NumPy ဖြင့် Array တစ်ခု ပြုလုပ်ခြင်း

Fixed-Type Arrays in Python

Python တွင် NumPy ဖြင့် ဒေတာများကို ကောင်းစွာ သိမ်းဆည်းပြီး ပေါင်း၊ နှုတ်၊ မြှောက် ၊ စား လုပ်နိုင်သည့်နည်းများရှိသည်။ အောက်တွင် Array တစ်ခု ပြုလုပ်ပုံကို ဖော်ပြထားသည်။

In [1]:
```
import numpy as np # numpy ကို import လုပ်သည်။
```

Creating Arrays from Python Lists

`np.array` ဖြင့် Python lists ကို Array တစ်ခု ပြုလုပ်ပုံကို ဖော်ပြထားသည်။

In [2]:
```
# integer array:
np.array([1, 4, 2, 5, 3])
```

Out[2]:
```
array([1, 4, 2, 5, 3])
```

Python မှ list နှင့် မတူသည့်အချက်မှာ NumPy မှ array ထဲ တွင် ရှိသည့် ဒေတာများအားလုံးသည် အမျိုးအစားတူညီသည့် ဒေတာများ(same data type) ဖြစ်ကြသည်။

Python မှ list ထဲတွင် အမျိုးအစား မတူညီသည့် ဒေတာများ(different data type) ထည့်ထားနိုင်သည်။ NumPy မှ array ထဲတွင် အမျိုးအစားတူညီသည် ဒေတာများသာဖြစ်သည်။

Python list မှ NumPy မှ array သို့ ပြောင်းလဲရာတွင် အမျိုးအစား မတူညီသည့် ဒေတာများကို အမျိုးအစားတူညီသည် ဒေတာများ ဖြစ်အောင် up-cast လုပ်လိုက်သည်။ ဥပမာ integer များနှင့် floating point တို့ ပါဝင်နေသည့် Python မှ list ကို floating point အားလုံးဖြစ်အောင် ပြုလုပ်သည်။ (integers are up-cast to floating point)

In [3]:
```
np.array([3.14, 4, 2, 3])
```

Out[3]:
```
array([3.14, 4.  , 2.  , 3.  ])
```

NumPy array ကို မိမိ အလိုရှိသည့် ဒေတာအမျိုးအစား(data type) ရရန်အတွက် dtype keyword ကို အသုံးပြုနိုင်သည်။ we can use the `dtype` keyword:

အောက်တွင် float32 ဒေတာ အမျိုးအစား ရရန် `dtype` keyword အသုံးပြုပုံကို ဖော်ပြထားသည်။

In [4]:
```
np.array([1, 2, 3, 4], dtype='float32')
```

Out[4]:
```
array([1., 2., 3., 4.], dtype=float32)
```

Initializing of NumPy multidimensional array

အောက်တွင် multi-dimensional NumPy arrays တစ်ခု တည်ဆောက် ပုံ နည်းတစ်မျိုး(one way of initializing)ကို ဖော်ပြထားသည်။

```
In [5]: # nested lists result in multi-dimensional arrays
        np.array([range(i, i + 3) for i in [2, 4, 6]])
```

```
Out[5]: array([[2, 3, 4],
               [4, 5, 6],
               [6, 7, 8]])
```

အပေါ်မှ two-dimensional array တွင် inner lists သည် row များဖြစ်ကြသည်။ (The inner lists are treated as rows of the resulting two-dimensional array.)

Creating Arrays from Scratch

အောက်တွင် ကြိုးမားသည့် NumPy array များ တည်ဆောက်ပုံကို ဖော်ပြထားသည်။

np.zeros() ဖြင့် သုညများသာပါသည့် အရေတွက် ၁၀ခုပါသည့် NumPy array တည်ဆောက်ပုံကို ဖော်ပြထားသည်။

```
In [6]: # Create a length-10 integer array filled with zeros
        np.zeros(10, dtype=int)
```

```
Out[6]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

np.ones() ဖြင့် တစ်များသာပါသည့် 3x5 floating-point array တည်ဆောက်ပုံကို ဖော်ပြထားသည်။

```
In [7]: # Create a 3x5 floating-point array filled with ones
        np.ones((3, 5), dtype=float)
```

