

Mutability

Announcements

List Mutation

(Demo)

List Mutation

(Demo)

List Mutation

Create a new list:

List literal: `s = [1, 2, 3]`

List constructor: `t = list(s)`

List comprehension: `u = [x for x in s]`

+: v = s + t

Slicing: `w = t[1:]`

Modify a list:

`s.append(4)`

`s[1] = 4`

`s.extend(t)` # Modifies s but not t

`s.remove(2)` # Removes the first 2

`s.pop()` # Removes the last element

(Demo)

Mutation and Identity

Identity Operators

Identity

`<exp0> is <exp1>`

evaluates to `True` if both `<exp0>` and `<exp1>` evaluate to the same object

Equality

`<exp0> == <exp1>`

evaluates to `True` if both `<exp0>` and `<exp1>` evaluate to equal values

Identical objects are always equal values

(Demo)

