

# PENGANTAR PEMROGRAMAN (21H07110202) (Python)



# Table of content

- Identifies
- Keywords
- Statements
- Expressions.
- Variables
- Operators
- Data types
- Indentation
- Comments
- Reading Input
- Print Output
- Conversi type data



# Identfier

Identfier atau python identfier adalah nama yang digunakan untuk mengidentifikasi atau memberikan identitas untuk variabel, function, class, module, ataupun object lainnya. Identfier bisa diawali dengan huruf A-Z atau a-z atau underscore (\_) diikuti huruf, underscore, dan digit (0-9). Adapun aturan penulisan identfier di Python yaitu sebagai berikut:

- Identfier class diawali dengan huruf kapital dan identfier lainnya dengan huruf kecil.
- Identfier yang diawali underscore dimaksudkan untuk penggunaan pribadi.
- Identfier yang diawal dengan dua buah underscore adalah identfier yang sangat pribadi.
- Identfier yang diawali dan diakhiri dengan dua underscore merupakan nama khusus yang telah ditetapkan Python.

# Keywords

Keywords pada python adalah kata-kata khusus yang memiliki arti dan tujuan tertentu dan tidak dapat digunakan untuk apa pun selain tujuan khusus tersebut. Berikut adalah keywords pada python:

```
In [1]: help("keywords")
```

Here is a list of the Python keywords. Enter any keyword to get more help.

False	class	from	or
None	continue	global	pass
True	def	if	raise
and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield
break	for	not	

# Statements

Statement adalah pernyataan atau instruksi yang diberikan untuk dieksekusi oleh mesin. Interpreter Python bertugas menginterpretasikan statement menjadi perintah yang sesuai. Penulisan statement di Python tidak diakhiri dengan tanda titik koma (;). Contohnya sebagai berikut:

```
In [2]: a = 5  
        b = 4  
        c = a + b  
        print(c)  
  
9
```



# Statements

Dari code tersebut setiap statement dipisahkan oleh karakter atau karakter menandakan suatu baris telah selesai. Namun tanda titik koma (;) bisa digunakan untuk kasus tertentu. Misalnya menuliskan statement dalam satu baris seperti berikut:

```
In [3]: a = 5; b = 4; c = a + b  
        print(c)  
9
```

Selain itu ada kalanya penulisan statement bisa sangat panjang jika dituliskan dalam satu baris. Solusinya kita bisa menuliskannya dengan multiple baris menggunakan tanda (;).

# Expressions

Expressions adalah kombinasi dari operator yang diinterpretasikan untuk menghasilkan beberapa nilai lain. Dalam bahasa pemrograman apa pun, expressions dievaluasi sesuai dengan prioritas operatornya. Sehingga jika ada lebih dari satu operator dalam suatu expressions, maka prioritasnya menentukan operasi mana yang akan dilakukan terlebih dahulu.

Contoh:

Constant Expressions	Arithmetic Expressions	Integral Expressions	Floating Expressions
<pre>x = 5 y = 4.5 z = x + y print(z)</pre> <p>9.5</p>	<pre>x = 5 y = 2 Output_1 = x * y Output_2 = x / y  print(Output_1) print(Output_2)</pre> <p>10 2.5</p>	<pre>x = 5 y = int(2.5) Output_1 = x + y  print(Output_1)</pre> <p>7</p>	<pre>x = 5 y = 2 Output_1 = x / y  print(Output_1)</pre> <p>2.5</p>

# Expressions

## Relational Expressions

```
x = 5
y = 2
a = 3
b = 4

Output_1 = (a + b) > (x-y)

print(Output_1)
```

True

## Logical Expressions

```
x = True
y = False

Output_1 = x and y

print(Output_1)
```

False

## Bitwise Expressions

```
a = 12
x = a << 2
y = a >> 1

print(x, y)
```

48 6

## Combinational Expressions

```
a = 12
x = (a * 2) + (a >> 1)

print(x)
```

30



# Variables

Sebuah variabel dapat memiliki nama pendek (seperti x dan y) atau nama yang lebih deskriptif. Aturan untuk variabel Python:

