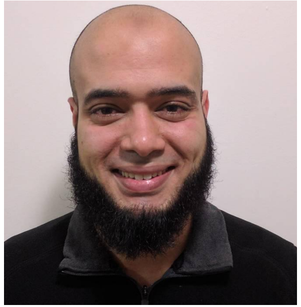
Python Programming

Tuples

Mostafa S. Ibrahim

Teaching, Training and Coaching since more than a decade!

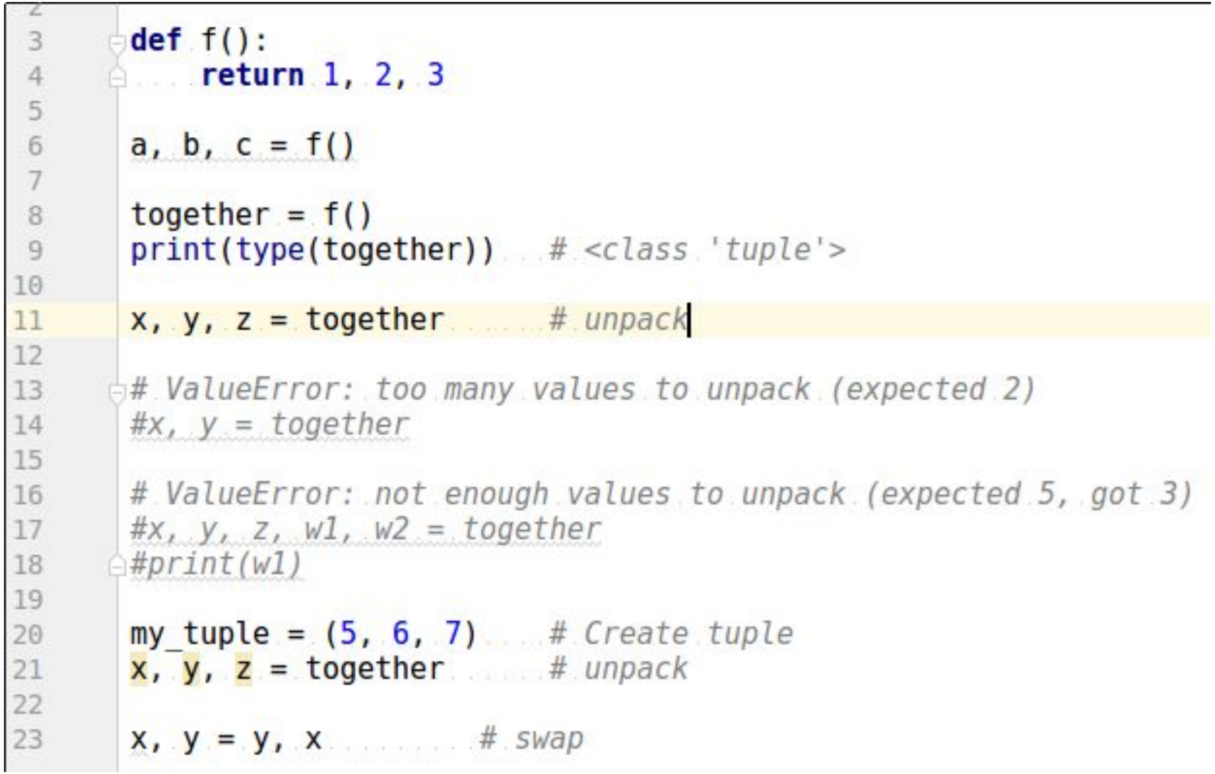
Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Tuples

