Tutorial 3f – Combine Core Transform in Apache Beam:

Combine:

- `Combine` is a Beam transform for combining collections of elements or values in your data.
- `Combine` has variants that work on entire `PCollection`s, and some that combine the values for each key in `PCollection`s of key/value pairs.
- When you apply a `Combine` transform, you must provide the function that contains the logic for combining the elements or values.
- The combining function should be commutative and associative.
- The Beam SDK also provides some pre-built combine functions for common numeric combination operations such as sum, min, and max.
- complex combination operations might require you to create a subclass of `CombineFn` that has an accumulation type distinct from the input/output type.

Advanced combinations using CombineFn:

- A general combining operation consists of four operations. When you create a subclass of `CombineFn`, you must provide four operations by overriding the corresponding methods:
  - **Create Accumulator -** creates a new "local" accumulator
  - **Add Input -** adds an input element to an accumulator, returning the accumulator value.
  - **Merge Accumulators -** merges several accumulators into a single accumulator; this is how data in multiple accumulators is combined before the final calculation.
  - **Extract Output -** performs the final computation.

Three types of Aggregator function is supported by beam. They are.

CombineGlobally:

- Combines all elements in a collection.

CombinePerKey:

- Combines all elements for each key in a collection.

CombineValues:

- Combines an iterable of values in a keyed collection of elements.

Resources:

- https://beam.apache.org/documentation/programming-guide/#combine
    - https://beam.apache.org/documentation/transforms/python/aggregation/combineglobally/
    - https://beam.apache.org/documentation/transforms/python/aggregation/combineperkey/
    - https://beam.apache.org/documentation/transforms/python/aggregation/combinevalues/