

# TABLE OF CONTENT



## PYTHON 1

|                                      |    |
|--------------------------------------|----|
| <b>Meeting 1</b>                     |    |
| INTRO TO PYTHON .....                | 1  |
| <b>Meeting 2</b>                     |    |
| VARIABLE AND LISTS .....             | 12 |
| <b>Meeting 3</b>                     |    |
| LOOPS .....                          | 23 |
| <b>Meeting 4</b>                     |    |
| CONDITIONAL & CONDITIONAL LOOP ..... | 31 |
| <b>Meeting 5</b>                     |    |
| FUNCTION .....                       | 39 |
| <b>Meeting 6</b>                     |    |
| RANDOM MODULE .....                  | 47 |
| <b>Meeting 7</b>                     |    |
| PYGAME .....                         | 57 |
| <b>Meeting 8</b>                     |    |
| EXAM PYTHON BASIC .....              | 64 |

|                               |   |
|-------------------------------|---|
| <b>Meeting 1</b>              |   |
| INTRO TO PYTHON CHATBOT ..... | 1 |

|                                  |    |
|----------------------------------|----|
| <b>Meeting 2</b>                 |    |
| API & QUOTES CHATBOT .....       | 15 |
| <b>Meeting 3</b>                 |    |
| TRANSLATOR CHATBOT .....         | 25 |
| <b>Meeting 4</b>                 |    |
| DAILY HOROSCOPE CHATBOT .....    | 33 |
| <b>Meeting 5</b>                 |    |
| RECOMMENDER SYSTEM CHATBOT ..... | 39 |
| <b>Meeting 6</b>                 |    |
| FOOD ORDERINGCHATBOT .....       | 45 |
| <b>Meeting 7</b>                 |    |
| EXAM PYTHON CHATBOT .....        | 51 |
| <b>Meeting 8</b>                 |    |
| EXAM PYTHON CHATBOT .....        | 52 |

# PYTHON 1

## Meeting 1

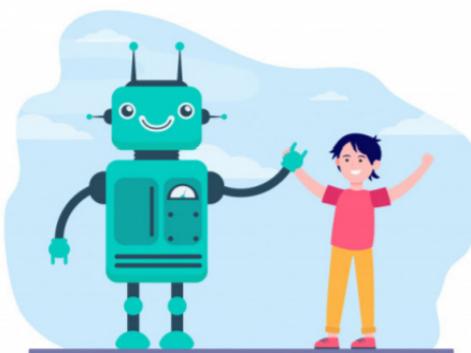
# INTRO TO PYTHON



## What will we learn today?

1. Introduction to Python and Replit
2. Introduction to basic Python programming

# PROGRAMMING LANGUAGE



What games have you played?

How do you give instructions to move the player?

## Computers communicate in a language different from ours!

So we have to use a language that could be understood by the computer

That language is called as CODING!

In Coding, there are several programming language that you can use!



To have a deeper understanding in programming languages

Let's watch the video below!

|              |         |       |
|--------------|---------|-------|
| Visual Basic | Fortran | Perl  |
| Mercury      | PHP     | Lisp  |
| Haskell      |         | Scala |
| Objective-C  | Rust    |       |
| Go           | C#      | SQL   |
| COBOL        | Dart    |       |
| Swift        | Curl    | Ruby  |

Source :  
[\[Youtube\] Computer Science Basics: Programming Languages](#)

From those languages mentioned in the video, have you ever used/heard one of them before?

Which languages do you think that you will often encounter?

# INTRO TO PYTHON



Have you heard of the Python programming language?

What can the Python programming language be used for?



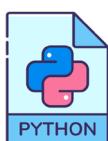
To have a deeper understanding about Python, let's watch the video below!

## WHAT IS PYTHON ?

Source :  
[Youtube] What is Python? Why Python is So Popular?



### VOCABULARY



Python is one of the most popular programming languages in the world, a multipurpose programming language that is very easy for beginners to learn.

Python can be used for:

- Basic programming
- Game
- Artificial Intelligence
- IOT
- Web and Desktop app

# WHY PYTHON?

## Who is the inventor of Python language?



This is Mr. Guido van Rossum, the Dutchman who was credited with creating the Python programming language which was released in 1991.

## Why do we use Python?



### 1. The most widely used programming language

Python is a great language to start with computer programming, many schools and universities use it to learn coding. In addition, Python can also be used to create various things.

### 2. Easy to learn

You can write code using English words, symbols, and numbers. This makes Python code easy to read, write, and understand.

### 3. Practical

Python has many tools that can speed up and make it easier for you to create programs. This tool is called the Standard Library.

### 4. Can be used anywhere

Python is portable. This means that you can write and run Python code on many computers that have different OS such as Windows, Mac, Linux, and IoT.

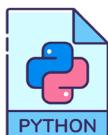
# MAKE REPLIT ACCOUNT



## What code editor do we use?



Replit is a very easy-to-use code editor that allows you to write and run code using a browser without having to install an IDE on your computer. Replit supports various programming languages



Now it's time for us to create a Replit Account to be able to access the features in Replit.

1. Open the link : <https://replit.com>
2. Click Start Coding then Sign Up with your Google account

The screenshot shows the Replit homepage with a dark background. The main text "Code, create, and learn together" is prominently displayed. Below it, a subtext reads: "Use our free, collaborative, in-browser IDE to code in 50+ languages – without spending a second on setup." A blue "Start coding" button is centered at the bottom, with a red box drawn around it to indicate the action step.

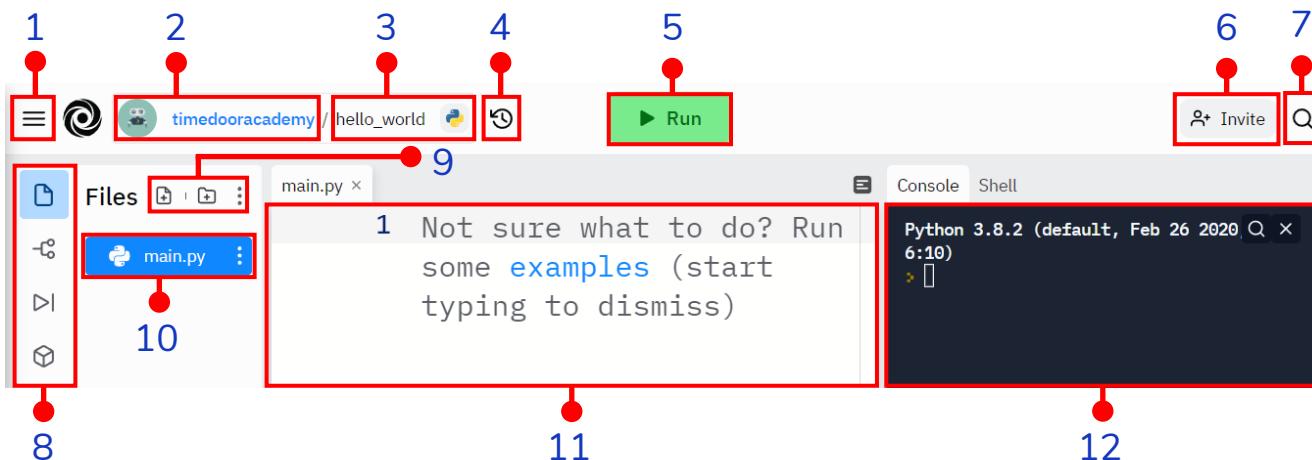
3. After successfully Sign Up it will appear like this in the browser

The screenshot shows the Replit dashboard. On the left, a sidebar menu includes options like "+ Create repl", "Upgrade", "Apps" (Beta), "Home" (selected), "My repls", "Talk", "Learn", "Teams", and "Curriculum". The main area features a "Replit Game Jam Oct. 18-25" banner with the text "Build a game on Replit, win prizes." Below the banner, there's a "Get started" section with three tasks: "Explore programming tutorials on Learn" (Incomplete), "See what people are building on Apps" (Incomplete), and "Create your first Repl" (Incomplete). The overall interface is clean and modern.

# BASIC PYTHON

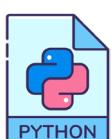


## Replic Parts



## VOCABULARY

**The console is used to execute short commands at a time and the code cannot be edited once it is executed.**



**Text Editor is used to compose long code and will be executed by python and display the results on the console.**



**After successfully creating a Replit Account, now is the time for you to learn School Basic Coding Python.**

# BASIC PYTHON



## Hello World!



We will learn to display the text “Hello World!” on the console.

1. Click [+Create](#)
2. Choose template Python, add title **hello\_world**, then [+Create repl](#)

3. Type the following code in the [console on the right](#) to display text “Hello World!”, then press Enter.



**Coding**

To display text in python we use the code:

**print("Text you want to display")**



**CHALLENGE**

Try to make code to display your name in the console!

# BASIC PYTHON



## Simple Calculator



In addition to displaying text, the console can also help us to operate mathematical operators to make calculations like a calculator.

1. We want to know the result of the calculation  $3 \times 4 : 2 = ?$
2. Type the following calculation in the console, then press the Enter key on the keyboard

```
> 3 * 4 / 2
6.0 → The result is 6
```



## Coding

| Operator | Keterangan                | Example |
|----------|---------------------------|---------|
| *        | Perkalian                 | 7 * 9   |
| /        | Pembagian                 | 8 / 2   |
| +        | Penjumlahan               | 6+3     |
| -        | Pengurangan               | 10 - 3  |
| %        | Modulus (sisa hasil bagi) | 7 / 2   |



## CHALLENGE

Try calculating the answers to the questions below with Python coding!

1.  $2 \times 9 + 4 = ...$
2.  $100 / 25 = ...$
3.  $250 - 2 \times 25 = ...$
4.  $225 / 5 - 5 \times 6 + 2 \times 7 = ...$

# BASIC PYTHON



## Text and Paragraph

66  
—  
— 99

After trying to code on the console, now we will learn to display text and paragraphs on the console but coded on the console text editor.

