

# CHAPTER 03

## VARIABLES & INSTRUCTIONS

[ACADEMY.CYBERKARTA.COM](https://ACADEMY.CYBERKARTA.COM)

# Variable dalam Programming

## Komponen Utama

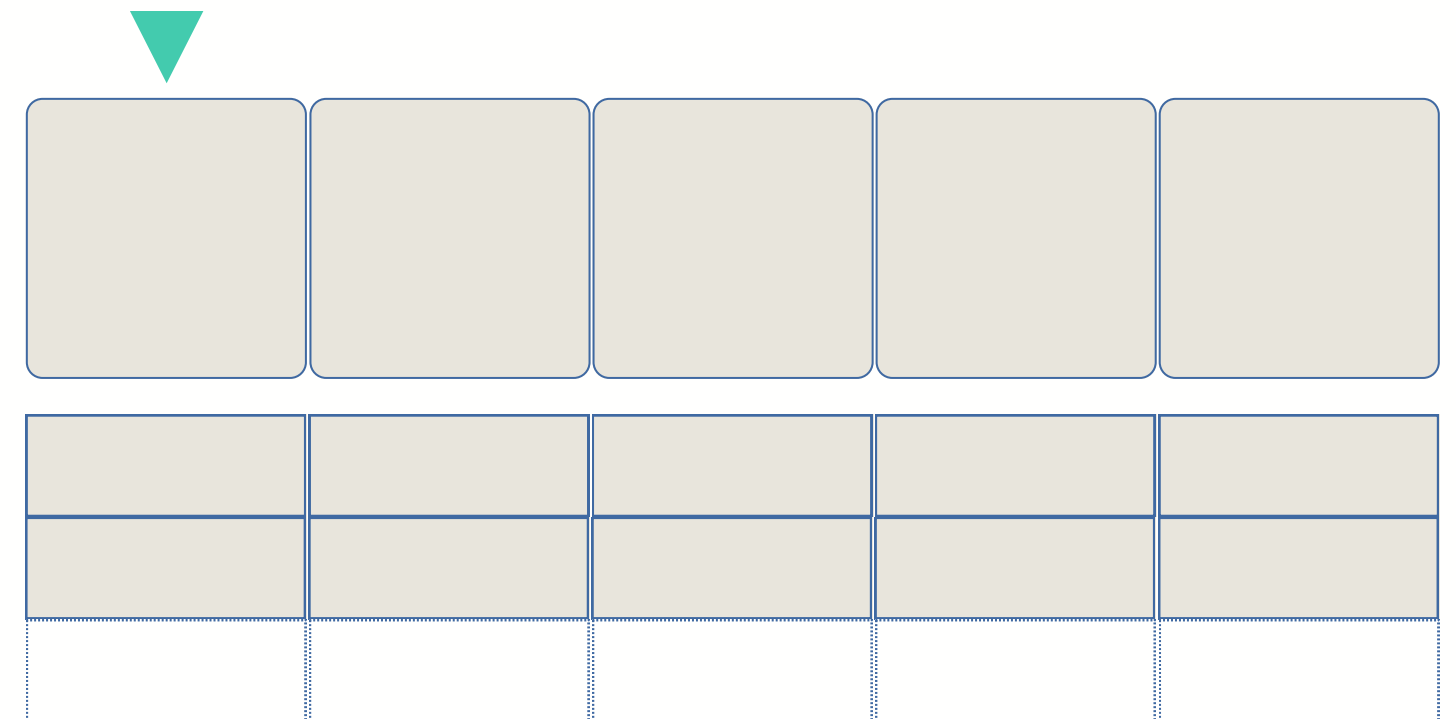
- Declaration: “pesan tempat” di memory
- Assignment: tentukan value di memory
- Di Python, declaration & assignment jadi satu