- Nama variabel harus dimulai dengan huruf atau karakter garis bawah
- Nama variabel tidak boleh diawali dengan angka
- Nama variabel hanya boleh berisi karakter alfanumerik dan garis bawah (A-z, 0-9, dan \_)
- Nama variabel sensitif huruf besar/kecil (usia, Usia, dan USIA adalah tiga variabel berbeda)

Contoh:

```
myname = "Fadil"  
Myname = "Jidil"  
MyName = "Ilham"  
MYNAME = "Ihza"  
  
print(myname, Myname, MyName, MYNAME)
```

```
Fadil Jidil Ilham Ihza
```

# Operators

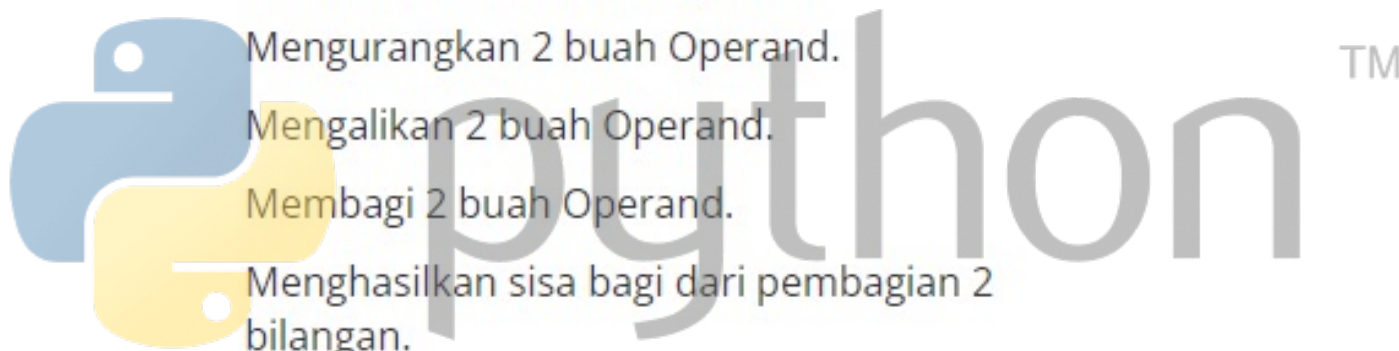
Operator pada Pemrograman Python dapat disimbolkan dengan tanda atau karakter seperti +, -, \*, /, \*\*, % dan sebagainya. Contoh sederhana seperti operasi penjumlahan dari  $1+2=3$ . Dimana angka 1 dan 2 disebut sebagai operand yaitu nilai yang dioperasikan oleh operator, sedangkan karakter + disebut sebagai operator. Bahasa Pemrograman Python mendukung berbagai macam jenis operator. Secara garis besar, Python memiliki tujuh jenis operator seperti:

- Operator Aritmatika
- Operator Pembanding
- Operator Penugasan
- Operator Logical
- Operator Keanggotaan
- Operator Identitas
- Operator Bitwise
- Operator Precedence
- Operator Associativity



# Operator Aritmatika

No	Operator dan Simbol	Deskripsi
1	Penjumlahan (+)	Menjumlahkan 2 buah Operand.
2	Pengurangan (-)	Mengurangkan 2 buah Operand.
3	Perkalian (*)	Mengalikan 2 buah Operand.
4	Pembagian (/)	Membagi 2 buah Operand.
5	Modulus (%)	Menghasilkan sisa bagi dari pembagian 2 bilangan.
6	Pemangkatan (**)	Memangkatkan nilai Operand.
7	Pembagian Bulat (//)	Sama dengan Pembagian hanya saja, angka dibelakang koma akan dihilangkan/dibulatkan.