1. Type the following code in the text editor, then press the Run button

```
nek / hello_world
```

main.py ×

```
1 print("Hello World!")
```

2. Take a look at the console, is there a difference in results with previous code?
3. We will try to make paragraphs with the same code as above, type the following code in the text editor, then press button Run

```
main.py ×
```

```
1 print("Hello World!")
2 print("Leonardo da Vinci spent 17 years
in Milan,
3 He was listed in the register of the
royal household as painter and engineer
of the duke.
4 Leonardo's gracious but reserved
personality and elegant bearing were
well-received in court circles.")
```



# BASIC PYTHON

4. How is the result? Is there an error like this in the console?

Console   Shell

```
File "main.py", line 2
    print("Leonardo da Vinci spent 17 years in
Milan.

 ^
SyntaxError: EOL while scanning string literal
```

5. To create code that can display paragraphs, we must add 2 more double quotes at the beginning and end of in brackets code.

```
print("""Leonardo da Vinci spent 17
years in Milan.
He was listed in the register of the
royal household as painter and engineer
of the duke.
Leonardo's gracious but reserved
personality and elegant bearing were
well-received in court circles.""")
```

6. Then click the Run button and see the results on the console



Coding

To display paragraphs in python we use code:

**print("““The paragraph you want to display.
The paragraph you want to display.””")**

# BASIC PYTHON



## Number and Operator



We will learn to display numbers and calculation results with operators on the console but coded in a text editor.

Type the following code in the text editor on lines 5 and 6, then click the Run button

```
5 print(60)
6 print(250 - 25)
```



### Coding

To display numbers and calculation results with operators in python we use code:

**print(Numbers or Calculations)**



### CHALLENGE

1. Try showing your own bio using Python coding with details:  
Name, age, school, hobbies, and why you study coding!
  
2. Try showing info about your favorite game using Python coding with details:  
Game name, platform (mobile, PC, consoles), how to do it play it, and what made you like the game



## Let's Play With TYNKER

We will play Tynker using Python concept!

Link : <https://www.tynker.com/>  
 Material : Python 1 - Lesson 1  
 Level : 1 - 7

# HOW WAS TODAY'S LEARNING ?



1

What can be made in Python?

Answer: \_\_\_\_\_  
\_\_\_\_\_

2

Who created the Python programming language? and  
what year was Python first released?

Answer: \_\_\_\_\_  
\_\_\_\_\_

3

Why do we use Python?

Answer: \_\_\_\_\_  
\_\_\_\_\_

4

How to code to display the text “How are you?”?

Answer: \_\_\_\_\_  
\_\_\_\_\_

## Meeting 2

# VARIABLE & LISTS



### What will we learn today?

1. Introduction to the Concept of Variables & Lists in Python

# INTRO TO VARIABLE



After successfully learning Basic Coding Python in the previous meeting, it's time for you to learn Variables and data types.

## Variable



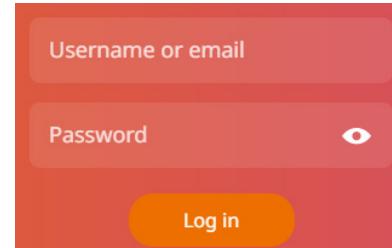
## VOCABULARY



Variable is a place to store data/value

## When do we use a Variable?

When creating an account, the username data will be stored in the Username variable, as well as the password data



When you want to log in, the system will match the data you entered with the data on the variable, if the data match then you will be able to log in to your account.

## What happens if you enter your home address instead of your email or username in the Username or email field?

Beside text, variables can also contain numbers!

Variables containing numbers can be called and operated using the operator. Let's answer the following questions using the concept of variables!

$$\text{flamingo} = 3$$

(A)

$$\text{fox} = 5$$

(B)

$$\text{elephant} = 7$$

(C)

$$\text{flamingo} + \text{elephant} - \text{fox} = ..... ?$$

(A)

(C)

# INTRO TO VARIABLE



## Rules for typing variables

# Aa

In Python, writing variables using both upper and lower case letters must be consistent, because Python is a programming language that is **Case Sensitive**.

Variables also cannot be typed using spaces, spaces can be replaced with underscores (\_), for example **first name** into **first\_name**.

```
first_name = "Steve"
last_name = "Job"

print("My name is " + first_name + " " + last_name) ✓
print("My name is " + First_Name + " " + LAST_Name) ✗
```

We will learn to display the data that is in the variable.

1. Open link : <https://replit.com>
2. Login with your Google account
3. Click **+Create repl**
4. Choose a Python template, add a title **variable**, then **+Create repl**
5. Type the following code in the text editor, then press the Run button and see the results on the console

```
x = 6
print(x) → To display number

greeting = "Hello World!"
print(greeting) → To display text
```



### CHALLENGE

Try to answer these following question with Python coding in Replit!

```
apel = 4
pisang = 10 * apel
jeruk = pisang - apel
```

# INPUT



## Recalling About Operator

| Operator | Keterangan  | Contoh |
|----------|-------------|--------|
| *        | Perkalian   | 7 * 9  |
| /        | Pembagian   | 8 / 2  |
| +        | Penjumlahan | 6+3    |
| -        | Pengurangan | 10 - 3 |

Input



## VOCABULARY

Input is response that we give to the program.

- Type the following code in the text editor to display the inputted name and combine the variable name with the text.

To display the input data on the variable name

To input data into a variable name

```
name = input("What's your name? ")
print(name)
print("Hello my name is " + name)
```

Separator between text and variables

To display text and input data on the variable name

- Press the Run button and see the results on the console, then input your name on the console then press enter

```
What's your name? Alexa
Alexa
Hello my name is Alexa
> []
```

Input from the user

# DISPLAY INPUT



3. Type the following code in the text editor to display the age and combine name and age variables with text, then press button Run and see the results on the console, then input it your name and age on the console then press enter.

```
age = input("How old are you? ")
print("Hello, I'm " + name +
", I'm " + age + " years old")
```



## CHALLENGE

Try adding:

1. Variable with the **name of the groceries** with text input  
“What do you want to buy?”
2. Variable with the name of the **number** with text input  
“How much do you want to buy?”

Then display the text as below:

**Hello I want to buy 10 apple**



## Coding

To display variable data in python we use code:

**Variable = Data**  
**Variable = input(Text)**  
**Print (Variable)**

# DATA TYPES



In coding, we find data in the form of words and numbers.

Can name and age data be grouped into the same data type?

We need to learn data types so that we can group data into several types according to program needs.

## Data Type



## VOCABULARY



**Data Types** is a grouping of data based on the type of data. Data types are needed so that the compiler can know how the data will be used.



To have a deeper understanding and recall the types of data types, let's watch the video below!

DataCamp

Intro to Python for Finance

INTRO TO PYTHON FOR FINANCE

### Variable Data Types

Source :  
[Youtube] Python Tutorial : Variable Data Types

Adina Howe  
Instructor

## Types of Data Type

| Variable Types | Example       | Abbreviations |
|----------------|---------------|---------------|
| Strings        | 'Tuesday'     | str           |
| Integers       | 40            | int           |
| Floats         | 3.1417        | float         |
| Booleans       | True or False | bool          |

# DATA TYPES



We will learn to display data types and change data types.

- Type the following code in the text editor to find out the types of data type on the variable, then press the Run button

```
p = 77.980
print(type(p))
```

- Type the following code in the text editor to change the data type variable p which was originally a float into a string and store it in the r variable, then press the Run button

```
r = str(p)
print(type(r))
```



## Coding

To display the data type in python we use the code:

**print(type(Data atau Variable))**

To change the data into the data type we want, we use code with an abbreviation according to the data type as below:

**float(Data atau Variable)**



## CHALLENGE

- Try to find out the data type below by coding Python in Replit!  
**25.5 = ...**  
**“hello” = ...**
- Try to change the following variables to float data then display the results with Python coding in Replit!  
**x = 12345**

# INTRO TO LISTS



## Lists



Do you remember what a variable is?  
How much data can a variable store?  
What should we do if we want to store more than 1 data?



## VOCABULARY



List is a place to store a lot of data sequentially.

Imagine you are making a game and want to save the name of the player in each team. If you code a variable for each player name, it will look as in the image below. It uses a lot of space, right?



main.py \*

```

1 rockets_player_1 = 'Rory'
2 rockets_player_2 = 'Rav'
3 rockets_player_3 = 'Rachel'
4 UFO_player_1 = 'Peter'
5 UFO_player_2 = 'Pablo'
6 UFO_player_3 = 'Polly'
```

With coding, we can make it simpler if we use a list concept.

main.py \*

```

1 rockets_players = ['Rory', 'Rav', 'Rachel']
2
3 UFO_players = ['Peter', 'Pablo', 'Polly']
```

In lists, the data is sorted (indexed) from the numbers 0, 1, 2, 3,... so to display the data in the first order we use [0], the second data uses [1], and so on.

To display the data in the last order and to make it faster we can use [-1], the 2nd last data [-2], and so on.

# LIST IN PYTHON



Animals = ["Lion", "Elephant", "monkey", "Koala", "flamingo"]

Index -> [0] [1] [2] [3] [4]

← →  
Length = 5

We will learn how to display data in lists.

1. Click +Create repl
2. Choose Python template, add title **lists**, then +Create repl
3. Type the following code in a text editor to display a list of fruits, then press the Run button

```
fruits = ["apple", "banana", "cherry"]
print(fruits)
```

Console Shell

['apple', 'banana', 'cherry']

4. Type the following code in the text editor, then press the Run button

print(len(fruits))

→ To display the amount of data

print(type(fruits))

→ To display the data type

How much is the data? \_\_\_\_\_

What is the data type? \_\_\_\_\_

5. Type the following code in the text editor, then press the Run button

print(fruits[0])

→ To display the first data

print(fruits[1])

→ To display the second data

print(fruits[-1])

→ To display the last data



# LIST IN PYTHON

What is the first data? \_\_\_\_\_

What is the second data? \_\_\_\_\_

What is the last data? \_\_\_\_\_

- In addition to storing data with the same data type, lists
- can also accommodate data with different data types.
- Let's try to practice!

6. Type the following code in the text editor, then press the Run button

```
list1 = ["abc", 34, true, 40, "male"]
print(list1)
```



## CHALLENGE

Try to display the data in the 2nd, 5th, and last order in the Pokemon list below by coding Python in Replit!

```
pokemon = [Pikachu, Bulbasaur, Charmeleon, Squirtle,
Wartortle, Kakuna, Kadabra]
```

5th data= .....

Last data= .....



