class **list**(object)

| list() -> new empty list

| list(iterable) -> new list initialized from iterable's

items

|

| Methods defined here:

|

| \_\_add\_\_(...)

| x.\_\_add\_\_(y) <==> x+y

|

| \_\_contains\_\_(...)

| x.\_\_contains\_\_(y) <==> y in x

|

| \_\_delitem\_\_(...)

| x.\_\_delitem\_\_(y) <==> del x[y]

|

| \_\_delslice\_\_(...)

| x.\_\_delslice\_\_(i, j) <==> del x[i:j]

|

| Use of negative indices is not supported.

|

| \_\_eq\_\_(...)

| x.\_\_eq\_\_(y) <==> x==y

|

| \_\_ge\_\_(...)

| x.\_\_ge\_\_(y) <==> x>=y

|

| \_\_getattribute\_\_(...)

| x.\_\_getattribute\_\_('name') <==> x.name

|

| \_\_getitem\_\_(...)

| x.\_\_getitem\_\_(y) <==> x[y]

|

| \_\_getslice\_\_(...)

| x.\_\_getslice\_\_(i, j) <==> x[i:j]

|

| Use of negative indices is not supported.

|

| \_\_gt\_\_(...)

| x.\_\_gt\_\_(y) <==> x>y

|

| \_\_iadd\_\_(...)

| x.\_\_iadd\_\_(y) <==> x+=y

|

| \_\_imul\_\_(...)

| x.\_\_imul\_\_(y) <==> x\*=y

|

| \_\_init\_\_(...)

| x.\_\_init\_\_(...) initializes x; see help(type(x)) for

signature

|

| \_\_iter\_\_(...)

| x.\_\_iter\_\_() <==> iter(x)

|

| \_\_le\_\_(...)

| x.\_\_le\_\_(y) <==> x<=y

|

| \_\_len\_\_(...)

| x.\_\_len\_\_() <==> len(x)

|

| \_\_lt\_\_(...)

| x.\_\_lt\_\_(y) <==> x<y

|

| \_\_mul\_\_(...)

| x.\_\_mul\_\_(n) <==> x\*n

|

| \_\_ne\_\_(...)

| x.\_\_ne\_\_(y) <==> x!=y

|

| \_\_mul\_\_(...)

| x.\_\_mul\_\_(n) <==> x\*n

|

| \_\_ne\_\_(...)

| x.\_\_ne\_\_(y) <==> x!=y

|

| \_\_repr\_\_(...)

| x.\_\_repr\_\_() <==> repr(x)

|

| \_\_reversed\_\_(...)

| L.\_\_reversed\_\_() -- return a reverse iterator over

the list

|

| \_\_rmul\_\_(...)

| x.\_\_rmul\_\_(n) <==> n\*x

|

| \_\_setitem\_\_(...)

| x.\_\_setitem\_\_(i, y) <==> x[i]=y

|

| \_\_setslice\_\_(...)

| x.\_\_setslice\_\_(i, j, y) <==> x[i:j]=y

|

| Use of negative indices is not supported.

|

| \_\_sizeof\_\_(...)

| L.\_\_sizeof\_\_() -- size of L in memory, in bytes

|

| append(...)

| L.append(object) -- append object to end

|

| count(...)

| L.count(value) -> integer -- return number of

occurrences of value

|

| extend(...)

| L.extend(iterable) -- extend list by appending elements from the iterable

|

| index(...)

| L.index(value, [start, [stop]]) -> integer – return

first index of value.

| Raises ValueError if the value is not present.

|

| insert(...)

| L.insert(index, object) -- insert object before

index

|

| pop(...)

| L.pop([index]) -> item -- remove and return item at

index (default last).

| Raises IndexError if list is empty or index is out

of range.

|

| remove(...)

| L.remove(value) -- remove first occurrence of value.

| Raises ValueError if the value is not present.

|

| reverse(...)

| L.reverse() -- reverse \*IN PLACE\*

|

| sort(...)

| L.sort(cmp=None, key=None, reverse=False) – stable

sort \*IN PLACE\*;

| cmp(x, y) -> -1, 0, 1

|

| --------------------------------------------------------