# Operator Pembanding

No	Operator	Simbol
1	Lebih Besar	>
2	Lebih Kecil	<
3	Sama Dengan	==
4	Tidak Sama dengan	!=
5	Lebih Besar Sama dengan	>=
6	Lebih Kecil Sama dengan	<=



python<sup>TM</sup>

# Operator Penugasan

No	Operator	Simbol
1	Pengisian	=
2	Penjumlahan	+=
3	Pengurangan	-=
4	Perkalian	*=
5	Pembagian	/=
6	Sisa Bagi	%=
7	Pemangkatan	**=

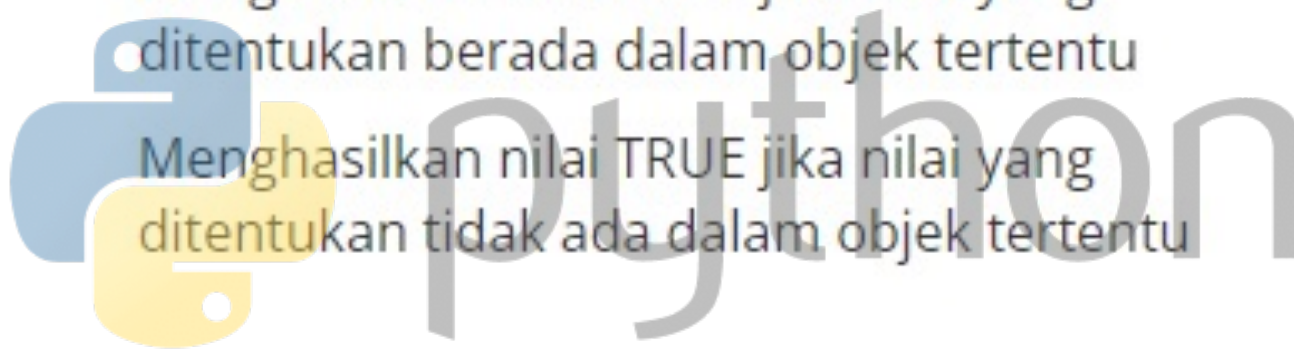


# Operator Logical

Operator	Tugas	Contoh
and	Mengembalikan True jika dua statement sama-sama benar	x and y
or	Mengembalikan True jika salah satu statement bernilai benar	x or y
not	Menegasikan hasil. True menjadi False dan sebaliknya	not x

# Operator Keanggotaan

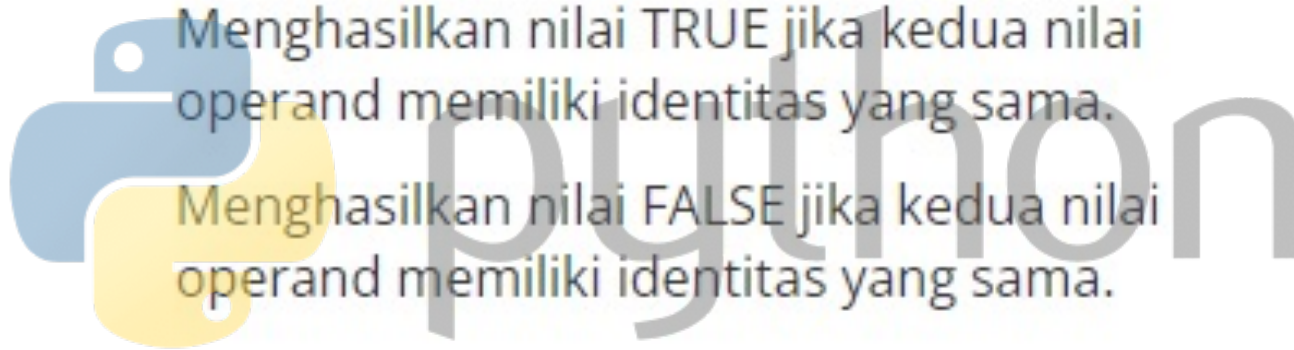
No	Operator	Tugas
1	in	Menghasilkan nilai TRUE jika nilai yang ditentukan berada dalam objek tertentu
2	not in	Menghasilkan nilai TRUE jika nilai yang ditentukan tidak ada dalam objek tertentu





# Operator Identitas

No	Operator	Tugas
1	is	Menghasilkan nilai TRUE jika kedua nilai operand memiliki identitas yang sama.
2	is not	Menghasilkan nilai FALSE jika kedua nilai operand memiliki identitas yang sama.





