In [1]:
```
#Tuple
a = ("apple", "banana", "cherry")
print(a)
```

```
('apple', 'banana', 'cherry')
```

In [2]:
```
#Allow Duplicates
t = ("apple", "banana", "cherry", "apple", "cherry")
print(t)
```

```
('apple', 'banana', 'cherry', 'apple', 'cherry')
```

In [3]:
```
#Tuple Length
c = ("apple", "banana", "cherry")
print(len(c))
```

```
3
```

In [7]:
```
#Create Tuple With One Item
a = ("apple",)
print(type(a))
```

```
<class 'tuple'>
```

In [6]:
```
a = ("apple")
print(type(a))
```

```
<class 'str'>
```

In [8]:
```
#type()
mytuple = ("apple", "banana", "cherry")
print(type(mytuple))
```

```
<class 'tuple'>
```

In [9]:
```
#The tuple() Constructor
tuple1 = tuple(("apple", "banana", "cherry")) # note the double round-brackets
print(tuple1)
```

```
('apple', 'banana', 'cherry')
```

In [10]:
```
#Access Tuple Items
W = ("apple", "banana", "cherry")
print(W[1])
```

```
banana
```

In [11]:
```
#Negative Indexing
A = ("apple", "banana", "cherry")
print(A[-1])
```

```
cherry
```

In [12]:
```
#Range of Indexes
t = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
print(t[2:5])
```

```
('cherry', 'orange', 'kiwi')
```

In [ ]:
```

```