Array creation routines¶

See also
Array creation

Ones and zeros

empty(shape[, dtype, order])	Return a new array of given shape and type, without initializing entries.
empty_like(prototype[, dtype, order, subok])	Return a new array with the same shape and type as a given array.
eye(N[, M, k, dtype, order])	Return a 2-D array with ones on the diagonal and zeros elsewhere.
identity(n[, dtype])	Return the identity array.
ones(shape[, dtype, order])	Return a new array of given shape and type, filled with ones.
ones_like(a[, dtype, order, subok])	Return an array of ones with the same shape and type as a given array.
zeros(shape[, dtype, order])	Return a new array of given shape and type, filled with zeros.
zeros_like(a[, dtype, order, subok])	Return an array of zeros with the same shape and type as a given array.
full(shape, fill_value[, dtype, order])	Return a new array of given shape and type, filled with *fill_value*.
full_like(a, fill_value[, dtype, order, subok])	Return a full array with the same shape and type as a given array.

From existing data

array(object[, dtype, copy, order, subok, ndmin])	Create an array.
asarray(a[, dtype, order])	Convert the input to an array.
asanyarray(a[, dtype, order])	Convert the input to an ndarray, but pass ndarray subclasses through.
ascontiguousarray(a[, dtype])	Return a contiguous array in memory (C order).
asmatrix(data[, dtype])	Interpret the input as a matrix.
copy(a[, order])	Return an array copy of the given object.
frombuffer(buffer[, dtype, count, offset])	Interpret a buffer as a 1-dimensional array.
fromfile(file[, dtype, count, sep])	Construct an arrayfrom data in a text or binary file.
fromfunction(function, shape, **kwargs)	Construct an arrayby executing a function over each coordinate.
fromiter(iterable, dtype[, count])	Create a new 1-dimensional arrayfrom an iterable object.
fromstring(string[, dtype, count, sep])	A new 1-D arrayinitialized from text data in a string.
loadtxt(fname[, dtype, comments, delimiter, ...])	Load data from a text file.

Creating record arrays (numpy.rec)

Note
numpy.rec is the preferred alias for **numpy.core.records**.

core.records.array(obj[, dtype, shape, ...])	Construct a record array from a wide-variety of objects.
core.records.fromarrays(arrayList[, dtype, ...])	create a record arrayfrom a (flat) list of arrays
core.records.fromrecords(recList[, dtype, ...])	create a recarray from a list of records in text form
core.records.fromstring(datastring[, dtype, ...])	create a (read-only) record array from binary data contained in
core.records.fromfile(fd[, dtype, shape, ...])	Create an array from binary file data

Creating character arrays (numpy.char)

Note
numpy.char is the preferred alias for **numpy.core.defchararray**.

core.defchararray.array(obj[, itemsize, ...])	Create a **chararray**.
core.defchararray.asarray(obj[, itemsize, ...])	Convert the input to a **chararray**, copying the data only if necessary.

Numerical ranges

arange([start,] stop[, step,][, dtype])	Return evenly spaced values within a given interval.
linspace(start, stop[, num, endpoint, ...])	Return evenly spaced numbers over a specified interval.
logspace(start, stop[, num, endpoint, base, ...])	Return numbers spaced evenly on a log scale.
geomspace(start, stop[, num, endpoint, dtype])	Return numbers spaced evenly on a log scale (a geometric progression).
meshgrid(*xi, **kwargs)	Return coordinate matrices from coordinate vectors.
mgrid	*nd_grid* instance which returns a dense multi-dimensional "meshgrid".
ogrid	*nd_grid* instance which returns an open multi-dimensional "meshgrid".

Building matrices

diag(v[, k])	Extract a diagonal or construct a diagonal array.
diagflat(v[, k])	Create a two-dimensional array with the flattened input as a diagonal.
tri(N[, M, k, dtype])	An array with ones at and below the given diagonal and zeros elsewhere.
tril(m[, k])	Lower triangle of an array.
triu(m[, k])	Upper triangle of an array.
vander(x[, N, increasing])	Generate a Vandermonde matrix.

The Matrix class

mat(data[, dtype])	Interpret the input as a matrix.
bmat(obj[, ldict, gdict])	Build a matrix object from a string, nested sequence, or array.