Table 2-1. Methods found in `list` or `tuple` (methods implemented by `object` are omitted for brevity)

Method call	Expression	Description	list	tuple
s.__add__(s2)	s + s2	Concatenation	●	●
s.__contains__(e)	e in s		●	●
s.__delitem__(p)		Remove item at position p	●	
s.__getitem__(p)	s[p]	Get item at position p	●	●
s.__getnewargs__()		Support for optimized serialization with `pickle`		●
s.__iadd__(s2)	s += s2	In-place concatenation	●	
s.__imul__(n)	s *= n	In-place repeated concatenation	●	
s.__iter__()		Get iterator	●	●
s.__len__()	len(s)	Number of items	●	●
s.__mul__(n)	s * n	Repeated concatenation	●	●
s.__reversed__()		Get iterator to scan items from last to first	●	
s.__rmul__(n)	n * s	Reversed repeated concatenation	●	●
s.__setitem__(p, e)	s[p] = e	Put e in position p, overwriting existing item	●	
s.append(e)		Append one element after last	●	
s.clear()		Delete all items	●	
s.copy()		Shallow copy of the list	●	
s.count(e)		Count occurrences of an element	●	●
s.extend(it)		Append items from iterable `it`	●	
s.index(e)		Find position of first occurrence of e	●	●
s.insert(p, e)		Insert element e before the item at position p	●	
s.pop([p])		Remove and return last item or item at optional position p	●	
s.remove(e)		Remove first occurrence of element e by value	●	
s.reverse()		Reverse the order of the items in place	●	
s.sort([key], [reverse])		Sort items in place with optional keyword arguments `key` and `reverse`	●	