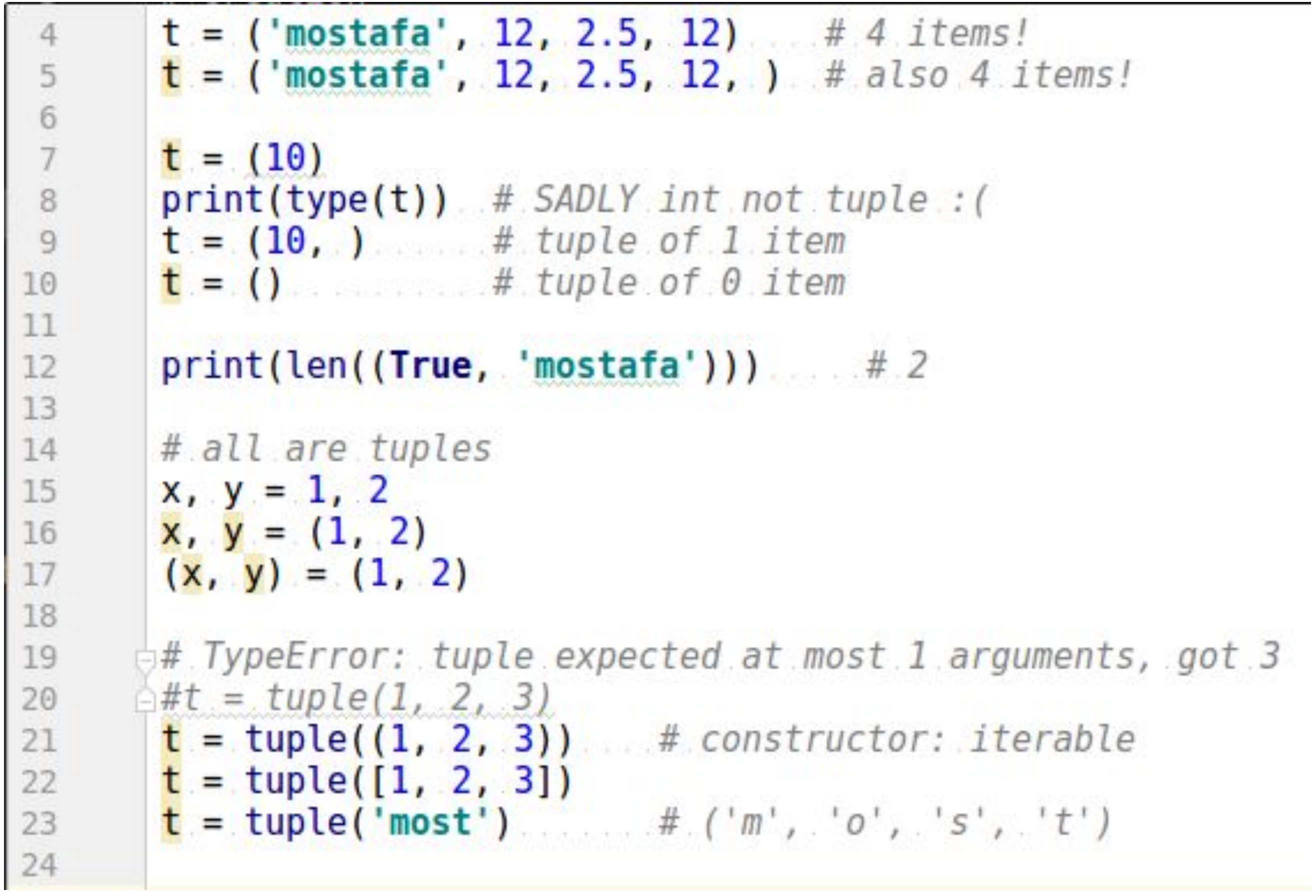
- Another an **ordered** collection of objects
  - Some pronounce it as though it were spelled **“too-ple”**
  - and others as though it were spelled **“tup-ple”**
- Several similarities with list
  - Iterating, Indexing, slicing, comparisons, multiple elements: min(), max(), sorted()
- More:
  - A im**mutable** data type: We can’t change or delete ith item
    - Many methods don’t exist: append, insert, remove
    - Though we can change the item internal content if mutable!
  - Fast iteration (visible with large collection)
  - Key with Dict. List can’t
  - Multiple return from a function or multiple assignments

