



INFOTECH

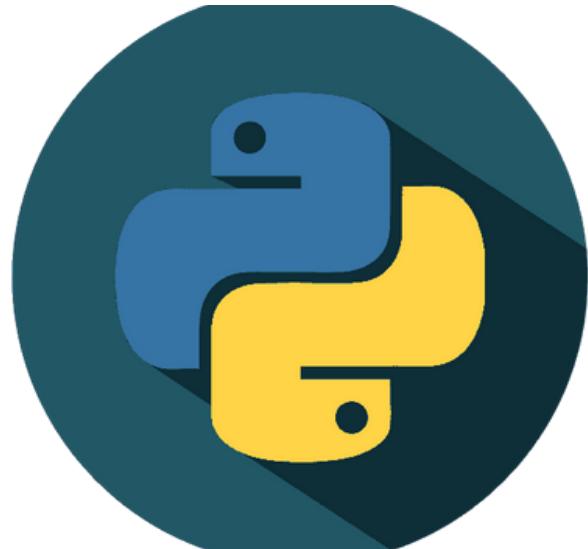


# Workshop Python

Ulfah Nur Oktaviana



# Kenapa Harus Python?



Mudah dimengerti

Dynamic

Multi Platform

Hype

- Web Development

- Data Analyst

- Mobile Apps

- Artificial Intelligence

- Desktop Apps

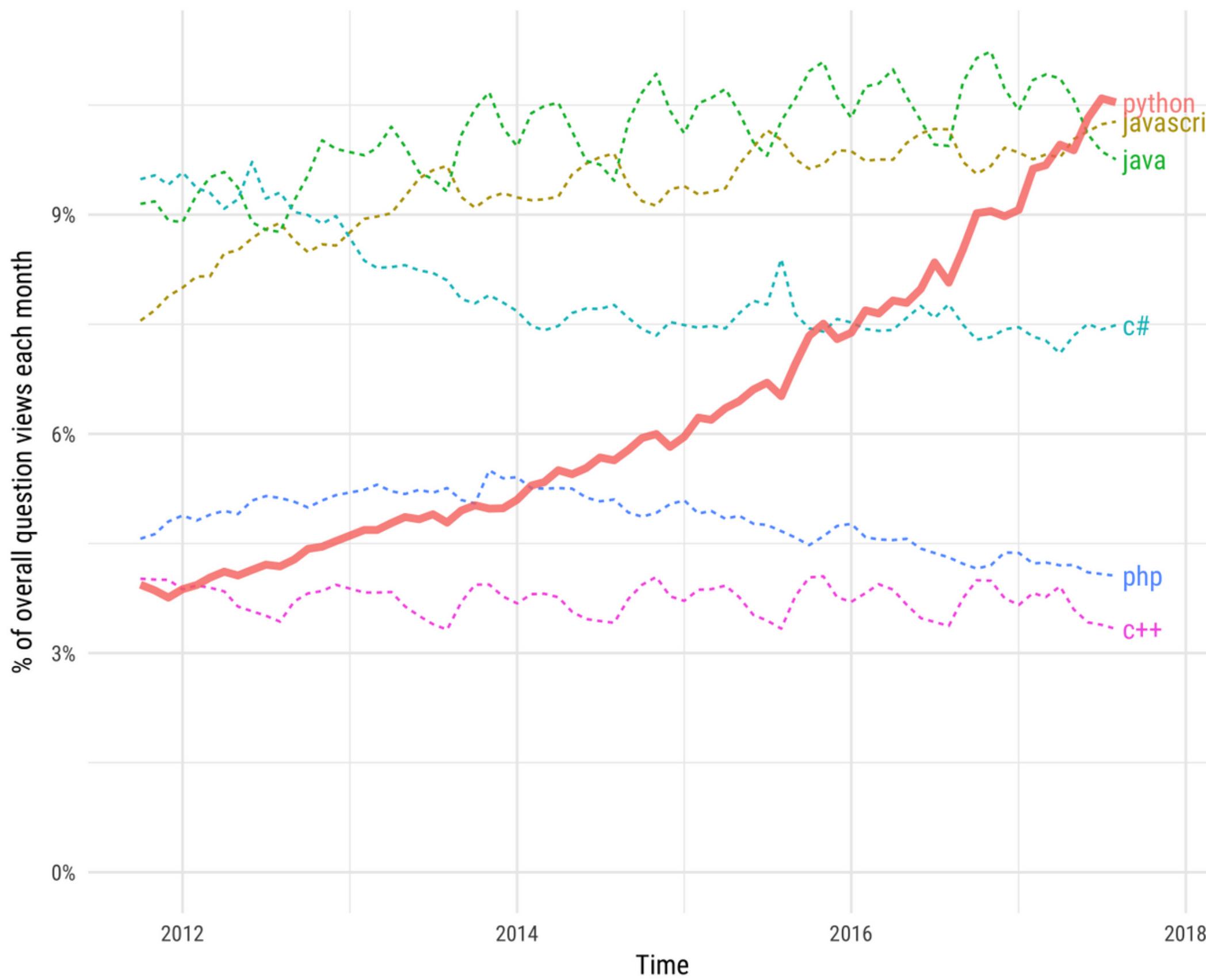
- Machine Learning

- Data Science

- Robotics

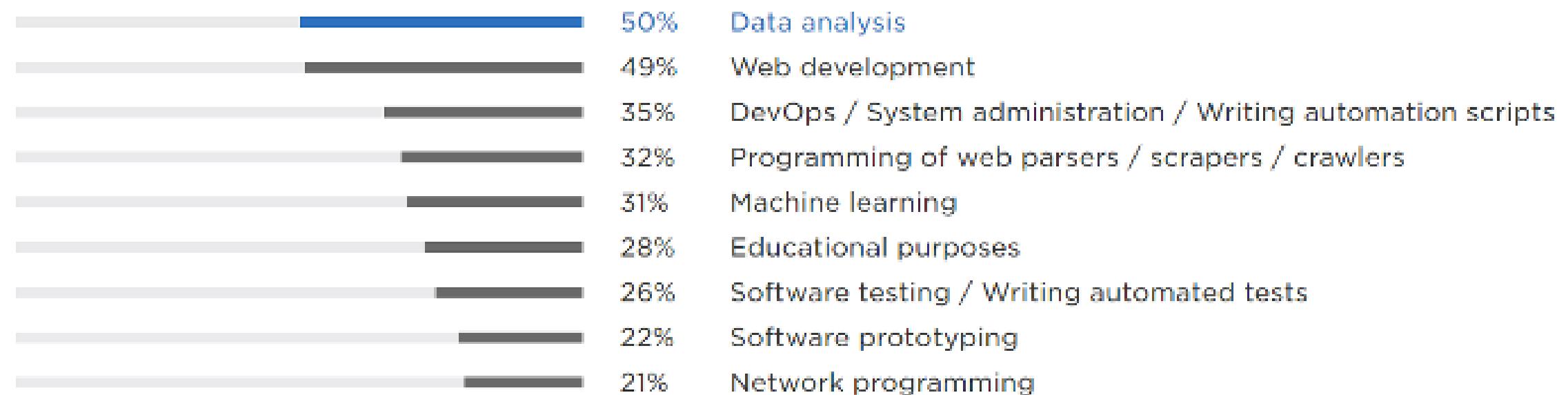
## Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries

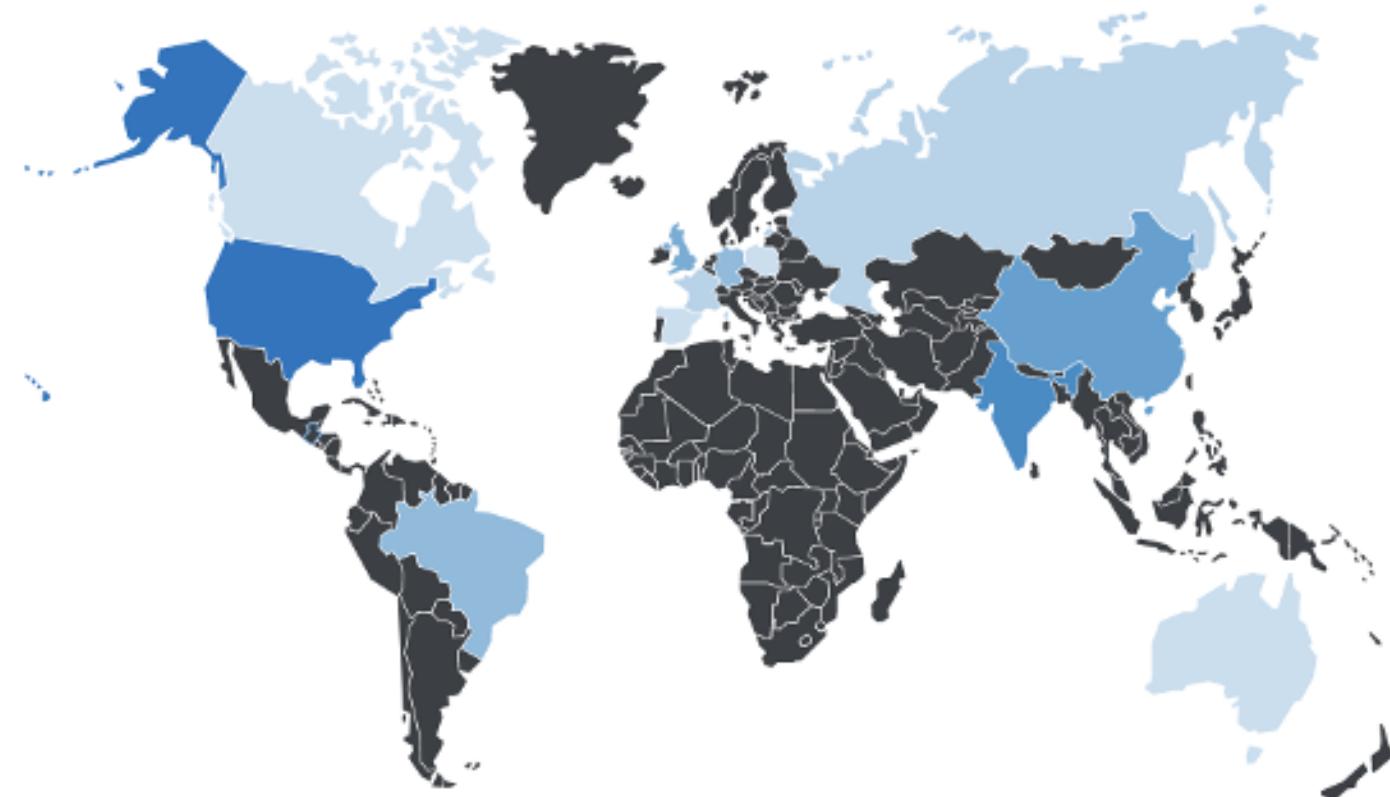


Ini adalah 6 dari 10 tag Stack Overflow yang paling banyak dikunjungi di negara-negara berpenghasilan tinggi.

