

Tugas LAB WEEK 1 Christopher Darren

```
In [1]: 1 #define variable
2 text = 'test'
3 angka = 5
4 angka2 = 5.5
```

```
In [2]: 1 #print variable value
2 print(text)
3 print(angka+angka2)
```

```
test
10.5
```

```
In [3]: 1 #Percabangan if else
2 if angka+angka2 <= 10:
3     print(10)
4 else:
5     print(angka+angka2)
```


```
10.5
```

```
In [4]: 1 #Membuat function
2 def penjumlahan(x,y):
3     hasil = x+y
4     return(hasil)
5
6 test = penjumlahan(angka,angka2)
7 print(test)
```

```
10.5
```

```
In [5]: 1 #Menerima input dari user dengan tipe data int
2 angka = int(input('angka baru :'))
3 angka2 = int(input('angka 2 baru: '))
4 test = penjumlahan(angka, angka2)
5 print(test)
```

```
angka baru :12
angka 2 baru: 20
32
```

 jupyter 00000054804_Christopher Darren_Week1 Last Checkpoint: 3 minutes ago (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3 (ipykernel) C

          Code 

```
In [6]: 1 #konversi number to string
2 angka_1 = 12
3 angka_2 = 20
4 print(angka_1+angka_2)
5 print(str(angka_1)+ str(angka_2))
```

```
32
1220
```


```
In [7]: 1 #melakukan Looping pada basic python
2 test_loop = 0
3
4 while(test_loop < 10):
5     print(test_loop)
6     test_loop += 1
```

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

```
In [8]: 1 import numpy as np
```

```
In [9]: 1 #membuat array numpy sederhana
2 a = np.array([[1,2,3],
3              [4,5,6]])
```

```
In [10]: 1 #array re-dimensioning
2 a = np.array([x for x in range(27)])
3
4 npshape = a.reshape((3,3,3))
5 print(npshape)
```

jupyter 00000054804_Christopher Darren_Week1 Last Checkpoint: 3 minutes ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [10]:

```
1 #array re-dimensioning
2 a = np.array([x for x in range(27)])
3
4 npshape = a.reshape((3,3,3))
5 print(npshape)
```

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

  [[ 9 10 11]
   [12 13 14]
   [15 16 17]]

  [[18 19 20]
   [21 22 23]
   [24 25 26]]]
```

In [11]:


```
1 #stacking numpy array
2 a1 = np.array([[1,2,3],
3               [4,5,6]])
4
5 a2 = np.array([[7,8,9],
6               [10,11,12]])
7
8 npstack = np.hstack((a1,a2))
9
10 print(npstack)
```

```
[[ 1  2  3  7  8  9]
 [ 4  5  6 10 11 12]]
```

In [12]:

```
1 #membuat array numpy dari angka 0 - 100 dengan jarak 2 angka
2 list_of_numbers = [x for x in range(0,101, 2)]
3
4 listnp = np.array(list_of_numbers)
5
6 print(listnp)
```

```
[ 0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34
 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70
 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100]
```

jupyter 00000054804_Christopher Darren_Week1 Last Checkpoint: 4 minutes ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [12]:

```
1 #membuat array numpy dari angka 0 - 100 dengan jarak 2 angka
2 list_of_numbers = [x for x in range(0,101, 2)]
3
4 listnp = np.array(list_of_numbers)
5
6 print(listnp)
```

```
[ 0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34
 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70
 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100]
```

In [13]:

```
1 #cara alternatif
2 npalt = np.arange(0,101,2)
3
4 print(npalt)
```

```
[ 0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34
 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70
 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100]
```

In [14]:

```
1 #membuat array numpy dengan random value
2 np.random.seed(123) #setting the seed
3
4 nprand = np.random.randint(0,10, size = (5,5))
5
6 print(nprand)
```

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

In [15]:

```
1 import pandas as pd
```

In [16]:

```
1 #membaca data dari file yang tersedia
2 data = pd.read_csv(r'D:\SEMESTER 4\IS411 Data Modelling\LAB\MBA_Python.csv', header=0, delimiter=',')
3
4 data.info()
```