Recall



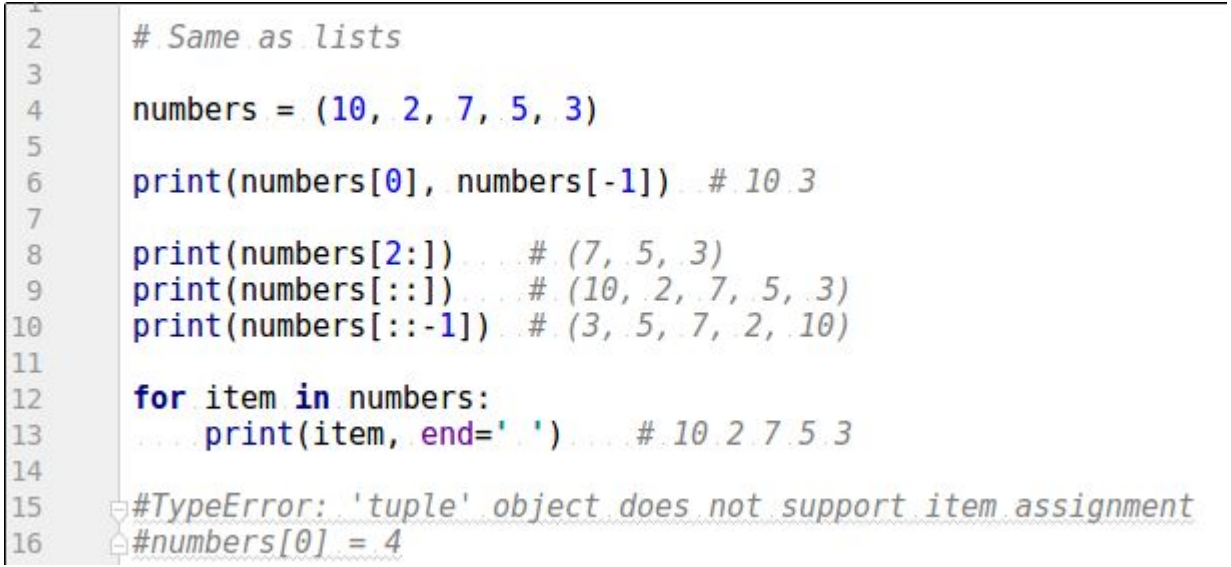
```python
def f():
    return 1, 2, 3

a, b, c = f()

together = f()
print(type(together))   # <class 'tuple'>

x, y, z = together      # unpack

# ValueError: too many values to unpack (expected 2)
#x, y = together

# ValueError: not enough values to unpack (expected 5, got 3)
#x, y, z, w1, w2 = together
#print(w1)

my_tuple = (5, 6, 7)    # Create tuple
x, y, z = together      # unpack

x, y = y, x         # swap
```

