**[Session 1] Python Basics Supplement**

**Indexing & Slicing of List / Tuple / String**

These three datatypes have in the following commons:

* **Can be iterated**: We can traverse each element (or character). (e.g. for statement)
* **Can be indexed:** We can access ith element directly, with [i]. (e.g. (1, 3, 2, 1)[2] => 2)
* **Can be sliced:** We can get a sub-data from the original, according to certain rules

Indexing and slicing are very important (and powerful) operations in Python. You should be accustomed to using these.

**(Index Rule)**

1. **Index starts at 0**. In order to access to ith element, we use (Data)[i-1], not (Data)[i].
2. Index out of range will cause error. For example, (1, 2, 3)[3] is an invalid indexing
3. Negative index can be valid. For example, list[-1] is the last element, and list[-2] is the second-last element. Note that, if the number of elements is n, then index under -n is invalid.
   * So, For some list with n items, list[i] = list[-(n-i)]

**(Slicing Rule)**

1. Basically, for some iterable data, A, use A[start:end:step]. All of them can be omitted. (even A[:] or A[::] are ok) The default of step is 1. For a list A, with n items
   1. A[:end] = A[0:end] = A[0:end:1]
   2. A[start:] = A[start:n-1] = A[start:n-1:1]
   3. A[:] = A[::] = A
2. In A[start:end], A[end] is not included.

We uploaded some exercises for practice of this.