<https://stackoverflow.blog/2017/09/06/incredible-growth-python/>



## Global Community



18% United States  
13% India  
7% China  
6% United Kingdom  
5% Germany  
4% Brazil  
3% France  
3% Russia  
2% Poland  
2% Canada  
2% Australia  
2% Spain  
33% Others

Python Developers Survey generates a wide range of data about the current use and trajectory of the Python programming language.

**"collection period, it received more than 10,000 responses. from 150+ countries included"**



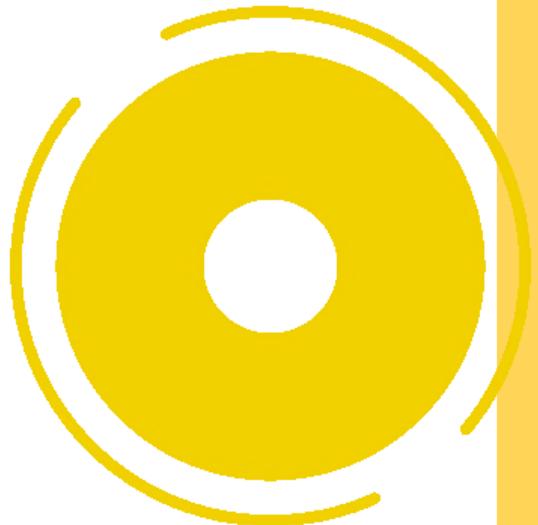
# Basic Python Syntax





## 1. Print With Python and Java

```
public class PythonandJava {  
    public static void main(String[] args)  
    {  
        System.out.println("Python and Java!");  
    }  
}
```



```
print("Python and Java !")
```





## 2. Python As a Calculator

### Arithmetic operators

Python can operate with numbers using the usual mathematical operators, and some special operators, too. These are all of them (we'll explore the last two in later videos).

- $a + b$  = Adds a and b
- $a - b$  = Subtracts b from a
- $a * b$  = Multiplies a and b
- $a / b$  = Divides a by b
- $a ** b$  = Elevates a to the power of b. For non integer values of b, this becomes a root (i.e.  $a**(1/2)$  is the square root of a)
- $a // b$  = The integer part of the integer division of a by b
- $a \% b$  = The remainder part of the integer division of a by b



## 2. Variables, Conversion, and Function

**VARIABLE**, Nama yang diberikan untuk nilai tertentu di sebuah program.

**ASSIGNMENT**, Proses menyimpan nilai didalam variabel.

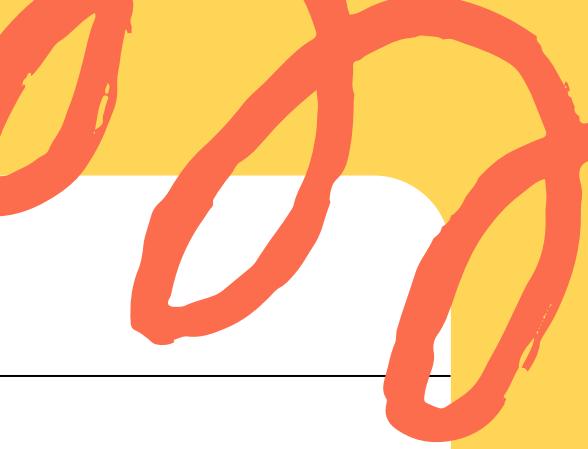
### Variable Name Restrictions

- 1. Jangan menggunakan **KEYWORD** atau **FUNCTION** yang sudah ada di python
- 2. Jangan menggunakan spasi
- 3. Diawali dengan huruf atau underscore (\_)
- 4. Hanya boleh terdiri atas huruf, angka, dan underscore (\_)

Example :

- **i\_am\_a\_variable** IS VALID
- **i\_am\_a\_variable2** IS VALID
- **1\_is\_a\_variable** INVALID
- **apples\_&\_orange** IS INVALID





**CONVENTION**, konversi dari satu tipe data ke tipe data lainnya dengan memanggil sebuah fungsi yang relevan dengan tipe data yang dikonversi.

Contoh : str() , merupakan function yang digunakan untuk melakukan konversi ke tipe data String.

```
print(type(str(luas_segitiga)))
```

```
def greeting(name):  
    print("Hello Good Morning, "+ name)  
  
greeting("Lorenzo")  
greeting("Diana")
```

**FUNCTION** , didefinisikan dengan keyword def, nama fungsi, parameter (disebut juga sebagai argumen). sebuah fungsi diapit oleh tanda kurung.





## 3. Conditionals



### Comparison Operator

- equality operator (==)
- not equals operator (!=)
- greater than (>)
- lesser than (<)
- and
- or
- not



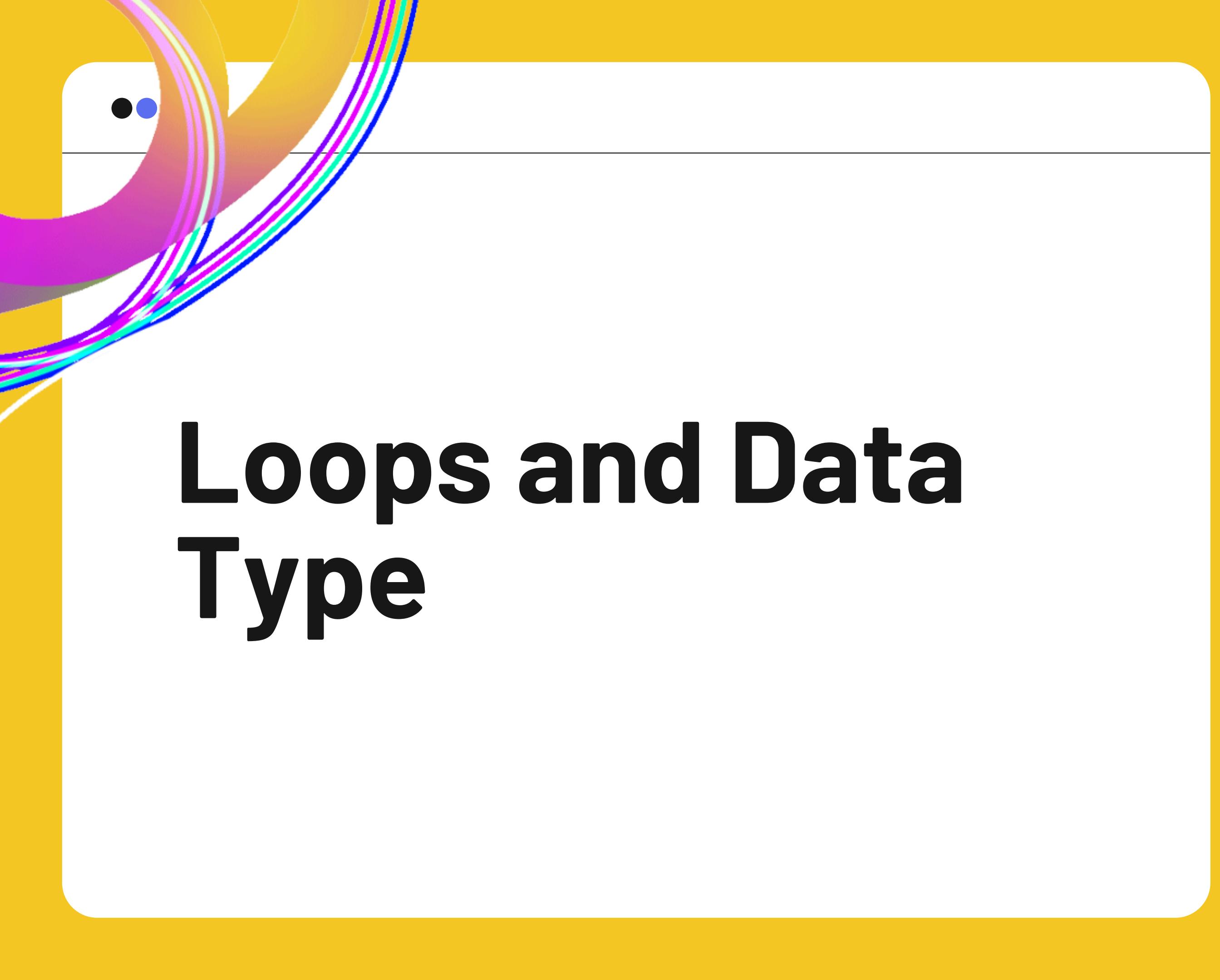
### Brancing

- IF statement
- ELSE statement
- ELIF statement



# Let's deeper dive in Colabolatory

[HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/1LOZNEYCOMDHTFJOLICBSWA5SPQIVTIRQ](https://colab.research.google.com/drive/1LozNEYCOMDHTfJolicBSWA5SPQIVTIRQ)