## Coding

To store data in the list we use the following code:

```
nama_list = [Data, Data, Data]
```

To display the number of data in the list we use the code:

```
print(len(nama_list))
```

To display the data type we use the code:

```
print(type(nama_list))
```

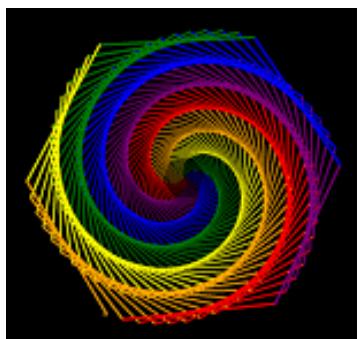
To display the order of data in the list we use the code:

```
print(nama_list[index])
```

# PYTHON AND TURTLE



Turtle



In addition to displaying text, we can also create simple images in Python. Python provides a Turtle module that moves the robotic “turtle” around the screen, as well as draws with a pen as it walks.



## VOCABULARY



Turtle is a coding package that can be used to draw lines, circles, and other more complex shapes in Python.

The Turtle module allows us to draw by giving instructions to the turtle

The instructions given may include when to start or stop drawing, where to move, what colors to use, and so on.

We will learn to draw a triangle using a turtle.

1. Click +Create repl
2. Choose the Python template, add title **myturtle**, then +Create repl
3. Type the following code in the text editor to draw a triangle with an angle of 120 degrees each, then press the Run button

```
import turtle ---  
  
t = turtle.Pen()---  
t.shape("turtle")---  
t.pencolor("blue")---  
t.forward(50)---
```

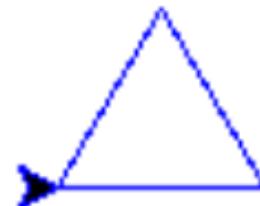
- To call the turtle coding package
- To draw with a pen
- To make a turtle shape
- To color the line in blue
- To move forward draw a 50px long straight line



# DRAWING WITH TURTLE

```
t.left(120) - - - - -  
t.forward(50)  
t.left(120)  
t.forward(50)  
t.left(120)
```

To turn left at an angle of 120 degrees



| Command   | Description             |
|-----------|-------------------------|
| forward() | Untuk bergerak maju     |
| left()    | Untuk berputar ke kiri  |
| right()   | Untuk berputar ke kanan |



## CHALLENGE

Try to create a square shape with 100px long straight lines and green lines using Python's coding in Replit!



## Let's Play With TYNKER

We're going to play Tynker using Python concepts!

Link : <https://www.tynker.com/>  
 Material : Python 1 - Lesson 1  
 Level : 8 - 14

# HOW WAS TODAY'S LEARNING ?



1

How to code to display data on variables?

Answer: \_\_\_\_\_

2

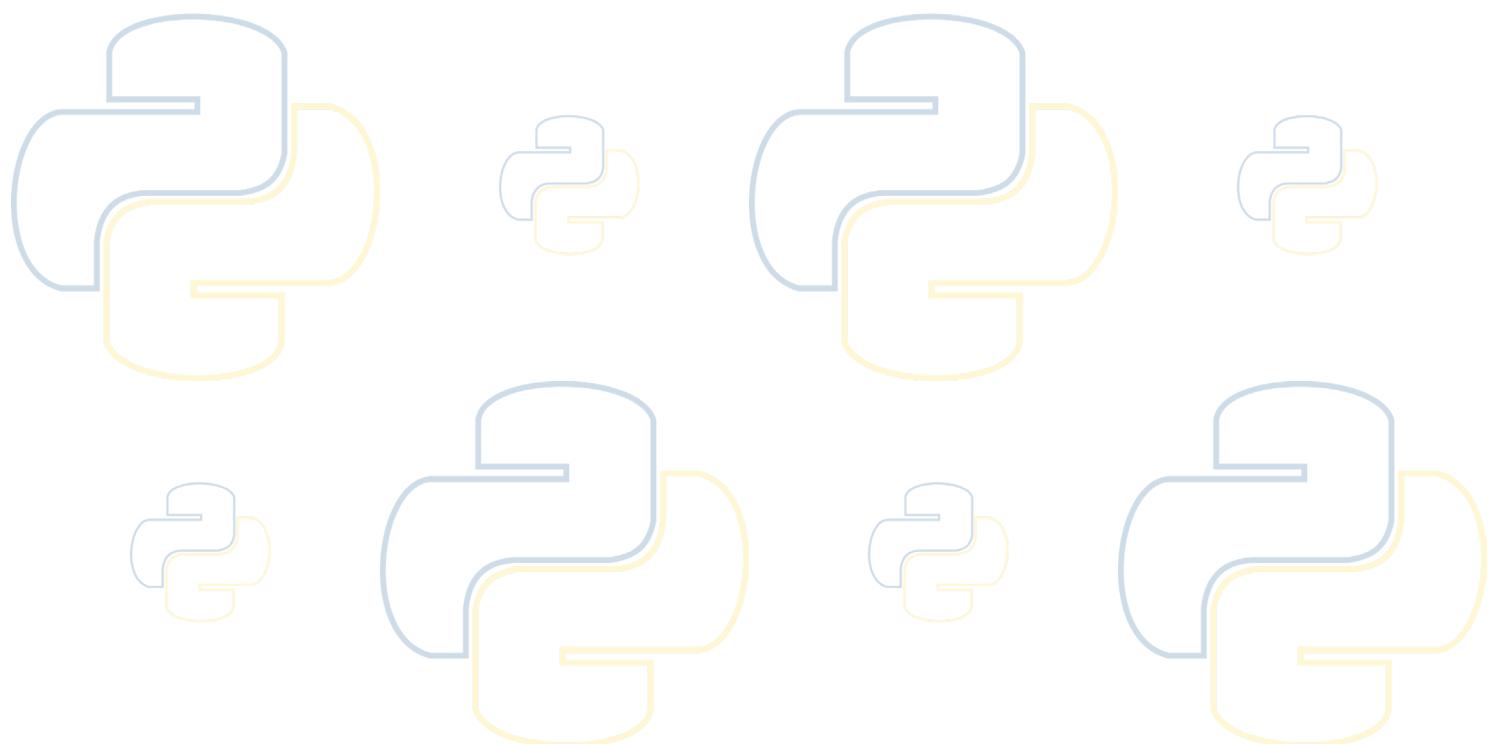
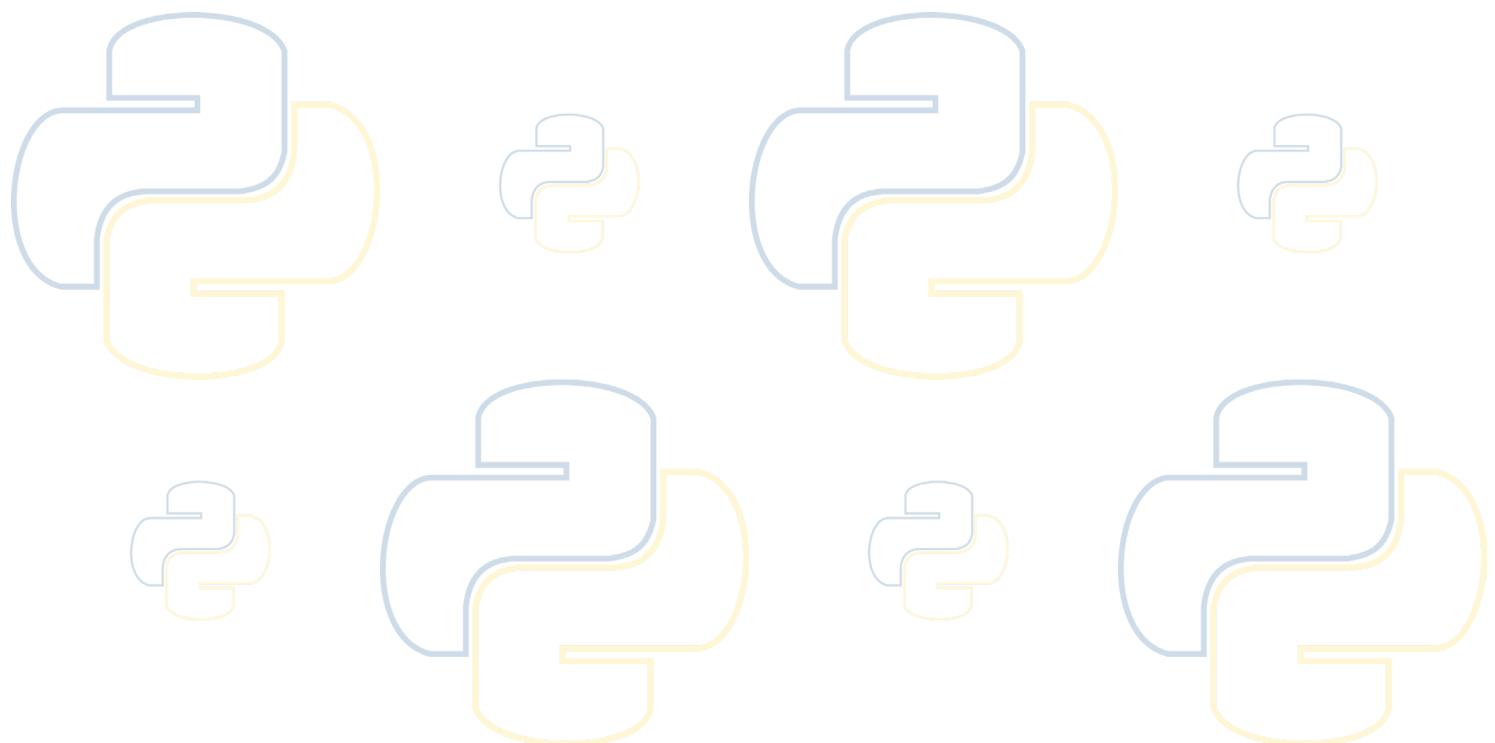
Mention the types of data types!

Answer: \_\_\_\_\_

3

How to code to display the amount of data in a list?

Answer: \_\_\_\_\_



## Meeting 3

# LOOPS



## What will we learn today?

1. Introduction of the Loops Concept in Python

# INTRO TO INDENTATION



After successfully learning variables and data types, it's time for you to learn about Indentation and Loops



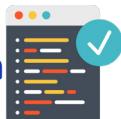
Have you ever seen messy and disorganized coding?

Of course, it will be difficult to understand and difficult to fix the error, right?

Python has a rule for typing code to make the code look neat which is called Indentation

## Do you know what Indentation is?

Indentation



### VOCABULARY



Indentation is writing paragraphs that are slightly indented to group blocks of program code.

