

▼ 1 Konsep String, List, dan Tuple

Tulis ringkasan atau penjelasan hal-hal berikut, dengan kata-kata kalian sendiri :

1. Tuple, akses variabel yang bertipe tuple (jelaskan indeks, dan pengaksesan setiap anggota yang terdapat pada tuple) dan berikan contoh
2. Perbedaan String, List, dan Tuple, dan berikan contoh kapan waktu yang tepat harus menggunakan tipe string, list, ataupun tuple
3. List comprehension (syntax dan contohnya)

1. Tuple merupakan kumpulan data yang terdiri dari banyak anggota (dapat terdiri dari tipe data yang berbeda). Tuple dapat diawali dan diakhiri tanda kurung (tuple=(1, 2, 'hi', 'hello')) atau juga bisa langsung datanya (tuple=1, 2, 'hi', 'hello'). Data dalam tuple bersifat Immutable atau tidak bisa diubah. Indeks dalam tuple dimulai dari 0. Untuk mengakses tuple adalah sebagai berikut:

```
contoftuple=(1, 2, 'hi', 'hello')
print(contoftuple[1]) #mengakses tuple index ke- 1
print(contoftuple[:]) #mengakses tuple semua index
print(contoftuple[1:]) #mengakses tuple dari index ke- 1 sampai index terakhir
print(contoftuple[:2]) #mengakses tuple dari index pertama sampai index ke- 3-1
print(contoftuple[-1]) #mengakses tuple index terakhir
print(contoftuple[:-1]) #mengakses tuple index pertama sampai index terakhir-1
```

```
2
(1, 2, 'hi', 'hello')
(2, 'hi', 'hello')
(1, 2)
hello
(1, 2, 'hi')
```

2. Perbedaan String, List, dan Tuple adalah, String berisi satu data dan selalu diawali dan diakhiri tanda petik (" ") atau (' '). Sedangkan List berisi banyak data (bisa berisi tipe data yang berbeda) dan diawali dan diakhiri dengan tanda kurung siku ([]). Sedangkan Tuple berisi banyak data (bisa berisi tipe data yang berbeda) dan dapat diawali dan diakhiri dengan tanda kurung atau tidak menggunakan tanda kurung.

```
#Contoh String
contohstring='namaku adalah ...'
print(contohstring)
```

namaku adalah ...

#Contoh List

```
contohlist=[1, 2, 'tiga', 'empat']  
print(contohlist[3])
```

empat

#Contoh Tuple

```
contohtuple1=(1, 2, 'tiga', 'empat')  
contohtuple2=1, 2, 'tiga', 'empat'
```

```
print(contohtuple1[2])  
print(contohtuple2[2])
```

tiga
tiga

3. List Comprehession adalah cara yang memudahkan kita dalam membuat list baru dari list yang sudah ada dengan menggunakan iterasi for. contohnya adalah sebagai berikut:

#menguadratkan anggota list1 dan dimasukkan kedalam list2

```
list1=[1, 2, 3, 4, 5]  
list2=[]
```

```
for i in list1:  
    list2.append(i**2)
```

```
print(list2)
```

[1, 4, 9, 16, 25]

#mengelompokkan anggota list1 yang merupakan bilangan ganjil dan genap dan

#dimasukkan kedalam list yang baru

```
list1=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
list2=[]  
list3=[]
```

```
for i in list1:  
    if i%2==0:  
        list2.append(i)  
    else:  
        list3.append(i)
```

```
print(list2)  
print(list3)
```

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

```
#melakukan perkalian setiap anggota pada dua list dan hasilnya dimasukkan ke
#dalam list3
list1=[1, 2, 3]
list2=[1, 2]
list3=[]

for i in list1:
    for j in list2:
        list3.append(i*j)

print(list3)

[1, 2, 2, 4, 3, 6]
```

▼ 2 Implementasi

▼ 2.1 faktor pembagi

Buatlah code untuk mendapatkan faktor pembagi dari dua buah bilangan, dan faktor pembagi yang sama dari dua buah bilangan tersebut, dengan ketentuan sebagai berikut :

- Dua bilangan yang akan dicari faktor pembaginya, adalah inputan dari user
- Inputan ini setelah kata -in-
- Output yang diinginkan ditampilkan setelah kata -out-

Contoh tampilan yang harus dihasilkan, dapat dilihat pada Gambar 1.

```
-in-
81
54
-out-
[1, 3, 9, 27, 81]
[1, 2, 3, 6, 9, 18, 27, 54]
[1, 3, 9, 27]
```

(a)

```
-in-
64
84
-out-
[1, 2, 4, 8, 16, 32, 64]
[1, 2, 3, 4, 6, 7, 12, 14, 21, 28, 42, 84]
[1, 2, 4]
```

(b)

```
-in-
121
225
-out-
[1, 11, 121]
[1, 3, 5, 9, 15, 25, 45, 75, 225]
[1]
```

(c)

```
-in-
95
85
-out-
[1, 5, 19, 95]
[1, 5, 17, 85]
[1, 5]
```

(d)

Gambar 1: Faktor Pembagi dan Faktor Pembagi yang sama

```
print('-in-')
bil1=int(input())
bil2=int(input())
pembagi1=[]
pembagi2=[]
pembagiSama=[]

if bil1>bil2:
    n=bil1
else:
    n=bil2

for i in range(1,n+1):
    if bil1%i==0:
        pembagi1.append(i)
    if bil2%i==0:
```

```
pembagi2.append(i)

for j in pembagi1:
    for k in pembagi2:
        if j==k:
            pembagiSama.append(j)

print('-out-')
print(pembagi1)
print(pembagi2)
print(pembagiSama)

-in-
64
84
-out-
[1, 2, 4, 8, 16, 32, 64]
[1, 2, 3, 4, 6, 7, 12, 14, 21, 28, 42, 84]
[1, 2, 4]
```

