

# Cheat sheet: Lists and tuples (#1)

## Tuple (immutable, ordered)

`()`, `(1, )`, `(1, 3.0, "Hello")`

## List (mutable, ordered)

`[]`, `[1, ]`, `[1, 3.0, "Hello"]`

## Tuple/List operations

<code>tpl[pos]</code>	access element
<code>tpl[from:to:incr]</code>	access range
<code>item in l</code>	<code>l</code> contains item?
<code>l1 + l2</code>	concatenate
<code>l1 * n</code>	repeat
<code>len(l)</code>	length

## List manipulation

<code>l[pos] = val</code>	Overwrite item
<code>l[i:j:k] = l2</code>	Overwrite range
<code>l.append()</code>	Append item
<code>l.extend()</code>	Extend with list
<code>l += [...]</code>	Extend with list
<code>l.sort()</code>	Sort
<code>del l[ind]</code>	Delete ind or range
<code>del l[from:to:incr]</code>	Delete range

# Cheat sheet: Lists and tuples (#2)

## Iterating over containers

**for *item* in *tpl*:** Tuple items

**for *item* in *lst*:** List items

## Enumeration iterator

**for *ind, item* in enumerate(*lst*):**

## List comprehension

***lst* = [expr for var in iterable if condition]**

## In place operators

**+=, -=, \*=, /=, //=**

## Function definition

```
def fname(arg1, arg2, ...):  
    """Documentation ..."""  
    ... # function code  
    return result
```

## Function call

```
fname(arg1, arg2, ...)
```