Getting started with lists

A list is an ordered and changeable sequence of elements. It can hold images, characters, floats, strings, and even objects.

Creating lists

# Create a list with 1 element separated by commas
x = [1, 3, 5]

List functions and methods

- `x.sort()` # Sort a list in-place (replaces x)
- `x.copy()` # Return a copy of the list
- `x.reverse()` # Reverse the list
- `x.count(a)` # Count the number of element a in the list

Selecting list elements

Python lists are zero-based (the first element is index 0). For ranges, the first element is included but the last is not.

- `x[1:]` # Get the rest of the list
- `x[1:3]` # Get the slice from index 1 to 3 (exclusive)
- `x[1:3:2]` # Get the slice from index 1 to 3 with step 2

Concating lists

# Define the x and y lists
x = [1, 3, 4]  
y = [3, 5, 6]
# Concatenate the lists
z = x + y

Getting started with dictionaries

A dictionary stores data in key-value pairs. That is, unlike lists which are accessed by position, dictionary are accessed by fixed-length the names of which must be unique.

Creating dictionaries

# Create a dictionary with ():
{ 'a': 1, 'b': 2, 'c': 3 }

Dictionary functions and methods

- `x.get(y)` # Get the value associated with key y
- `x.keys()` # Returns elements in the dictionary as a list of keys
- `x.values()` # Get the values of a dictionary, returns dict.values([1, 2, 3])

Selecting dictionary elements

- `x[k]` # Get a value from a dictionary by specifying the key

NumPy arrays

NumPy is a python package for scientific computing. It provides multidimensional array objects and efficient operations.

Creating arrays

# Convert a python list to a NumPy array
np.array([1, 2, 3]) # Returns array([1, 2, 3])
# Return a sequence from start to end (inclusive) to end (exclusive)
np.arange(0.0, 4.0, 1.0) # Returns array([0.0, 1.0, 2.0, 3.0])
# Return a sequence from start (inclusive) to end (exclusive)
np.linspace(0.0, 4.0, 4) # Returns array([0.0, 1.0, 2.0, 3.0])
# Return a sequence from start to end (inclusive)
np.logspace(0.0, 1.0, 10) # Returns array([0.001, 0.01, 0.1, 1.0, 10.0])
# Random values in a range
np.random.randint(0, 100, 3) # Returns array([35, 20, 85])

Math functions and methods

All functions take an array as input.

- `np.log10(x)` # Calculate logarithm
- `np.exp(x)` # Calculate exponential
- `np.sqrt(x)` # Get square root of each value
- `np.sin(x)` # Get sine of each value
- `np.cos(x)` # Get cosine of each value
- `np.tan(x)` # Get tangent of each value
- `np.sinh(x)` # Calculate sinh
- `np.cosh(x)` # Calculate cosh
- `np.tanh(x)` # Calculate tanh
- `np.round(x)` # Round to the nearest integer

Using NumPy functions on a list

# Calculate the mean of each column
np.mean(x)
# Calculate the variance of each column
np.var(x)
# Calculate standard deviation
np.std(x)
# Calculate the median of each column
np.median(x)
# Get summary statistics by column
np.describe(x)
# Get summary statistics for each function
np.describe2(x)
# Get summary statistics by column
np.stats.describe(x)
# Calculate the mean of each row
np.mean(x, axis=0)
# Calculate the variance of each row
np.var(x, axis=0)
# Calculate the standard deviation of each row
np.std(x, axis=0)
# Calculate the median of each row
np.median(x, axis=0)
# Get summary statistics for each function
np.stats.describe2(x, axis=0)
# Calculate the mean of each column
np.mean(x, axis=1)
# Calculate the variance of each column
np.var(x, axis=1)
# Calculate the standard deviation of each column
np.std(x, axis=1)
# Calculate the median of each column
np.median(x, axis=1)
# Get summary statistics for each function
np.stats.describe2(x, axis=1)

Using NumPy functions on an array

# Calculate the mean of each column
np.mean(x)
# Calculate the variance of each column
np.var(x)
# Calculate the standard deviation of each column
np.std(x)
# Calculate the median of each column
np.median(x)
# Get summary statistics for each function
np.stats.describe(x)
# Calculate the mean of each row
np.mean(x, axis=0)
# Calculate the variance of each row
np.var(x, axis=0)
# Calculate the standard deviation of each row
np.std(x, axis=0)
# Calculate the median of each row
np.median(x, axis=0)
# Get summary statistics for each function
np.stats.describe(x, axis=0)
# Calculate the mean of each column
np.mean(x, axis=1)
# Calculate the variance of each column
np.var(x, axis=1)
# Calculate the standard deviation of each column
np.std(x, axis=1)
# Calculate the median of each column
np.median(x, axis=1)
# Get summary statistics for each function
np.stats.describe(x, axis=1)