*Memory*

*name*

*data type*

*size\**



# Variable dalam Programming

```
# will throw error
varWithoutAssignment

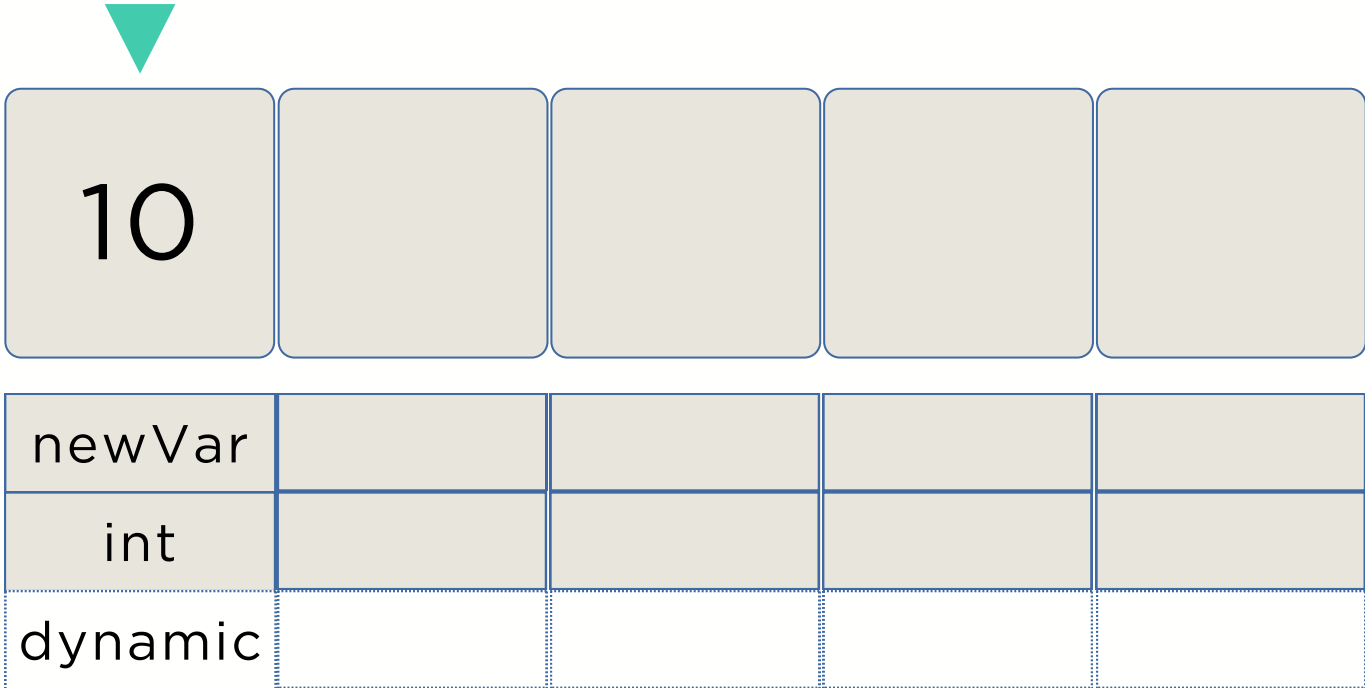
# var in Python
newVar = 10
```

Memory

name

data type

size\*



# Variable dalam Programming



```
# will throw error  
varWithoutAssignment  
  
# var in Python  
newVar = 10  
newVar = "112.74.74.190"
```

*Memory*

*name*

*data type*

*size\**

▼	"112.7 4..."				
	newVar				
	string				
	dynamic				

# Variable dalam Programming



```
# var in Python
newVar = 10

# convention for constant
LIMIT_VAR = 1000
```

*Memory*

*name*

*data type*

*size\**

10	1000			
newVar	LIMIT_VAR			
int	int			
dynamic	dynamic			

# Data Type dalam Programming

## Data Type Dasar

- int: bilangan bulat
- float: bilangan desimal
- string: teks
- boolean: true/false
- None: tak ada nilai

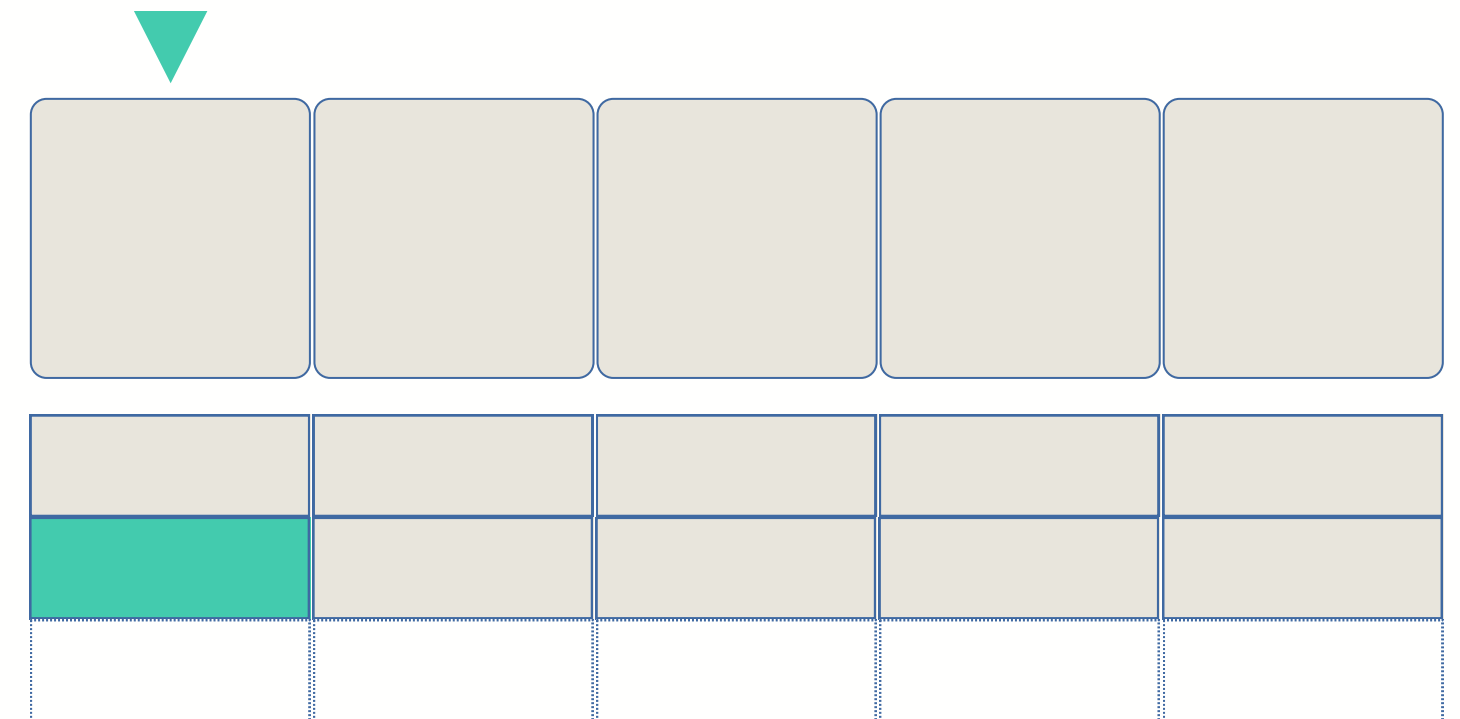
data type menentukan  
instruksi yang diperbolehkan  
oleh Python