# Loops and Data Type





## 1. Loops (Perulangan)

### While Loops

**"Instruct your computer to continuously execute your code based on the value of a condition"**

### For Loops

**"for loop not only works in number loops, but also loops over lists"**

### Recursion

**"The repeated application of the same procedure to a smaller problem. In programming, recursion is a way of doing a repetitive task by having a function call itself."**





## 2. Data Type (STRING)

### Define STRING

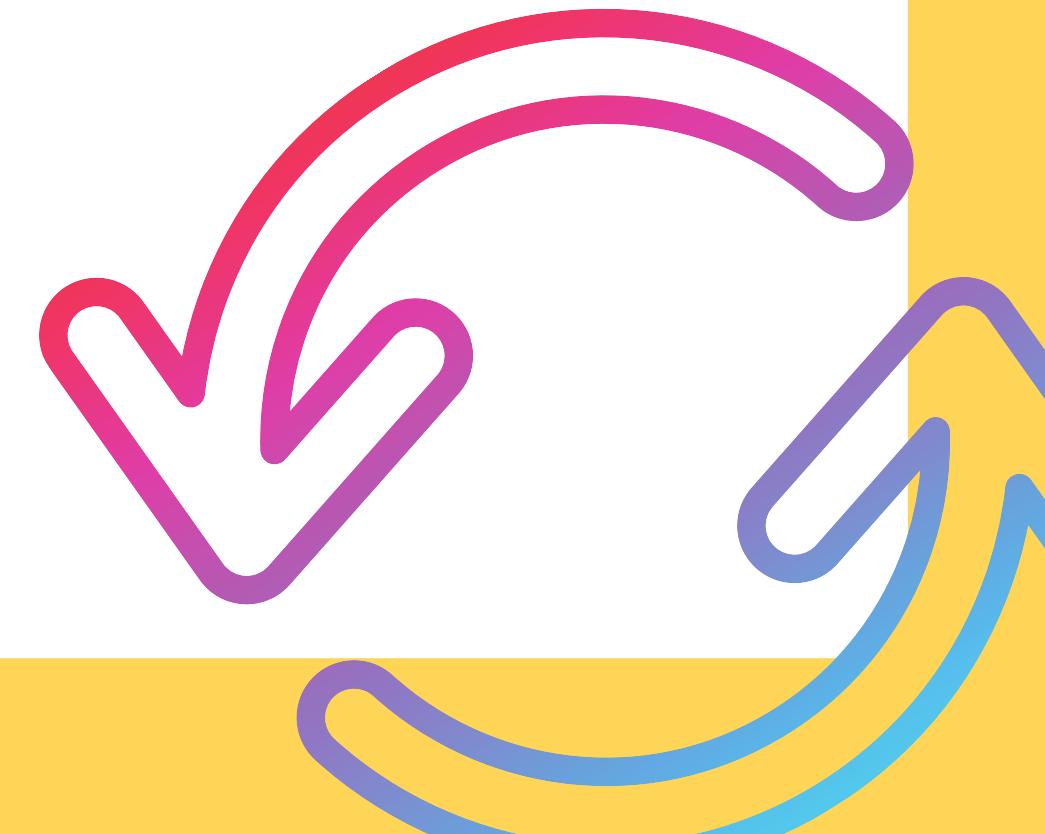
```
name = "jeje junior"
```

```
color = 'white'
```

```
place = "Harvard"
```



**"String in Python  
are IMMUTABLE  
(tidak dapat  
berubah)"**



## Common String Operation and String Method

### Slice

**"The Portion of a String that can contain more than one character; also sometimes called a substring "**



- `string[i]` access the character at index i of the string
- `string[i:j]` Access the substring starting at index i, ending at index j-1.

### Method

**"A function associate with a specific class "**



- `string.lower()` / `string.upper()`
- `string.lstrip()` / `string.rstrip()` / `string.strip()`
- `string.count(substring)`
- `string.isnumeric()`
- `string.isalpha()`
- `string.split()`
- `string.replace(old, new)`
- `delimiter.join(list of string)`

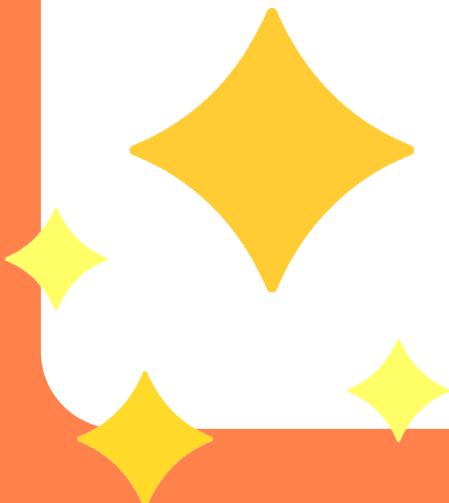


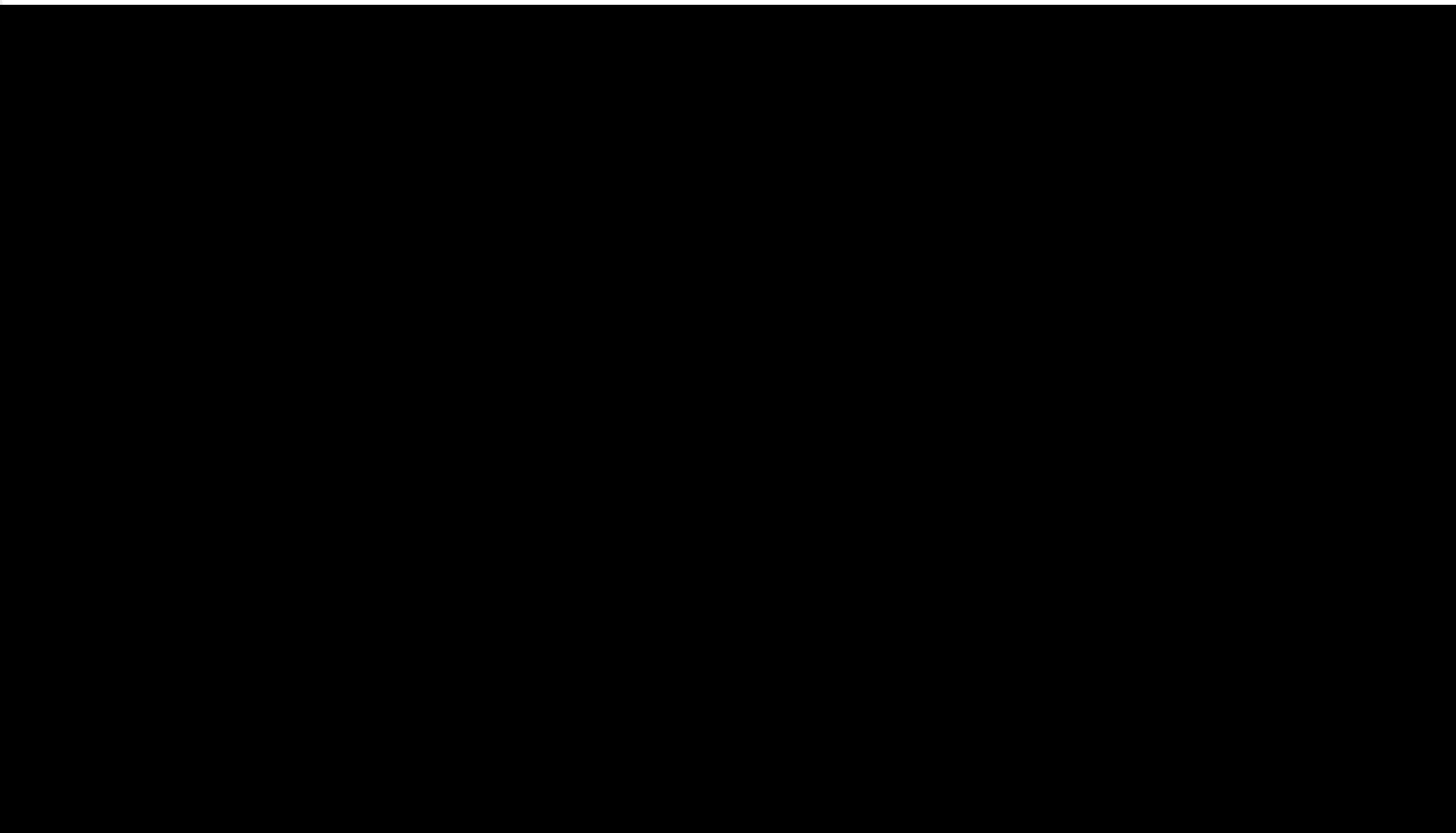
# QUIZ!!!!



