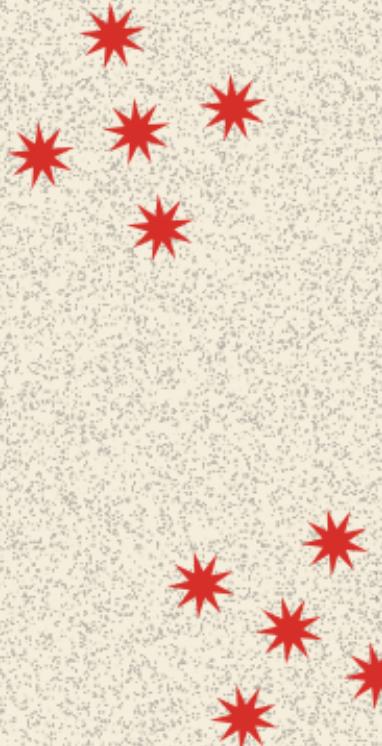


# WEEK 00

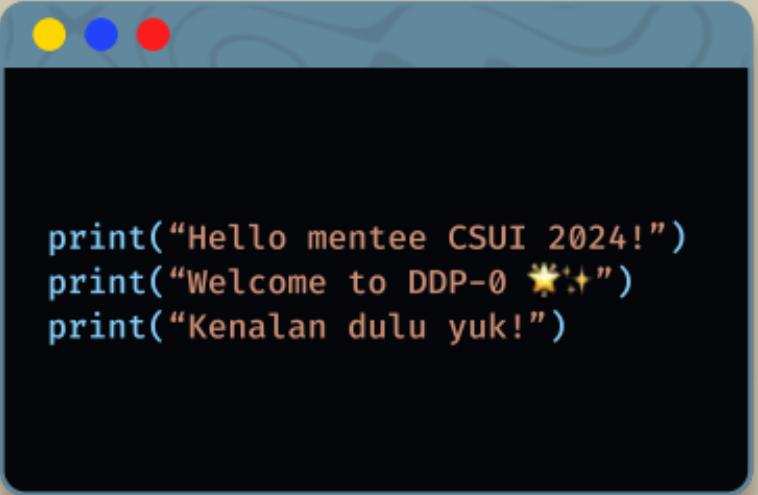
---

Data Types  
Input & Output  
Basic Arithmetic Operations



# Welcome, Mentees!

---



```
print("Hello mentee CSUI 2024!")
print("Welcome to DDP-0 ★★★")
print("Kenalan dulu yuk!")
```

# DDP-0

## Apa itu DDP-0?

DDP-0 adalah kegiatan belajar mengajar yang dipandu oleh kakak tingkat untuk membantu mahasiswa baru mempersiapkan diri menghadapi mata kuliah DDP-1.

## Mengapa Python?

Python adalah bahasa pemrograman yang ramah bagi pemula, membuatnya ideal untuk mahasiswa baru yang ingin belajar coding.

## Apa yang akan diajarkan pada DDP-0?

Materi yang dipelajari di DDP-0 adalah Basic of Python Programming yang mencakup materi hingga UTS DDP-1



# DDP-O Rules

1. Kalian akan dibimbing oleh mentor kalian setiap minggu untuk menjelaskan materi
2. Pada setiap akhir materi, ada sebuah **Lab** dan **Mini Quiz** yang akan dikumpul pada Google Form yang disediakan
3. Lab bersifat **wajib** untuk dikerjakan setiap minggunya, dikumpulkan dengan format **.py**
4. Mini Quiz bersifat **opsional** untuk dikerjakan, tetapi dapat memberikan poin plus, dikumpulkan dengan format **.doc** yang templatnya akan disediakan!
5. Lab dan Mini Quiz akan dikumpul secara bersamaan dengan di-**zip**
6. Have Fun & Enjoy! Jangan terlalu dibawa serius yaa. Anggap DDP-O sebagai persiapan kalian menuju perkuliahan nanti.



# PROGRAMMING\*

---

# Programming

**Program** adalah kumpulan instruksi atau perintah yang dikembangkan untuk dijalankan oleh suatu komputer!

Semua yang bisa dilakukan oleh **komputer** dan **mesin digital** itu berkat ratusan bahkan jutaan baris **instruksi program**. Bayangkan, semua keajaiban itu terjadi hanya karena sekumpulan kata-kata keren yang ditulis dalam bahasa mesin!

```
    event('on' + type, callback);
}

function decorate(event) {
  event = event || window.event;
  var target = event.target || event.srcElement;
  if (target && (target.getAttribute('action') || target.getAttribute('data-action'))) {
    ga(function (tracker) {
      var linkerPage = tracker.get('linkerPage');
      var cookie = 'shopify_' + linkerPage + '_linker';
      document.cookie = cookie + '=' + linkerPage;
    });
  }
}
addListener(window, 'load', function () {
  for (var i = 0; i < document.links.length; i++) {
    var link = document.links[i];
    if (link.getAttribute('data-action')) {
      link.setAttribute('data-action', link.getAttribute('action'));
    }
  }
});
```



Hey





**Contoh-contoh program di kehidupan!**



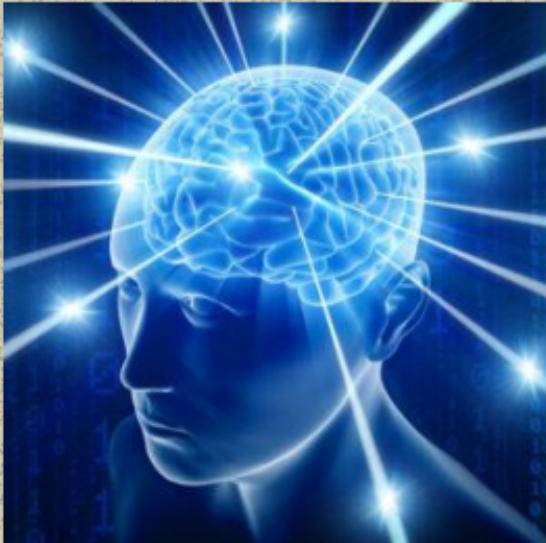
# Programming

Perantara untuk menghubungkan bahasa alami yang digunakan manusia untuk menulis instruksi-instruksi yang dapat dipahami oleh komputer disebut dengan **programming language**

Contoh dari beberapa bahasa pemrograman, yaitu C++, C, Rust, Java, Javascript, dan bahasa yang akan digunakan di DDP-O ini yaitu **Python**.



