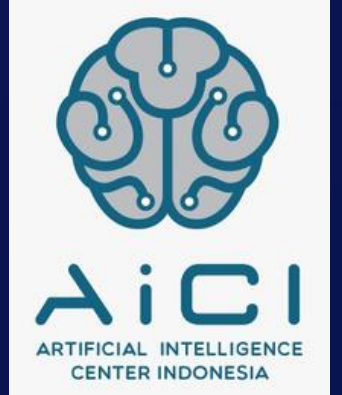




**Kampus
Merdeka**
INDONESIA JAYA



Data Operator dan Variabel



Penyusun Modul : Ratna Aditya Apsari
Editor : Delyanda Rahmadini
Hilma Arifah Roihan
M. Hamed Bagus P



Tipe Data

1. String

- Dapat berupa huruf, kata juga angka
- Ditandai dengan tanda kutip dua atau kutip satu: "Halo", 'merdekabelajar2022', 'Studi Independen'
- Tipe data ini tidak dapat melakukan operasi matematika

2. Integer

- Berupa bilangan bulat
- Tidak perlu menggunakan tanda kutip
- Dapat melakukan operasi matematika

3. Floating point number (float)

- berupa angka/bilangan pecahan atau desimal
- Tidak perlu menambahkan tanda kutip
- Python akan secara otomatis menyimpan angka dengan desimal sebagai *float*, dan *output* dari operasi *float* akan berupa *float* juga.

4. Boolean

- Istilah Boolean mengacu pada aljabar boolean yang dibuat oleh George Boole di tahun 1854. Nilai boolean hanya dapat berupa True atau False.



Variabel

Di python, kita bisa menyimpan string, integer, float, ataupun kombinasi dari ketiganya dalam suatu variabel.

String :

```
var1 = 'Hello World'
```

```
var2 = '40'
```

Integer :

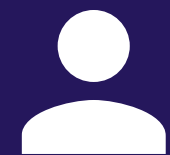
```
var3 = 10
```

Float :

```
var4 = 10.2
```

Kombinasi :

```
kombinasi = ['Hello World', 10, 10.2]
```



Penamaan Variabel

- Python bersifat *case-sensitive* yang berarti pemakaian huruf kapital atau tidak memengaruhi penamaan (variabel nama tidak sama dengan NAMA ataupun Nama)
- Beberapa keyword tidak dapat digunakan seperti:

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async',  
'await', 'break', 'class', 'continue', 'def', 'del',  
'elif', 'else', 'except', 'finally', 'for', 'from',  
'global', 'if', 'import', 'in', 'is', 'lambda',  
'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try',  
'while', 'with', 'yield']
```
- Tidak dimulai dengan angka
- Tidak menggunakan spasi
- Tidak menggunakan symbol special (@, -, +, *, dsb.)



Komentar

- Dalam pembuatan program seringkali kita memerlukan penulisan komentar untuk menambahkan informasi sehingga memudahkan pembacaan program
- Komentar adalah baris tulisan yang akan diabaikan oleh *interpreter*
- Pembuatan komentar dapat menggunakan tanda pagar (#) atau diapit dengan *triple* kutip

```
# One Line Comment
```

```
# This is one line comment
```

```
"""
```

```
Multi-Line Comment
```

```
In Python, Multi-Line comments,  
are created via a pair of triple quotes: """ """
```