*Memory*

*name*

***data type***

*size\**



# Instructions & Operations

Instructions: perintah yang dijalankan komputer

- operations
- control flow
- function call
- data movement

Operations: instructions di level dasar / CPU

- aritmatika (+, -, \*, /)
- komparasi (>, <, ==, !=)
- logika (AND, OR, NOT)
- assignment (=, +=, -=)
- bitwise (XOR, SHIFT)
- bagian dari (in, not in)
- identitas (is, is not)

## Behavior

- operations berlaku untuk variabel-variabel dengan data type yang sama
  - bandel? throw **TypeError**
  - ada beberapa pengecualian, misal float + int → float
- beberapa operations hanya berlaku untuk tipe data tertentu
- operations yang sama bisa menghasilkan output berbeda
  - “string” + “string” VS int + int
- berbagai jenis operations dapat digabungkan untuk memperoleh hasil yang diinginkan
  - perhatikan urutan, gunakan tanda kurung() untuk membantu
- Python sangat fleksibel dan operations kompleks bisa membingungkan. **Pahami sambil jalan.**



# Data Type Lanjutan

- list  
menyimpan elemen berurut, mutable
- tuple  
menyimpan elemen berurut, immutable
- range  
menyimpan rentang int, immutable
- dictionary  
menyimpan elemen secara key-value, mutable
- set  
menyimpan elemen unik, mutable

# Data Type Lanjutan

## Behavior List

- anggota dari list disebut element
  - element dipanggil dengan index → `list[index]`
  - index mulai dari 0
- operation dasar list
  - `append(anggota)` → tambah anggota, jadi elemen terakhir
  - `pop(index)` → hapus anggota, bisa pilih index
  - `count(anggota)` → hitung banyak anggota list
  - dan banyak lainnya: `index()`, `insert()`, `sort()`, `reverse()`, ...
- assignment anggota list
  - contoh: `list[1] = 1`
- error yang umum
  - `IndexError` → index di luar ukuran list
  - `ValueError` → value tidak ditemukan dalam list

# Data Type Lanjutan

## Behavior Tuple

- menangani element
  - element dipanggil dengan index → `tuple[index]`
  - index mulai dari 0
- operation dasar tuple
  - `count()` → hitung banyak anggota tuple
  - `index(x)` → cari index dari elemen dengan nilai x
- assignment anggota tuple → tidak boleh ❌
- error yang umum
  - `TypeError` → assignment element tuple

# Data Type Lanjutan

## Behavior Range

- menangani element
  - definisi → `range(m, n)`
  - element dipanggil dengan index → `tuple[index]`
  - index mulai dari 0
  - hanya untuk int
- operation dasar range
  - `count()` → hitung banyak anggota range
  - `index(x)` → cari index dari elemen dengan nilai x
- assignment anggota tuple → tidak boleh ✗
- error yang umum
  - `TypeError` → assignment element range, definisi dengan float

# Data Type Lanjutan

## Behavior Dictionary

- tipikal element dictionary
  - element terdiri dari key dan value
  - element dipanggil dengan `dict[key]` atau `dict.get(key)`
- operation dasar dictionary
  - `get(key, default)` → panggil elemen, lebih aman dari error
  - `keys()` → list keys
  - `values()` → list values
- assignment anggota dictionary
  - contoh: `dict["foo"] = "bar"`
- error yang umum
  - `KeyError` → key tidak ditemukan
  - `TypeError` → invalid key

# Data Type Lanjutan

## Behavior Set

- handling element dari set
  - element pasti unik
  - element tidak dapat dipanggil
  - element dapat dibandingkan
- operation dasar set
  - `add(element)` → tambah element
  - `remove(element)` → hapus element
  - `update(element)` → update element
  - operations set/himpunan → `union()`, `intersection()`, `difference()`
- set tidak support assignment → `TypeError`
- error yang umum
  - `KeyError` → elemen tidak ditemukan
  - `TypeError`