Code 1 :

Indentation Level 1      ↔      Code 2 :

Indentation Level 2      ↔      Code 3

Code 3

Code 2

Code 1

# INTRO TO LOOPS



## When is indentation used?

Indentation is very important used in several coding concepts such as **Loops** (for & while), **conditional** (if, else, elif), and **function**.

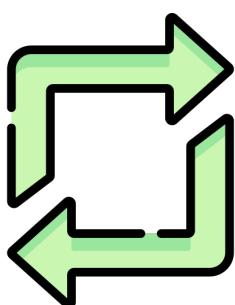
Indentation is used to make it easier for humans to read program code and is tasked with defining the structure of program code blocks

**Making mistakes in indenting Python code can cause program errors.**



We will start using loops in coding so pay attention to your indents in coding.

The indent is created by pressing the **Tab** key on the keyboard.



**Have you ever coded with the same command over and over?**

**Of course, it takes a lot of time and effort, doesn't it make the same code over and over again?**



**Using repeat can shorten the coding that is compiled**

# LOOPS



Loops

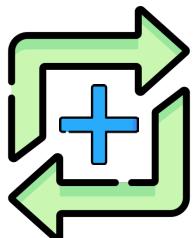


## VOCABULARY



Loops are a method for running the same code over and over again.

There are 2 types of loops in python, they are:



### For Loops

For loops are used when you know how many times you want to run the code loop.



### While Loops

While loops are used when you don't know how many times you want to run the coding loop and the coding will loop over and over as long as the condition is met.



What do you think is the difference between the two types of loops?

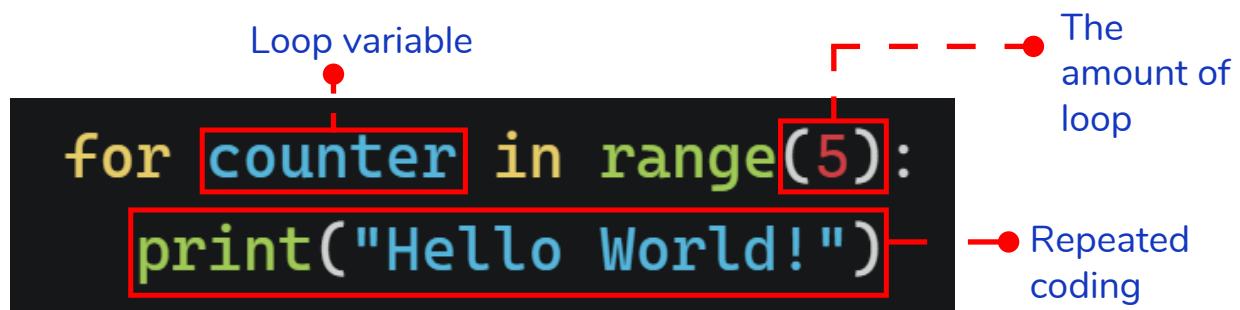
Do you know what code to use to loop in Python?

# FOR LOOPS



We will learn to display data using For Loops.

1. Open the link : <https://replit.com>
2. Log in with your Google account
3. Click +Create
4. Choose the Python template, add title **for\_loops**, then +Create repl
5. Type the following code in the text editor to display the text 5 times, then press the Run button

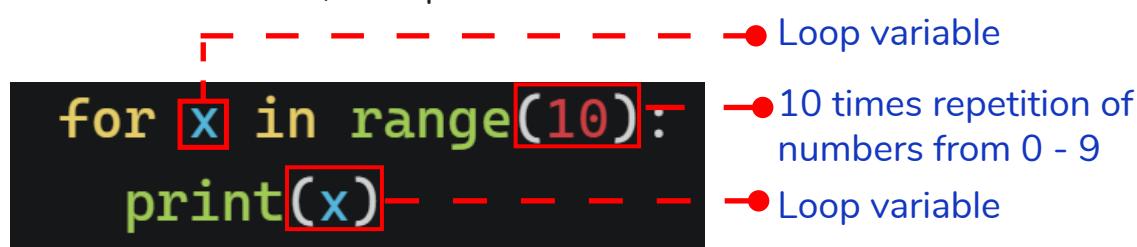


6. Type the following code in the text editor to display the text + variable 5 times, then press the Run button

```

name = "Kakuna"
for counter in range(5):
    print("Hello " + name)
  
```

7. Type the following code in the text editor to display the numbers 0-9, then press the Run button



Starts at 0 by default, increases by 1 (by default), and ends at the specified number.

to display the numbers 1-10, change the code `print(x)` to be like this:

```
print(x+1)
```

8. Type the following code in the text editor to count down the numbers 10 - 1 with the concept of decrement, then press the Run button



# FOR LOOPS

```
for i in range(10, 0, -1):
    print(i)
```

Start   End

(-) Decrement : subtraction  
score  $x = x - 1$

8. Type the following code in the text editor to calculate multiples numbers with the concept of increment, then press the Run button

```
for i in range(0, 10, +2):
    print(i)
```

(+) Increment : addition  
score  $x = x + 2$

9. Type the following code in the text editor to display the data in list pokemon in sequence, then press the Run button

```
pokemon = ["Pikachu", "Bulbasaur", "Squirtle"]
for x in pokemon:
    print(x)
```

10. Type the following code in the text editor to display the number of data loops entered, then press the Run button

```
person = input("What's your name? ")
loops = input("How many times you want to repeat?")
for counter in range(int(loops)):
    print("Hello, " + person)
    print("Good Luck!")
```

To convert a string variable to an integer so that the loop can be processed



Coding

To create a coding loop in python we use the code:

**for Counter\_Variable in range (Number) :**

or

**for Counter\_Variable in range (Start, End, Increment/decrement) :**

or

**for Counter\_Variable in Variable :**

# WHILE LOOPS



We will learn to display data using While Loops.

1. Click +Create
2. Choose the Python template, add title **while\_loops**, then +Create repl
3. Type the following code in the text editor to display the numbers 0-9, then press the Run button

```
i = 0
while i < 10:
    print(i)
    i += 1
```

Start

Condition end when i is less than 10

Increment which has the same meaning as **i = i + 1**

4. Type the following code in the text editor to count down the numbers 5-1, then press the Run button

```
b = 5
while not b == 0:
    print(b)
    b -= 1
```

Condition end when b is not equal to 0

Decrement which has the same meaning as **i = i - 1**



To create a coding loop in python we use the code:

**while Condition end :**  
or  
**while not Condition end :**

# WITH OR WITHOUT INDENTATION

How important is the influence of indentation in writing coding?

Type the following code, then press the Run button!



```
for counter in range(5):  
    print("Hello World!")
```

What is the result from coding you made?

Also try to create and run the following code!



```
for counter in range(5):  
print("Hello World!")
```

Is there a difference in the results of coding with and without indentation?



What do you think is the difference between both codings?

# TURTLE

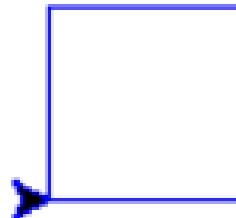


We will learn how to draw a square using a turtle.

1. Click [+Create repl](#)
2. Choose Python template, add title **myturtle**, then [+Create repl](#)
3. Type the following code in the text editor to draw a square with a 90 degree angle using the for loop concept, then press the Run button

```
import turtle

t = turtle.Pen()
t.shape("turtle")
t.pencolor("blue")
t.speed("fast")-- -
for x in range(4):
    t.forward(50)
    t.left(90)
```



We can adjust the speed of the turtle when drawing from the slowest to the fastest:  
“slowest”, “slow”, “normal”, “fast”, and “fastest”



## CHALLENGE

Try to display a triangle image using the concept of a for loop with a green line!



## Let's Play With TYNKER

We're going to play Tynker using Python concepts!

Link : <https://www.tynker.com/>  
Material : Python 1 - Lesson 2

| Command             | Description                          |
|---------------------|--------------------------------------|
| <b>forward()</b>    | Bergerak satu langkah ke depan       |
| <b>fire()</b>       | Meluncurkan panah 5 langkah ke depan |
| <b>jump()</b>       | Melompat                             |
| <b>turn_right()</b> | Berbekok ke kanan                    |
| <b>turn_left()</b>  | Berbekok ke kiri                     |

# HOW WAS TODAY'S LEARNING ?



**1** What is the difference between for loops and while loops?

Answer: \_\_\_\_\_  
\_\_\_\_\_

Complete the following code format!

**2** `for _____ in range (Start, End, _____) :`

Answer: \_\_\_\_\_  
\_\_\_\_\_

What is the coding format for while loops?

**3** Answer: \_\_\_\_\_  
\_\_\_\_\_

## Meeting 4

# CONDITIONAL & CONDITIONAL LOOP



## What will we learn today?

1. Introduction of Conditional Concepts in Python
2. Introduction of Conditional Loops in Python



# INTRO TO CONDITIONAL



Have you ever felt confused when choosing a toy?

How do you decide which toy you want to choose?



After successfully learning Loops, it's time for you to learn Conditional and Conditional Loops.

Conditional



**IF RAIN,**  
then Andre can't play outside

**ELSE**

Andre can play outside

**So what are the conditions for Andre to be able to play outside?**



## VOCABULARY



Conditional is coding to carry out actions according to conditions



## Coding

| Code | Deskripsi   |
|------|-------------|
| if   | Jika        |
| elif | Selain jika |
| else | Selain itu  |

# INTRO TO DATA TYPES



The use of if, elif, else

```
if (condition 1):
    action 1
else :
    action 2
```

Coding if and else is used to check only one condition. It can be said to only check whether the condition is true or false.

```
if (condition 1):
    action 1
elif (condition 2):
    action 2
else :
    action 3
```

Coding if, elif, and else is used to check more than 1 condition. For the 2nd, 3rd, and so on conditions use elif coding.

## Logical Operators



## VOCABULARY



Logical operators are symbols that tell the computer to make comparisons.

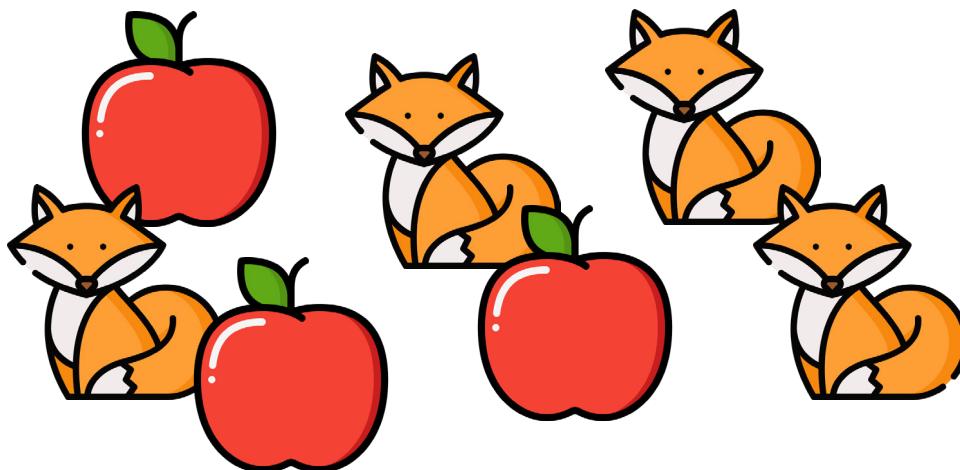
| Symbol            | Meaning             |
|-------------------|---------------------|
| <code>==</code>   | <b>equal to</b>     |
| <code>!=</code>   | <b>not equal to</b> |
| <code>&lt;</code> | <b>less than</b>    |
| <code>&gt;</code> | <b>greater than</b> |



# INTRO TO CONDITIONAL

We will learn to compare data with logical operators

1. Open link : <https://replit.com>
2. Log in with your Google account
3. Click +Create repl
4. Choose Python template, add title **logical\_oprtr**, then +Create repl
5. Let's check the number of Apples and Foxes below use **logical operator**.



Type the following code in a text editor to declare variables

```
apple = 3
fox = 4
```

6. Type the following code in the text editor, then press the Run button

|                                    |                             |
|------------------------------------|-----------------------------|
| <code>print(apple == fox)</code>   | <input type="radio"/> False |
| <code>print(apple != fox)</code>   | <input type="radio"/> True  |
| <code>print(apple &lt; fox)</code> | <input type="radio"/> True  |
| <code>print(apple &gt; fox)</code> | <input type="radio"/> False |

Are the number of Apples and Foxes the same? \_\_\_\_\_

Is the number of Apples less than the number of Foxes? \_\_\_\_\_

# INTRO TO DATA TYPES



7. You can use **and**, **or** to combine more than one comparison.  
 If you use **and**, both the comparison part must be true for the statement to be **True**. If you use **or**, only one part that should be correct

One part is wrong, the result will be False

```
print((apple == 3) and (fox == 2)) → False
print((apple == 3) and (fox == 4)) → True
print((apple == 3) or (fox == 2)) → True
```

One part is correct, the result will be True

We will learn to compare data with 2 conditions.

1. Click +Create repl
2. Choose Python template, add title **light\_detectr**, then +Create repl
3. Let's check the situation outside is it dark? if so then the system prints "Good night", otherwise the system will print "Good morning".

Comment : coding that is not executed by the system, usually to provide notes

In python, comments start with the **#** symbol.

```
#light detection
is_dark = input("Is it dark outside? y/n ")
if is_dark == "y":
    print("Good night! Zzz....")
else:
    print("Good morning !")
```

y : yes  
n : no





# INTRO TO CONDITIONAL

We will learn to compare data with more than 2 conditions.

1. Click [+Create repl](#)
2. Choose Python template, add title **weather**, then [+Create repl](#)
3. Let's check today's weather, if it's Raining, Snowy or Sunny, then the print system according to the following codes!

```
weather = input ("What is the weather for
today? (rainy/snowy/sunny)")
if weather == "rainy":
    print("Remember your umbrella!")
elif weather == "snowy":
    print("Remember your wooly gloves!")
else:
    print("Remember your sunglasses!")
```



We will learn to combine more than 1 comparison with conditionals.

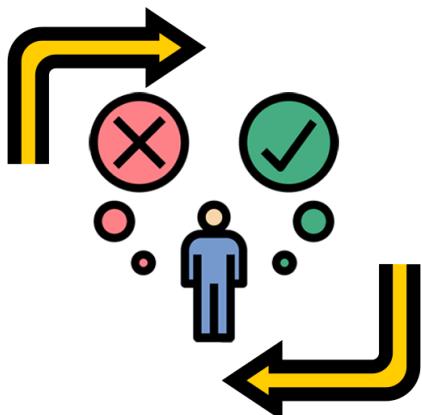
1. Open link : <https://replit.com>
2. Log in with your Google account
3. Click [+Create repl](#)
4. Choose Python template, add title **roller\_coaster**, then [+Create repl](#)
5. Let's select the person who will play the roller coaster if the age is > 10 years and the height is > 100cm then he can play the roller coaster, if the age and height are below those numbers then he is not allowed to play the roller coaster.



```
age = int(input("How old are you?"))
height = int(input("How tall are you?"))
if age > 10 and height > 100:
    print("Congratulation! You may ride a roller coaster")
else:
    print("Sorry, You may not ride a roller coaster")
```

**int** is used to convert the input string variable into an **integer**

# INTRO TO DATA TYPES



Do you remember what Loop is?

We will combine the use of Conditional and Loop concepts to make coding with looping conditions

Conditional Loop



## VOCABULARY



Conditional Loop is coding to make decisions according to conditions repeatedly.

We will learn to code using the concept of conditional loops.

1. Open link : <https://replit.com>
2. Log in with your Google account
3. Click +Create repl
4. Choose Python template, add title **BMI**, then +Create repl
5. Let's calculate Body Mass Index (BMI) to calculate ideal body weight based on body weight in kilograms and height in meters. Then group the BMI into several categories.

|               |             |
|---------------|-------------|
| Less Weight   | = <18.5     |
| Normal Weight | = 18.5-24.9 |
| Overweight    | = 25-29.9   |
| Obesity       | = >30       |





# INTRO TO CONDITIONAL

Using while loop because it has no loop limit

```
while True:
```

```
    print("Body Mass Index (BMI)")
    weight = float(input("How much do you weight in Kilograms? "))
    height = float(input("What is your height in meters? "))

    BMI = weight / (height*height)

    if BMI < 18.5:
        print("Less Weight")
    elif 18.5 < BMI < 24.9:
        print("Normal Weight")
    elif 25 < BMI < 29.9:
        print("Overweight")
    else:
        print("Obesity")
    print("Your BMI = " + str(BMI))

    ans = input("Do you want to input again? y/n --> ")
    if ans == "n":
        break
```

To stop the looping process



## Let's Play With TYNKER

We're going to play Tynker using Python concepts!

Link : <https://www.tynker.com/>  
 Material : Python 1 - Lesson 3 & 4

| Command                       | Description               |
|-------------------------------|---------------------------|
| <code>has_path_right()</code> | Ada jalur ke kanan        |
| <code>has_path_left()</code>  | Ada jalur ke kiri         |
| <code>has_path_ahead()</code> | Ada jalan di depan        |
| <code>enemy_in_sight()</code> | Ada musuh di depan        |
| <code>reached_goal()</code>   | Telah mencapai pixie dust |
| <code>is_gap_ahead()</code>   | Ada lubang                |

# HOW WAS TODAY'S LEARNING ?



1

What are the types of conditional coding?

Answer: \_\_\_\_\_

---

2

What is the difference between using “and” and “or” coding?

Answer: \_\_\_\_\_

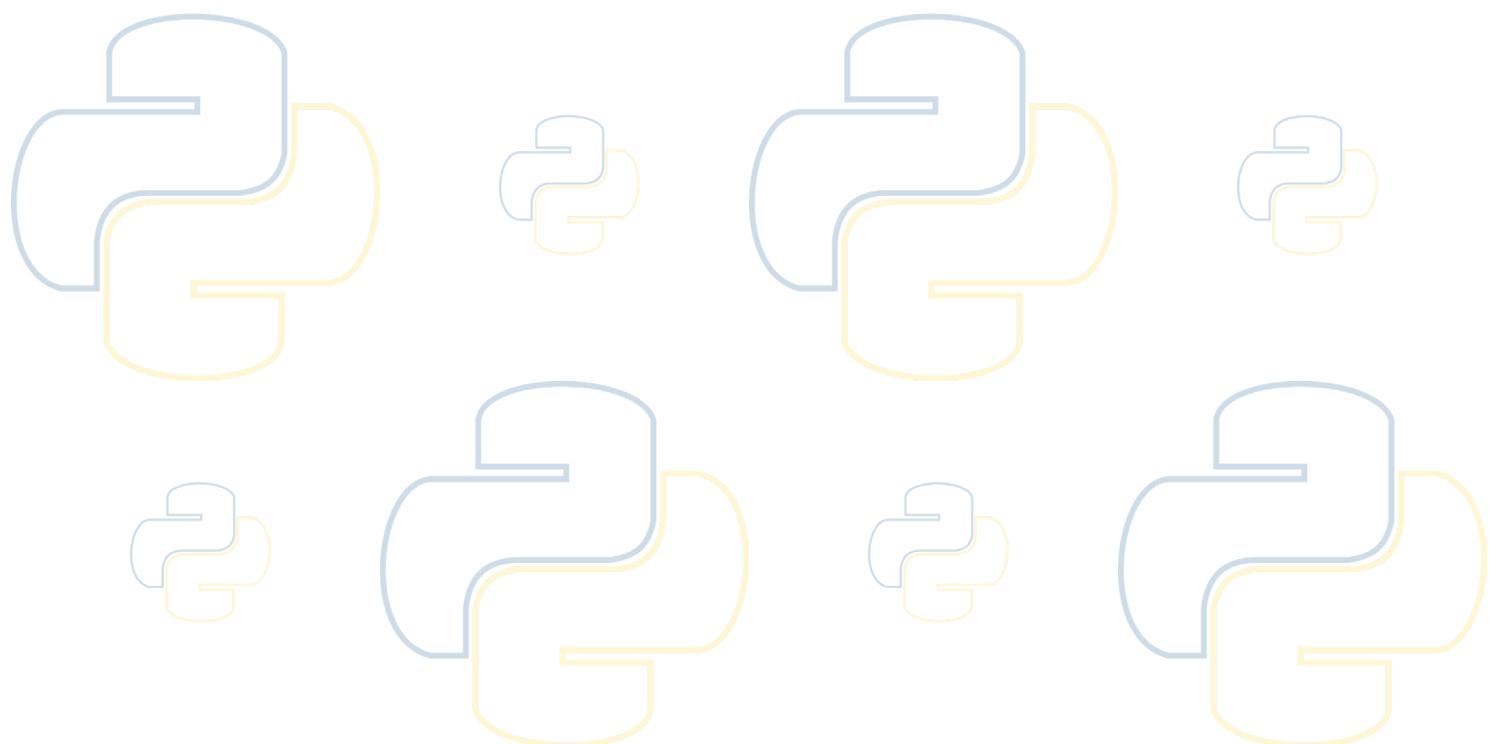
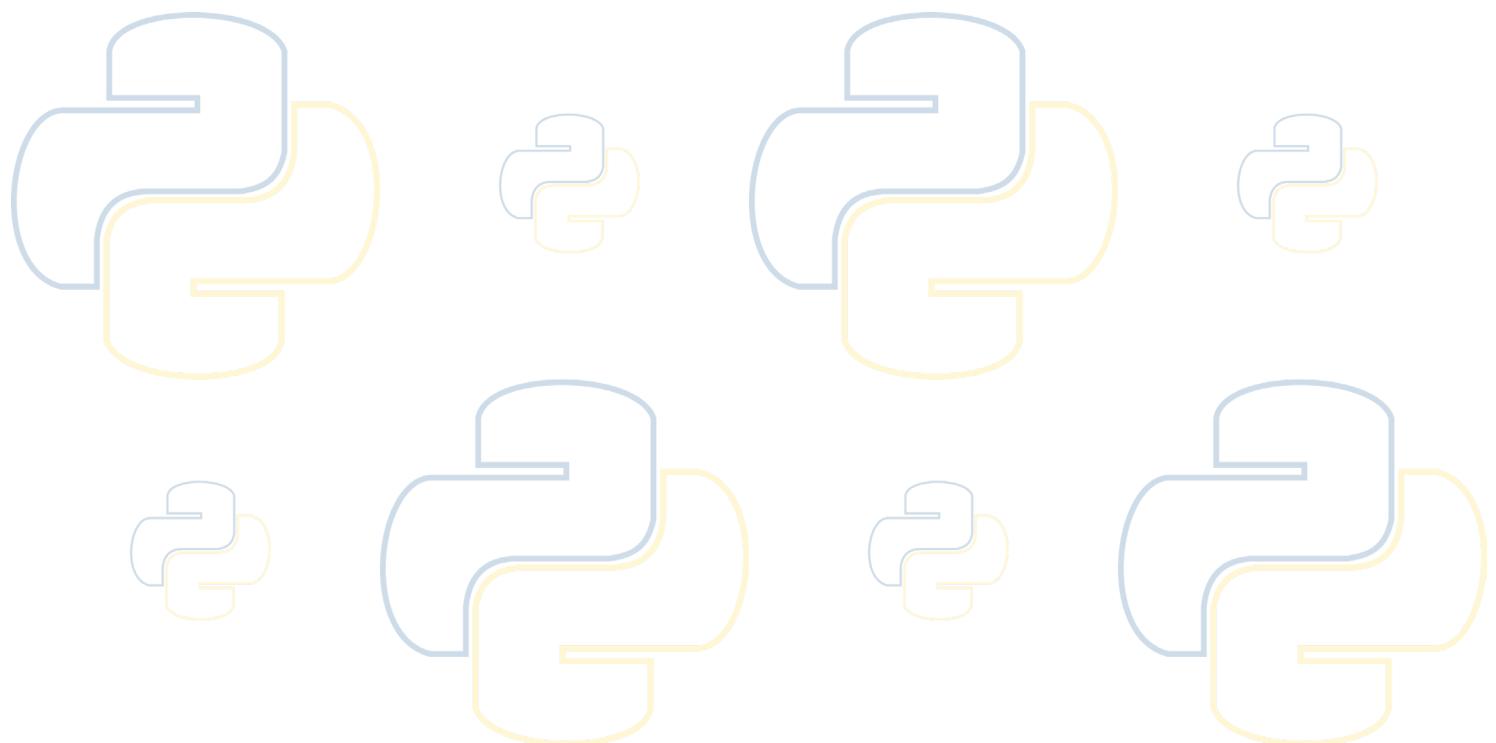
---

3

What is the difference between conditional and conditional loop?

Answer: \_\_\_\_\_

---



## Meeting 5

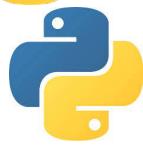
# FUNCTION



## What will we learn today?

1. Introduction of Function Concepts in Python

# INTRO TO FUNCTION



After successfully learning Conditional and Conditional Loops in previous meeting, its time for you to learn Function in Python



Have you ever made the same code over and over and thought about shortening it?

Function  $f(x)$

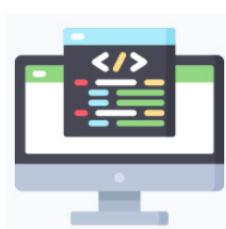
## Watch the following story!

Charlotte asked her mother to make fried eggs

Her mother did the following steps to make fried eggs:

1. Put the pan on the stove
2. Turn on the stove
3. Pour a little oil
4. Crack the egg
5. Give salt and pepper to taste
6. Remove the egg when it is cooked
7. Turn off the stove

Mom, could you please make a fried egg for me?



The command **to make fried eggs** already represents the sequence of activities from **putting the pan in - to turning off the stove**.

Next time Charlotte wants more fried eggs, Charlotte just needs to ask her mother **to make fried eggs** without mentioning the order from start to finish.

# INTRO TO FUNCTION



In coding it is called as **Function**

**Can you mention another example of function?**

**In python, you can compose code and abbreviate it using functions, by labeling the code name and then calling its name on a certain line.**



## VOCABULARY



A function is a command that represents a sequence of code to carry out a specific task and can be used repeatedly.

The coding process can be more efficient because the commands that have been compiled can be called in a single function.

There are 2 types of functions in python, they are:



### Built-in Function

Built-in Functions are the default functions of python that can be used directly without having to make special coding. Such as **print()**, **input()**, etc.



### Manual Function

Manual functions are functions that you create yourself using certain coding and can be named as you want.



# BUILT-IN FUNCTION

# We will learn to use the Built-in Function

1. Open link : <https://replit.com>
  2. Log in with your Google account
  3. Click [+Create repl](#)
  4. Choose Python template, add title **built-in\_function**, then [+Create repl](#)
  5. Type the following code in the text editor to display the value of the largest and the smallest in the list, then press the Run button

```
number=[10, 16, 30, 21]
print(max(number)) - - - - • To display the max value
print(min(number)) - - - - • To display the min value
```

6. Type the following code in the text editor to change the text to uppercase and lowercase letters, then press the Run button.

```
print("hello".upper()) -->  
print("WELCOME".lower()) -->
```

- To change text to uppercase
- To change text to lowercase

7. Type the following code in the text editor to replace the word in variable, then press the Run button

```
message = "I love coding!"  
print(message.replace("coding", "python"))
```

## To replace the word

The word  
you want to  
replace

## Substitute words



# MANUAL FUNCTION



We will learn to create a Manual Function

1. Click +Create repl
2. Choose Python template, add title **manual\_function**, then +Create repl
3. Type the following code in the text editor to create a function to calculate the total price of ordering 1 pizza with a price of \$ 10, then press the Run button



**Manual function coding always starts with def**

```

def order_pizza():
    quantity = 1
    price = 10
    total_order = quantity * price
    print(total_order)

order_pizza()

```

• Coding wrapped in a function

• Call the function

4. If you want to calculate the total price for 7 pizzas automatically you can change the code to be like this, then press the Run button

Variable to store the data of number of pizza

```

def order_pizza(quantity):
    price = 10
    total_order = quantity * price
    print(total_order)

order_pizza(7)

```

The number of pizzas is accommodated into the quantity variable.



# MANUAL FUNCTION

5. Jika kamu ingin membuat perhitungan total harga yang interaktif untuk memesan pizza kamu dapat mengubah code menjadi seperti berikut, lalu tekan button Run

```
def order_pizza(quantity):
    price = 10
    total_order = quantity * price
    print(total_order)

order = input("How many pizzas do you want to order? ")
pizza = int(order)
order_pizza(pizza)
```

The data on the pizza variable is accommodated into the quantity variable.

To change the input data type from string to integer



## CHALLENGE

Try to display the output when ordering 9 pizzas as below:

Your total order for 9 pizza is \$90



Have you ever played quiz with your friends?

What quizzes do you know and often play?

# GUESS THE COUNTRY QUIZ



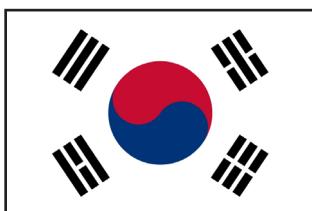
In this special project, you will make Guess The Country Quiz!

Where the user will guess the name of the country according to the clue given

Before that let's play Guess The Flag Quiz!



In this quiz, you will guess 5 names of countries that have flags below, who is the fastest to answer correctly and completely will be the winner!



Country Name:



Country Name:



Country Name:



Country Name:



Country Name:

## Let's make Guess The Country Quiz!

1. Click +Create repl
2. Choose Python template, add title **guess\_country**, then +Create repl
3. Type the following code in a text editor to create a function that can check the quiz answers, if the guess is the same as the answer the quiz then the score will increase by 1, if the guess is wrong then the user does not get a score.

# GUESS THE COUNTRY QUIZ

Guess data variable      Answer data variable

```
def check_guess(guess, answer):
    global score
    if guess.lower() == answer.lower():
        print("Correct answer")
        score = score + 1
    else:
        print("Sorry wrong answer")
```

To avoid errors because the answer contains case-sensitive.

To declare score as a global variable

**Do you remember what the difference between global and local variables is?**

When a variable is inside a function it is usually said to be a local variable and that variable can only be used within that function.

To create a global variable inside a function that can be used outside a function, you can use the keyword **global**.

- Type the following code in the text editor to add questions and the answer on the quiz and also to display the score, then press the Run button.

```
score = 0
print("Guess the Country!")

guess1 = input("By size, what is the largest country in
the world? ")
check_guess(guess1, "russia")
guess2 = input("Which country has a unicorn as its
national animal? ")
check_guess(guess2, "scotland")
guess3 = input("In which country would you find the
currency Baht? ")
check_guess(guess3, "thailand")

print("Your score is " + str(score))
```

# GUESS THE COUNTRY QUIZ



5. To make the user have 3 chances to answer the quiz, and if the opportunity runs out then the correct answer will appear, you can change the code to be like this, then press the Run button.

```
def check_guess(guess, answer):
    global score
    still_guessing = True
    attempt = 0
    while still_guessing and attempt < 3:
        if guess.lower() == answer.lower():
            print("Correct answer")
            score = score + 1
            still_guessing = False
        else:
            if attempt < 2:
                guess = input("Sorry wrong answer. Try again.")
                attempt = attempt + 1
            if attempt == 3:
                print("The correct answer is " + answer)
```



# HOW WAS TODAY'S LEARNING ?



1

What are functions?