# Operator Bitwise

No	Operator	Simbol	Tugas
1	AND	&	Mengembalikan bit 1 jika dua bit bernilai 1.
2	OR		Mengembalikan bit 1 jika salah satu bit bernilai 1.
3	XOR	^	Mengembalikan bit 1 jika hanya satu bit saja yang bernilai 1.
4	NOT	~	Membalikkan semua bit.
5	Left Shift	«	Menggeser bit ke kiri dengan mendorong digit 0 dan membiarkan bit paling kiri terlepas.
6	Right Shift	»	Menggeser bit ke kanan dengan mendorong salinan digit sebelah kiri dan membiarkan digit sebelah kanan terlepas.

# Operator Precedence

```
x,y,z = 2,3,4
print(x + y)
```

5

# Operator Associativity

```
x = 2
y = x * 3 + 3 // 3
print(y)
```

7

# Data types

Dalam pemrograman, tipe data merupakan konsep penting. Variabel dapat menyimpan data dari tipe yang berbeda, dan tipe yang berbeda dapat melakukan hal yang berbeda. Berikut tipe data pada python:

Text Type:

`str`

Numeric Types:

`int`, `float`, `complex`

Sequence Types:

`list`, `tuple`, `range`

Mapping Type:

`dict`

Set Types:

`set`, `frozenset`

Boolean Type:

`bool`

Binary Types:

`bytes`, `bytearray`, `memoryview`

None Type:

`NoneType`



# Data types

Example	Data Type
<code>x = "Hello World"</code>	str
<code>x = 20</code>	int
<code>x = 20.5</code>	float
<code>x = 1j</code>	complex
<code>x = ["apple", "banana", "cherry"]</code>	list
<code>x = ("apple", "banana", "cherry")</code>	tuple
<code>x = range(6)</code>	range
<code>x = {"name" : "John", "age" : 36}</code>	dict
<code>x = {"apple", "banana", "cherry"}</code>	set
<code>x = frozenset({"apple", "banana", "cherry"})</code>	frozenset
<code>x = True</code>	bool
<code>x = b"Hello"</code>	bytes
<code>x = bytearray(5)</code>	bytearray
<code>x = memoryview(bytes(5))</code>	memoryview
<code>x = None</code>	NoneType

# Indentation

Indentasi adalah penulisan yang agak menjorok masuk ke dalam. Dalam sebuah artikel biasanya indentasi digunakan untuk menunjukkan paragraf baru atau paragraf selanjutnya. Indentasi pada Python penting dipahami agar tidak terjadi error saat menjalankan program, contohnya pada penulisan code untuk looping atau perulangan seperti berikut:

```
print("Hello")
print("Im Python")
    print("Error")
```

Input In [68]  
 print("Error")  
 ^

**IndentationError:** unexpected indent

```
x = 3
if x < 4:
    print("Good")
```

Good

# Comments

Penulisan code dalam Python kita juga bisa menambahkan komentar. Komentar adalah sebuah baris code atau statement yang diabaikan interpreter Python. Komentar biasanya dituliskan hanya sebagai catatan atau penjelasan suatu baris code. Penulisan komentar pada Python terdiri dari 2 jenis yaitu satu baris dan multi baris. Komentar satu baris dituliskan dengan tanda pagar (#), sedangkan multi baris dituliskan dengan tanda petik dua sebanyak tiga kali. Contohnya sebagai berikut:

```
# Ini adalah komentar  
a = 2 # a bernilai 2  
b = 3 # b bernilai 3  
  
## Ini juga masih komentar  
print(a + 3)
```