```
In [20]: 1 def encode(qty):
2         if qty >= 1:
3             return 1
4         if qty <= 0:
5             return 0
6
7         basketSimplified = basket.applymap(encode)
8         basketSimplified
```

Out[20]:

Description	PURPLE FLOCK DINNER CANDLES	4 CHRISTMAS GIFT BAG LARGE	50'S DOLLY GIRL BEAKER	I LOVE LONDON MINI BACKPACK	I LOVE LONDON MINI RUCKSACK	NINE DRAWER OFFICE TIDY	OV WALL MIRROR DIAMANTE	RED SPOT GIFT BAG LARGE	SET 2 TEA TOWELS I LOVE LONDON	SPACEBO BABY GIFT SET	...	returned	taig adjust	test	to f o throu s s'
Invoice															
536365	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
536366	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
536367	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
536368	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
536369	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
...
581586	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
581587	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
A563185	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
A563186	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
A563187	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0	0

20136 rows x 4077 columns

```
In [21]: 1 finalBasket = basketSimplified[(basketSimplified > 0).sum(axis=1) >= 2]
```

```
In [22]: 1 from mlxtend.frequent_patterns import apriori
```

```
In [22]: 1 from mlxtend.frequent_patterns import apriori
2
3 frequent_item = apriori(finalBasket, min_support=0.03, use_colnames=True).sort_values('support', ascending=False).reset_index()
4
5 frequent_item
```

C:\Users\Darren\anaconda3\lib\site-packages\mlxtend\frequent_patterns\fpcommon.py:111: DeprecationWarning: DataFrames with non-bool types result in worse computational performance and their support might be discontinued in the future. Please use a DataFrame with bool type
warnings.warn(

Out[22]:

	support	itemsets
0	0.122205	(WHITE HANGING HEART T-LIGHT HOLDER)
1	0.113317	(JUMBO BAG RED RETROSPOT)
2	0.107264	(REGENCY CAKESTAND 3 TIER)
3	0.091122	(PARTY BUNTING)
4	0.085233	(LUNCH BAG RED RETROSPOT)
...
160	0.030320	(LUNCH BAG RED RETROSPOT, LUNCH BAG SUKI DESIGN)
161	0.030210	(HAND WARMER BIRD DESIGN)
162	0.030210	(LUNCH BAG CARS BLUE, LUNCH BAG RED RETROSPOT)
163	0.030101	(CHILDRENS APRON SPACEBOY DESIGN)
164	0.030101	(PAPER CHAIN KIT VINTAGE CHRISTMAS, PAPER CHAI...

165 rows x 2 columns

```
In [23]: 1 from mlxtend.frequent_patterns import association_rules
2
3 association_rules(frequent_item, metric='lift', min_threshold=1).sort_values('lift', ascending=False).reset_index(drop=True)
```

10	(LUNCH BAG PINK POLKADOT)	(LUNCH BAG RED RETROSPOT)	0.059385	0.085233	0.033046	0.556474	6.528866	0.027985	2.062487
----	---------------------------	---------------------------	----------	----------	----------	----------	----------	----------	----------

```
In [23]: 1 from mlxtend.frequent_patterns import association_rules
2
3 association_rules(frequent_item, metric='lift', min_threshold=1).sort_values('lift',ascending=False).reset_index(drop=True)
```

Out[23]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(PINK REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.041499	0.055350	0.034518	0.831800	15.028131	0.032222	5.616242
1	(GREEN REGENCY TEACUP AND SAUCER)	(PINK REGENCY TEACUP AND SAUCER)	0.055350	0.041499	0.034518	0.623645	15.028131	0.032222	2.546804
2	(ROSES REGENCY TEACUP AND SAUCER)	(PINK REGENCY TEACUP AND SAUCER)	0.057967	0.041499	0.032664	0.563500	13.578784	0.030259	2.195877
3	(PINK REGENCY TEACUP AND SAUCER)	(ROSES REGENCY TEACUP AND SAUCER)	0.041499	0.057967	0.032664	0.787122	13.578784	0.030259	4.425229
4	(GREEN REGENCY TEACUP AND SAUCER)	(ROSES REGENCY TEACUP AND SAUCER)	0.055350	0.057967	0.041880	0.756650	13.053107	0.038672	3.871107
5	(ROSES REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.057967	0.055350	0.041880	0.722484	13.053107	0.038672	3.403944
6	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.057258	0.053168	0.034900	0.609524	11.464049	0.031856	2.424813

In []: 1