```
As in this case.
```

```
"""
```



Tipe-tipe Operator

Aritmatik

Operator aritmatik python berupa operasi-operasi matematika.

Operator	Deskripsi	Contoh (misalnya A= 40 dan B= 10)
Penambahan (+)	Melakukan operasi penambahan	$A+B=50$
Pengurangan (-)	Melakukan operasi pengurangan	$A-B=30$
Perkalian (*)	Melakukan operasi perkalian	$A*B=400$
Pembagian (/)	Melakukan operasi pembagian	$A/B=4$
Modulus (%)	Melakukan operasi pembagian dan mengembalikan sisa dari pembagian tersebut	$A\%B=0$
Eksponen (**)	Melakukan operasi eksponensial terhadap suatu nilai	$A**B=40^{10}$
Pembagian floor (//)	Melakukan pembulatan ke nilai terendah untuk hasil pembagian	$13//2=6$



Tipe-tipe Operator Relasional

Opreator relasional digunakan untuk membandingkan nilai/variable antara kedua sisi dari operator tersebut dan menentukan hubungan atau relasi antara kedua nilai/variable.

Operator	Contoh (misalnya A= 40 dan B= 10)
==	(A==B)adalah true
!=	(A!=B)adalah true (A<>B)adalah true
>	(A>B)adalah true
<	(A<B)adalah true
>=	(A>=B)adalah true
<=	(A<=B)adalah true



Tipe-tipe Operator

Penetapan (assignment)

Operator penetapan digunakan untuk menetapkan suatu nilai pada suatu variable.

Operator	Deskripsi	Contoh
=	Menetapkan nilai dari sisi kanan operator untuk sisi kiri operator	$C=A+B$ akan menetapkan nilai C sebagai hasil dari penambahan A dan B
