Lists: basic-level questions

> If you need help reviewing Lists, take a look at these resources:
>
> - Albert's and Robert's slides (https://docs.google.com/presentation/d/1skXespdgvJRzwUmW1oSxHP54_bKkJPGkF1-N9poNRt4/edit#slide=id.p)

Each question has a "Toggle Solution" button -- click it to reveal that question's solution.

What would Python print?

Question 1

```
>>> L = [1, 2, 3, 4]
>>> L[0]
______
>>> L[100]
______
>>> L[-1]
______
>>> L[2] = 100
>>> L
______
```

Toggle Solution

Question 2

```
>>> L = [1, 2, 3, 4]
>>> L[1:3]
______
>>> L[:2]
______
>>> L[1:]
______
>>> L[:]
______
>>> L[0:3:2]
______
>>> L[::-1]
______
```

Toggle Solution

Question 3

```
>>> L = [1, 2, 3, 4]
>>> [1, 2] + [3, 4]
______
>>> [1, 2] * 2
______
>>> L.append(5)
>>> L
______
>>> L.extend([6, 7])
>>> L
______
>>> L.index(5)
______
>>> L.remove(3)
>>> L
______
>>> L.pop()
______
>>> L
______
```

Toggle Solution

Code-Writing questions

Question 4

Implement a function `reverse` that takes a list as an argument and reverses the list. You should mutate the original list, without creating any new lists. Do NOT return anything.

```
def reverse(L):
    """Reverses L in place (i.e. doesn't create new lists).

    >>> L = [1, 2, 3, 4]
    >>> reverse(L)
    >>> L
    [4, 3, 2, 1]
    """
    "*** YOUR CODE HERE ***"
```

Toggle Solution

Question 5

Implement a function `map_mut` that takes a list as an argument and maps a function `f` onto each element of the list. You should mutate the original lists, without creating any new lists. Do NOT return anything.

```
def map_mut(f, L):
    """Mutatively maps f onto each element in L.

    >>> L = [1, 2, 3, 4]
    >>> map_mut(lambda x: x**2, L)
    >>> L
    [1, 4, 9, 16]
    """
    "*** YOUR CODE HERE ***"
```

Toggle Solution