

## 5 ) Accessing Tuples

- **Accessing tuple elements using positive and negative indexing.**

### Introduction to Tuple Indexing

Tuples are ordered sequences, meaning each element has a fixed position. Python provides two indexing methods:

- Positive indexing (left to right, starting at 0)
- Negative indexing (right to left, starting at -1)

### 1. Positive Indexing

#### Characteristics:

- Starts at 0 for the first element
- Increases by 1 for each subsequent element
- Last element at index  $\text{len}(\text{tuple}) - 1$

```
animals = ("cat", "dog", "elephant", "lion")
```

```
print(animals[0]) # Output: cat
```

```
print(animals[2]) # Output: elephant
```

## 2. Negative Indexing

### Characteristics

- Starts at -1 for the last element.
- Decreases by 1 moving leftward
- First element at index `-len(tuple)`

```
animals = ("cat", "dog", "elephant", "lion")
```

```
print(animals[-1]) # Output: lion
```

```
print(animals[-3]) # Output: dog
```

- **Slicing a tuple to access ranges of elements**

### Introduction to Tuple Slicing

Tuple slicing extracts a subsequence (sub-tuple) from a tuple using the 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)
- 
- Returns a new tuple (original remains unchanged)
  - Works similarly to list slicing but produces immutable results

## Indexing Systems

Slicing works with both indexing schemes:

- **Positive indices:** Count from left (0 = first element)
- **Negative indices:** Count from right (-1 = last element)

## Positive indices

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

```
# Slicing examples
```

```
print("t[2:6] →", t[2:6]) # Output: (2, 3, 4, 5)
```

```
print("t[:4] →", t[:4]) # Output: (0, 1, 2, 3)
```

```
print("t[5:] →", t[5:]) # Output: (5, 6, 7, 8, 9)
```

```
print("t[:] →", t[:]) # Output: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) — full
```

## Negative indices

```
# Tuple initialization
```

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

```
# Slicing with negative indices
```

```
print("t[-4:-1] →", t[-4:-1]) # Output: (6, 7, 8)
```

```
print("t[-6:] →", t[-6:]) # Output: (4, 5, 6, 7, 8, 9)
```

```
print("t[:-7] →", t[:-7]) # Output: (0, 1, 2)
```

```
print("t[-5:-2] →", t[-5:-2]) # Output: (5, 6, 7)
```

```
print("t[::-1] →", t[::-1]) # Output: (9, 8, 7, 6, 5, 4, 3, 2, 1, 0) → reversed tuple
```