Buatlah sebuah fungsi yang dapat mengganti domain email dengan menggunakan konsep yang sudah kita terapkan sebelumnya !!!!!!!!

```
replace_domain("ulfah@gmail.com", "gmail.com", "yahoo.com")  
'ulfah@yahoo.com'
```







## 2. Data Type (LIST, Tuple)

### LIST

**"Sequences of elements of any type, and are mutable (dapat berubah)"**



### TUPLE

**"Sequences of elements of any type, and are immutable (tidak dapat berubah)"**



## List and Tuple Operations



- `len(sequences)` return the length of the sequence
- `sequence[i]` access element at index i of the sequences, starting at 0
- `sequence[i:j]` access a slice starting at index i , ending at index j-1
- `for index, element in enumerate(sequence)` iterate over both the indexes and the elements in the sequences at the same time

## List Specific Operations and Methods



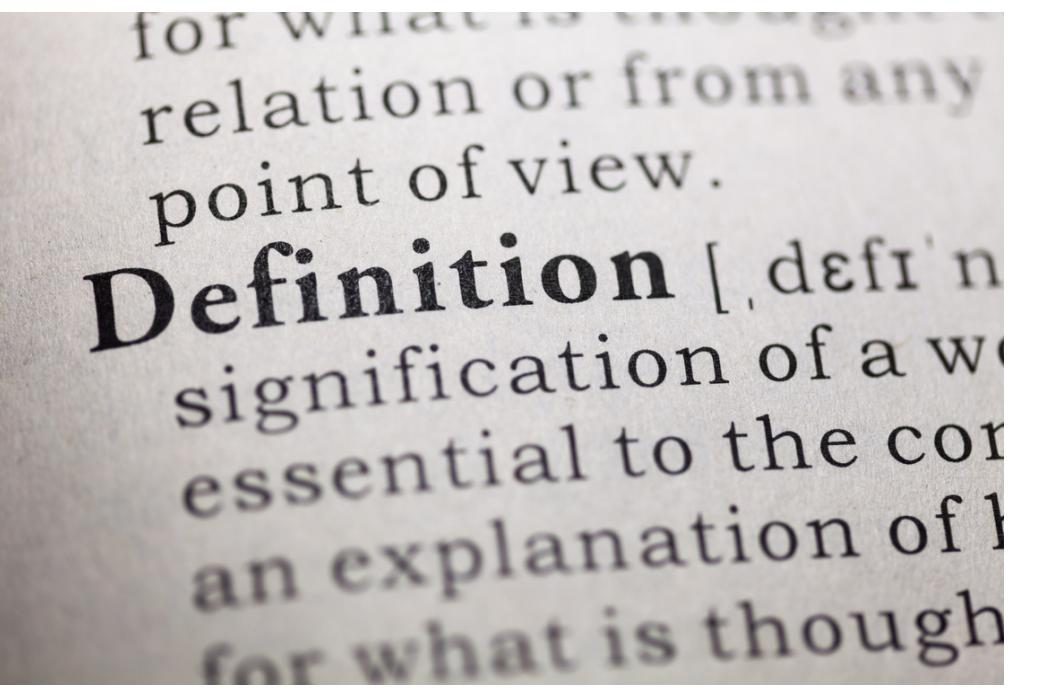
- `list[i] = x` , Replace the element at index i with x
- `list.append(x)` Insert x at the end of the list
- `list.insert(i, x)` Insert X at index i
- `list.pop(i)` return the element of index i, also removing it from the list
- `list.remove(x)` Remove the first occurrence of X on the list



## 2. Data Type (Dictionary)

### DICTIONARY

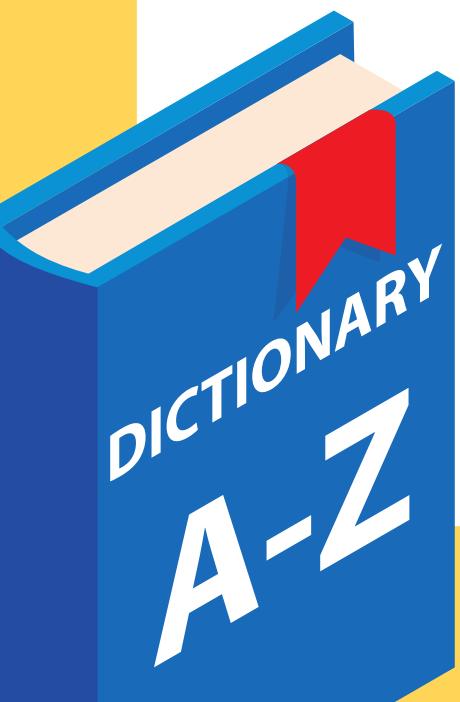
"The data inside dictionaries take the form of pairs keys and values, and dictionary is **MUTABLE** data type"



**When to use dictionary ?**

"You want to use ductionaries when you plan on searching for a specific element."

```
ip_address = ['192.168.1.1', '127.0.0.1', '8.8.8.8']
host_address = {"router" : "192.168.1.1", "localhost" : '127.0.0.1', 'google' : '8.8.8.8'}
```





## OPERATION IN DICTIONARY



- len(dictionary) - Returns the number of items in the dictionary
- for key in dictionary - Iterates over each key in the dictionary
- for key, value in dictionary.items() - Iterates over each key,value pair in the dictionary
- dictionary[key] - Accesses the item with key key of the dictionary
- dictionary[key] = value - Sets the value associated with key
- del dictionary[key] - Removes the item with key key from the dictionary

## OPERATION IN DICTIONARY



- dict.keys() - Returns a sequence containing the keys in the dictionary
- dict.values() - Returns a sequence containing the values in the dictionary



# Let's deeper dive in Colabolatory

[HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/197JZTHCWK1I6JEBOFKZFRMWBCEC-KPJZ#SCROLLTO=TDFQ1RLNEHTS](https://colab.research.google.com/drive/197JZTHCWK1I6JEBOFKZFRMWBCEC-KPJZ#scrollTo=TDFQ1RLNEHTS)





# Object Oriented Programming



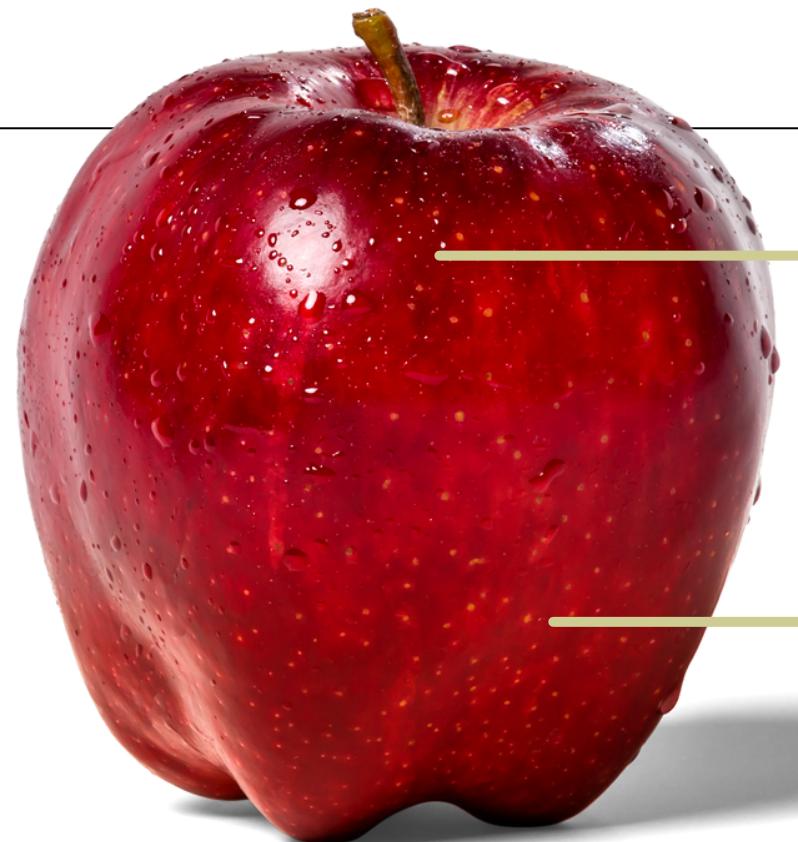
OBJECT ORIENTED PROGRAMMING



# What is OOP?

**"A FLEXIBLE, POWERFULL  
PARADIGM WHERE CLASSES  
REPRESENT AND DEFINE CONCEPT  
, WHILE OBJECT ARE INSTANCES OF  
CLASSES"**





Color



Flavor



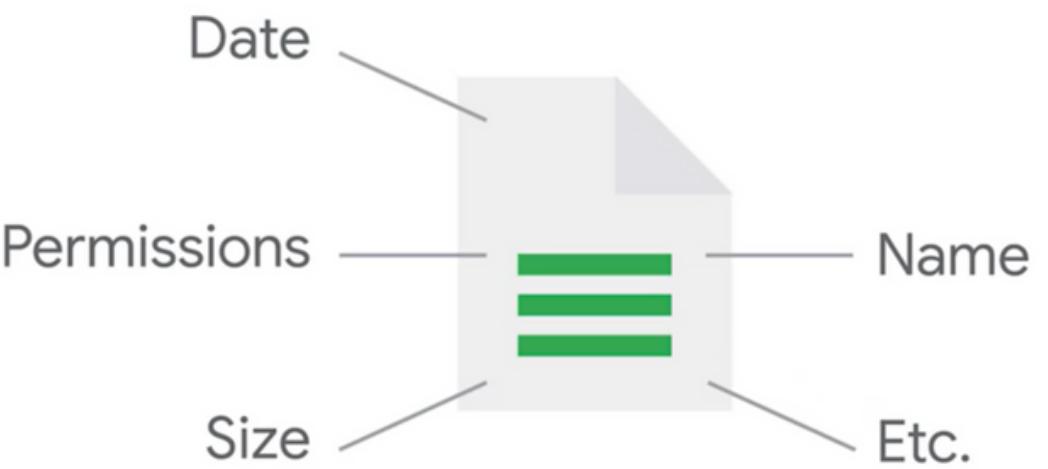


## Attributes

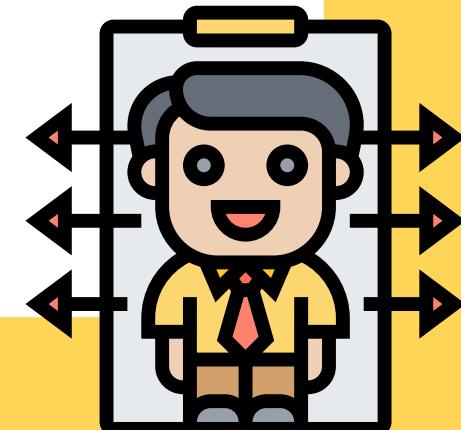
**"The attributes are the characteristics associated to a type"**

