Notation and Terminology

Here are some common phrases that appear in the instructions for computational exercises, with examples of the corresponding code.

Files

- Open X for reading (or writing) using file handle Y:

  ```
  Y = open(X)
  Y = open(X, 'w')
  ```

Loops

- Use X to loop over Y:
  or: Loop over Y using X:

  ```
  for X in Y:
  ```

Calling functions

- Set Z to the result of calling F on X and Y:

  ```
  Z = F(X, Y)
  ```

- Call F on X and Y:

  ```
  F(X, Y)
  ```

Calling methods of an object

- To emphasize something is a method call it will be shown with a dot; for example, method `met` would be shown as `.met()`. That's a reminder that the actual call will have the form `obj.met()` where `obj` is the object whose method is being called.
- Set `R` to the result of calling the `.met()` method of `obj` with argument `A`
  or: Set `R` to the result of calling `obj`'s `.met()` method with argument `A`
  or: Set `R` to the result of calling `.met()` on `obj` with argument `A`:

  ```
  R = obj.met(A)
  ```

- Call the `.met()` method of `obj` with argument `A`
  or: Call `obj`'s `.met()` method with argument `A`
  or: Call `.met()` on `obj` with argument `A`:

  ```
  obj.met(A)
  ```

Working with dictionaries

- Set the value of X in Y to Z:

  ```
  Y[X] = Z
  ```

- Set Z to the value of X in Y:

  ```
  Z = Y[X]
  ```

- The keys or values of X:

```
X.keys()
X.values()
```

Working with Pandas Series and DataFrames

- Set element Y of series S to X:

  ```
  S[Y] = X
  ```

- Set X to element Y of series S; X will be a scalar:

  ```
  X = S[Y]
  ```

- Set column C of dataframe D to X:

  ```
  D[C] = X
  ```

- Set X to column C of dataframe D; X will be a series:

  ```
  X = D[C]
  ```

- Set X to row R of dataframe D; X will be a series:

  ```
  X = D.loc[R]
  ```

- Set X to the result of using selector .loc[S] on Y
  or: Set X to the result of using selector .iloc[S] on Y:

  ```
  X = Y.loc[S]
  X = Y.iloc[S]
  ```