter notebook, please prepare a full application with your class structures.

You should be able to return your code structure, as well as a 1-page summary of your approach.

## Questions.

- There is no unique identifier for a property in the data. How would you approach this to come up with a column that can be used as a unique id for each property? Would you combine any

columns for instance? Can you test your method that it returns unique values? Are there any issues?

- Once you have defined a property unique id (unfortunately this doesn't exist in the data so it needs to be defined by you), how would you store the data in your SQL database? What table structure would you use?
- How would you work on improving the performance of the queries? Would you use primary keys, indexes?
- Can you write a query that returns the transactions that took place in `EC1A` between `2018-04-01` and `2019-12-31`?
- Utilizing the class structure in python you have defined, create methods to
    - return the number of properties that have been sold in a postcode, and which transaction_ids refer to those. Test with `ST10 4BS`. Were there 2 transaction in 2019?
    - Given a transaction_id, return which property it refers to. Test with `{7C2D0701-0253-4963-E053-6B04A8C07B97}`. Does it return a property in Cornwall?
- Which postcodes have seen the highest increase in transactions during the last 5 years? No need to do the analysis at the full postcode level; the first part is sufficient. Thus instead of e.g. `SE13 5HA`, consider only `SE13`.
- Can you come up with an indication of a 'migration' metric in the UK? Perhaps it would be best if you combined the postcode coordinates dataset for this exercise (url #6). Where is the 'centre of gravity' in terms of number of transactions of the population moving to every year? Where is the 'centre of gravity' in terms of value moving to? For now, consider that the 'centre of gravity' is a weighted average function, so for each year determine the weighted average of the coordinates based on number of transactions, or value.
- Assume that `EC1A` is the centre of London. Can you plot the average transaction price of a postcode as a function of distance from `EC1A`? There are numerous postcodes within `EC1A`, so try to use the centre of all of the postcodes within it. The distance between two points `a`, and `b`, with coordinates $(x_1, y_1)$ and $(x_0, y_0)$ is $d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$
- Can you find any correlation between the average house prices (url #4), and a CPI indicator (url #5)? Do not be concerned about the lag in the CPI and the house prices not being synchronized. Assume that all data points are normalized (an average price in Aug 2014 can be matched to an Aug 2014 entry in the CPI dataset). Also, just use United Kingdom as a region from the first file.

## Suggestions.

- If storage space and/or processing is an issue, use only 2019 and 2020 from the transaction data.
- Do not try to find the best solution. A decent solution is fine.
- Try to generalize your solution. For example have a parameter in your functions that the user can use to specify a different postcode, or transaction id.
- If you make assumptions, please make them clear in the code or report.