# Computational Thinking



## Computational Thinking

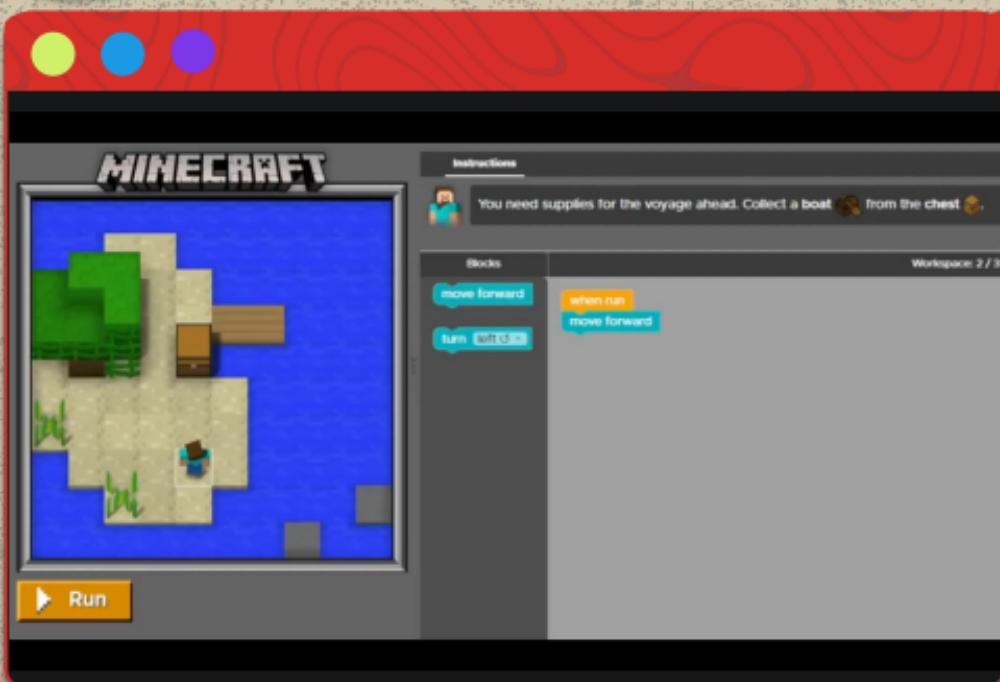
Kemampuan untuk merumuskan masalah secara tepat, memecahkan masalah secara sistematis, dan merancang solusi dengan menggunakan konsep-konsep komputasional.

Oleh karena itu, penting bagi seorang programmer untuk memiliki kemampuan computational thinking yang memadai.

# Computational Thinking

## Take a look at an example:

Nih, coba perhatiin video di sebelah. Di situ ada contoh keren tentang Computational Thinking. Jadi, langkah-langkahnya harus diikuti dengan cermat supaya bisa buka peti harta karun di akhirnya!



# Computational Thinking

---

Try it yourself at:

**code.org/minecraft**



# Let's Code!

---

Eitsss ...

Tapi kok kayaknya ada yang  
ketinggalan, apa yaa (◦·ω·◦)?

# Python Installation

Yuk, sebelum kita mulai belajar, mari kita instal Python dulu di laptop kalian! Di bawah ini ada beberapa video tutorial yang bisa diikuti mentee, tergantung sistem operasi yang kalian gunakan. Jangan sampai salah pilih yaa! ✘(\*^\_\_^\*)/



[ristek.link/PyWin](http://ristek.link/PyWin)



[ristek.link/PyMac](http://ristek.link/PyMac)

