# Programming III:

Functions in Python

# Hello, everyone!

## Sebelum kita memulai kelas, kita awali dengan:

1. Berdoa
2. siapkan diri

# Pertemuan lalu...

## Let’s Recall

pertemuan sebelumnya kita sudah belajar intermediate python. Ada yang inget apa aja?

# Target

1. What is functional programming?
2. Create functions in Python
3. Introduction to library
4. String manipulation library

# Tools

https://colab.research.google.com/

# 1. What is functional programming?

# Apaitu *functional programming* ?

- suatu bagian dari program yang digunakan untuk menjalankan suatu tugas tertentu dan letaknya terpisah dari bagian program yang menggunakannya.
- Suatu fungsi/prosedur dipanggil/digunakan dengan tujuan khusus, yaitu untuk mengerjakan suatu tugas tertentu, dapat berupa :
    - Tugas input (menyimpan hasil ke dalam suatu array atau file) dan/atau
    - Tugas output (menampilkan hasil di layar monitor) ataupun
    - melakukan penyeleksian dan perhitungan
- Suatu Fungsi dapat memberikan suatu hasil balik ke program yang memanggilnya atau tidak memberikan hasil balik sama sekali.
- Hasil balik ini biasanya berupa suatu nilai yang dibutuhkan oleh bagian program yang memanggilnya.

# 2. Create functions in Python

# Apaitu *functional programming* di Python?

- Function dalam Python berguna untuk menjalankan sebuah blok berisikan script Python yang berfungsi untuk menjalankan suatu task tertentu.
- Jenis-jenis function di python lho, yaitu:
  - Function tanpa Parameter
  - Function dengan Parameter
- fungsi dalam Python dapat melakukan tugas spesifik dan kemudian mengembalikan hasilnya. Artinya, kita dapat mengisolasi bagian kode untuk melakukan tugas tertentu, lalu menggunakan ulang bagian tersebut di tempat lain dalam program kita.
- Program yang baik terdiri dari beberapa fungsi. Fungsi merupakan blok kode yang diberi nama dan dirancang untuk melakukan suatu tugas tertentu

# Jenis-jenis Fungsi

Fungsi dibedakan menjadi dua, yaitu fungsi internal/bawaan (built in function) dan user-defined functions (fungsi eksternal):

- Fungsi bawaan merupakan fungsi yang sudah tersedia dalam bahasa pemrograman tanpa perlu mendefinisikan terlebih dahulu. Kita dapat langsung menggunakan fungsi ini dengan memanggil nama fungsi tersebut. Sebelumnya kita sudah menggunakan beberapa fungsi bawaan seperti `print()`, `input()`, `int()`, `bool()`, `list()`, dll. Lengkapnya bisa membaca dokumentasi resmi Python.
- User-defined functions (fungsi eksternal) adalah fungsi yang kita definisikan sendiri. Fungsi yang kita definisikan sendiri digunakan untuk membuat sebuah modul terpisah dari kode program utama. Bagian ini akan dibahas lengkap di bagian 3.

# Keyword ‘def’

- Untuk membuat fungsi di Python, Kita dapat menggunakan kata kunci 'def' diikuti dengan nama fungsi dan tanda kurung yang berisi parameter (jika ada). Lalu diakhiri dengan titik dua (:).
- Blok kode yang menjalankan tugas khusus dari fungsi tersebut ditulis pada baris baru dengan indentasi (penulisan yang menjorok ke dalam).
- Keyword 'def' adalah fungsi berperan penting dalam struktur dan logika dari kode yang kita tulis. Saat kita menulis 'def' diikuti oleh nama fungsi dan tanda kurung (), Python akan mendefinisikan sebuah fungsi.
- langkah-langkah dalam mendefinisikan fungsi:
    - Tentukan nama fungsi: memiliki aturan sama seperti penamaan pada variabel.
    - Tentukan parameter: merupakan variabel yang tercantum di dalam tanda kurung dan dipisahkan dengan tanda koma jika ada lebih dari satu (opsional).
    - Titik dua: diperlukan pada akhir dari header. Header merupakan baris pertama dari fungsi yang mencakup nama fungsi dan parameter.
    - Buat blok kode fungsi: mendefinisikan badan fungsi (body) dimulai pada baris berikutnya dari header dan harus memiliki tingkat indentasi yang sama.
    - Tentukan output dengan 'return'