Tambah (+=)	Menambahkan nilai dari sisi kanan operator dan menetapkan hasilnya sebagai variabel di sisi kiri	$C+=A$ adalah $C=C+A$
Kurang (-=)	Mengurangkan nilai dari sisi kanan operator dan menetapkan hasilnya sebagai variabel di sisi kiri	$C-=A$ adalah $C=C-A$
Kali (*=)	Mengalikan nilai dari sisi kanan operator dan menetapkan hasilnya sebagai variabel di sisi kiri	$C*=A$ adalah $C=C*A$
Bagi (/=)	Membagi nilai dari sisi kanan operator dan menetapkan hasilnya sebagai variabel di sisi kiri	$C/=A$ adalah $C=C/A$

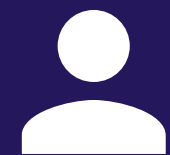


Tipe-tipe Operator

Penetapan (assignment)

Operator	Deskripsi	Contoh
Modulus (%=)	Melakukan operasi modulo nilai dari sisi kanan operator dan menetapkan hasilnya sebagai variabel di sisi kiri	$C\%=A$ adalah $C=C\%A$
Eksponen (**=)	Melakukan operasi eksponensial pada nilai dari sisi kanan operator dan menetapkan hasilnya sebagai variabel di sisi kiri	$C^{**}=A$ adalah $C=C^{**}A$
Pembagian floor (//=)	Melakukan pembagian floor pada nilai dari sisi kanan operator dan menetapkan hasilnya sebagai variabel di sisi kiri	$C//=A$ adalah $C=C//A$





Tipe-tipe Operator Logika

operator logika berguna untuk membandingkan dua nilai atau variabel untuk mendapatkan hasil Boolean (true and false). beberapa operator logikal yang dapat digunakan di python antara lain:

Operator	Deskripsi
Logical DAN (and)	Kondisinya bernilai <i>true</i> hanya jika kedua variabel bernilai <i>true</i>