Answer: \_\_\_\_\_

---

2

What is the difference between built-in function and manual function?

Answer: \_\_\_\_\_

---

3

Manual function coding always begins with?

Answer: \_\_\_\_\_

---

## Meeting 6

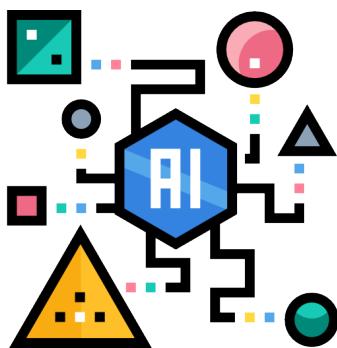
# RANDOM MODULE



## What will we learn today?

1. Introduction of Random Module in Python

# INTRO TO AI & RANDOM



Have you ever played games against AI or bots that can play automatically?

Have you ever been given a recommendation for your favorite video while watching YouTube?

All of these things use Artificial Intelligence technology (Artificial Intelligence)



## VOCABULARY



AI is a technology that can do things automatically that is created by coding.



Have you ever played guessing numbers? When the numbers that appear match your guess, then you win

You can make the game using AI technology with a random module to automatically choose numbers that will appear randomly.

Random



## VOCABULARY



Random is a coding module that can display numbers and words randomly.

# SIMPLE PROJECT



There are 2 types of random in python, they are:



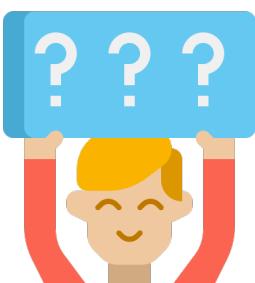
## random.choice(list name / data)

The function of this coding is to randomly select data in lists and strings.



## random.randint([start], [end])

The function of this coding is to randomly select data within a certain range of numbers.



Let's play number guessing with the random module

1. Open link : <https://replit.com>
2. Log in with your Google account
3. Click +Create repl
4. Choose Python template, add title **guessing\_number**, then +Create repl
5. Type the following code in the text editor to display random numbers, then press the Run button

To download the random module which will be used to randomly pick adjectives, nouns, and numbers.

```
import random
number_choice = random.randint(1, 5)
print(number_choice)
```

To take data randomly in the range 1-5

6. Let's play word guessing with the random module!  
Type the following code in the text editor to display random words, then press the Run button

```
word_choice = ["hello", "night", "morning", "good"]
choice = random.choice(word_choice)
print(choice)
```

To take data randomly in the word\_choice list

# PASSWORD MAKER



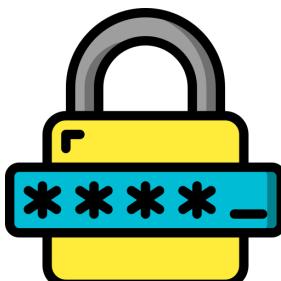
What do you think needs to use a password?

Do you know how to make a good and secure password?



## Let's Learn to Create a Secure Password!

1. Start from a combination of words you remember without spaces
2. Use at least 8 characters
3. Use letters, numbers, and symbols
4. Use a capital letter in one of the letters



## Password Maker

We are going to make a password production engine AI project!

The machine can recommend good and complicated passwords randomly with a combination of letters, numbers, and symbols so that your account is not easily hacked.

1. Click [+Create repl](#)
2. Choose Python template, add title **password\_maker**, then [+Create repl](#)
3. Type the following code in the text editor to download the module and declare a list variable.

```
import random
import string
print("Welcome to Password Maker!")

adjectives = ["sleepy", "slow", "red",
              "orange", "yellow", "green",
              "blue", "purple", "fluffy", "white"]
```

To download the string module that provides symbol characters (punctuation)

Variables to hold adjectives

# PASSWORD MAKER



```
nouns = ["Apple", "Dinosaur", "Ball",
"Goat", "Dragon", "Hammer",
"Duck", "Panda", "Carrot"]
```

Variables to accommodate nouns

- Type the following code in the text editor to create a function that can produce passwords

To provide symbol character data

```
def password_maker():
    adjective = random.choice(adjectives)
    noun = random.choice(nouns)
    number = random.randint(0,100)
    special_char = random.choice(string.punctuation)

    password = adjective + noun + str(number)
    + special_char
    print("Your new password is: " + password)
```

- Type the following code in the text editor to call the function and give the user the option to repeat, then press the Run button

```
while True:
    password_maker()
    response = input("Would you like another password?
Type y or n: ")
    if response == "n":
        break
    elif response != "y" and "n":
        print("What do you mean ?")
        break
```



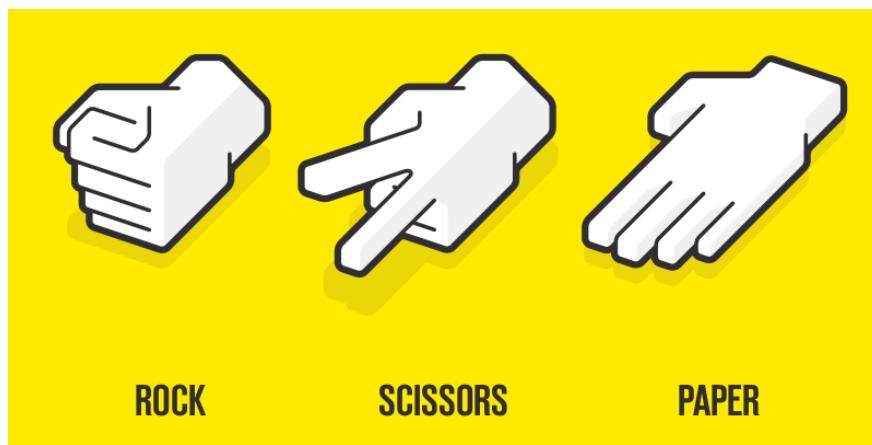
## CHALLENGE

Try adding a list of the names of pokemon/game characters that you like and randomly select the data to add to the password!



# ROCK, SCISSORS, PAPER

## Rock Scissors Paper Game



Do you know  
how to play rock  
paper scissors?

Have you ever  
played it with  
your friends?

### What is the function of coding “\n“ ?

This coding is used to create a new line. When entered into a string, all characters after coding “\n“ are added to a new line. Basically, coding “\n“ indicates that the line ends here and the remaining characters will be displayed on a new line.

We are going to make an AI Rock Scissors Paper Game project!

Users can play rock paper scissors with AI bots as opponents, where rock, scissors, and paper are represented by numbers:

- 1 = Rock (rock)
- 2 = Scissors (scissors)
- 3 = Paper

The AI bot uses the random module to randomly pick 1-3 data to play.

1. Click [+Create repl](#)
2. Choose Python, add title **rock\_scissors\_paper**, then [+Create repl](#)
3. Type the following code in the text editor to download the module random and displays the game guide.

```
print("""Winning Rules of the Rock paper scissor
game as follows:
Rock vs paper->paper wins
Rock vs scissor->Rock wins
paper vs scissor->scissor wins""")
```

# ROCK, SCISSORS, PAPER



4. Type the following code in the text editor to input Answer user and display it, if the answer does not match then the user have to re-enter the answer. Answer users are represented by numbers:

1 = Rock (rock)  
 2 = Scissors (scissors)  
 3 = Paper

Then click Run

```

while True:
    print("""Enter choice:
        1 = Rock
        2 = scissor
        3 = paper""")

    choice = int(input("User turn: "))
    while choice > 3 or choice < 1:
        choice = int(input("enter valid input: "))

    if choice == 1:
        choice_name = "Rock"
    elif choice == 2:
        choice_name = "scissor"
    else:
        choice_name = "paper"

    print("User choice is: " + choice_name)
  
```



# ROCK, SCISSORS, PAPER

5. Type the following code in the text editor to make the computer choose numbers 1-3 randomly as the Answer and display it, if the number selected by the computer is the same as the number selected by the user, the computer will repeat choosing numbers 1-3 randomly once again. Then click Run

```
print("\nNow its computer turn")

comp_choice = random.randint(1, 3)

while comp_choice == choice:
    comp_choice = random.randint(1, 3)

if comp_choice == 1:
    comp_choice_name = "Rock"
elif comp_choice == 2:
    comp_choice_name = "scissor"
else:
    comp_choice_name = "paper"

print("Computer choice is: " + comp_choice_name)
```

6. Type the following code in the text editor to display Answer selected by the user and the computer, as well as to match Answer and determine who wins (whether rock, scissors,papers).

```
print("\n" + choice_name + " V/s " +
comp_choice_name)
```

# ROCK, SCISSORS, PAPER



```

if((choice == 1 and comp_choice == 2) or
(choice == 2 and comp_choice ==1 )):
    print("Rock wins")
    result = "Rock"
elif((choice == 1 and comp_choice == 3) or
(choice == 3 and comp_choice == 1)):
    print("Paper wins")
    result = "Paper"
else:
    print("Scissor wins", end = "")
    result = "Scissor"

```

7. Type the following code in the text editor to determine who wins (whether the user or the computer) and also to ask the user whether to repeat the game or not. Then click Run

```

if result == choice_name:
    print("<== User wins ==>")
else:
    print("<== Computer wins ==>")

response = input("\nDo you want to play
again? (Y/N)")

if response == "n" or response == "N":
    break

print("Thanks for playing")

```

# HOW WAS TODAY'S LEARNING ?



1

What is AI?

Answer: \_\_\_\_\_

---

2

How to code to download random module?

Answer: \_\_\_\_\_

---

3

How to code to take data randomly in the range 5-25?

Answer: \_\_\_\_\_

---

## Meeting 7

# PYGAME



## What will we learn today?

1. Introduction to Pygame

# INTRO TO PYGAME

Did you know that we can make games with the Python programming language?

We're going to make a game using a module called Pygame!