Keyword 'def'

```
def nama_fungsi(argumen1, argumen2,...):
  # Blok kode yang menjalankan tugas
  return hasil
```

# Contoh Keyword ‘def’

- Contoh disamping merupakan fungsi yang paling sederhana. Pada baris pertama, kita menggunakan keyword def untuk memberitahu Python bahwa kita mendefinisikan suatu fungsi. Kemudian, kita beri nama salam pada fungsi tersebut. Fungsi tersebut tidak membutuhkan informasi/parameter untuk melakukan tugasnya, sehingga tanda kurungnya kosong (tanda kurung tetap wajib meski kosong). Akhirnya, definisi fungsi tersebut diakhiri dengan tanda titik dua (:). Hal ini juga menandakan akhir dari header fungsi tersebut.

- Selanjutnya, kita isi bagian dari body yang memiliki indentasi sama dengan suatu expression atau instruksi tertentu. Pada fungsi tersebut, baris pertama dari body kita isi dengan doc string yang menjelaskan mengenai apa yang dilakukan fungsi tersebut. Pada baris setelahnya, kita isi dengan instruksi print(“Assalamualaikum wr.wb”). Terakhir, kita mengetikkan salam() untuk menggunakan/memanggil fungsi yang telah kita buat.


# Function dengan Parameter

- Di dalam tanda kurung dari salam(user), user merupakan sebuah parameter. Ketika sebuah fungsi memiliki parameter, kita harus memberikan argumen saat memanggilnya atau fungsi tersebut tidak akan bekerja (error).
- Seperti contoh di samping line 19 ketika mengetik salam() dan dirunning output nya eror
- sedangkan line 23, ketika mengetik salam(“Nadya”) maka outputnya sesuai dengan blok kode yang didefinisikan.

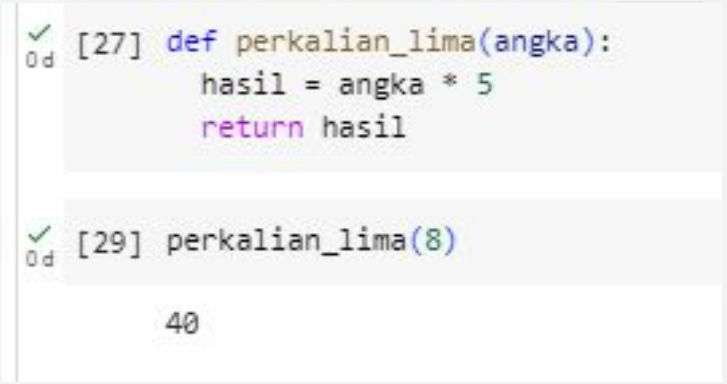
```
[17] def salam(user):
       print("Assalamualaikum wr.wb {user}")
```

```
[19] salam()

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-19-5e16eb799c18> in <cell line: 1>()
----> 1 salam()

TypeError: salam() missing 1 required positional argument: 'user'

TELUSURI STACK OVERFLOW
```

```
[23] salam("Nadya")

Assalamualaikum wr.wb Nadya
```

# Function dengan mengembalikan suatu nilai (return)

- Untuk membuat sebuah fungsi yang mengembalikan suatu nilai, kita dapat menggunakan keyword return. Keyword return digunakan untuk mengirim hasil kalkulasi fungsi kembali ke pemanggil. Fungsi yang tidak memiliki keyword return dinamakan void function dan dia mengembalikan None



# Perbedaan return dan print

- fungsi print berguna untuk memperlihatkan nilainya
- return adalah keyword khusus untuk mengeluarkan output dari sebuah fungsi (def )
- Tipe dari perkalian_lima_print itu disebut NoneType sementara, perkalian_lima_return hasilnya int
- Ketika membuat fungsi dengan keyword def sering disandingkan dengan return. Karena dengan return kita bisa melanjutkan dengan fungsi di python lainnya, seperti split, dll.