Logical OR (or)	ketika salah satu dari kedua variabel adalah true, maka kondisinya menjadi true
Logical NOT (not)	mengubah kondisi logika dari suatu variabel





Bitwise

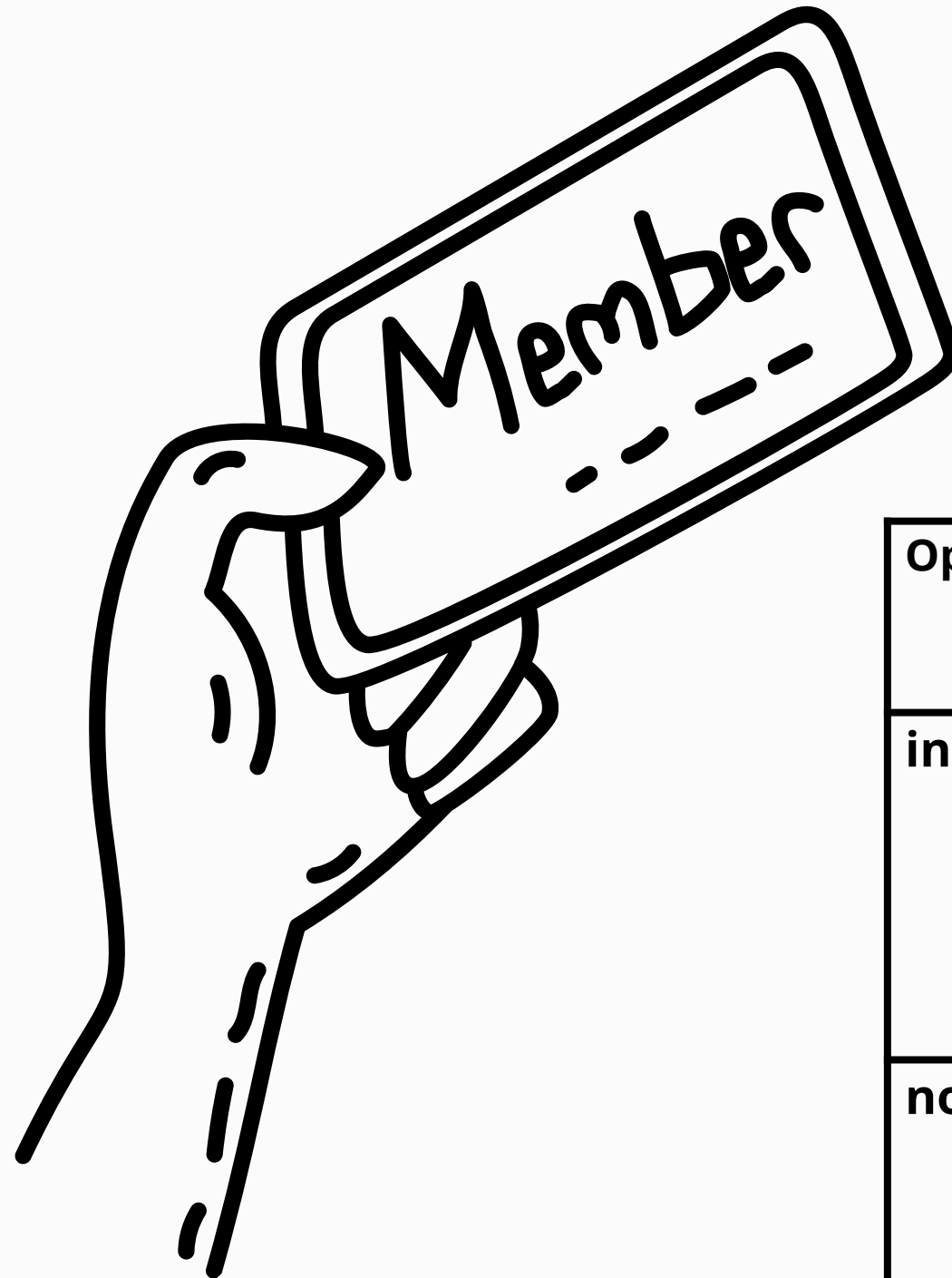
Operator bitwise melakukan operasi logika bilangan biner secara bit-per-bit. Python akan secara otomatis mengkonversi nilai menjadi bilangan biner (0 dan 1) melakukan operasi bitwise.

Operator	Deskripsi	Contoh (A = 0000 1101, B = 0001 1001, dan C = 1111)
Binary AND (&)	Melakukan operasi AND secara bit-per-bit	A & B = 0000 1001
Binary OR ()	Melakukan operasi OR secara bit-per-bit	A B = 0001 1101
Binary XOR (^)	Melakukan operasi XOR secara bit-per-bit	A ^ B = 0001 0100
Binary Ones Complement (~)	Melakukan flipping atau pembalikan bit (bit 0 menjadi 1 dan 1 menjadi 0)	~C = 0000
Binary Left Shift (<<)	Melakukan shifting atau penggeseran bit ke kiri, sesuai dengan jumlah bit yang ditentukan pada sisi kiri operator	C << 2 = 1100
Binary Right Shift (>>)	Melakukan shifting atau penggeseran bit ke kanan, sesuai dengan jumlah bit yang ditentukan pada sisi kanan operator	C >> 2 = 0011

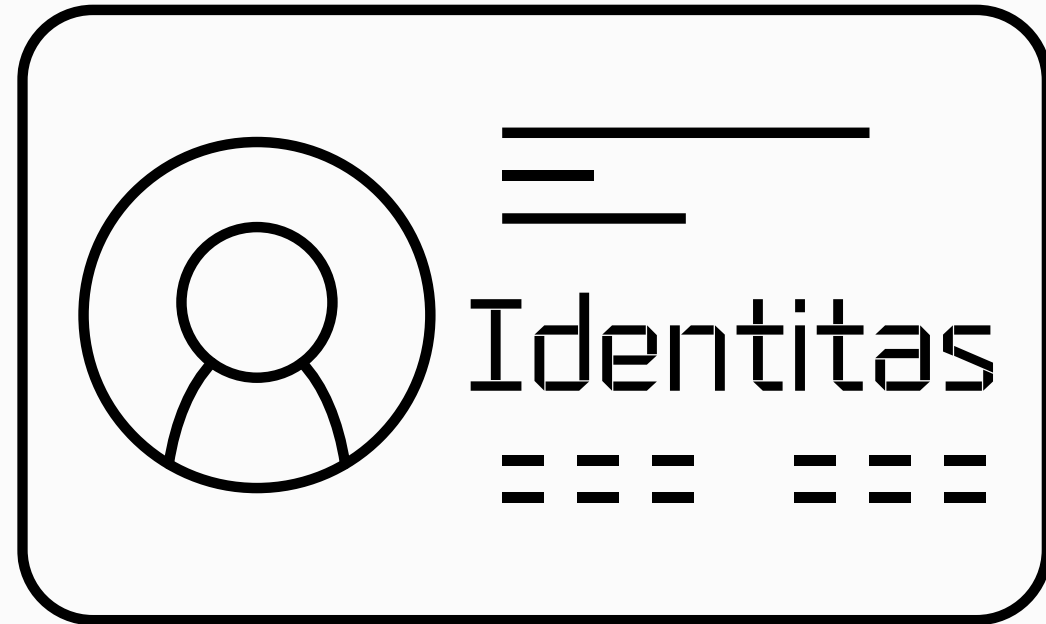


MEMBERSHIP

Operator *membership* adalah operator yang mengecek apakah suatu objek berada dalam suatu sekuens, yaitu list, tuple, atau string



Operator	Contoh
in	x in y jika ditemukan x dalam sekuens y