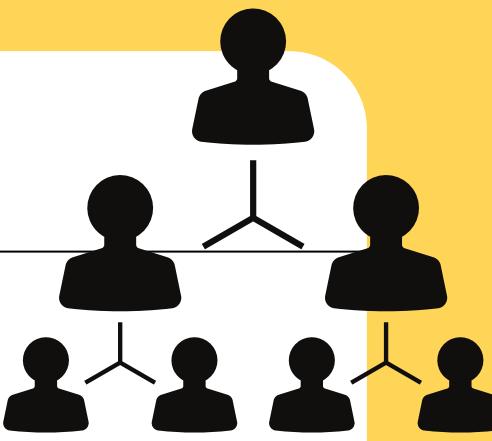
## Method

**"The methods are the functions associated to a type"**



**A D D**

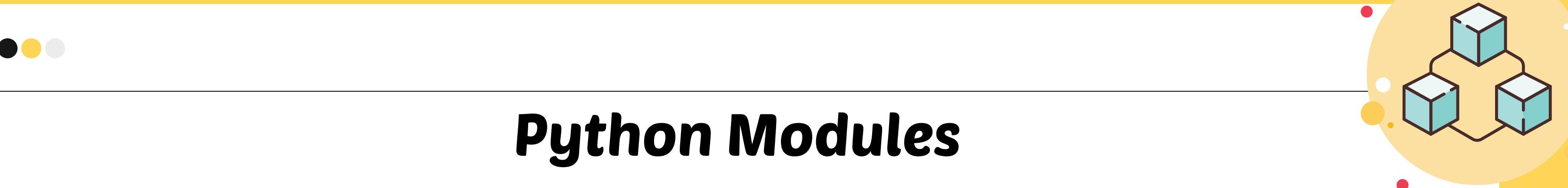




# Inheritance

```
class Fruits:  
    def __init__(self, color, flavor):  
        self.flavor = flavor  
        self.color = color  
  
class Apple(Fruits):  
    pass  
  
class Banana(Fruits):  
    pass  
  
manalagi = Apple("Red", "sweet")  
ambon = Banana("Yellow", "sweet")  
  
print(manalagi.color)  
Red
```





# Python Modules

```
import datetime

now = datetime.datetime.now() # module, class, method

type(now)

datetime.datetime

print(now)

2021-11-17 12:19:45.010378

now.year

2021
```

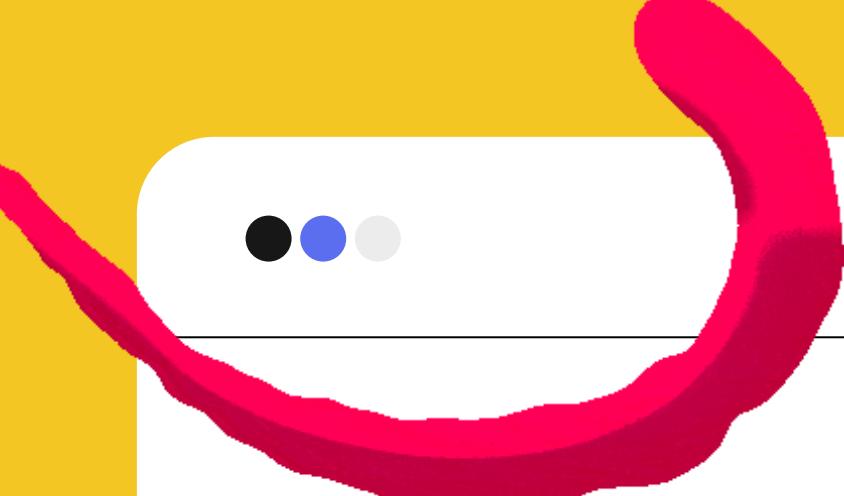


Documentation : <https://pypi.org/>

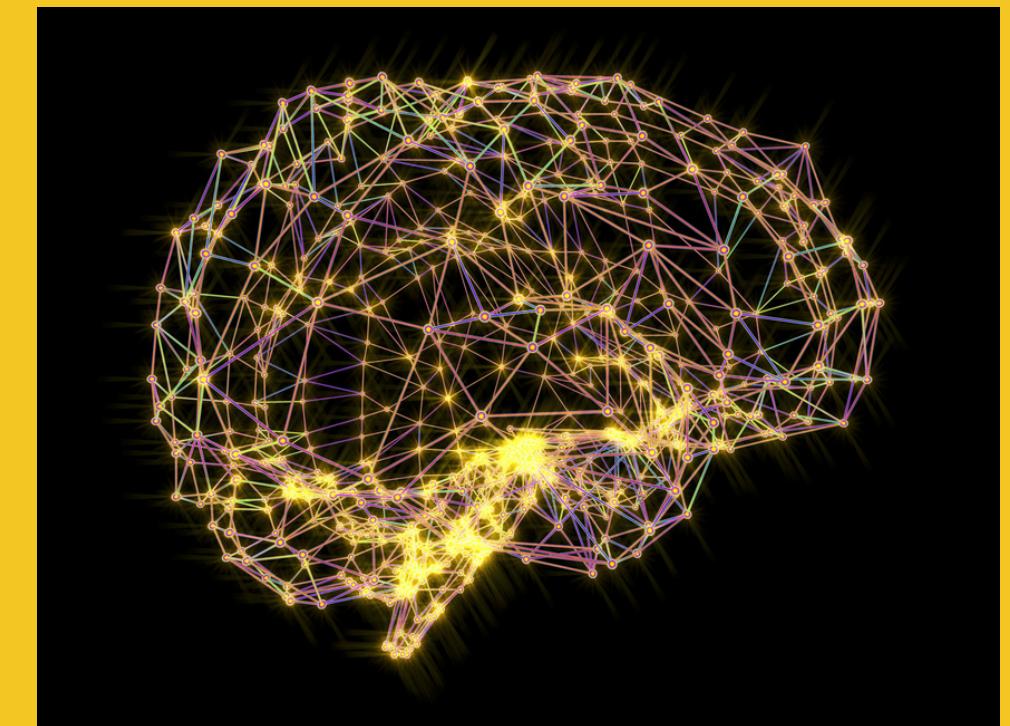
# Let's deeper dive in Colabolatory

[HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/1XUOLDOL2QFFOSUFZDMMDCATYIU4YF5FW#SCROLLTO=RDXHTZSR1EYF](https://colab.research.google.com/drive/1XUOLDOL2QFFOSUFZDMMDCATYIU4YF5FW#scrollTo=RDXHTZSR1EYF)





