1 Tuples

1.1 Creating and adding to tuples

Tuples are like lists in many ways, apart from they cannot be changed - they are immutable. Tuples are defined used curved brackets (unlike square brackets for lists). Tuples may be returned, or be required, by various functions in Python, and tuples may be chosen where immutability would be an advantage (for example, to help prevent accidental changes).

```
my_tuple = ('Hobbit','Elf','Ork','Dwarf')
print (my_tuple[1])

OUT:
Elf
```

It is possible to add to a tuple. Note that if adding a single item an additional comma is used to indicate to Python that the variable being added is a tuple.

```
my_tuple = my_tuple + ('Man',)
my_tuple += ('Wizard','Balrog') # Note that the += is short hand to add something to itslef
print (my_tuple)

OUT:
('Hobbit', 'Elf', 'Ork', 'Dwarf', 'Man', 'Wizard', 'Balrog')
```

It is not possible to change or delete an item in a tuple. To change or delete a tuple a new tuple must be built (but if this is going to happen then a list would be a better choice).

```
my_new_tuple = my_tuple[0:2] + ('Goblin',) + my_tuple[4:]
print (my_new_tuple)

OUT:
('Hobbit', 'Elf', 'Goblin', 'Man', 'Wizard', 'Balrog')
```

1.2 Converting between tuples and lists, and sorting tuples

A tuple may be turned into a list. We can recognise that it is a list by the square brackets.

```
my_list = list(my_tuple)
print (my_list)

OUT:
['Hobbit', 'Elf', 'Ork', 'Dwarf', 'Man', 'Wizard', 'Balrog']
```

A list may also be converter into a tuple.

```
my_list.sort()
my_new_tuple = tuple(my_list)
print (my_new_tuple)

OUT:
my_list.sort()

my_new_tuple = tuple(my_list)

print (my_new_tuple)

('Balrog', 'Dwarf', 'Elf', 'Hobbit', 'Man', 'Ork', 'Wizard')
```

In the above example we sorted a list and converted it to a tuple. Tuples cannot be changed (apart from being added to), so it is not possible to directly sort a tuple. The sorted command will act on a tuple to

sort it, but returns a list.

```
print (sorted(my_tuple))
```

```
OUT:
['Balrog', 'Dwarf', 'Elf', 'Hobbit', 'Man', 'Ork', 'Wizard']
```

The tuple may be sorted by converting back to a list in a single step (but think of using lists, rather than tuples, is sorting will be common.

```
my_tuple = tuple(sorted(my_tuple))
print (my_tuple)
```

```
OUT:
('Balrog', 'Dwarf', 'Elf', 'Hobbit', 'Man', 'Ork', 'Wizard')
```