not in	x not in y jika x tidak ditemukan dalam sekuens y



Selain operator *membership*, operator identitas membandingkan memori dari dua obyek lokasi

Operator	Contoh
is	x is y jika indeks x sama dengan indeks y
is not	x is not y jika indeks x tidak sama dengan indeks y





Operator	Description
**	Exponentiation (raise to the power)
~ + -	Complement, unary plus and minus (method names for the last two are +@ and -@)
* / % //	Multiply, divide, modulo and floor division
+ -	Addition and subtraction
>> <<	Right and left bitwise shift
&	Bitwise 'AND'
^	Bitwise exclusive 'OR' and regular 'OR'
<= < > >=	Comparison operators
<> == !=	Equality operators
= %= /= //= -= += *= **=	Assignment operators
is is not	Identity operators
in not in	Membership operators
not or and	Logical operators

URUTAN OPERASI (OPERATOR PRECEDENCE)

Python memiliki urutan operator mana yang akan dieksekusi terlebih dahulu dibanding yang lain. Hal ini disebut juga sebagai *operator precedence* atau *order of operations*.

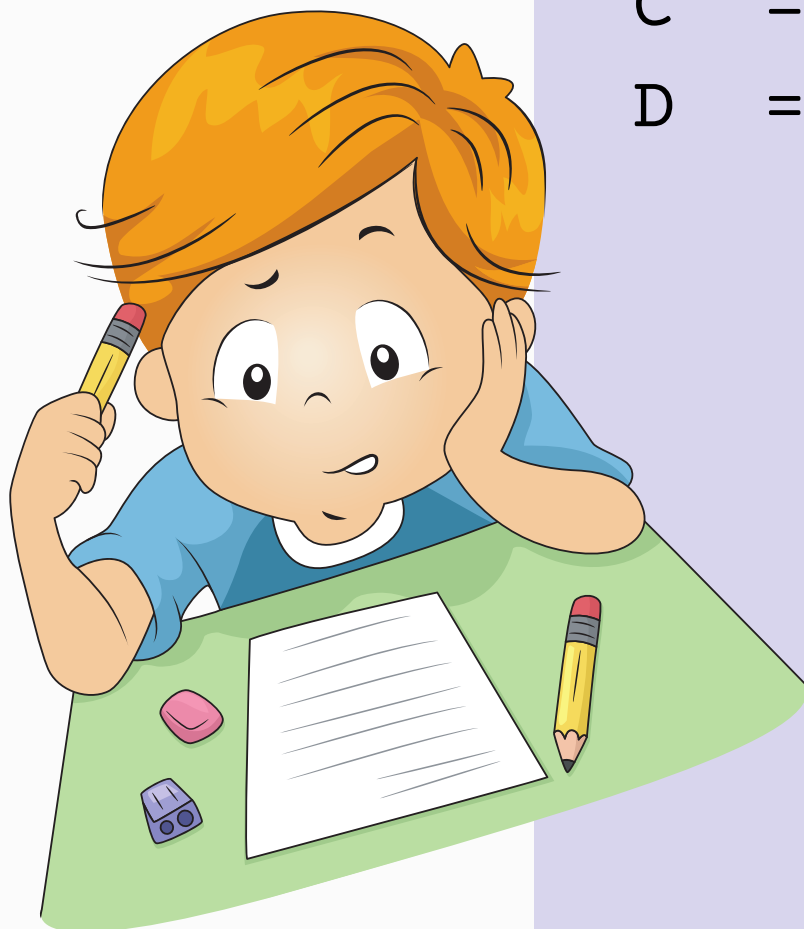
TRY IT!
OUT

Lihatlah potongan *code* di bawah. Sebelum meng-copy *paste* potongan *code* tersebut, cobalah prediksi hasil dari tiap variabel sesuai dengan urutan operasi yang telah dipelajari

A = 2
B = 5
C = 10
D = 3

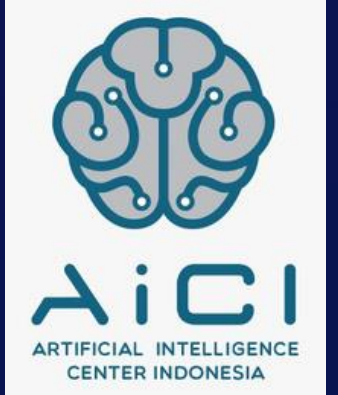
latihan_1 = (A+D)*C/B
latihan_2 = (A+D)*D /B
latihan_3 = ((A+D)*(C/B))*B
latihan_4 = A*(D*C)/B

