Big Daddy's

(c) John A Oakey 2017 v010917aR1

Data Container Quick Chart

Lists (sequence type, mutable)

ACTION/EVENT	Key Term/Symbol/Method/Attribute	Example
Create initially	[item, item,...]	LPreps=["of", "over","with","in","under"]
		Numlist=(47,3,12.23,96,52)
Combine/merge containers	~concatenation~	newlist=list1 + list2
Combine overlap items only	N/A	
Combine non-overlap items only	N/A	
Add at first position	.insert(0,item)	Lpreps[0]="about"
Add items at end	loop and append	for i in ("at","about"): ~ Lpreps.append(i)
Add one item at end	.append(item)	LPreps.append("behind")
Add at position in container	.insert(position, item)	LPreps.insert(2,"of")
Add somewhere inbetween	list[i]=x ~replace item i with x~	LPreps[5]="at"
Add multiple items	.extend(multiple items in a list)	LPreps.extend(["under","above"])
or simple concatenation...	list+= [item, item, ...] ~ concatenation~	LPreps+=["down","by","over"] ~add 2nd over~
Remove a known value or key	.remove(first item x)	LPreps.remove("over")
Remove item(s) by index	del list[index : index]	del Lpreps[2]
Remove and return the last item	.pop()	FetchedItem = LPreps.pop()
Remove and return a known item	N/A	
Remove and return a random item	N/A	
Remove and return item number i	.pop(i)	FetchedItem = LPreps.pop(7)
Replace an item/pair or value	list(index]="new value"	Lpreps[2]= "among"
Replace a group of items	list[index i: index j]=new list of same length	Lpreps[2:3]=['around','by']
Retrieve sequential items	=list[from index i : to index j : step by k]	NewList=Lpreps[1:5]
Retrieve values, keys, or pairs	N/A	
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=LPreps.index("under")
Compare overlap	N/A -or programmed function	
Compare subset	N/A -or programmed function	
*compare as true subset(not equal)	N/A -or programmed function	
Compare superset	N/A -or programmed function	
*compare as true superset(not equal)	N/A -or programmed function	
Iteration (loop)	for int in list	for i in LPreps | print(i)
Iteration (iter, next)	iter(list); next(itervariable, default)	x=iter(LPreps) | print(next(x,"end"))
Return number of items/pairs	len(list)	len(Lpreps)
Find count of x values	.count(x) ~number of item values == x~	LPreps.count('in')
Find maximum value	.max(list)	print(max(Numlist))
Find minimum value	.min(list)	print(min(Numlist))
Determine membership	if/in ~if then else~	if "at" in LPreps:
-		
Copy	.copy() ~return shallow copy~	Lcopy=LPreps.copy()
Sort	.sort(key=None, reverse=False)~rev is low->hi~	LPreps.sort() ~key ex: key= str.lower~
Reverse items	.reverse	LPreps.reverse()
Clear all	.clear or Lpreps=[]	Lcopy.clear()
Delete the object	del list	del LPreps
Convert	list(tuple)	Newlist = list(SomeTuple)
Other: setdefault		