Big Daddy's

Data Container Quick Chart

(Attributes, Methods, Processes, Workarounds)

Tuples (sequence type, immutable)

ACTION/EVENT	Key Term/Symbol/Method/Attribute	Example
Create initially	(items) ~note example tuple objects~; ("1item",) must have comma; value trailing comma issue	TupPreps=('with','with','under', ['on', 'of', 'in'], 'down', ('by', 'at'), 'among', 'about')
		TupNum=(4,2.4, 7/3, 1,20.2) or Testtup=('a','b','c')
Combine/merge containers	~concatenation~	newtup=tup1+tup2
Combine overlap items only	N/A	
Combine non-overlap items only	N/A	
Add at first position	~tuples are "immutable" - can't add by position	none
Add items at end	+= can only concat tuple - not string/number	Testtup+=(1,2,3)
Add one item at end	+=(x,) ~note comma~	Testtup+=(4,)
Add at position in container	N/A - or programmed function	-
Add somewhere inbetween	N/A -or programmed function	-
Add multiple items	+=	Testtup+=(1,2,3) ↳('a', 'b', 'c', 1, 2, 3)
or simple concatenation...	+=	Testtup+=(1,2,3) ↳('a', 'b', 'c', 1, 2, 3)
Remove a known value or key	N/A -or programmed function	-
Remove item(s) by index	N/A	
Remove and return the last item	N/A -or programmed function	-
Remove and return a known item	N/A	
Remove and return a random item	N/A	
Remove and return item number i	N/A -or programmed function	-
Replace an item/pair or value	N/A -or programmed function	
Replace a group of items	N/A -or programmed function	
Retrieve sequential items	tuple[i:j] ~start is 0, end is last item +1~	NewList = TupPreps[0:3]
Retrieve values, keys, or pairs	value for each item = tuple if no [x[x]] defined	x1,x2,x3,x4,x5=TupNum ~must have 1 for1~
Retrieve value from known key	N/A	
Retrieve all keys, values, pairs	N/A	
Retrieve index number of first value x	.index(x[,at or after index i [,before index j]])	MyIndex = Testtup.index("c")
Compare overlap		
Compare subset		
*compare as true subset(not equal)		
Compare superset		
*compare as true superset(not equal)		
Iteration (loop)	for int in tuple	for x in Testtup: | print (x)
Iteration (iter, next)	iter(tuple) | x=iter(TupPreps) | for i in TupPreps: | print (next (x, "defalut if no value"))	
Return number of items/pairs	len(tuple)	len(TupPreps)
Find count of x values	.count(x) ~number of item values == x~	TupPreps.count("with")
Find maximum value	max(tuple)	max(TupNum) ~TupPreps would give error
Find minimum value	min(tuple)	min(TupNum) due to inclusion of list objects~
Determine membership	value in tuple name or value in (item,item,item)	BooleanVal = "xx" in Testtup ↳ False
-		
Copy	Why copy an immutable object?	-
Sort	sorted(tuple) ~if tup=("C","c","a","z","f","zz","x")~	sorted(tup) ↳['C', 'a', 'c', 'f', 'x', 'z', 'zz']
Reverse items	tuple[::-1] ~ok, it's a trick but it works~	tup[::-1] ↳('zz', 'z', 'x', 'f', 'c', 'a', 'C')
Clear all	=() ~clears the tuple ~	tup=()
Delete the object	del tuple	del tup ~after gives name not defined error~
Convert	tuple(list) ~convert list to tuple~	mytup = tuple(mylist)
Other: setdefault		