**Hello World ! TO  
Neural Network**

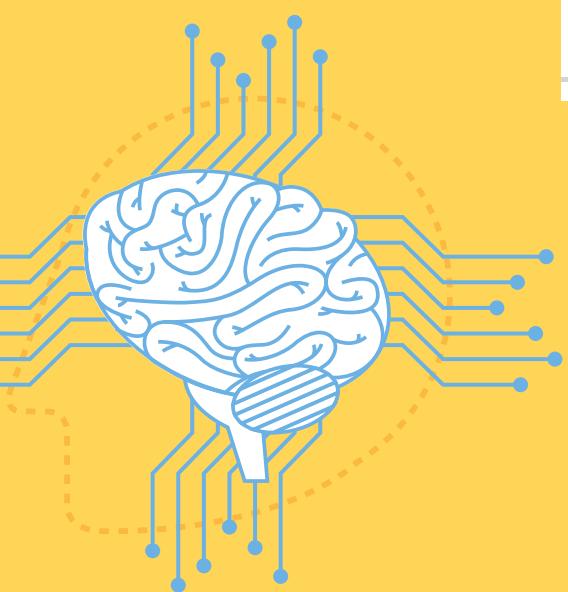




# Neural Network

x	y
-1	-3
0	-1
1	1
2	3
3	5
4	7

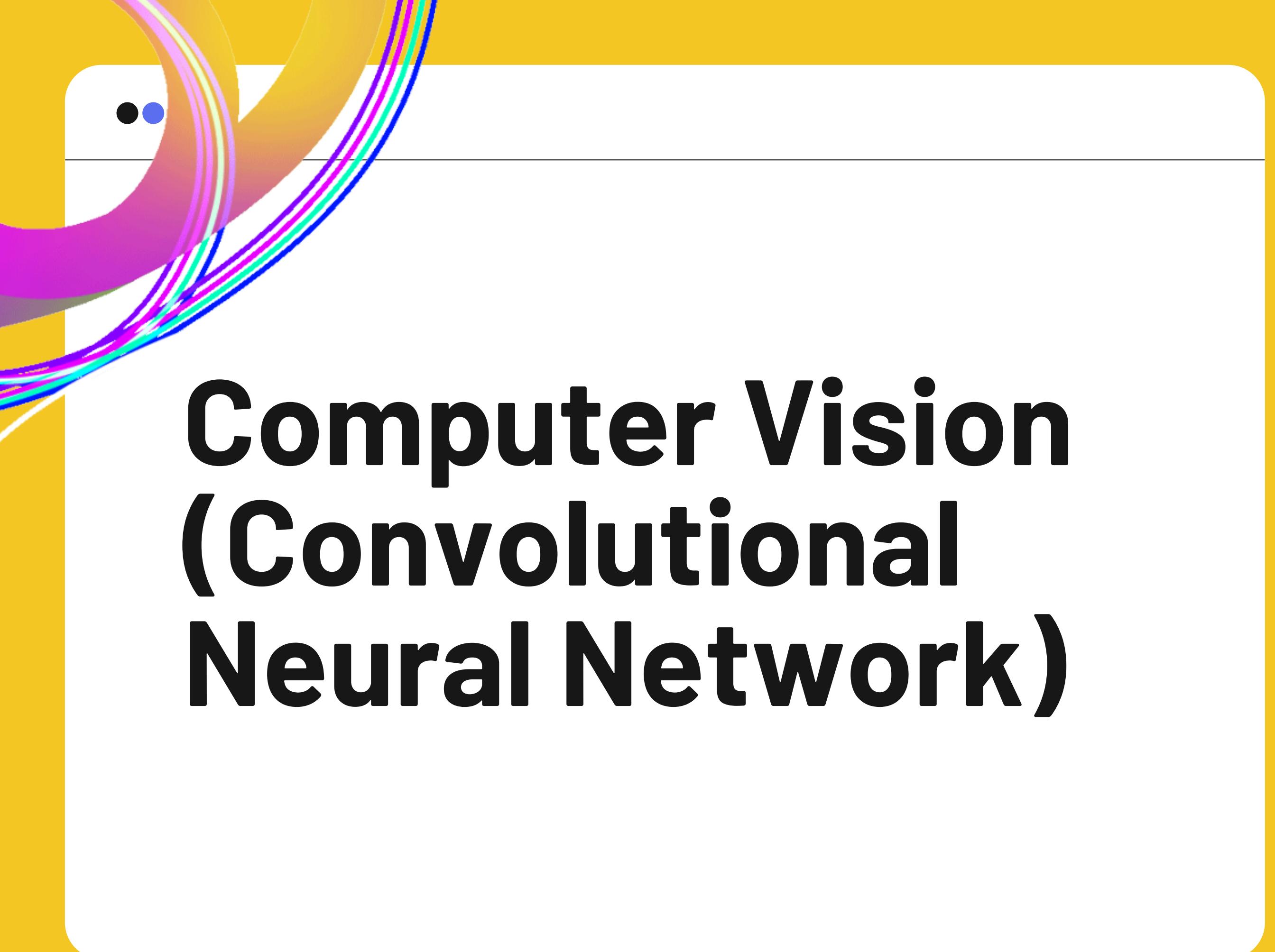
$$y = 2x - 1$$



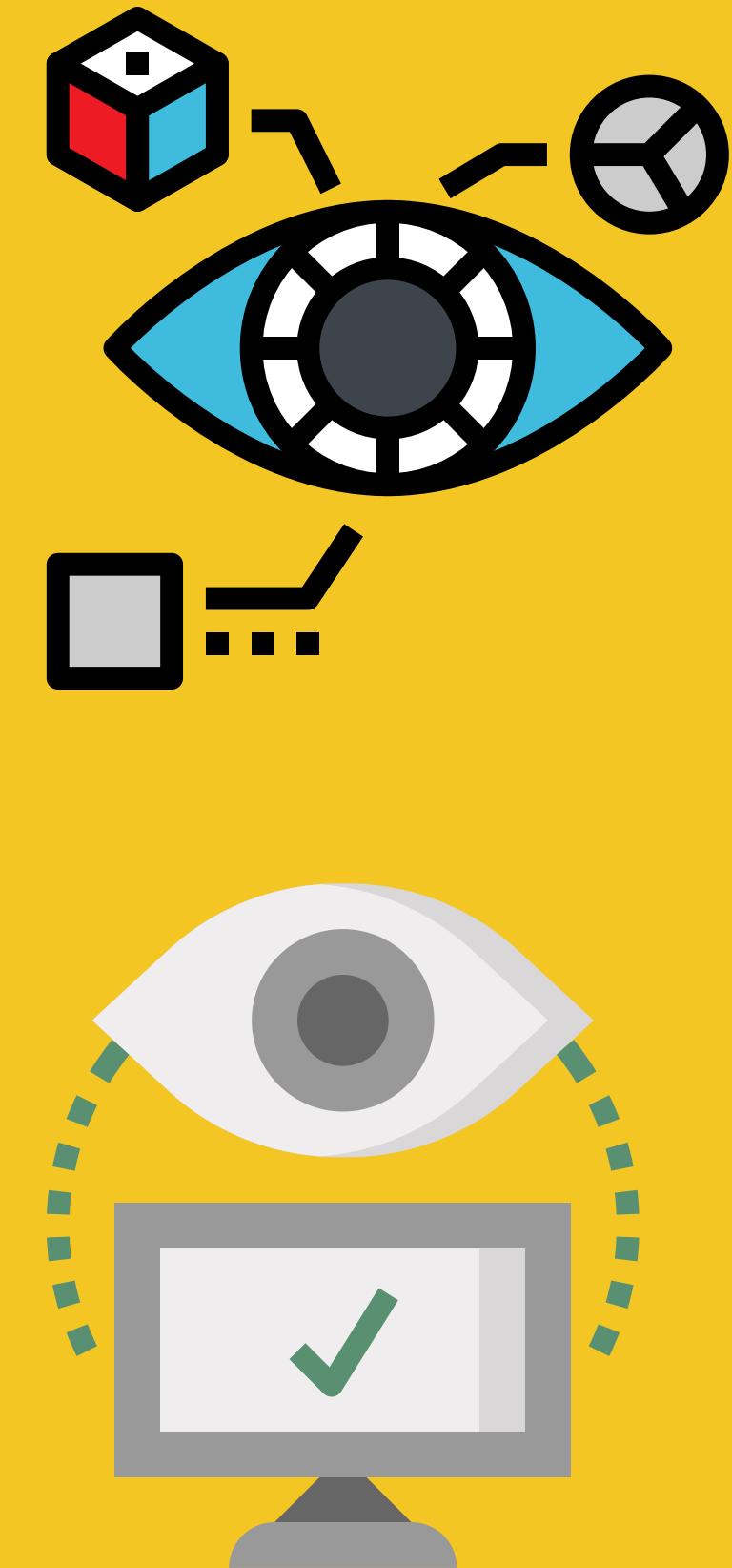
# Let's deeper dive in Colabolatory

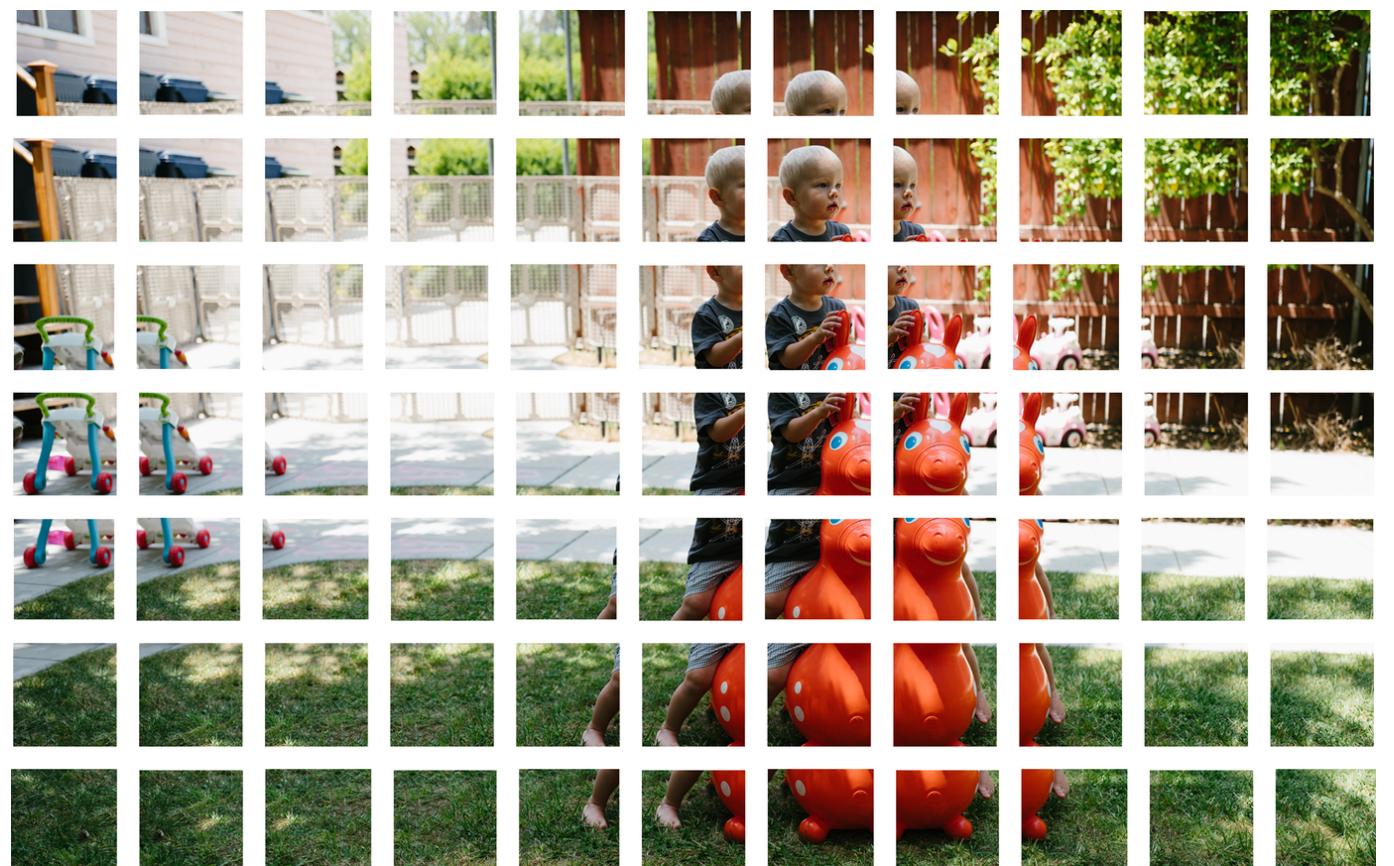
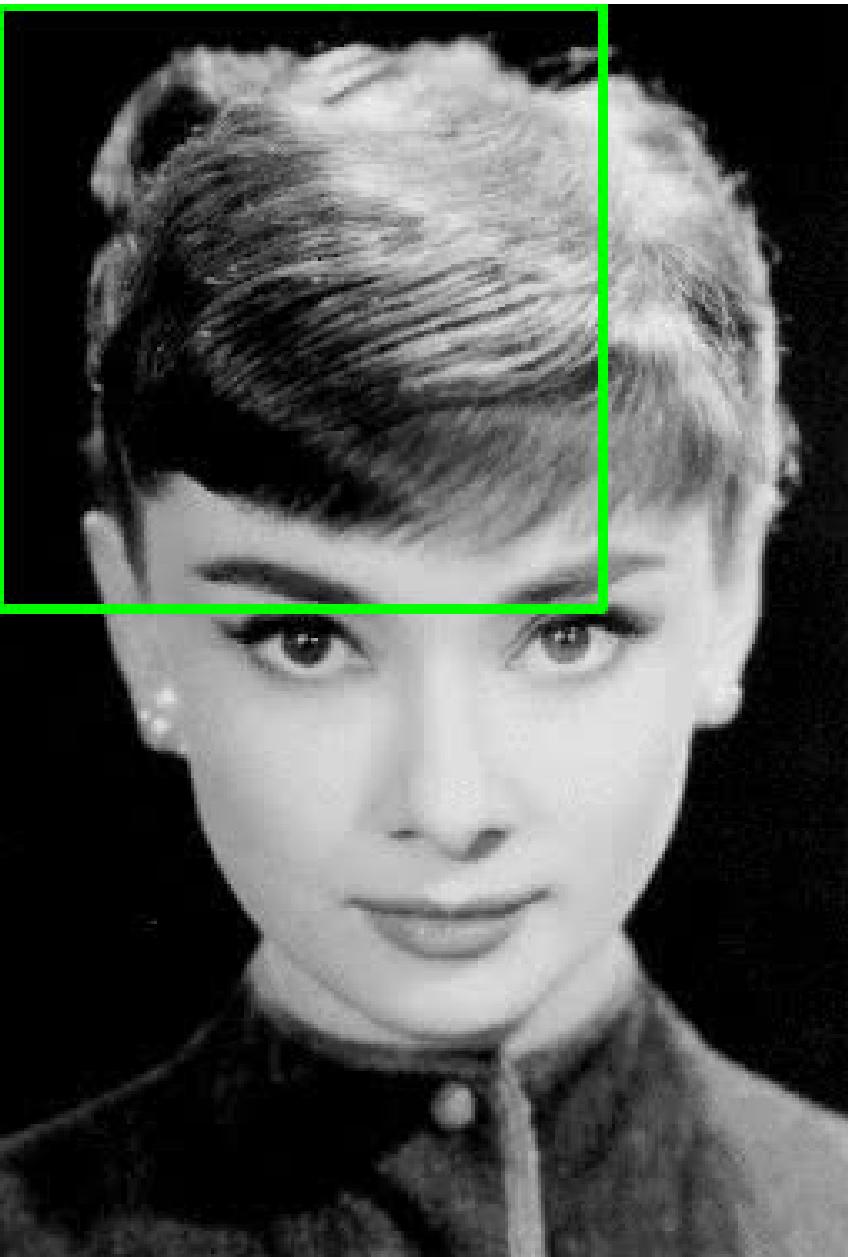
[HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/1ARFA5351RRSUQG8ZBNKF5IZBJOZBK7V#SCROLLTO=FA93WUY1ZZ](https://colab.research.google.com/drive/1ARFA5351RRSUQG8ZBNKF5IZBJOZBK7V#scrollTo=FA93WUY1ZZ)  
WF





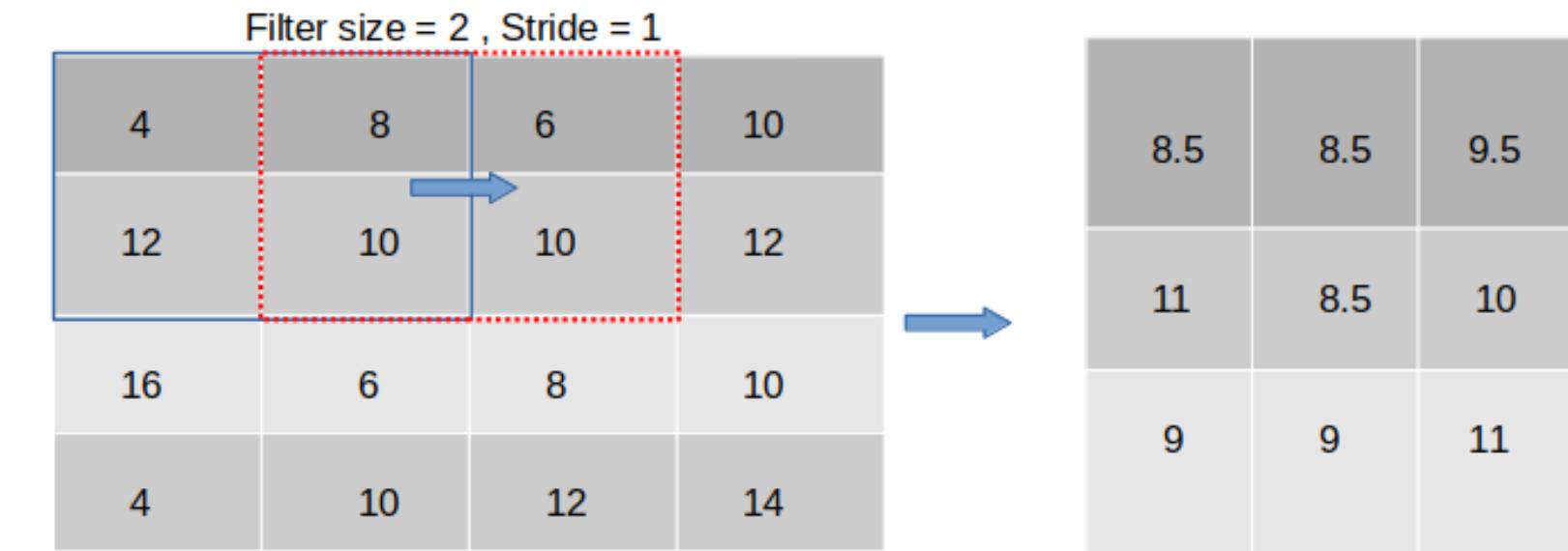
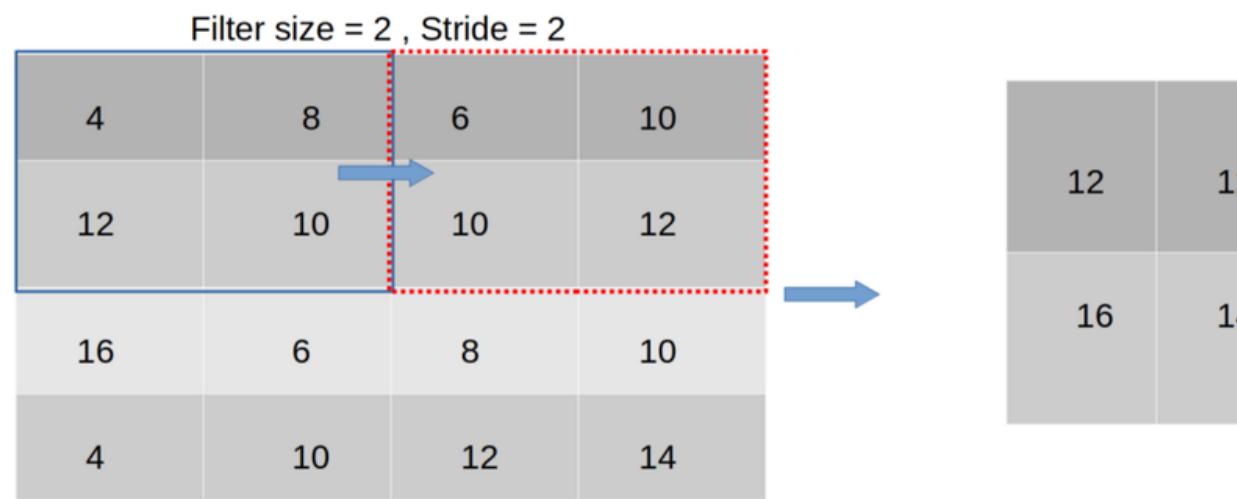
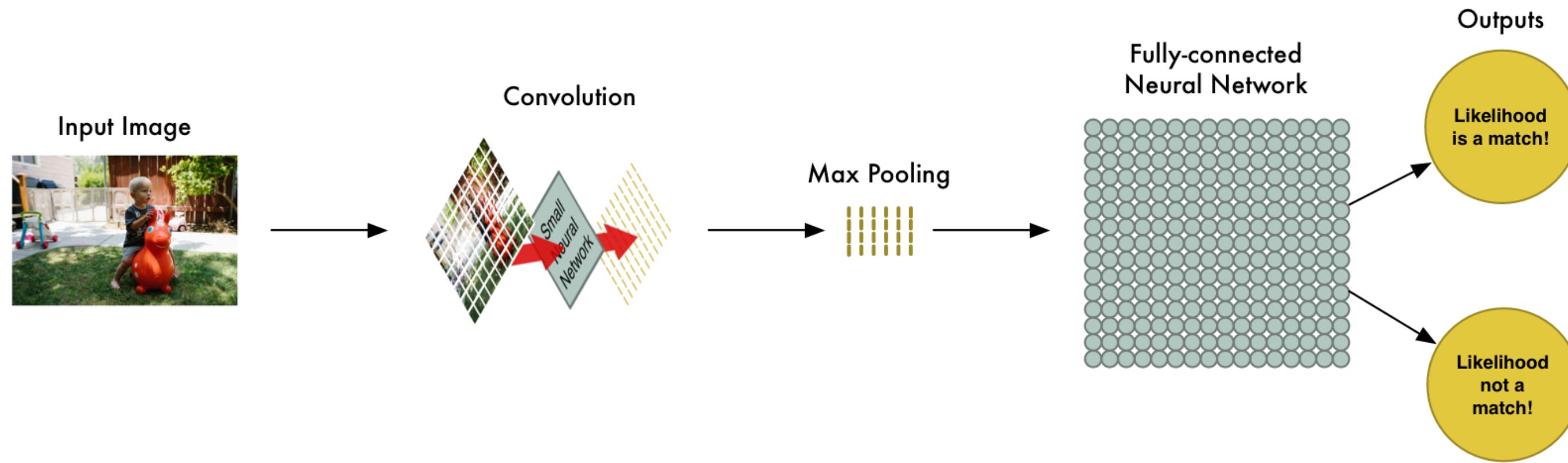
# **Computer Vision (Convolutional Neural Network)**





