Tuples

- A tuple consists of a number of values separated by commas

```
>>> t = 'intro to python', 'amey karkare', 101
```

- Empty and Singleton Tuples

Nested Tuples

- Tuples can be nested

- Note that course tuple is copied into student.
  - Changing course does not affect student

Length of a Tuple

- len function gives the length of a tuple

```
>>> course = 'Python', 'Amey', 101
>>> student = 'Prasanna', 34, course
>>> empty = ()
>>> singleton = 1,
>>> len(empty)
0
>>> len(singleton)
1
>>> len(course)
3
>>> len(student)
3
```

More Operations on Tuples

- Tuples can be concatenated, repeated, indexed and sliced

```
>>> 2*course1
('Python', 'Amey', 101, 'Python', 'Amey', 101)
```

Unpacking Sequences

- Strings and Tuples are examples of sequences
  - Indexing, slicing, concatenation, repetition operations applicable on sequences
- Sequence Unpacking operation can be applied to sequences to get the components
  - *Multiple assignment* statement
  - LHS and RHS must have equal length

Unpacking Sequences

```
>>> student
('Prasanna', 34, ('Python', 'Amey', 101))
>>> name,roll,regdcourse=student
>>> name
```

Mutable and Immutable Types

- Tuples and List types look very similar
- However, there is one major difference: Lists are **mutable**
  - Contents of a list can be modified
- Tuples and Strings are **immutable**
  - Contents can not be modified

Summary of Sequences

Operation	Meaning
seq[i]	i-th element of the sequence
len(seq)	Length of the sequence
seq1 + seq2	Concatenate the two sequences
num*seq seq*num	Repeat seq num times
seq[start:end]	slice starting from **start**, and ending at **end-1**
e in seq	True if e is present is seq, False otherwise
e not in seq	True if e is not present is seq, False otherwise
for e in seq	Iterate over all elements in seq (e is bound to one element per iteration)

Sequence types include String, Tuple and List.
Lists are mutable, Tuple and Strings immutable.

Summary of Sequences

- For details and many useful functions, refer to:

https://docs.python.org/3.2/tutorial/datastructures.html

How do you assign a tuple of length 1 to the variable a?

A. a = (1,)

B. a = [1]

C. a = tuple(1)

D. a = 1

E. a = 1,

Suppose you have the following tuple definition:

t = ('foo', 'bar', 'baz')
Which of the following statements replaces the second element ('bar') with the string 'qux':

A. It’s a trick question—tuples can’t be modified.

B. t(1) = 'qux'

C. t[1] = 'qux'

D. t[1:1] = 'qux'

Consider this assignment statement:

a, b, c = (1, 2, 3, 4, 5, 6, 7, 8, 9)[1::3]

Following execution of this statement, what is the value of b:

A. 4

B. 5

C. 2

D. 6

Assume x and y are assigned as follows:

x = 5
y = -5
What is the effect of this statement:

x, y = (y, x)[::-1]

A. The values of x and y are swapped

B. Both x and y are -5

C. Both x and y are 5

D. The values of x and y are unchanged