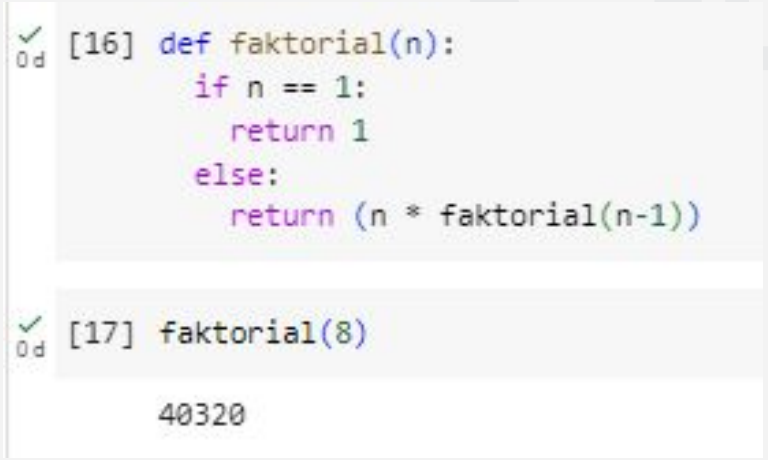
```
[15] type(perkalian_lima_return(8))

     int

     type(perkalian_lima_print(5))

     25
     NoneType
```

```
[5]  def perkalian_lima_return(angka_):
       hasil_ = angka_ * 5
       return hasil_

[6]  perkalian_lima_return(8)

     40
```

```
def perkalian_lima_print(angka):
  hasil = angka * 5
  print(hasil)

[2]  perkalian_lima_print(10)

     50
```

# Fungsi Rekursif

- Di Python, suatu fungsi dapat memanggil dirinya sendiri.
- Fungsi yang memanggil dirinya sendiri disebut recursive, sedangkan proses eksekusinya disebut recursion
- Salah satu contoh paling umum digunakan dari fungsi rekursif adalah menghitung nilai faktorial. Faktorial (n!) adalah hasil kali semua bilangan dari 1 sampai dengan n.
- Agar fungsi rekursif tidak memanggil dirinya sendiri secara terus menerus, kita juga harus menentukan kasus disaat nilai n = 0. Pada contoh tersebut, kita mengetikkan sintaks return 1 pada saat n = 0. Jika sebuah fungsi rekursif tidak pernah mencapai base case, ia akan terus melakukan pemanggilan rekursif selamanya, dan program akan terus berjalan. Hal ini disebut sebagai infinite recursion



# 3. Introduction to library

# Library Python

- Pada pertemuan Programming I kita sudah sekilas membahas library
- libraries pada python menyimpan kode-kode yang dapat diambil dan digunakan
- libraries berisi kumpulan modul dan kode yang bisa kamu gunakan berulang-ulang untuk berbagai kebutuhan
- libraries python memudahkan mengambil kode tertentu yang kita butuhkan.
- Python mempunyai banyak *library* pihak ketiga yang dapat digunakan untuk menambahkan fungsionalitas tambahan ke program. *Library* ini sering disebut *"package"* dan dapat diinstal menggunakan keyword PIP, yang disertakan dengan Python.
- Misalnya, Kita dapat menginstal perpustakaan numpy, yang menyediakan dukungan untuk komputasi numerik
- Kita mengimpor dan menggunakan fungsi dan kelas yang disediakan oleh numpy di program seperti ini:

```
import numpy as np
x = np.array([1, 2, 3])
print(x) # prints [1 2 3]
```

# Macam-macam Library Python

Ada banyak jenis library pada python, salah satu contohnya adalah:

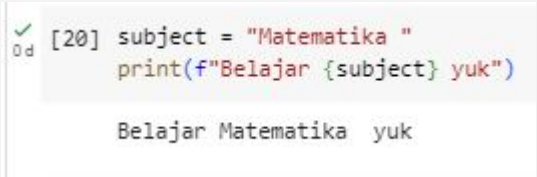
- Numpy (numerical python): array atau tempat penyimpanan dan pengelompokkan data dengan tipe yang sama.