Creation



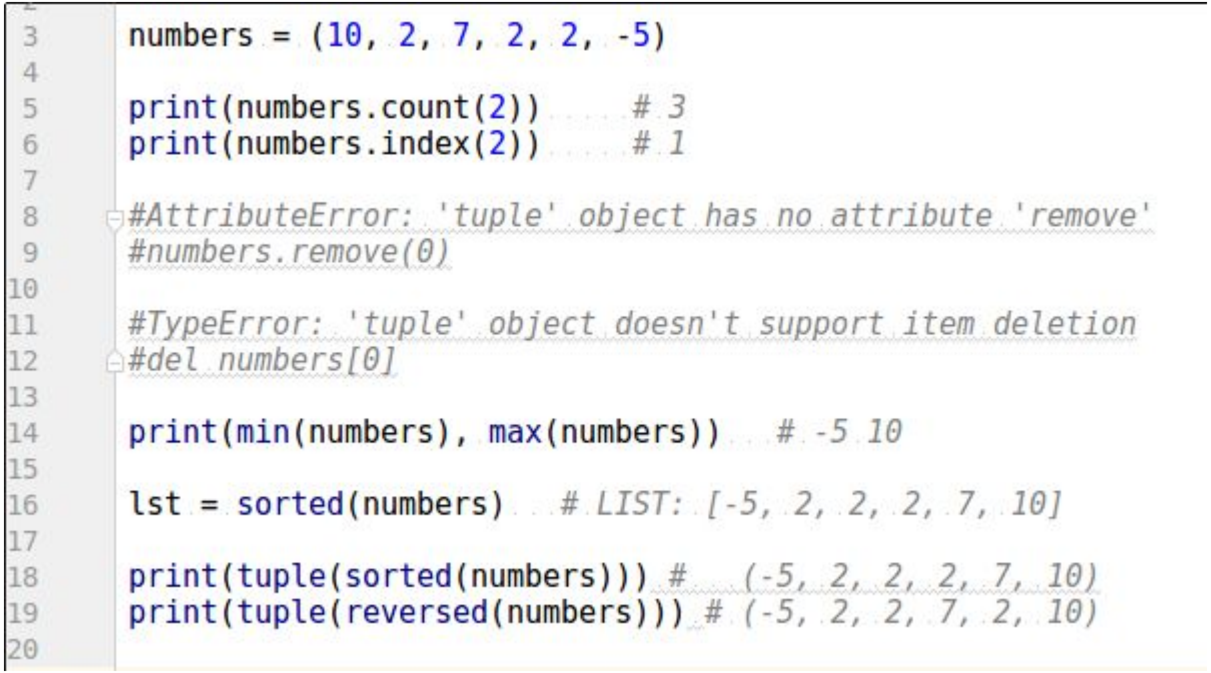
```
t = ('mostafa', 12, 2.5, 12)    # 4 items!
t = ('mostafa', 12, 2.5, 12, )  # also 4 items!

t = (10)
print(type(t))  # SADLY int not tuple :(
t = (10, )      # tuple of 1 item
t = ()          # tuple of 0 item

print(len((True, 'mostafa')))     # 2

# all are tuples
x, y = 1, 2
x, y = (1, 2)
(x, y) = (1, 2)

# TypeError: tuple expected at most 1 arguments, got 3
#t = tuple(1, 2, 3)
t = tuple((1, 2, 3))    # constructor: iterable
t = tuple([1, 2, 3])
t = tuple('most')       # ('m', 'o', 's', 't')
```