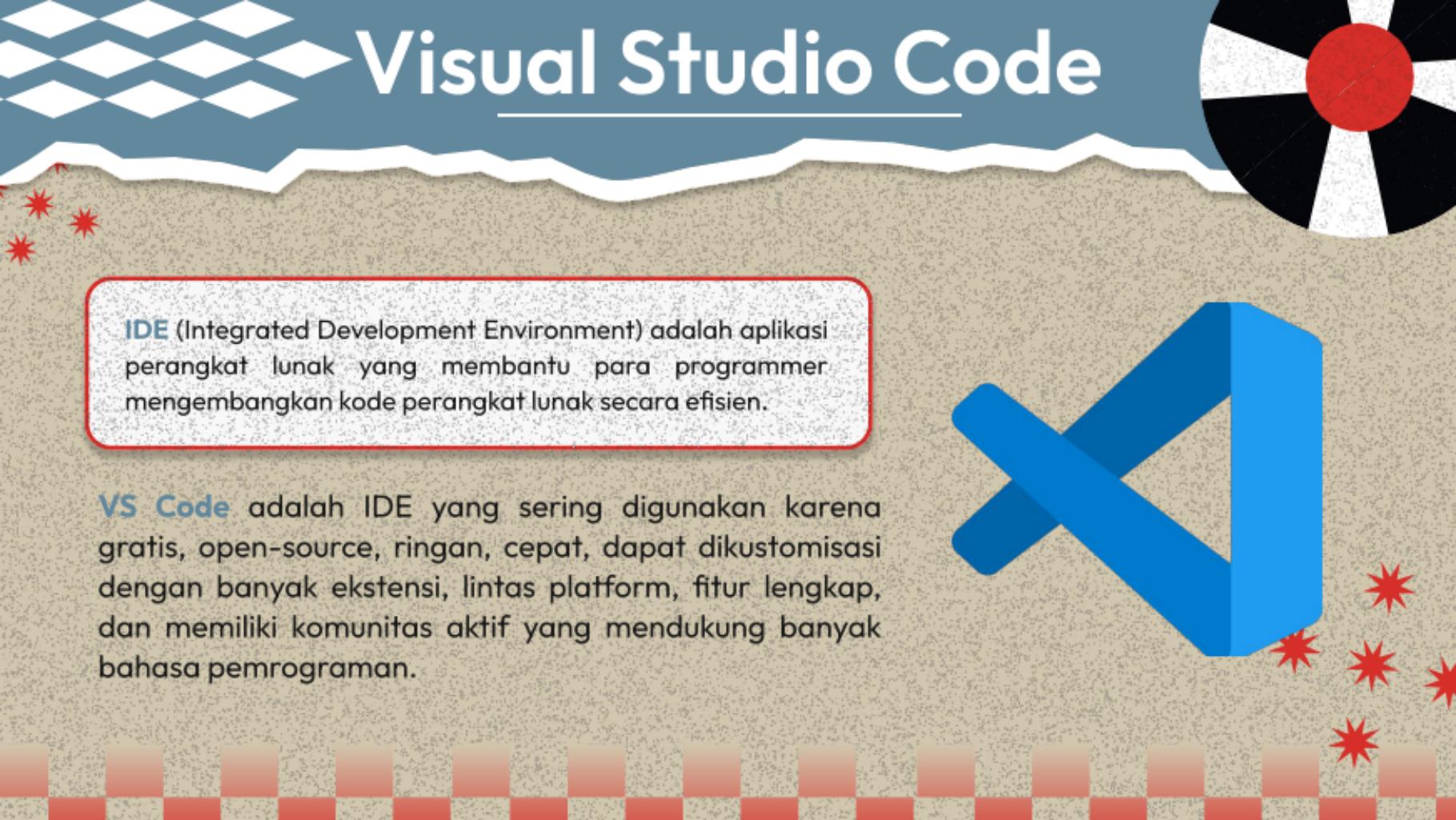
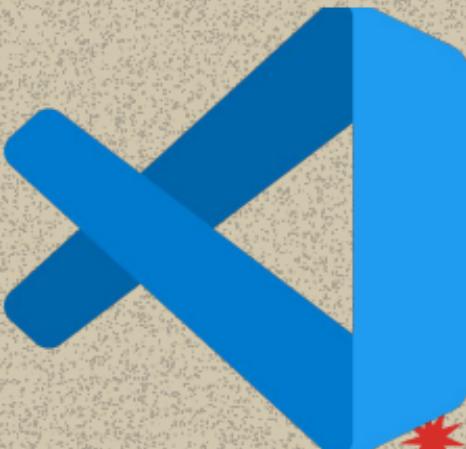


[ristek.link/PyLin](http://ristek.link/PyLin)

# Visual Studio Code

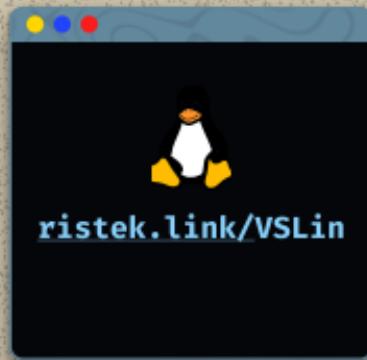
**IDE** (Integrated Development Environment) adalah aplikasi perangkat lunak yang membantu para programmer mengembangkan kode perangkat lunak secara efisien.

**VS Code** adalah IDE yang sering digunakan karena gratis, open-source, ringan, cepat, dapat dikustomisasi dengan banyak ekstensi, lintas platform, fitur lengkap, dan memiliki komunitas aktif yang mendukung banyak bahasa pemrograman.



# VS Code Installation

Setelah berhasil menginstal Python, ayo kita lanjutkan dengan menginstal VS Code di laptopmu! Di bawah ini tersedia beberapa video tutorial yang dapat kamu ikuti. Pastikan memilih OS yang sesuai agar proses instalasi berjalan lancar! \(\sim\ \sim\)/



# VS Code Extension

## Python for VSCode Extension

Install ekstensi Python untuk VSCode melalui link [ristek.link/ExPy](https://ristek.link/ExPy)

1. Jika muncul pesan seperti ini, tekan Continue.
2. Jika muncul pesan seperti ini, tekan Open Visual Studio Code.

Visual Studio Code is required to install this extension.

Don't have Visual Studio Code? [Get it now.](#)



Don't show this message again

Continue

Open Visual Studio Code?