```
print ("Hasil 1: ", latihan_1)
print ("Hasil 2: ", latihan_2)
print ("Hasil 3: ", latihan_3)
print ("Hasil 4: ", latihan_4)
```





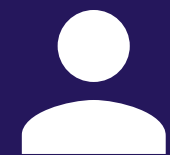
**Kampus
Merdeka**
INDONESIA JAYA



Fungsi



Penyusun Modul : Ratna Aditya Apsari
Editor : Delyanda Rahmadini
Hilma Arifah Roihan
M. Hamed Bagus P



Function

Fungsi adalah kumpulan perintah/code yang dibuat untuk melakukan pekerjaan spesifik tertentu. Perintah tersebut hanya akan dijalankan pada saat ia dipanggil. Python sendiri menyediakan beberapa fungsi built-in, seperti :

- `int ()` : mengubah string atau float menjadi integer
- `float ()` : mengubah string atau integer menjadi float.
- `input ()` : mengambil informasi yang dimasukkan oleh pengguna dan menyimpannya
- `str ()` : mengubah integer, float atau tipe data lainnya menjadi string
- `help ()` : meminta bantuan Python



Pembuatan Fungsi

Cermati *code* fungsi berikut :

```
def additon ( ) :  
    first_number = 30  
    second_number = 60  
    print (first_number + second_number)
```

- **def** berasal dari kata *define*, digunakan sebagai *keyword* ketika ingin mendefinisikan sebuah fungsi baru
- **def** harus diikuti dengan nama fungsi, dalam contoh ini, nama fungsinya adalah *addition*.
- Setelah nama fungsi, terdapat kurung () yang wajib ditulis. Kita dapat mengisi kurung () ini dengan suatu parameter input yang diperlukan, atau bisa juga dikosongkan.
- Baris pertama dari fungsi diakhiri dengan titik dua (:)
- Setiap baris memiliki fungsi (kecuali baris pertama harus diberikan
- Semua baris kode yang ada di dalam fungsi diberi indentasi untuk menandakan mana saja program yang termasuk ke dalam fungsi.



untuk menggunakan fungsi yang telah kita buat, hal yang perlu kita lakukan hanyalah memanggil fungsi tersebut. Sebagai berikut :

addition ()

Dengan memanggil fungsi tersebut, kita akan mendapatkan output dari fungsi addition sesuai dengan hasil yang telah kita program sebelumnya, yaitu 90.





Meminta input dari pengguna

Seringkali, kita membutuhkan input dari pengguna. Python menyediakan dua cara untuk meminta dan membaca input dari pengguna, menggunakan fungsi built-in berikut :

`input (prompt)`

fungsi `input ()` akan membaca input dan menyimpannya sebagai bentuk *string*. Contohnya, ketika kita membutuhkan input nama dari pengguna, kita dapat menulis *code* seperti berikut :

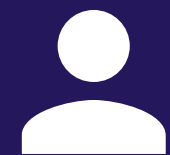
```
name = input ("What is your name ? " )  
print (name)
```




Scope

- *Scope* adalah cakupan dimana variabel tertentu berlaku
- Suatu variabel yang dideklarasikan di dalam fungsi hanya berlaku di fungsi tersebut. Di luar fungsi, variabel tersebut dianggap tidak ada
- Seluruh variabel yang dideklarasikan di luar fungsi adalah variabel *global*
- Jika suatu variabel di dalam fungsi ingin dibuat menjadi variabel *global* maka perlu dideklarasikan seperti berikut

```
def ucapan_selamat():  
    global tahun  
    tahun = 2022  
    print("Selamat Tahun Baru", tahun)
```



Return

- Ketika sebuah fungsi dipanggil, tidak hanya menjalankan perintah tapi juga dapat me-*return* suatu nilai
- Perhatikan perbedaan dua fungsi berikut:

```
def penjumlahan_tanpa_return_value(a,b):  
    print("hasil penjumlahan =",a+b)
```

```
def penjumlahan(a,b):  
    print("hasil penjumlahan =",a+b)  
    return a+b
```

```
c = penjumlahan_tanpa_return_value(3, 4)  
print(c)  
d = penjumlahan(5, 6)  
print(d)
```

✓ 0.1s

hasil penjumlahan = 7

None

hasil penjumlahan = 11

11



Try it Out!

Buatlah fungsi python untuk fungsi pengurangan, perkalian dan pembagian dari dua bilangan dengan variable `first_number` dan `second_number` seperti contoh di atas. kemudian, ubahlah fungsi tersebut agar pengguna dapat meng-input bilangan mereka sendiri.

