Module	Description	Example	Script
core	dictionary, adding a new entry	co['po'] = 'CO'	g05/demo.py
core	dictionary, creating	co = {'name':'Colorado', 'capital':'Denver'}	g05/demo.py
core	dictionary, length of	n = len(to_nato)	g05/nato.py
core	dictionary, looking up a value	name = ny['name']	g05/demo.py
core	dictionary, making a list of	list1 = [co,ny]	g05/demo.py
core	dictionary, obtaining a list of keys	names = super_dict.keys()	g05/demo.py
core	f-string, using a formatting string	print(f"PV of {payment} with T={year} and r={r} is ${pv}")	g07/demo.py
core	file, closing	fh.close()	g02/demo.py
core	file, opening for reading	fh = open('states.csv')	g05/demo.py
core	file, opening for writing	fh = open(filename,"w")	g02/demo.py
core	file, output using print	print("It was written during",year,file=fh)	g02/demo.py
core	file, output using write	fh.write("Where was this file was written?\n")	g02/demo.py
core	file, reading one line at a time	for line in fh:	g05/demo.py
core	for, looping through a list	for n in a_list:	g04/demo.py
core	function, calling	d1_ssq = sumsq(d1)	g06/demo.py
core	function, calling with an optional argument	sample_function(100, 10, r=0.07)	g07/demo.py
core	function, defining	def sumsq(values):	g06/demo.py
core	function, defining with optional argument	def sample_function(payment,year,r=0.05):	g07/demo.py
core	function, returning a result	return values	g06/demo.py
core	list, appending an element	a_list.append("four")	g03/demo.py
core	list, create via comprehension	cubes = [n**3 for n in a_list]	g04/demo.py
core	list, creating	a_list = ["zero","one","two","three"]	g03/demo.py
core	list, determining length	n = len(b_list)	g03/demo.py
core	list, extending with another list	a_list.extend(a_more)	g03/demo.py
core	list, generating a sequence	b_list = range(1,6)	g04/demo.py
core	list, joining with spaces	a_string = " ".join(a_list)	g03/demo.py
core	list, selecting an element	print(a_list[0])	g03/demo.py
core	list, selecting elements 0 to 3	print(a_list[:4])	g03/demo.py
core	list, selecting elements 1 to 2	print(a_list[1:3])	g03/demo.py
core	list, selecting elements 1 to the end	print(a_list[1:])	g03/demo.py
core	list, selecting last 3 elements	print(a_list[-3:])	g03/demo.py
core	list, selecting the last element	print(a_list[-1])	g03/demo.py

Module	Description	Example	Script
core	list, sorting	c_sort = sorted(b_list)	g03/demo.py
core	list, splitting on whitespace	b_list = b_string.split()	g03/demo.py
core	math, raising a number to a power	a_cubes.append(n**3)	g04/demo.py
core	math, rounding a number	rounded = round(ratio,2)	g05/demo.py
core	string, concatenating	name = s1+" "+s2+" "+s3	g02/demo.py
core	string, convert to lower case	line = line.lower()	g05/nato.py
core	string, converting to an int	values.append(int(line))	g06/demo.py
core	string, creating	filename = "demo.txt"	g02/demo.py
core	string, including a newline character	fh.write(name+"!\n")	g02/demo.py
core	string, printing	print("Hello, World!")	g02/hello1.py
core	string, remove spaces	line = line.strip()	g05/nato.py
core	string, splitting on white space	parts = line.split(',')	g05/demo.py
core	string, stripping blank space	clean = [item.strip() for item in parts]	g05/demo.py
csv	setting up a DictReader object	reader = csv.DictReader(fh)	g08/demo.py
json	importing the module	import json	g05/demo.py
json	using to print an object nicely	print(json.dumps(list1,indent=4))	g05/demo.py
scipy	calling newton's method	cr = opt.newton(find_cube_root,xinit,maxiter=20,args=[y])	g07/demo.py
scipy	importing the module	import scipy.optimize as opt	g07/demo.py