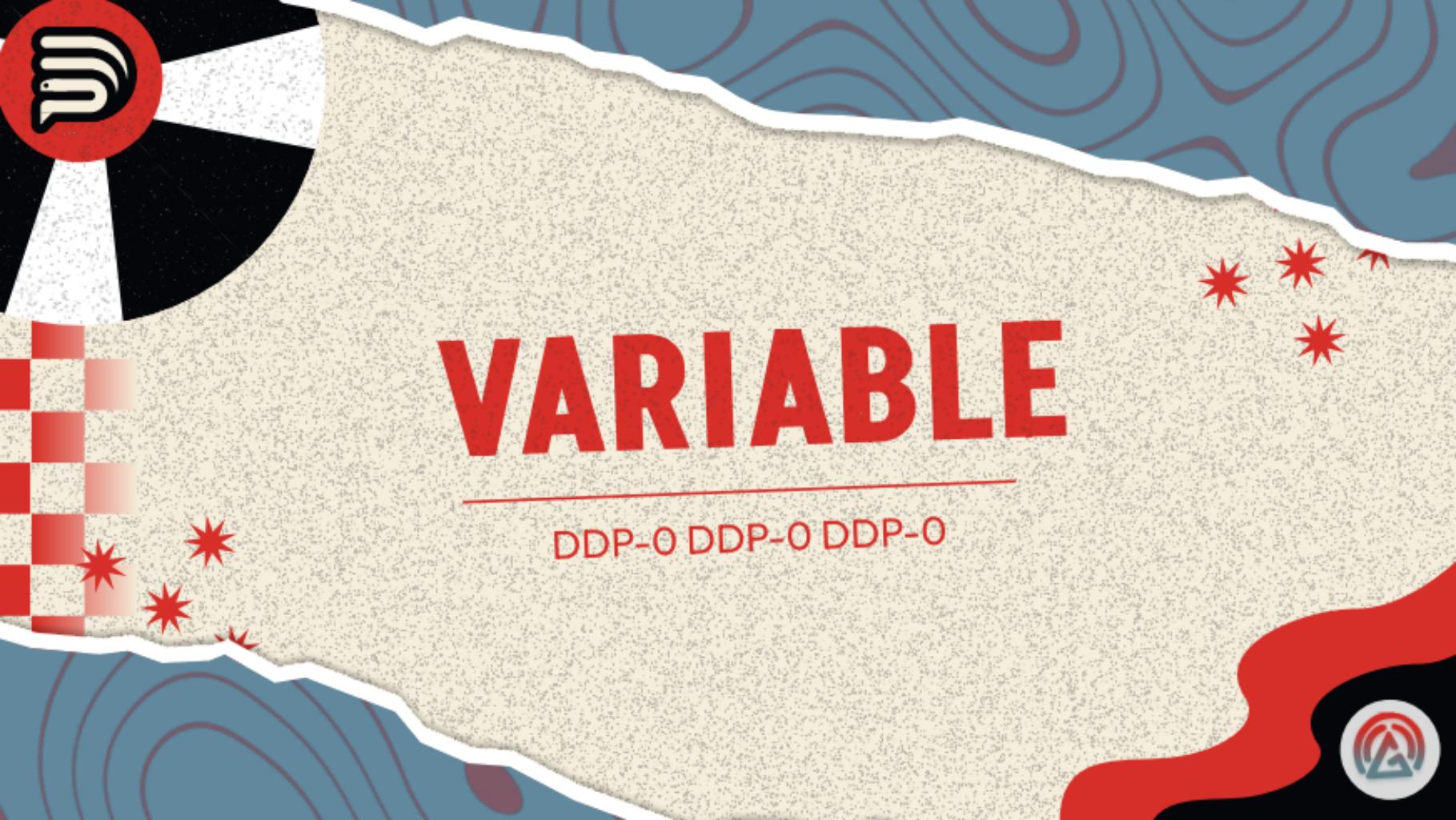
<https://marketplace.visualstudio.com> wants to open this application.

Always allow marketplace.visualstudio.com to open links of this type in the associated app

2.

[Open Visual Studio Code](#)

Cancel



# VARIABLE

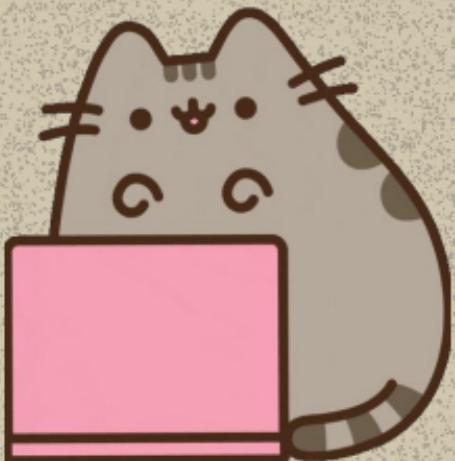
---

DDP-O DDP-O DDP-O

# Variable

**Variabel** adalah wadah bernama dalam program yang digunakan untuk menyimpan dan mengakses berbagai jenis data, seperti angka atau teks.

Lesgoooo!!!



```
# Variabel  
my_number = 1  
my_name = "John Doe"  
is_male = True
```

