

List and Array

Python (<i>recommended: Jupyter</i>)	Mathematica
[a,b,c] (note: {a,b,b} is Set, not List in Mathematica)	{a,b,c}
[5, 'string', 1.4]	{anything, all objects, multi-media...}
len()	Length[]
["list", ["nested list item", a, b], ...]	Same. Use Depth[] to see nest level.
indexing: my_list[n]	mylist[[n]]
negative index	same
list.append()	AppendTo[...] Prepend, PrependTo
list.remove()	Drop[,n] ... (* see details *)
del my_list[n]	see Drop[] above
insert()	Insert[]
clear()	not Clear[]
my_list=['p','y','t','h','o','n']; my_list[0:2]= ['m','a','r','a'] list1 + list2	Join[{"m", "a", "r", "a"}, Drop[mylist, 2]] Join[list1, list2]
extend()	Join[]
sorted()	Sort[]
reverse	Reverse[]
list=[function(x) for x in range(n)]	mylist=Table[f[x]..., {x, n}]
range(n)	Range[n], but starting from 1 not zero and end with n, not n-1.
np.arange(beg, end, step)	Range[beg, end, step]
np.linspace(beg, end, npoint)	Range[0, npoint]*(end-beg)/(npoint-1)
x=['a', 2, 'b', 4.2] for u in x: if isinstance(u, numbers.Number): print(u**2) else: print("not a number")	Do[If[NumberQ[u], Print[u^2], Print[u, " is not a number"]], {u, x}]
break()	Break[]
min(), max()	Min[], Max[], MinMax[]
sum()	Total[]
all(), any()	AllTrue[], AnyTrue[]
u=range(5); v=range(3); [f(x,y) for x in u for y in v] [[f(x,y) for x in u] for y in v]	Table[x y, {x, 0, 4}, {y, 0, 2}] output is 5 x 3 matrix. Flatten[] to get Python 1-D list or numpy 1D array.
np.ndarray.flatten	Flatten[]
list.index(), enumerate()	Position[]
list1+list2	Join[list1, list2]
list(map(operator.add, list1, list2)) np.add(list1, list2) np.array[list1]+np.array[list2]	List1+List2 MapThread[Plus, {list1, list2}] List1+List2
list*n	Flatten[ConstantArray[List, n], 1] or Flatten[Join[Table[List, n], 1]
[x*const for x in a] if a is np.array, treat like List in Mathematica	a*constant
x in list	MemberQ[]

List and Array

	Select[]
myset={a,b,c}	set in Mathematica is an ordered List
union(), difference(), intersection	Union[], Complement[], Intersection[]
//is// isdisjoint()	//Q// DisjointQ, IntersectingQ
isinstance(,)	...Q... StringQ, NumberQ, ListQ, FileExistsQ
np.array() np.ndarray()	Table[] . Note that in Mathematica, List is already like an np.array for most operations.
np.random.randint(n1,n2,size=(j,k)) np.random.rand()	RandomInteger[{n1,n2},{j,k}] RandomReal[]
np.random.normal(mu,s,n)	RandomVariate[NormalDistribution[<i>mu,s</i>],n] see Mathematica docum for full list of distributions
np.dot(a,b)	a . b
np.matmul(m,a) m @ a	m . a a . m
my_array.shape()	Dimensions[]
np.full()	ConstantArray[]
my_array[i,k:m] mnemonic technique: a : is = ;;	my_array[i,k;;m] [[1]] Take[my_array[[i]},{k,m}]
m.T, or np.transpose or np.matrix.transpose()	Transpose[]
np.linalg.inv()	Inverse[]

Other things

(to be added later)