Basic Data Science

1. Import the numpy package under the name `np`

In [1]:
```
import numpy as np
```

2. Create a null vector of size 20

In [5]:
```
t=np.zeros(20)
print(t)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

3. Create a Ones Vector of size 20

In [7]:
```
np.ones(20)
```

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

4. Create a boolean array of 3X4.

In [11]:
```
t=np.random.randint(0,2,size=(3,4),dtype=bool)
print(t)
```

```
[[ True  True  True False]
 [False False False False]
 [ True False False False]]
```

5. Create a vector with values ranging from 100 to 200 of float64 data type

In [12]:
```
t=np.arange(100,200,dtype=float)
print(t)
```

```
[100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113.
 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127.
 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141.
 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155.
 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169.
 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183.
 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197.
 198. 199.]
```

6. Create an array of five values evenly spaced between 0 and 1

In [16]:
```
t=np.linspace(0,1,5)
print(t)
```

```
[0.   0.25 0.5  0.75 1.  ]
```

7. Reverse a given Vector

In [17]:
```
myarray = np.array([9, 8, 7, 6, 5, 4, 3, 2, 1, 0])
myarray=myarray[::-1]
print(myarray)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

8. Find indices of non-zero elements from [12,34,0,4,0,2,3,0,123]

In [19]:
```
t=np.array([12,34,0,4,0,2,3,0,123])
print(np.nonzero(t))
```

```
(array([0, 1, 3, 5, 6, 8], dtype=int64),)
```

9. Replace all even numbers in given arr vector with -1

In [20]:
```
arr = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14])
print(np.where(arr%2==0,-1,arr))
```

```
[ 1 -1  3 -1  5 -1  7 -1  9 -1 11 -1 13 -1]
```

10. Create a 5x3 array with random values (In - between 100 to 300) and find the minimum and maximum values (Hints : Use np.random.random)

In [21]:
```
r = np.random.randint(100,300,size=(5,3))
print(r)
print("the max value is:",np.max(r))
print("the min value is:",np.min(r))
```

```
[[114 163 222]
 [249 165 205]
 [245 247 200]
 [269 277 186]
 [272 138 241]]
the max value is: 277
the min value is: 114
```

11. Create a random vector of size 30 and find the mean value

In [24]:
```
t=np.random.randint(1,100,30)
print(t)
print(np.mean(t))
```

```
[10 81 39 78 98 53 66 60 63 76 16 66 38 45  5 37 86 84 90 91 97 39 85 52
 38 37 92 49 73 18]
58.733333333333334
```

12. What is the result of the following expression?

```
0 * np.nan
np.nan == np.nan
np.inf > np.nan
np.nan - np.nan
np.nan in set([np.nan])
0.3 == 3 * 0.1
```

In [26]:
```
print(0 * np.nan)
print(np.nan == np.nan)
print(np.inf > np.nan)
print(np.nan - np.nan)
print(np.nan in set([np.nan]))
print(0.3 == 3 * 0.1)
```

```
nan
False
False
nan
True
False
```

13. Normalize a 5x5 random matrix (Hints - fourmula (x - mean) / std)

In [27]:
```
Z = np.random.random((5,5))
zmean,zstd=np.mean(Z),np.std(Z)
Z=(Z-zmean)/(zstd)
print(Z)
```

```
[[ 1.17611742  0.1972721  -0.44561194  1.38738229 -1.28372845]
 [ 0.73975461 -1.01070645 -0.91343477 -0.0314753   1.37248975]
 [-1.34260917 -1.10862552 -0.94166182  1.40429835 -1.43208696]
 [-0.43866435 -0.73625046  0.16182519  1.15339861  1.98659003]
 [-0.59777915 -0.62253379  0.64217713  0.0692938   0.61456884]]
```

14. Multiply a 5x3 matrix by a 3x2 matrix (real matrix product)

In [34]:
```
t1=np.random.randint(1,20,size=(5,3))
t2=np.random.randint(1,20,size=(3,2))
print("random matrix t1:",t1)
print("random matrix t2:",t2)
t3=np.dot(t1,t2)
print("multiplication of matrix t1 and t2 :",t3)
```

```
random matrix t1: [[13  7 18]
 [13  5  5]
 [ 1 13  8]
 [ 5  3  6]
 [ 6  9 11]]
random matrix t2: [[14  7]
 [ 3 11]
 [ 5 19]]
multiplication of matrix t1 and t2 : [[293 510]
 [222 241]
 [ 93 302]
 [109 182]
 [166 350]]
```

15. How to find common values between two arrays?

In [39]:
```
r=np.random.randint(15,40,10)
t=np.random.randint(25,50,20)
print(r)
print(t)
print(np.intersect1d(r,t))
```

```
[33 24 37 32 30 18 27 16 28 29]
[27 48 38 32 32 40 42 43 27 48 26 46 33 39 30 34 34 32 36 35]
[27 30 32 33]
```

16. Convert a 1D array to a 2D array with 4 rows

In [41]:
```
one = np.arange(2,22)
t=one.reshape(4,5)
print(t)
```

```
[[ 2  3  4  5  6]
 [ 7  8  9 10 11]
 [12 13 14 15 16]
 [17 18 19 20 21]]
```

17. Create two array (a and b) and stack them vertically?.(concatenate vertically?)

In [42]:
```
a=np.random.randint(1,10,size=(2,2))
b=np.random.randint(1,10,size=(2,2))
print(a)
print('\n')
print(b)
print('\n')
print('Stacked arrays:')
print(np.concatenate((a,b),axis=0))
```

```
[[7 1]
 [8 3]]


[[9 6]
 [3 4]]


Stacked arrays:
[[7 1]
 [8 3]
 [9 6]
 [3 4]]
```

18. Create two 2Darray (a and b) and stack them horizontally.(concatenate horizontally)

In [43]:
```
a=np.random.randint(1,10,size=(2,2))
b=np.random.randint(1,10,size=(2,2))
print(a)
print('\n')
print(b)
print('\n')
print('Stacked arrays:')
print(np.concatenate((a,b),axis=1))
```

```
[[5 4]
 [1 7]]


[[9 9]
 [8 3]]


Stacked arrays:
[[5 4 9 9]
 [1 7 8 3]]
```

19. Create a 2darray of 4X4 and swap 2nd and 4th column .

In [46]:
```
my_array=np.random.randint(1,50,size=(4,4))
print(my_array)
my_array[:,[1, 3]] = my_array[:,[3, 1]]
print(my_array)
```

```
[[15  2  7 13]
 [47 24  2  3]
 [14  5 28 38]
 [ 5 31 33 45]]
[[15 13  7  2]
 [47  3  2 24]
 [14 38 28  5]
 [ 5 45 33 31]]
```

20. Create a 2darray of 4X4 and swap 2nd and 4th rows

In [47]:
```
my_array=np.random.randint(1,50,size=(4,4))
print(my_array)
my_array[[1,3],:] = my_array[[3,1],:]
print(my_array)
```

```
[[32 36  3 46]
 [40  6 16 39]
 [ 1 22 37 33]
 [34  1 29 39]]
[[32 36  3 46]
 [34  1 29 39]
 [ 1 22 37 33]
 [40  6 16 39]]
```