# Variable

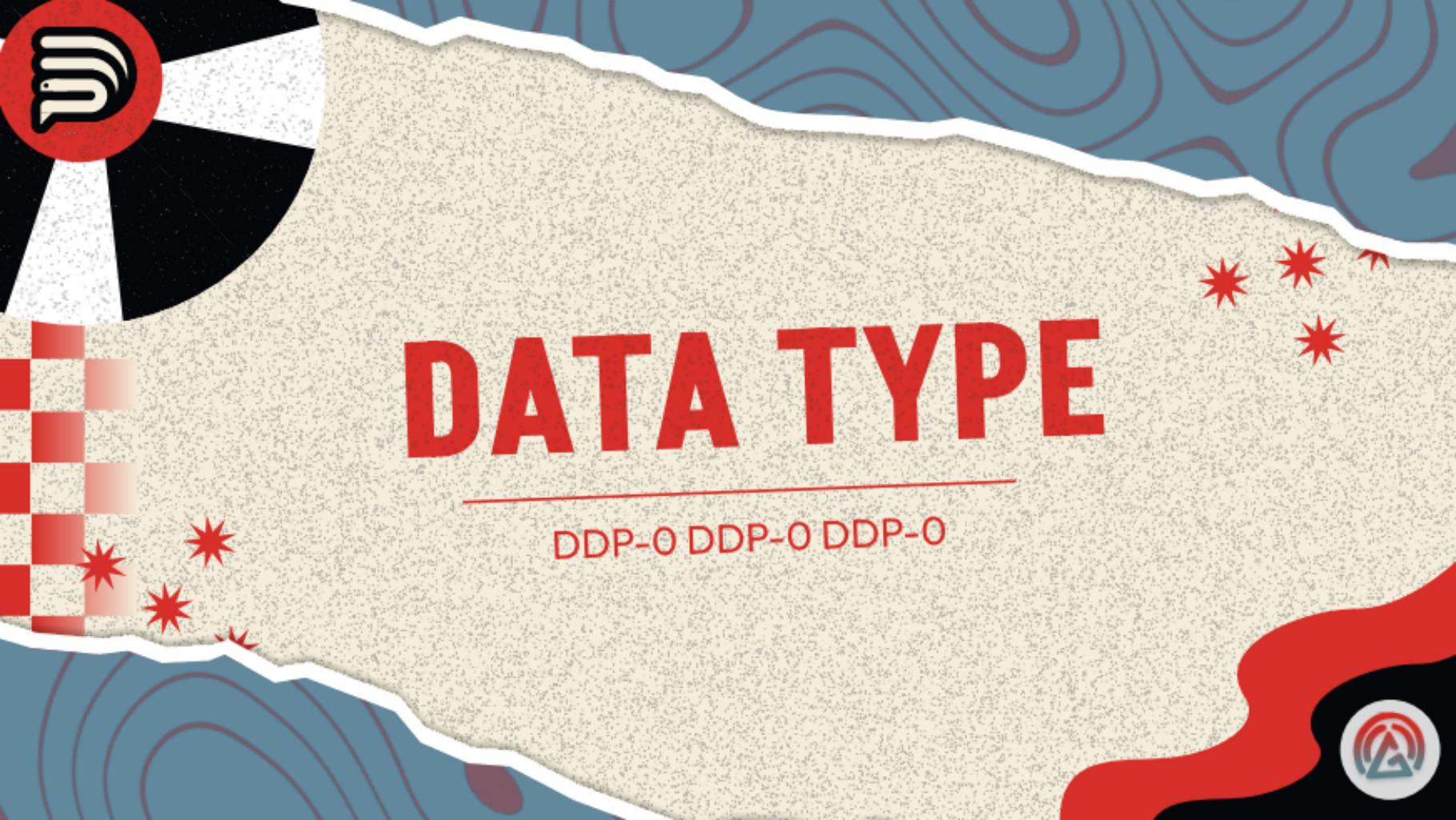
Berikut adalah syntax bahasa pemrograman Python dalam melakukan **Declaration** (membuat) dan **Assignment** (menyatakan nilai) suatu variabel.

```
# Mendeklarasikan variabel dan value
name = "Dek Depe Ganteng"
angkatan = 2024

# re-assignment variabel nama dengan value baru
name = "Dek Penol"
```

Gampang kan meong~





# DATA TYPE

---

DDP-O DDP-O DDP-O

# Data Type

**Data type** atau tipe data adalah klasifikasi data yang menentukan jenis nilai yang disimpan dan operasi apa yang bisa dilakukan pada nilai tersebut. Python memiliki berbagai tipe data.

Walau begitu, Python adalah bahasa **dinamis**, sehingga kita tidak perlu menentukan tipe data apa pun saat mendeklarasikan variabel .

Integer? String?



# Data Type

Tipe Data (Dibaca)	Tipe Data (Singkatan)	Contoh
String	str	z = "Hello World"
Integer	int	z = 481
Float	float	z = 2.7182
List	list	z = ["Euler", "Hamilton", "Dirac"]
Tuple	tuple	z = ("Euler", "Hamilton", "Dirac")
Range	range	z = range(5)
Dictionary	dict	z = {"first name": "Leonhard", "surname": "Euler"}
Set	set	z = {"Euler", "Hamilton", "Dirac"}

Boolean	bool	z = True
Bytes	bytes	z = b"Hello"
Nonetype	NoneType	z = None

Semua tipe data dapat dibagi menjadi beberapa type: **Text Type**, **Numeric Types**, **Sequence Types**, **Mapping Type**, **Set Types**, **Boolean Type**, **Binary Types**, dan **None Type**.

# INPUT & OUTPUT

---

DDP-0 DDP-0 DDP-0



# Input & Output

**Input & Output** adalah kemampuan program untuk menerima data dan menghasilkan informasi. Dengan fitur ini, programmer dapat berinteraksi dengan kode yang dibuat.

Lagi makan apa?  
(Prompt)



Ayam goyeng  
(Input)



# Input & Output

Sebuah fungsi **print()** akan mencetak pesan yang ditentukan ke layar/konsol. Pesan yang dicetak dapat berupa string atau objek lainnya.

Sebuah fungsi **input()** akan menangkap masukan yang diberikan oleh user. Masukan yang diterima akan memiliki tipe data string.

```
name = input("Give me input: ")  
print('You Said: ' + name)
```

Terminal

```
# Give me input: Dek Depenol  
# You said: Dek Depenol
```

# Input & Output

**Prompt** adalah teks yang muncul ketika meminta input dari pengguna. Prompt dibuat dengan menulis string di dalam fungsi **input()**.

Jika ingin melanjutkan ke input berikutnya atau merasa input sudah benar, cukup tekan **ENTER**.

#Ini adalah Terminal

Give me input: Dek Depe  
You said: Dek Depe

Yellow adalah **Prompt**  
Blue adalah **Input**  
Red adalah **Output**

# Input & Output

```
#Meminta input nama, makanan favorit, dan harga tanpa pajak
nama = input("Input nama: ")
makanan_suka = input("Makanan favorit: ")
harga = int(input("Harga: "))

harga_asli = 1.25*harga

#Mencetak kalimat berikut, tanda + disebut string concatenation
print("Halo, " + nama + ". Anda suka " + makanan_suka +
      ". Total harga adalah Rp" + str(harga_asli))

#Alternatif dengan f-string formatting:
print(f'Halo, {nama}. Anda suka {makanan_suka}' +
      f'. Total harga adalah Rp{harga_asli}'')
```

