Lists are Sequences

- Like a string, a list is a sequence of values
- In a string the values are characters
- In a list the values can be any type, even other lists
- The values in a list are called elements
Creating a List

- The easiest way to create a list is to enclose some items in square brackets.

- Some examples:
  ```python
  nums = [12, 23, 34, 45, 56, 67, 78]
  strs = ['one', 'two', 'three']
  ```

- The empty list is created with empty brackets.
  ```python
  empty = []
  ```
Accessing List Elements

- The syntax for accessing elements in a list is the same as for accessing the characters of a string
- The index operator
  \texttt{print(nums[0])}
- Remember indices start with 0!
Lists are Mutable

- Unlike string, we can change the value of an element in a list
  
  \[ \text{nums}[0] = 98 \]

- Now the element at index 0, which used to have the value 12, has the value 98
List Indices

- List indices work like string indices
- Any integer expression can be used
- If you try to read or write from an element that doesn't exist you get an IndexError
- You can use negative indices

```python
print(nums[-1])
```
Traversing a List

• The most common way to traverse the elements of a list is with a `for` loop

```python
for num in nums:
    print(num)
```
List Operations

- The '+' operator concatenates lists

```python
a = [1, 2, 3]
b = [4, 5, 6]
c = a + b
print(c)
```
List Operations

• The '/*' operator repeats a list a given number of times

```python
    t = [0] * 4
print(t)
```
List Slices

• The slice operator also works on lists
  
  \[ t = ['a', 'b', 'c', 'd', 'e'] \]
  
  \[ \text{print}(t[1:4]) \]

• If you omit the first index the slice starts at the beginning

• If you omit the last index the slice goes to the end
List Methods

- The 'append' method allows us to add new items to the end of an existing list
  ```python
t.append('f')
  print(t)
  ```

- The same thing can be done with the '+=' operator
  ```python
t += ['g']
  ```
List Methods

• The 'extend' method takes a second list as an argument and appends all of its items to the first list

```python
t1 = [12, 23, 34]
t2 = [45, 56, 67]
t1.extend(t2)
print(t1)
```