▼ 2.2 Penjumlahan List

Buatlah code untuk menjumlahkan dua buah list, dengan ketentuan sebagai berikut :

- banyaknya anggota dan anggota dari masing-masing adalah inputan dari list
- inputan ini setelah kata -in-, dimana baris pertama adalah banyaknya anggota dari list, kemudian baris berikutnya adalah anggota atau nilai dari list tersebut
- output yang diinginkan ditampilkan setelah kata -out-
- Terdapat tiga buah skenario penjumlahan, yaitu ketika banyaknya anggota adalah sama, banyaknya anggota list pertama lebih banyak dari pada list kedua, dan banyaknya anggota list kedua lebih banyak dari pada list pertama.

Contoh tampilan yang harus dihasilkan, dapat dilihat pada Gambar 2.

```

-in-
4

1
2
3
4

4

4
3
2
1

-out-
[1, 2, 3, 4]
[4, 3, 2, 1]
[5, 5, 5, 5]

```

(a) Banyaknya anggota dua buah list adalah sama

```

-in-
3

1
4
7

5

4
5
2
6
3

-out-
[1, 4, 7]
[4, 5, 2, 6, 3]
[5, 9, 9, 6, 3]

```

(b) Banyaknya anggota list kedua lebih banyak

```

-in-
6

4
9
1
5
3
8

4

7
3
4
1

-out-
[4, 9, 1, 5, 3, 8]
[7, 3, 4, 1]
[11, 12, 5, 6, 3, 8]

```

(c) Banyaknya anggota list pertama lebih banyak

Gambar 2: Penjumlahan dua buah list

```

list1=[]
list2=[]
penjumlahan=[]

print('-in-')
sizeList1=int(input())
print('')
for i in range(sizeList1):
    list1.append(int(input(' ')))

print('')

sizeList2=int(input())

```

```

print('')
for i in range(sizeList2):
    list2.append(int(input(' ')))

print('\n-out-')
print(list1)
print(list2)

if sizeList1>sizeList2:
    listBanyak=list1
    listSedikit=list2
    tambah0=sizeList1-sizeList2
else:
    listBanyak=list2
    listSedikit=list1
    tambah0=sizeList2-sizeList1

for i in range(tambah0):
    listSedikit.append(0)

for i in range(len(listBanyak)):
    jumlah=listBanyak[i]+listSedikit[i]
    penjumlahan.append(jumlah)

print(penjumlahan)

```

-in-

6

4

9

1

5

3

8

4

7

3

4

1

-out-

[4, 9, 1, 5, 3, 8]

[7, 3, 4, 1]

[11, 12, 5, 6, 3, 8]

▼ 2.3 Mendapatkan nama hari dari suatu tanggal

Buatlah code dengan menggunakan Python mendapatkan nama hari (gunakan data dengan bentuk tuple), dengan ketentuan sebagai berikut

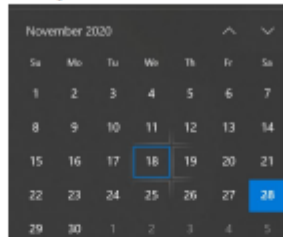
- user memberikan informasi bahwa tanggal pertama bulan ini, jatuh pada hari tertentu
- tanggal yang ingin dicari nama harinya, adalah inputan dari user
- untuk melihat apakah output yang dihasilkan adalah benar, dapat dilihat di calendar

Petunjuk : untuk mendapatkan index dari suatu data tuple atau list, bisa digunakan perintah `namaTuple.index(anggota)`.

Contoh output yang dihasilkan dapat dilihat pada Gambar 3.

Masukkan informasi, hari pertama bulan ini, jatuh pada hari : minggu
 Masukkan tanggal yang ingin diketahui harinya : 18
 Tanggal 18 adalah hari Rabu

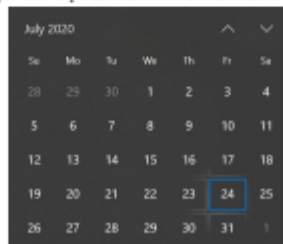
(a) Hari pertama pada bulan tersebut adalah hari Minggu



(b) Nama Hari pada tanggal 18 pada Calendar

Masukkan informasi, hari pertama bulan ini, jatuh pada hari : rabu
 Masukkan tanggal yang ingin diketahui harinya : 24
 Tanggal 24 adalah hari Jumat

(c) Hari pertama pada bulan tersebut adalah hari rabu



(d) Nama Hari pada tanggal 24 pada Calendar

Gambar 3: Nama Hari dari suatu Tanggal

```
hari = ('Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu', 'Minggu')
hariPertama=input('Masukkan informasi, hari pertama bulan ini, jatuh pada hari : ')
tanggal=int(input('Masukkan tanggal yang ingin diketahui harinya : '))
```

```
n=0
```

```
for i in hari:
    if hariPertama.capitalize()==i:
        hari=hari[n:]+hari[:n]
    n+=1
```

```
hitung=tanggal%7
hariTanggal=hari[hitung-1]
print('Tanggal',tanggal,'adalah hari',hariTanggal)
```

⇒ Masukkan informasi, hari pertama bulan ini, jatuh pada hari : rabu
 Masukkan tanggal yang ingin diketahui harinya : 24
 Tanggal 24 adalah hari Jumat

[Produk berbayar Colab](#) - [Batalan kontrak di sini](#)

✓ 0 d selesai pada 19.44

