lists_Tuples

October 17, 2019

[3]:
```
NT=(1,2,('abc','aa'),44,5,('bb','aa'))
NT[5][1]
```

[3]:
```
'aa'
```

[9]:
```
L=[1,44,'acdb',[22,'aaa'],11,12]
L[3:5]
```

[9]:
```
[[22, 'aaa'], 11]
```

[12]:
```
L1=L+['zzz','pop']
```

[13]:
```
L1
```

[13]:
```
[1, 44, 'acdb', [22, 'aaa'], 11, 12, 'zzz', 'pop']
```

[14]:
```
L1.extend(['push','sss'])
```

[15]:
```
L1
```

[15]:
```
[1, 44, 'acdb', [22, 'aaa'], 11, 12, 'zzz', 'pop', 'push', 'sss']
```

[16]:
```
A=['ee','ddd','aa']
A[0]=['zz']
```

[17]:
```
A
```

[17]:
```
[['zz'], 'ddd', 'aa']
```

[18]:
```
del(A[0])
```

[19]:
```
A
```

[19]:
```
['ddd', 'aa']
```

[20]:
```
'aaa ggg'.split()
```

[20]:
```
['aaa', 'ggg']
```

[21]:
```
a=['baba','ma','aaa']
b=a
```

[22]:
```
b
```

[22]:
```
['baba', 'ma', 'aaa']
```

[23]:
```
b[0]='ccc'
```

[24]:
```
b
```

[24]:
```
['ccc', 'ma', 'aaa']
```

[25]:
```
a
```

[25]:
```
['ccc', 'ma', 'aaa']
```

[26]:
```
c=a[:]
```

[27]:
```
c
```

[27]:
```
['ccc', 'ma', 'aaa']
```

[28]:
```
help(a)
```

```
Help on list object:

class list(object)
 |  list() -> new empty list
 |  list(iterable) -> new list initialized from iterable's items
 |
 |  Methods defined here:
 |
 |  __add__(self, value, /)
 |      Return self+value.
 |
 |  __contains__(self, key, /)
 |      Return key in self.
 |
 |  __delitem__(self, key, /)
 |      Delete self[key].
 |
 |  __eq__(self, value, /)
 |      Return self==value.
 |
 |  __ge__(self, value, /)
 |      Return self>=value.
```

```
|
|  __getattribute__(self, name, /)
|      Return getattr(self, name).
|
|  __getitem__(…)
|      x.__getitem__(y) <==> x[y]
|
|  __gt__(self, value, /)
|      Return self>value.
|
|  __iadd__(self, value, /)
|      Implement self+=value.
|
|  __imul__(self, value, /)
|      Implement self*=value.
|
|  __init__(self, /, *args, **kwargs)
|      Initialize self.  See help(type(self)) for accurate signature.
|
|  __iter__(self, /)
|      Implement iter(self).
|
|  __le__(self, value, /)
|      Return self<=value.
|
|  __len__(self, /)
|      Return len(self).
|
|  __lt__(self, value, /)
|      Return self<value.
|
|  __mul__(self, value, /)
|      Return self*value.
|
|  __ne__(self, value, /)
|      Return self!=value.
|
|  __new__(*args, **kwargs) from builtins.type
|      Create and return a new object.  See help(type) for accurate signature.
|
|  __repr__(self, /)
|      Return repr(self).
|
|  __reversed__(…)
|      L.__reversed__() -- return a reverse iterator over the list
|
|  __rmul__(self, value, /)
|      Return value*self.
```

```
 |
 |  __setitem__(self, key, value, /)
 |      Set self[key] to value.
 |
 |  __sizeof__(…)
 |      L.__sizeof__() -- size of L in memory, in bytes
 |
 |  append(…)
 |      L.append(object) -> None -- append object to end
 |
 |  clear(…)
 |      L.clear() -> None -- remove all items from L
 |
 |  copy(…)
 |      L.copy() -> list -- a shallow copy of L
 |
 |  count(…)
 |      L.count(value) -> integer -- return number of occurrences of value
 |
 |  extend(…)
 |      L.extend(iterable) -> None -- 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) -> None -- remove first occurrence of value.
 |      Raises ValueError if the value is not present.
 |
 |  reverse(…)
 |      L.reverse() -- reverse *IN PLACE*
 |
 |  sort(…)
 |      L.sort(key=None, reverse=False) -> None -- stable sort *IN PLACE*
 |
 |  ----------------------------------------------------------------------
 |  Data and other attributes defined here:
```

```
 |
 |  __hash__ = None
```

[ ]: