A Guide to Apache Beam for Humans

- **Runs jobs on Google DataFlow**
  - DataFlow processes large amounts of data
  - batch data processing - doing the job by breaking it down into "batches" of data and processing each batch concurrently

- **pipeline** - a data processing task
  - consists of PCollections of data and Transforms that perform an action on the data

- **PCollection** - a collection of data elements
  - think of these as holding a bag of data - there is no sense of "order"

- **Transform** - an operation that is applied to each PCollection, and returns a transformed PCollection
  - **Element-wise transform** - maps an input value to a varying amount of output values
    - i.e: ParDo, Map, FlatMap
  - **Aggregation transform** - maps multiple input values into one or zero output values
    - i.e: GroupByKey, CoGroupByKey
  - **Composite transform** - element-wise + aggregation transforms
  - Python overloaded symbols:
    
    - | - "apply" a transform onto a PCollection, i.e:
      
      | collection | beam.ParDo
      | bag of apples | take.a.bite.out.of

    - >> - "name" a transform, i.e:
      
      "Read a file" >> beam.io.Read
      "Eat apple" >> take.a.bite.out.of

- **Sink** - a place to write the data back to

**The ParDo Transform**

- ParDo - "Parallel Do", do some function in parallel
- Input: a PCollection, and a user-defined function
- Output: a PCollection where the user-defined function has been applied to all values inside the input PCollection

**The GroupByKey Transform**

- GroupByKey - group values together based on a specific key value
- Input: a PCollection of key-value pair tuples, (first value is a key, and second value is the corresponding value)
Output: a PCollection of tuples, where the first value is the key, and the second value is a list of values that correspond with the key