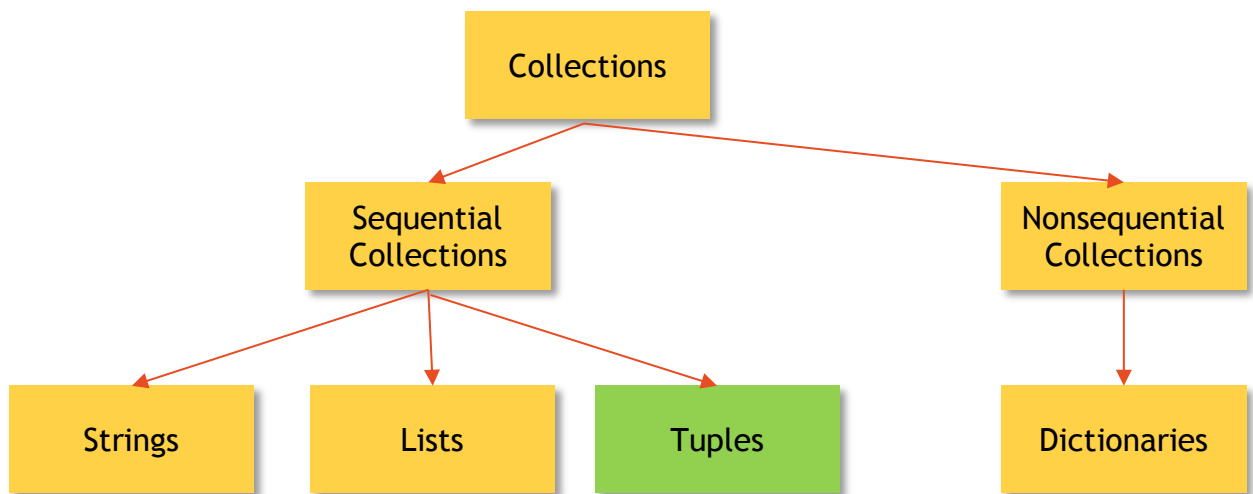


**tuples**

Like lists, but immutable



## Tuple Literals

```
T = (1, 2, 3)
```

```
type(T)
```

→ <class 'tuple'>

**One item - gives an int**

```
T = (1)
```

```
type(T)
```

→ <class 'int'>

**Need to use comma**

```
T = (1,)
```

```
type(T)
```

→ <class 'tuple'>

## Can omit parenthesis

```
T = 1, 2, 3
```

→ T: (1, 2, 3)

```
T = 1,
```

```
len(T)
```

→ T:(1,)

→ 1

## How multiple assignment really works

```
a, b, c = 1, 2, 3
```

is equivalent to

```
(x, y, z) = (1, 2, 3)
```

## Can't sort - use sorted (tuples are immutable)

```
T = (2, 3, 1, 6, 4)
```

```
sorted(T)
```

→ a list: [1, 2, 3, 4, 6]

Save it back to the tuple

```
T = tuple(sorted(T))
```

→ T: (1, 2, 3, 4, 6)

## Why Use Tuples

**Dictionary keys must be immutable**

```
D = {}
```

```
D[[1, 2, 3]] = 12
```

→ Error: unhashable type: 'list'

```
D[(1, 2, 3)] = 12
```

→ No problem

To prevent inadvertent changes to collection