# Reading

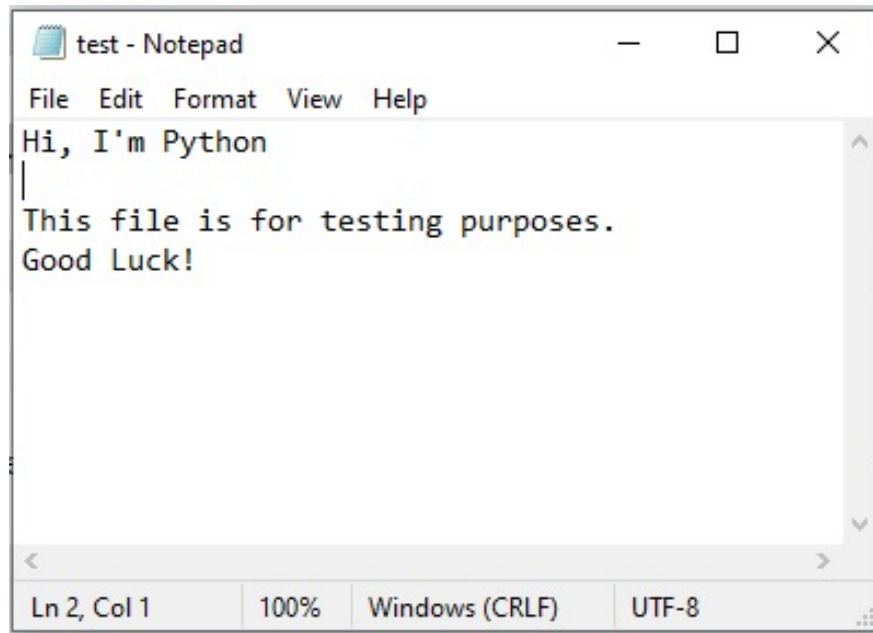
Reading adalah mode akses untuk mengatur jenis operasi yang mungkin dilakukan dalam file yang dibuka. Ini mengacu pada bagaimana file akan digunakan setelah dibuka. Mode ini juga menentukan lokasi File yang akan dibuka. Ada 6 mode akses di python yaitu:

- ('r') : Buka file teks untuk membaca. Jika file tidak ada, memunculkan kesalahan I/O. Ini juga merupakan mode default di mana file dibuka.
- ('r+') : Buka file untuk membaca dan menulis. Menimbulkan kesalahan I/O jika file tidak ada.
- ('w') : Buka file untuk menulis. Untuk file yang ada, data dipotong dan ditulis ulang. Membuat file jika file tidak ada.
- ('w+') : Buka file untuk membaca dan menulis. Untuk file yang sudah ada, data dipotong dan ditulis ulang.
- ('a') : Buka file untuk menulis. File dibuat jika tidak ada. Data yang ditulis akan disisipkan di akhir, setelah data yang ada.
- ('a+') : Buka file untuk membaca dan menulis. File dibuat jika tidak ada. Data yang ditulis akan disisipkan di akhir, setelah data yang ada.



# Reading

Contoh:



```
read_file = open('test.txt', 'r')
print(read_file.read())
```

Hi, I'm Python

This file is for testing purposes.  
Good Luck!



# Input

Contoh:

```
name = input('Nama      : ')
nim  = input('Nim       : ')
univ = input('Universitas : ')

print('\nOutput:')
print(name, nim, univ)
```

Nama :

```
name = input('Nama      : ')
nim  = input('Nim       : ')
univ = input('Universitas : ')

print('\nOutput:')
print(name, nim, univ)
```

Nama : Muhammad Fadil  
Nim : H12115006  
Universitas : Unhas

Output:  
Muhammad Fadil H12115006 Unhas

# Conversi type data

Contoh:

```
a = 3.0
b = int(a)

x = ('Muhammad Fadil', 'Fadil')
y = str(x)
```

```
print(type(a))
print(type(b))
print(type(x))
print(type(y))
```

```
<class 'float'>
<class 'int'>
<class 'tuple'>
<class 'str'>
```

# References

- [1] <https://www.dqlab.id/>
- [2] <https://www.faceprep.in/python/>
- [3] <https://www.geeksforgeeks.org/expressions-in-python/>
- [4] <https://www.w3schools.com/python/>