## Contoh Input dan Output

```
Input nama: Exusiai
Makanan favorit: Apple Pie
Harga: 50000
Halo, Exusiai. Anda suka Apple Pie. Total harga adalah Rp62500.0
Halo, Exusiai. Anda suka Apple Pie. Total harga adalah Rp62500.0
```

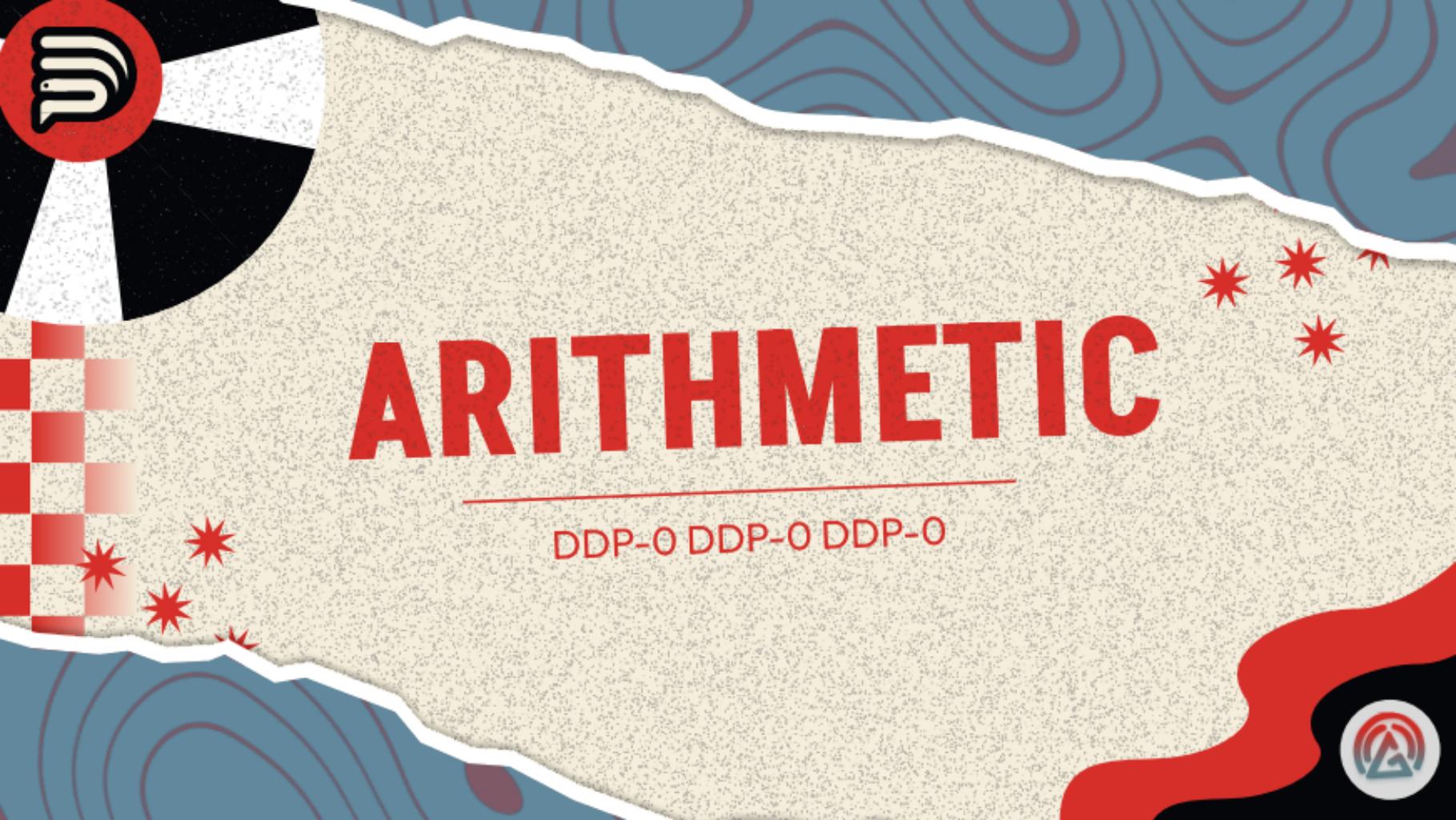
\*Jangan lupa ubah suatu input menjadi numeric type sebelum dihitung.

# Try it Out!!!



Nama: Nelvin Tan  
NPM: 310301051509  
Jalur Masuk: SNBT  
Umur: 19  
Tinggi: 156.76

Mari mencoba untuk membuat program **Input dan output** sederhana untuk memperkenalkan diri kalian sendiri! Serta diskusikan dengan mentormu tentang **data type** dari setiap variabel.



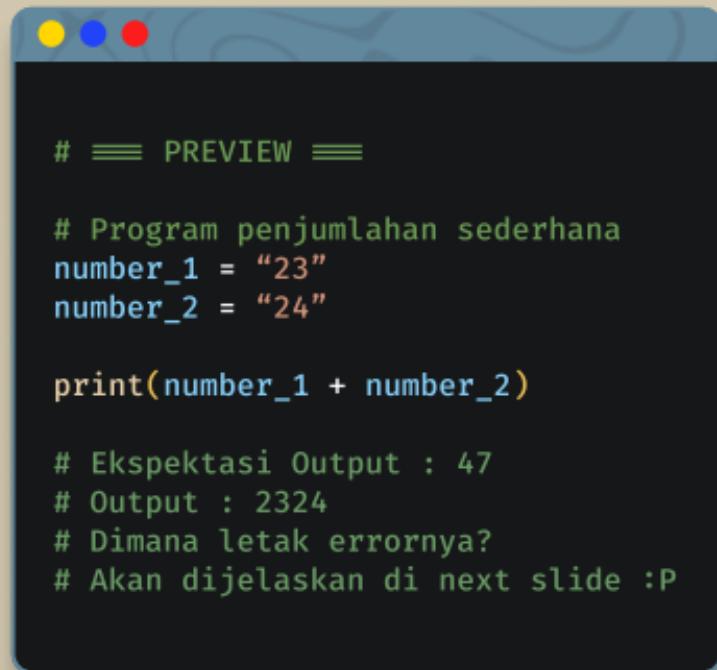
# ARITHMETIC

---

DDP-O DDP-O DDP-O



# Basic Arithmetic Operations



The image shows a windowed terminal interface with a blue header bar featuring three colored dots (yellow, blue, red). The main area is black and contains the following text:

```
# === PREVIEW ===

# Program penjumlahan sederhana
number_1 = "23"
number_2 = "24"

print(number_1 + number_2)

# Ekspektasi Output : 47
# Output : 2324
# Dimana letak errornya?
# Akan dijelaskan di next slide :P
```

