Generate

Rangos

numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None)

>>> np.linspace(2.0, 3.0, num=5)

array([ 2. , 2.25, 2.5 , 2.75, 3. ])

>>> np.linspace(2.0, 3.0, num=5, endpoint=False)

array([ 2. , 2.2, 2.4, 2.6, 2.8])

>>> np.linspace(2.0, 3.0, num=5, retstep=True)

(array([ 2. , 2.25, 2.5 , 2.75, 3. ]), 0.25)

numpy.logspace(start, stop, num=50, endpoint=True, base=10.0, dtype=None)

>>> np.logspace(2.0, 3.0, num=4)

array([ 100. , 215.443469 , 464.15888336, 1000. ])

>>> np.logspace(2.0, 3.0, num=4, endpoint=False)

array([ 100. , 177.827941 , 316.22776602, 562.34132519])

>>> np.logspace(2.0, 3.0, num=4, base=2.0)

array([ 4. , 5.0396842 , 6.34960421, 8. ])

**numpy.geomspace(start, stop, num=50, endpoint=True, dtype=None)**

**Return numbers spaced evenly on a log scale (a geometric progression).**

**>>> np.geomspace(1, 1000, num=4)**

array([ 1., 10., 100., 1000.])

>>> np.geomspace(1, 1000, num=3, endpoint=False)

array([ 1., 10., 100.])

>>> np.geomspace(1, 1000, num=4, endpoint=False)

array([ 1. , 5.62341325, 31.6227766 , 177.827941 ])

>>> np.geomspace(1, 256, num=9)

array([ 1., 2., 4., 8., 16., 32., 64., 128., 256.])

**numpy.arange([start, ]stop, [step, ]dtype=None)**

**>>> np.arange(3)**

array([0, 1, 2])

>>> np.arange(3.0)

array([ 0., 1., 2.])

>>> np.arange(3,7)

array([3, 4, 5, 6])

>>> np.arange(3,7,2)

array([3, 5])