Indexing and Slicing



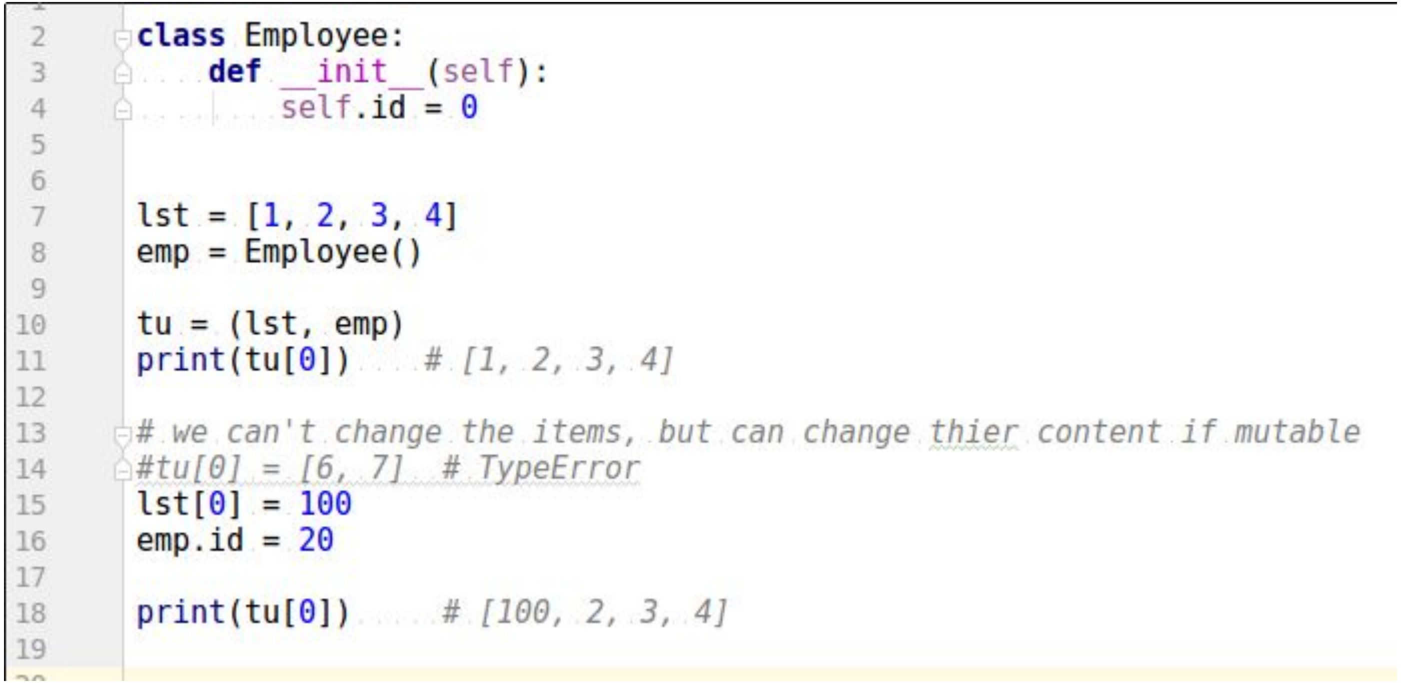
```python
# Same as lists

numbers = (10, 2, 7, 5, 3)

print(numbers[0], numbers[-1])  # 10 3

print(numbers[2:])     # (7, 5, 3)
print(numbers[::])     # (10, 2, 7, 5, 3)
print(numbers[::-1])  # (3, 5, 7, 2, 10)

for item in numbers:
    print(item, end=' ')    # 10 2 7 5 3

#TypeError: 'tuple' object does not support item assignment
#numbers[0] = 4
```

