

5. Accessing Tuples

❖ Accessing tuple elements using positive and negative indexing.

Positive Indexing

- Starts from **0** (first element), **1, 2, ...** up to **$n-1$** where n is tuple length.
- Example:

```
tup = (10, 20, 30)
```

```
tup[0]
```

```
tup[2]
```

Negative Indexing

- Starts from **-1** for the **last** element, then **-2, -3, ...** up to **- n** .
- Example:

```
tup = (10, 20, 30)
```

```
tup[-1]
```

```
tup[-3]
```

- **Positive indices** start at 0 and go up to $\text{length}-1$.

- **Negative indices** map back-to-front: -1 is last, -2 second-last, etc.

❖ Slicing a tuple to access ranges of elements.

Syntax: `tuple[start:stop:step]`

- **start** (inclusive): index where slicing begins
- **stop** (exclusive): index where slicing ends
- **step** (optional): stride between elements (can be negative)

`t = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)`

- `t[2:6]` \rightarrow (2, 3, 4, 5)
- `t[:4]` \rightarrow (0, 1, 2, 3)
- `t[5:]` \rightarrow (5, 6, 7, 8, 9)
- `t[:]` \rightarrow full copy (0...9)

negative Indexing + Slicing

- `t[-3:]` \rightarrow last 3 elements (7,8,9)
- `t[:-3]` \rightarrow everything except last 3 (0...6)
- `t[-3:-1]` \rightarrow (7,8)