# Arithmetic Operators

Untuk membuat suatu program, kita sering kali membutuhkan perhitungan matematika di dalam kode kita.

Python mengikuti aturan prioritas operasi matematika yang dikenal sebagai **PEMDAS**, yaitu:

- Parentheses
- Exponents
- Multiplication dan Division
- Addition dan Subtraction

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	$x / y$
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

# Arithmetic Operation

Di Python, operasi aritmatika dengan tipe data numerik itu seru dan mudah, mirip seperti bermain matematika sehari-hari!

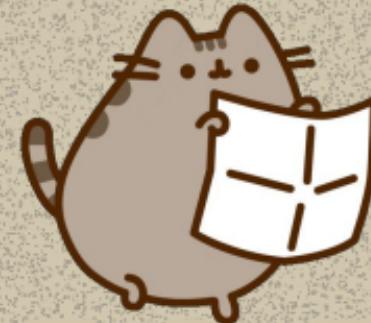
```
integer_var = 26
float_var = 7.10

print(integer_var + integer_var)
print(integer_var * float_var * 10 % 24)
print((integer_var - float_var) // 2)
```

## Output

```
52
22.0
9.0
```

$$1+1 = 3$$



# Arithmetic Operation

Untuk tipe data **non-numerik** seperti string, ada trik-trik seru! Kamu bisa menggabungkan dua string dengan mudah, seperti contoh berikut

```
integer_var = 5
string_var_1 = "CEES"
string_var_2 = "UWIW"

print(string_var_1 + string_var_2 + "DUWA EMPAT")
print(integer_var * string_var_1)
```

## Output

```
CEEW UWIW DUWA EMPAT
CEES CEES CEES CEES CEES
```

**Penjumlahan** bisa menghasilkan gabungan antara string. Kalau **perkalian** dengan bilangan bulat? Siap-siap aja, karena bisa bikin string terulang-ulang sebanyak yang kamu mau!

# Assignment Operation



Operator	Name	Example	Equivalent
<code>+=</code>	Addition assignment	<code>i += 8</code>	<code>i = i + 8</code>
<code>-=</code>	Subtraction assignment	<code>i -= 8</code>	<code>i = i - 8</code>
<code>*=</code>	Multiplication assignment	<code>i *= 8</code>	<code>i = i * 8</code>
<code>/=</code>	Float division assignment	<code>i /= 8</code>	<code>i = i / 8</code>
<code>//=</code>	Integer division assignment	<code>i //= 8</code>	<code>i = i // 8</code>
<code>%=</code>	Remainder assignment	<code>i %= 8</code>	<code>i = i % 8</code>
<code>**=</code>	Exponent assignment	<code>i **= 8</code>	<code>i = i ** 8</code>

# Math Library

$A = \pi r^2$

$C = 2\pi r$

$V = \frac{1}{3} \pi r^2 h$

$V = \pi r^2 h$

$\begin{array}{|c|c|c|} \hline & 30^\circ & 45^\circ & 60^\circ \\ \hline \sin & \frac{1}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{3}}{2} \\ \hline \cos & \frac{\sqrt{3}}{2} & \frac{\sqrt{2}}{2} & \frac{1}{2} \\ \hline \tan & \frac{\sqrt{3}}{3} & 1 & \sqrt{3} \\ \hline \end{array}$

$\begin{array}{l} \text{sin } x = -\cos x + C \\ \frac{dx}{\cos^2 x} = \operatorname{tg} x + C \\ \int \operatorname{tg} x dx = -\ln |\cos x| + C \end{array}$

$\begin{array}{l} \frac{dx}{\sin x} = \ln \left| \frac{x}{2} \right| + C \\ \int \frac{dx}{a^2 + x^2} = \frac{1}{a} \operatorname{arctg} \frac{x}{a} + C \\ \int \frac{dx}{x^2 - a^2} = \frac{1}{a} \ln \left| \frac{x-a}{x+a} \right| + C \end{array}$

$\tan(\theta)$

$\alpha x^3 + \beta x + c = 0$

$a(x^3 + \frac{b}{a}x + \frac{c}{a}) = 0$

$x^3 + 2\frac{b}{2a}x + (\frac{b}{2a})^2 = (\frac{b}{2a})^2$

$(x + \frac{b}{2a})^3 - \frac{b^3 - 4ac}{2a} = 0$

# Library

Tadi kita sudah belajar mengenai basic arithmetic. Namun, bagaimana jika kita ingin melakukan perhitungan yang lebih kompleks? **math library** adalah solusinya

**Library** adalah kumpulan kode yang sudah dibuat sebelumnya dan dapat digunakan kembali jika dibutuhkan.

Python memiliki **Standard Library** yang berisi banyak library yang bermanfaat. Contohnya adalah **Math**, **Tkinter**, dan **Turtle**.

```
# CARA MENGGUNAKAN LIBRARY
# import [nama Library]

import math
import turtle
import tkinter

# Bisa juga dengan
import math as m
import tkinter as tk
```

# Attribute & Constant

Math Library berisi beberapa angka **konstan** dalam matematika, seperti pi.