Methods and Functions



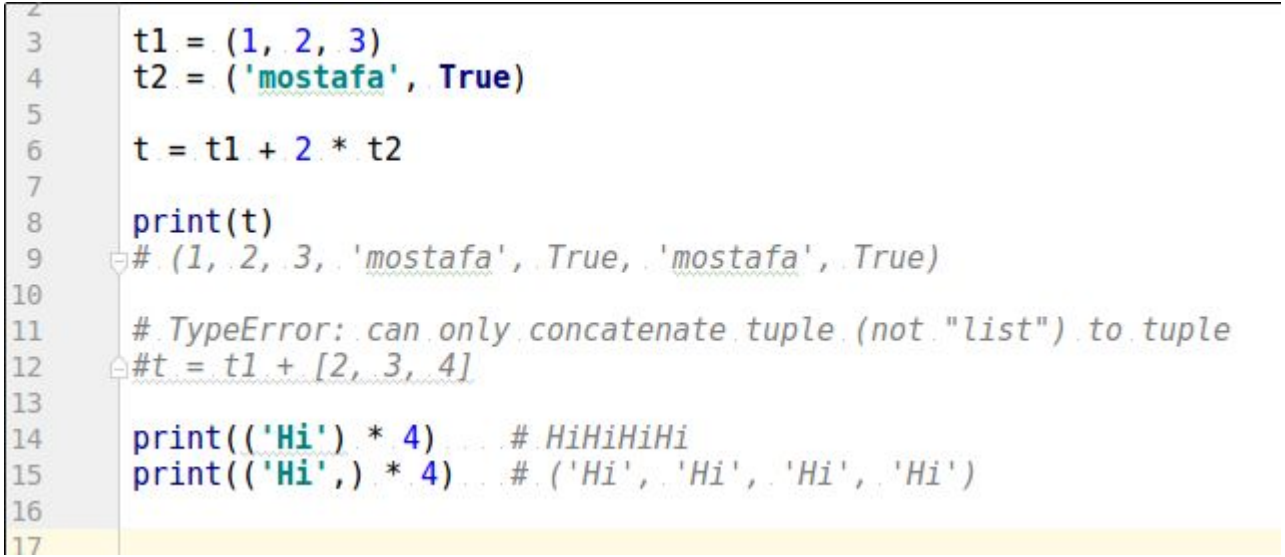
```
numbers = (10, 2, 7, 2, 2, -5)

print(numbers.count(2))     # 3
print(numbers.index(2))     # 1

#AttributeError: 'tuple' object has no attribute 'remove'
#numbers.remove(0)

#TypeError: 'tuple' object doesn't support item deletion
#del numbers[0]

print(min(numbers), max(numbers))   # -5 10

lst = sorted(numbers)   # LIST: [-5, 2, 2, 2, 7, 10]

print(tuple(sorted(numbers))) #   (-5, 2, 2, 2, 7, 10)
print(tuple(reversed(numbers))) # (-5, 2, 2, 7, 2, 10)

```

Change or not?!



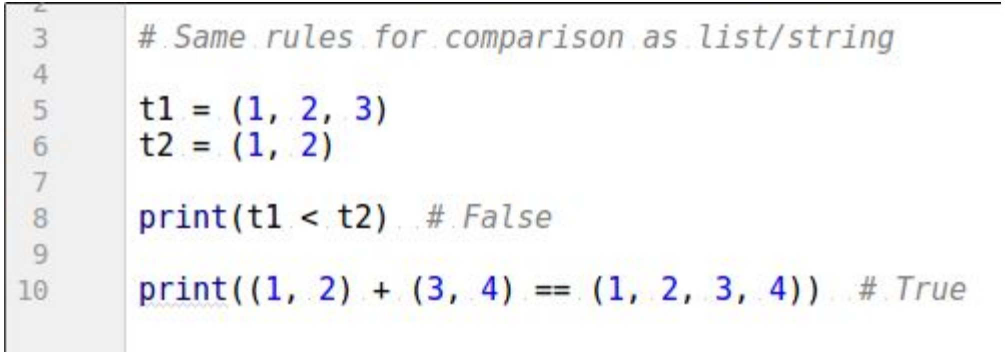
```python
class Employee:
    def __init__(self):
        self.id = 0


lst = [1, 2, 3, 4]
emp = Employee()

tu = (lst, emp)
print(tu[0])    # [1, 2, 3, 4]

# we can't change the items, but can change thier content if mutable
#tu[0] = [6, 7]  # TypeError
lst[0] = 100
emp.id = 20

print(tu[0])     # [100, 2, 3, 4]

```

+ And * Operators



```python
t1 = (1, 2, 3)
t2 = ('mostafa', True)

t = t1 + 2 * t2

print(t)
# (1, 2, 3, 'mostafa', True, 'mostafa', True)

# TypeError: can only concatenate tuple (not "list") to tuple
#t = t1 + [2, 3, 4]

print(('Hi') * 4)    # HiHiHiHi
print(('Hi',) * 4)   # ('Hi', 'Hi', 'Hi', 'Hi')

```

Comparisons



```
# Same rules for comparison as list/string

t1 = (1, 2, 3)
t2 = (1, 2)

print(t1 < t2)  # False

print((1, 2) + (3, 4) == (1, 2, 3, 4))  # True
```

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”