

```

# coding: utf-8

# In[1]:

import numpy as np

# In[4]:

# The key feature of NumPy is the array object class. Here Arrays are
very similar to the lists in Python,
# except that every element of an array must be of the same type,
typically a numeric type like float or int.
#Arrays make operations with large amounts of numeric data very fast and
are generally much more efficient than lists

# In[10]:

x = np.array([10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
#This is the simplest way to create the numpy array object with set of
values

# In[11]:

print 'The values of the x numpy array object is \n', x

# In[12]:

print 'The type specification of the array object is :', type(x)

# In[13]:

print 'The dimension of the array object is : ', x.shape

# In[15]:

print 'The slicing numpy operation of the array object', x[:5]

# In[16]:

print 'Extracting a specific element of the array : ', x[5]

# In[19]:

x[-1] = 121

# In[20]:

print 'The new values of the numpy Array is :', x

```

```

# In[25]:

#Numpy also provides a mechanism to define the multi dimensional values
y = np.array([[10,20,30],[40,50,60],[60,70,80]])

# In[26]:

print 'The value of the new numpy array object is\n',y

# In[27]:

print 'The dimension of the numpy array object is ', y.shape

# In[30]:

print 'Extracting the specific values of multi dimension array object is
', y[1][1]

# In[31]:

print 'The first row elements of the multi dimension array is :', y[0]

# In[33]:

print 'The second row elements of the multi dimension array is :', y[1]

# In[34]:

print 'The third row elements of the multi dimension array is :', y[2]

# In[39]:

print 'The entire first and second row elements of multidimension array
is ', y[[0,1]]

# In[43]:

print 'The reverser elements of multidimension array is ', y[::-1]

# In[45]:

print 'The element types of the multidimension array is :', y.dtype
#The dtype property tells you what type of values are stored by the
array:

# In[46]:

```

```

print 'The lenght of the mutidimensional array is :', len(y)

# In[48]:

print 'The conditional logic can be applied on the multidimensional array
which returns boolean value : ', 50 in y

# In[52]:

#Arrays can be reshaped and also can created using range function
z = np.array(range(10,100,10))

# In[53]:

print 'The type of z is :', type(z)

# In[54]:

print 'The dimension of z is ', z.shape

# In[55]:

# Reshaping of the above created numpy array "z" can be done using the
following
z.reshape((3,3))

# In[56]:

print 'The dimension of th array is ', z.shape

# In[57]:

# Copy of the existing array object can be created using copy function
newarray = z.copy()

# In[58]:

print 'The values of newarray is :', newarray

# In[61]:

#Arrays can be also converted to a list
newlist = z.tolist()

# In[62]:

print 'The type of newlist object is :', type(newlist)

```

```
# In[63]:  
  
print 'The values of the list is :', newlist
```

```
# In[67]:  
  
# Numpy provides tostring and fromstring functions which can be used to  
# convert array into binary representation  
# for quick storage and faster retrieval.  
a = newarray.tostring()
```

```
# In[68]:  
  
print 'The tostring converted value is :', a
```

```
# In[ ]:
```

```
# In[ ]:
```