- Pandas: menyediakan struktur data dan analisis data yang mudah digunakan

# 4. String manipulation library

# Creating string using ‘ ‘ character

- Di Python, string adalah jenis variabel yang mewakili urutan karakter. kita dapat membuat string menggunakan karakter spasi, ” atau ' '.
- Pada contoh line 18, kita membuat string “Belajar Yuk” menggunakan karakter spasi. Kami menggunakan karakter spasi untuk memisahkan kata-kata dalam string.
- Dalam contoh 20, Kita menggunakan sintaks f untuk memasukkan nilai variabel “subject” ke dalam string.
- Jadi Python dimungkinkan untuk membuat string menggunakan karakter spasi ‘ ‘. Ini berguna ketika kita perlu memisahkan elemen berbeda dalam sebuah string.

# Formatting strings with ‘+’ and format()

- Dengan Python, dimungkinkan untuk memformat string menggunakan operator '+' dan fungsi format().
- fungsi format() digunakan untuk memformat string dengan nilai yang telah ditentukan sebelumnya. Ini memungkinkan kita memasukkan nilai numerik dengan dua tempat desimal, nilai numerik dengan eksponen, string, dan lain-lain.
- {} untuk mengizinkan penyisipan variabel dalam string.
- Singkatnya, dengan Python dimungkinkan untuk memformat string menggunakan operator '+' atau fungsi format(). Operator '+' digunakan untuk memformat string dengan nilai numerik dan string sederhana, sedangkan fungsi format() digunakan untuk memformat string dengan nilai numerik dan string yang lebih kompleks.

```
[24] number = 111.222
     date = "2024-02-14"
     print("The value is: {} {}".format(number, date))

     The value is: 111.222 2024-02-14
```

# String operators and attributes

- Operator String memungkinkan untuk melakukan operasi string sederhana seperti penggabungan, penggabungan, pemotongan, dan penggantian. seperti contoh line 26
- Pada contoh 30, kita dapat memilih substring yang diinginkan
- Pada contoh 38, kita dapat menggunakan fungsi replace untuk string
- Pada contoh 39, kita dapat mengetahui jumlah karakter dalam sebuah string.
- Pada contoh 40, kita dapat mengubah semua huruf dari string menjadi huruf besar.
- Singkatnya, dengan Python dimungkinkan untuk melakukan operasi dengan string menggunakan operator dan atribut. Operator string memungkinkan kita melakukan operasi seperti penggabungan, memangkas, dan mengganti, sedangkan atribut string memungkinkan kita mendapatkan jumlah karakter dalam string dan mengekstrak sebagian string.

```
[26] string1 = "Semangat"
     string2 = "Belajar"
     string3 = string1 + " " + string2
     print(string3)

     Semangat Belajar
```

```
[30] string = "Belajar Yuk"
     substring = string[:7]
     print(substring)

     Belajar
```

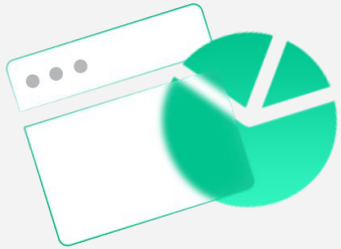
```
[38] string = "Belajar Yuk"
     substring = "Belajar"
     new_string = string.replace(substring, "Python")
     print(new_string)

     Python Yuk
```

```
[39] string = "Belajar Yuk"
     print(len(string))

     11
```

```
[40] string = "Belajar Yuk"
     upper_string = string.upper()
     print(upper_string)

     BELAJAR YUK
```

# NEXT!!

Introduction to NumPy



# Terima Kasih