CONSTANT	DESCRIPTION
math.pi	Returns PI (3.1415...)
math.e	Returns Euler's number (2.7182...)
math.tau	Returns tau (6.2831...)
math.inf	Returns a floating-point positive infinity
math.nan	Returns a NaN value

# Method

Berisi fungsi-fungsi untuk melakukan perhitungan matematika, seperti untuk menghitung square root, trigonometri, dan lain-lain

Method	Deskripsi
math.ceil(x)	Menghitung nilai ceil dari x *ceil adalah pembulatan ke atas  x — Sebuah bilangan integer/float.
math.floor(x)	Menghitung nilai floor dari x *floor adalah pembulatan ke bawah  x — Sebuah bilangan integer/float.

# Method

Method	Deskripsi
<code>math.pow(x, y)</code>	Menghitung $x$ pangkat $y$  $x$ — Sebuah bilangan integer/float. $y$ — Sebuah bilangan integer/float.
<code>math.sqrt(x)</code>	Menghitung akar kuadrat dari $x$  $x$ — Sebuah bilangan integer/float.

# Method

Method	Deskripsi
<code>math.sin(x)</code>	Menghitung nilai sin dari x radian  $x$ — Sebuah bilangan integer/float
<code>math.cos(x)</code>	Menghitung nilai cos dari x radian  $x$ — Sebuah bilangan integer/float.

# Implementasi

Daripada pusing kebanyakan teori mending langsung implementasi gasi :D

Berikut contoh implementasi program menghitung luas lingkaran dengan bantuan math library

```
import math  
PI = math.pi  
radius = 10  
  
luas_lingkaran = PI * math.pow(radius, 2)  
print(luas_lingkaran)
```

Output

314.1592653589793

# Eksplorasi Mandiri

Masih banyak method math dan contoh-contoh library yang belum bisa dimasukkan di slide.  
Oleh karena itu, silakan lakukan **eksplorasi secara mandiri :3**

More about Math Library :

<https://docs.python.org/3/library/math.html>

More about Standard Library :

<https://docs.python.org/3/library/index.html>

Selain dari yang diberikan, ada juga library diluar Standard Library yang sangat berguna. Seperti **numpy** untuk melakukan komputasi pada array, **pandas** untuk analisis data, **numerize** untuk komputasi **angka yang besar**, dan lain-lain.

# Try it Out!!!

```
Enter the sphere's radius: 7  
The volume of the sphere is 1436.76
```

Coba buat **kalkulator sederhana** untuk menghitung **volume bola pejal**. Kode harus meminta **radius** dari user berupa tipe data float.



# COMMENT

---

DDP-O DDP-O DDP-O

# Comment

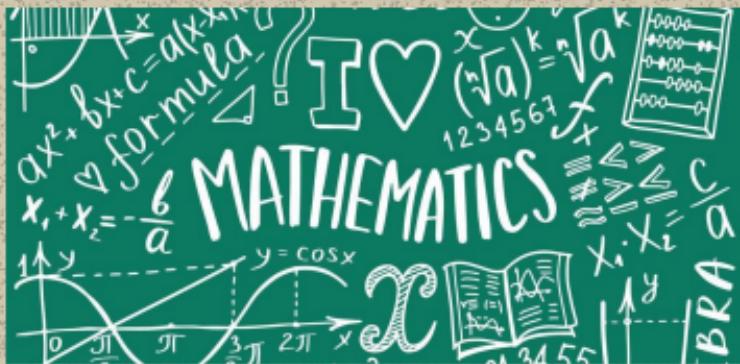
**Comment** adalah catatan kecil yang kita tulis di kode untuk menjelaskan apa yang sedang kita lakukan. Comment dalam Python ditulis dengan mengawali kalimat dengan tanda **#**.

Comment digunakan agar kita dapat memberitahu diri sendiri atau orang lain apa yang sedang terjadi di bagian tertentu dari program.

```
# Ini adalah komentar, tidak  
akan dieksekusi oleh Python  
  
print("Hello, world!")
```

# Don't Forget!

Jangan lupa eksplorasi math library di dokumentasi Python untuk mengetahui method dan konstanta dalam math library. <https://docs.python.org/3/library/math.html>



# Done for the Week

Yeay! Kita sudah menguasai dasar-dasar pemrograman di Python seperti **variabel, input, output, dan math library**.

Tapi, program kita masih sederhana banget, belum bisa menangani kasus-kasus rumit yang butuh keputusan atau pengulangan berulang. Di Week 1 nanti, kita bakal belajar **kondisional dan loop** untuk bikin program kita makin keren dan berguna!



# Mini Quiz (1)

a. Apakah **hanya** ada **satu cara** untuk menaklukkan setiap level di mini game Minecraft di code.org? Atau mungkin ada banyak cara kreatif untuk mencapai kemenangan? Yuk, kita bahas dan temukan kesimpulannya!

b. Sekarang, cek nama-nama di tabel samping. Menurutmu, apakah semuanya **sah** untuk dijadikan **nama variabel**? Let's find out!

Nama Variabel	Valid/Invalid
__var__	
2var	
var@	
!	

# Mini Quiz (2)

Isilah tabel dengan **datatype** yang sesuai dengan masing-masing variabel

Variabel	Datatype
a = 123	
b = "123"	
c = False	
d = "123" + "456"	
e = "True"	
f = None	
g = [1, 2, 3]	
h = 123.45	

# Mini Quiz (3)

Apa sih yang akan terjadi jika kita jalankan operasi aritmatika di tabel ini di Python? Apakah kita akan melihat hasilnya atau malah mengalami error? Ayo, kita cek sama-sama!

## Operasi

$0.7 + 0.6$

$9928765403 + 4.0$

`"CEES UWIW" + 2 ** 0 ** 2 ** 4`



**TERIMA  
KASIH**

---