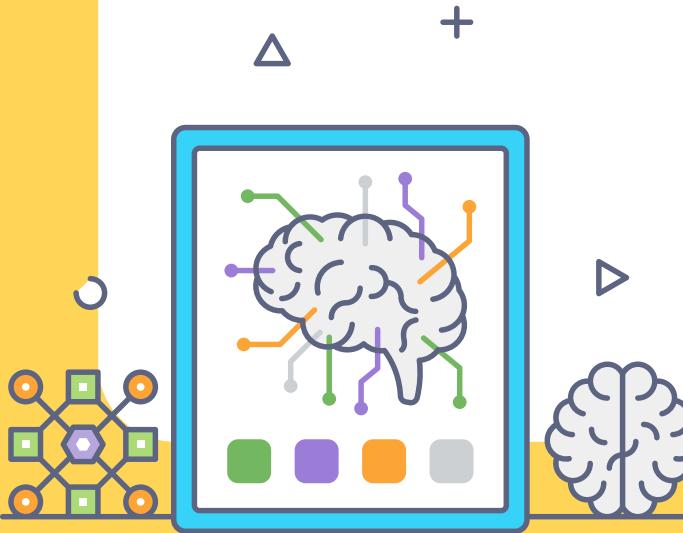
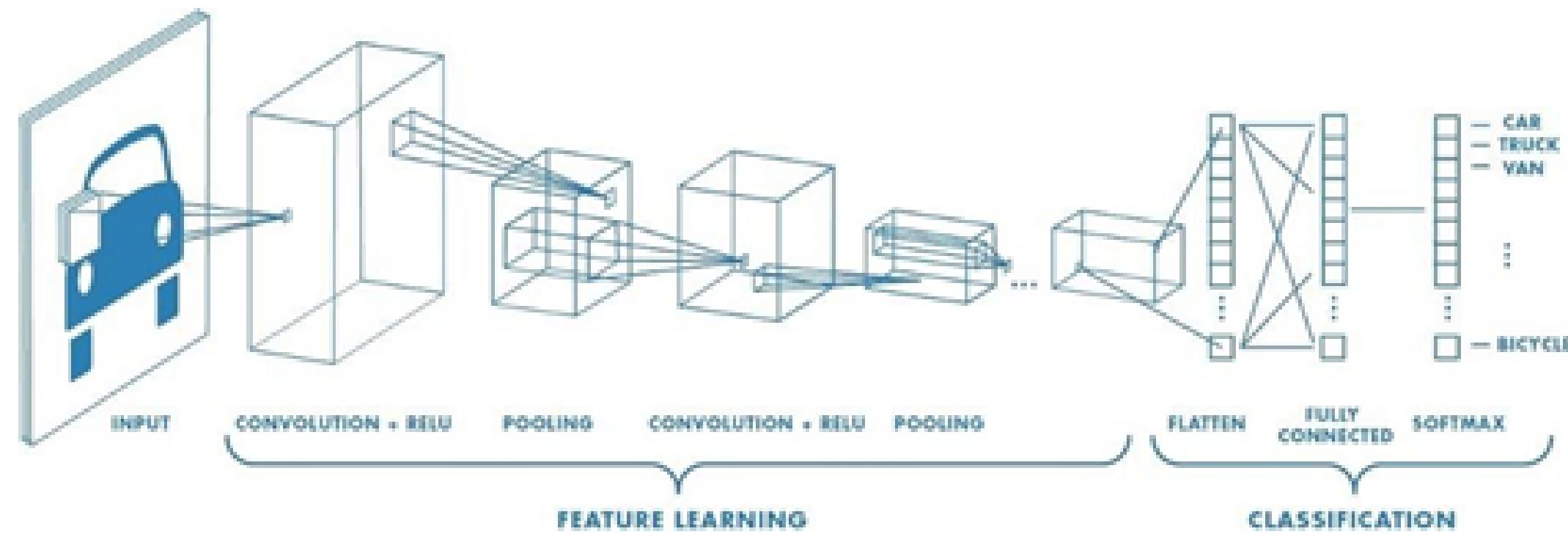


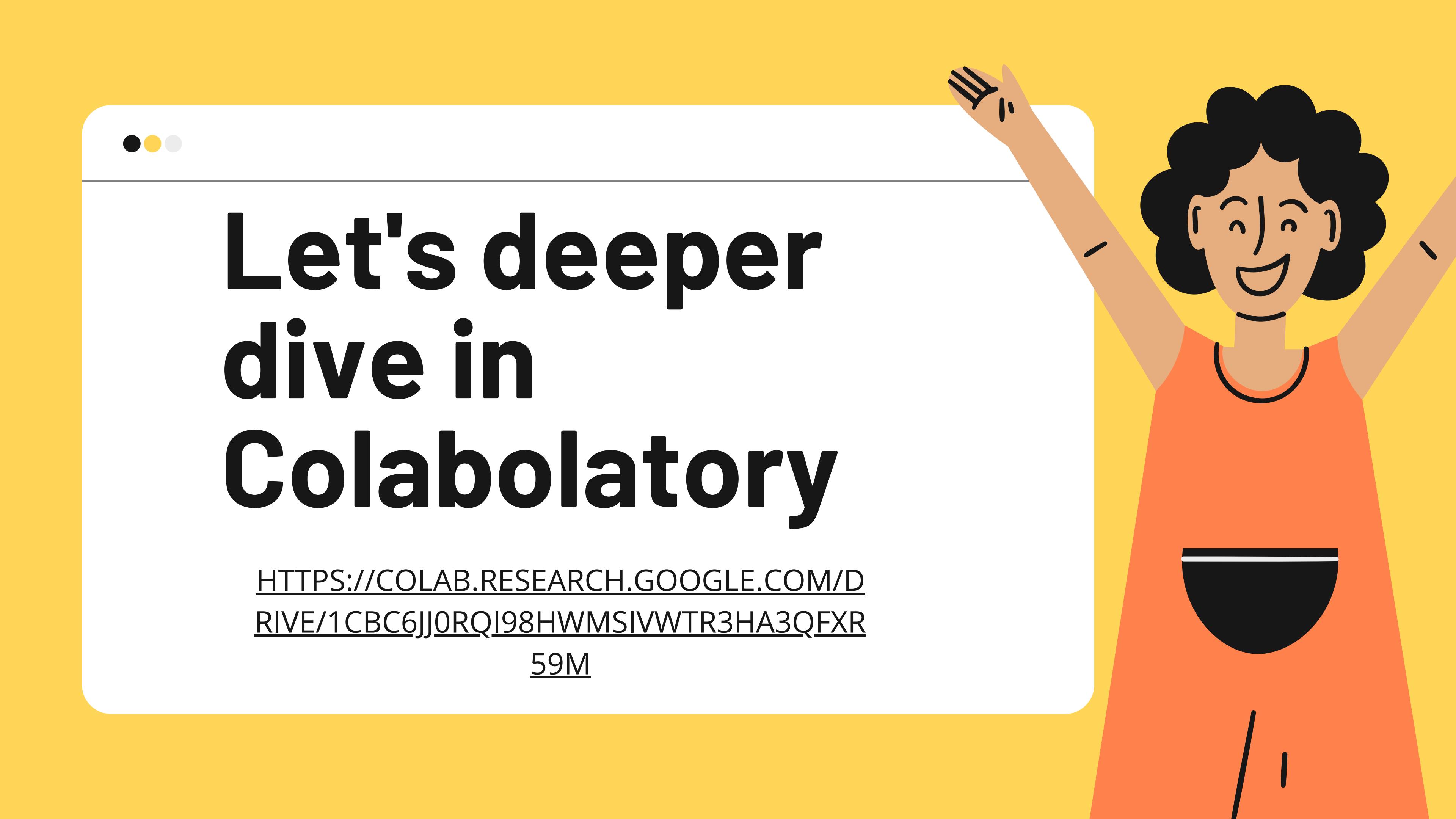
# CNN Architecture





# CNN Architecture





# Let's deeper dive in Colabolatory

[HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/1CBC6JJ0RQI98HWMSIVWTR3HA3QFXR](https://colab.research.google.com/drive/1CBC6JJ0RQI98HWMSIVWTR3HA3QFXR)

59M



# Terima kasih

Tetap Semangat !!!

“ Don’t be afraid to move, because  
the distance of 1000 miles starts  
by a single step.  
”