```
Out[7]: array([[1., 1., 1., 1., 1.],
               [1., 1., 1., 1., 1.],
               [1., 1., 1., 1., 1.]])
```

np.full() ဖြင့် များသာပါသည့် 3x5 floating-point array တည်ဆောက်ပုံကို ဖော်ပြထားသည်။

```
In [8]: # Create a 3x5 array filled with 3.14
        np.full((3, 5), 3.14)
```

```
Out[8]: array([[3.14, 3.14, 3.14, 3.14, 3.14],
               [3.14, 3.14, 3.14, 3.14, 3.14],
               [3.14, 3.14, 3.14, 3.14, 3.14]])
```

၀ မှ ၂၀ အထိ နှစ်ခုခြားစီဖြစ်သည့် ကိန်းများဖြင့်တည်ဆောက်ထားသည့် array

```
In [9]: # Create an array filled with a linear sequence
        # Starting at 0, ending at 20, stepping by 2
        # (this is similar to the built-in range() function)
        np.arange(0, 20, 2)
```

```
Out[9]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])
```

အစ ၀ နှင့် အဆုံး ၁ ပါဝင်ပြီး ညီမျှစွာ ခြားနားသည့် ကိန်း(၅)ခုဖြင့်တည်ဆောက်ထားသည့် array

In [10]:
```
# Create an array of five values evenly spaced between 0 and 1
np.linspace(0, 1, 5)
```

Out[10]:
```
array([0.  , 0.25, 0.5 , 0.75, 1.  ])
```

ဝ မှ ၁ အတွင်း ကိန်းများသာ ပါဝင်သည့် random ကိန်းများဖြင့် တည်ဆောက်ထားသည့် (3x3) array , uniformly distributed

In [11]:
```
# Create a 3x3 array of uniformly distributed
# random values between 0 and 1
np.random.random((3, 3))
```

Out[11]:
```
array([[0.84186821, 0.12114871, 0.63768114],
       [0.65492378, 0.22531327, 0.47192503],
       [0.79216983, 0.83087306, 0.66309743]])
```

ဝ မှ ၁၀ အတွင်း random ကိန်း ပြည့်(integer)များသာ ပါဝင်သည့် (3x3) array တည်ဆောက်ခြင်း

In [12]:
```
# Create a 3x3 array of random integers in the interval (0, 10)
np.random.randint(0, 10, (3, 3))
```

Out[12]:
```
array([[5, 5, 1],
       [6, 4, 7],
       [6, 3, 0]])
```

3x3 identity matrix တည်ဆောက်ခြင်း

In [13]:
```
# Create a 3x3 identity matrix
np.eye(3)
```

Out[13]:
```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

NumPy Standard Data Types

NumPy arrays contain values of a single type, so it is important to have detailed knowledge of those types and their limitations. Because NumPy is built in C, the types will be familiar to users of C, Fortran, and other related languages.

The standard NumPy data types are listed in the following table. Note that when constructing an array, they can be specified using a string:

```
np.zeros(10, dtype='int16')
```

Or using the associated NumPy object:

```
np.zeros(10, dtype=np.int16)
```

Data type	Description
`bool_`	Boolean (True or False) stored as a byte
`int_`	Default integer type (same as C `long`; normally either `int64` or `int32`)
`intc`	Identical to C `int` (normally `int32` or `int64`)
`intp`	Integer used for indexing (same as C `ssize_t`; normally either `int32` or `int64`)
`int8`	Byte (-128 to 127)
`int16`	Integer (-32768 to 32767)
`int32`	Integer (-2147483648 to 2147483647)
`int64`	Integer (-9223372036854775808 to 9223372036854775807)
`uint8`	Unsigned integer (0 to 255)
`uint16`	Unsigned integer (0 to 65535)
`uint32`	Unsigned integer (0 to 4294967295)
`uint64`	Unsigned integer (0 to 18446744073709551615)
`float_`	Shorthand for `float64`.
`float16`	Half precision float: sign bit, 5 bits exponent, 10 bits mantissa
`float32`	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa
`float64`	Double precision float: sign bit, 11 bits exponent, 52 bits mantissa
`complex_`	Shorthand for `complex128`.
`complex64`	Complex number, represented by two 32-bit floats
`complex128`	Complex number, represented by two 64-bit floats

Python Data Science Handbook by Jake VanderPlason GitHub (https://github.com/jakevdp/PythonDataScienceHandbook)

ကောင်းထက်ညွှန့်