## VOCABULARY



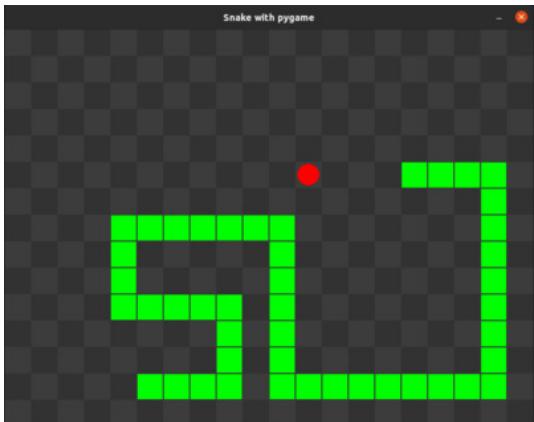
Pygame is a python module used to create games.

With this module, you can add and control game characters, update scores and timers.

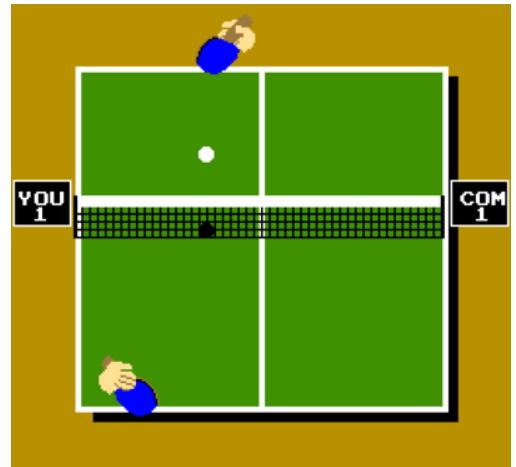


What games can we make with Pygame?

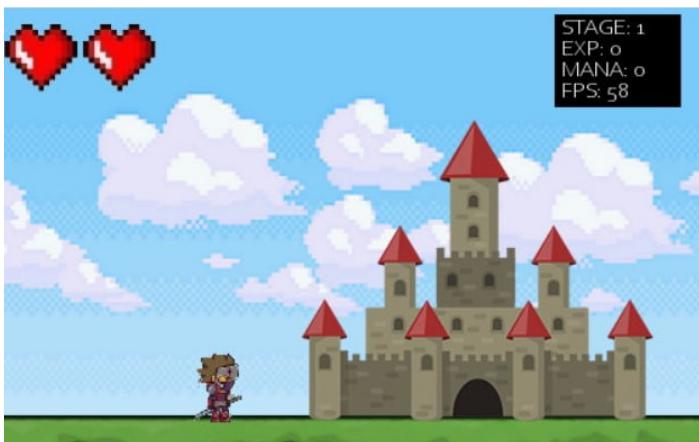
# GAME USING PYGAME



SNAKE GAME



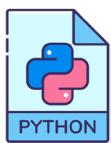
PONG GAME



RPG GAME



DANCE CHALLENGE



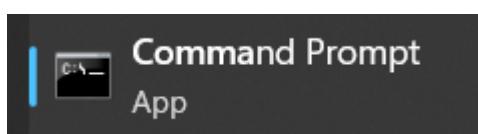
# INSTALLING PYTHON

Now it's time for us to install the Pygame module on the computer so we can code the game.

1. Open link : <https://www.python.org/downloads/>
2. Click Download Python on the yellow button as below



3. Install software Python
  4. Click windows -> open Command prompt
  5. Type the following code at the Command prompt then press Enter to check if pip is available on your computer
- py -m pip --version atau python -m pip --version**



```
Command Prompt
App

C:\ Command Prompt

Microsoft Windows [Version 10.0.22000.434]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kusum>py -m pip --version
```

```
C:\Users\kusum>py -m pip --version
pip 22.0.3 from C:\Users\kusum\AppData\Local\Programs\Python\Python39\lib\
site-packages\pip (python 3.9)
```

6. Type the following code at the Command prompt press Enter to install pygame
- py -m pip install pygame atau python -m pip install pygame**

# INSTALLING PYTHON



```
C:\Users\kusum>py -m pip install pygame
Collecting pygame
  Downloading pygame-2.1.2-cp39-cp39-win_amd64.whl (8.4 MB)
----- 8.4/8.4 MB 1.1 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.1.2
```

7. Type the following code at the Command prompt to install pygame zero

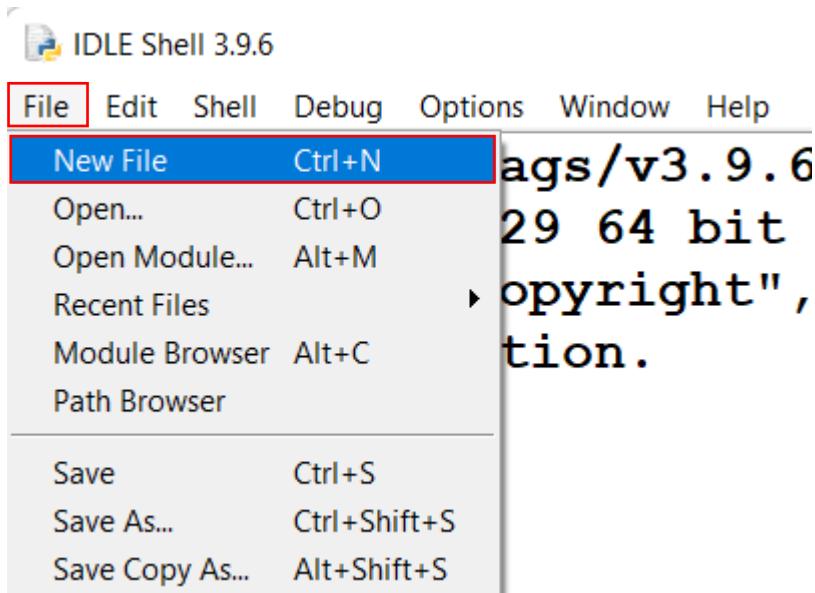
**py -m pip install pgzero** or **python -m pip install pgzero**

```
C:\Users\kusum>py -m pip install pgzero
Collecting pgzero
  Using cached pgzero-1.2.1-py3-none-any.whl (71 kB)
Requirement already satisfied: numpy in c:\users\kusum\appdata\local\programs\python\python39\lib\site-packages (from pgzero) (1.21.2)
Requirement already satisfied: pygame~=2.0 in c:\users\kusum\appdata\local\programs\python\python39\lib\site-packages (from pgzero) (2.1.2)
Installing collected packages: pgzero
  WARNING: The script pgzrun.exe is installed in 'C:\Users\kusum\AppData\Local\Programs\Python\Python39\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pgzero-1.2.1
```

8. Click windows -> search IDLE then click



9. In Python IDLE, click File -> New File to create a document  
New Python





# MONEY COLLECTOR PROJECT

## Money Collector Game



We will create a game project to collect as much money as possible in 10 seconds.

To play the game you can use the keyboard arrows to move the player to earn money.

1. Save the file with the name `money_collector.py`, then type the following code In the text editor to download the pygame zero modules, and the module random is also used to declare variables.

```
import pgzrun
from random import randint

WIDTH = 400
HEIGHT = 400
score = 0
game_over = False
```

Layout height and length in pixels

2. Create a **folder** with the name **images** in the folder that stores the `money_collector.py` file, then copy and paste the `money.png` and `pig.png` files into the images folder. Then type the following code in the text editor to input the sprite file name and sprite coordinate position.

```
pig = Actor("pig")
pig.pos = 100, 100

money = Actor("money")
money.pos = 200, 200
```

To enter the sprite file name

To set the sprite's starting position

3. Type the following code in the text editor to color the screen, displays sprite images and text scores.

# MONEY COLLECTOR PROJECT



To make the screen yellow

```

def draw():
    screen.fill("yellow")
    pig.draw()
    money.draw()
    screen.draw.text("Money: $" + str(score),
                     color="black", topleft=(10, 10))

    if game_over:
        screen.fill("blue")
        screen.draw.text("You got: $" + str(score),
                         topleft=(10, 10), fontsize=60)

```

To display the score

- Type the following code in a text editor to create a sprite position money becomes random.

```

def place_money():
    money.x = randint(20, (WIDTH - 20))
    money.y = randint(20, (HEIGHT - 20))

```

- Type the following code in the text editor to move the player as far as 2 pixels using the keyboard arrow and when the player get money then the score increases by 10 points

```

def update():
    global score
    To declare score variable

    if keyboard.left:
        pig.x = pig.x - 2
    elif keyboard.right:
        pig.x = pig.x + 2
    elif keyboard.up:
        pig.y = pig.y - 2
    elif keyboard.down:
        pig.y = pig.y + 2
    To move the player

```



# MONEY COLLECTOR PROJECT

```

    money_collected = pig.colliderect(money)

    if money_collected:
        score = score + 10
        place_money()
    
```

When the player touch the money

When the player gets money,  
the score increases by 10  
points and the money changes  
positions

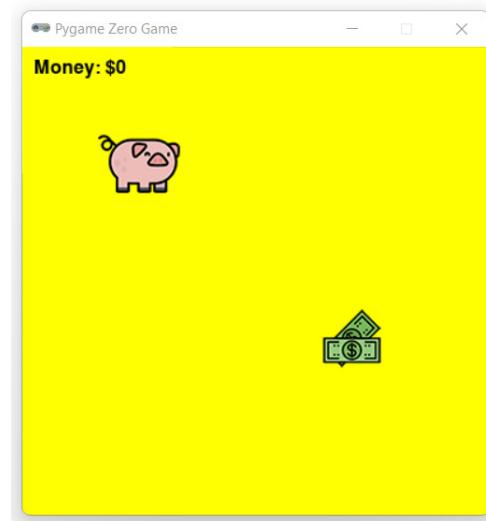
6. Type the following code in the text editor to change the scene into the scene of the game over when the game has lasted for 10 seconds, then press the F5 key on the keyboard.

```

def time_up():
    global game_over
    game_over = True

    clock.schedule(time_up, 10)
    place_money()

pgzrun.go()
    
```



## CHALLENGE

1. Try to make the layout size bigger!
2. Try to make the player move faster!
3. Try increasing game play time into 30seconds!
4. Try changing the sprite player according to the character you want! (you can search for it on google or flat icons)

# HOW WAS TODAY'S LEARNING ?



1

What is Pygame?

Answer: \_\_\_\_\_

---

2

How to code to input sprite file name?

Answer: \_\_\_\_\_

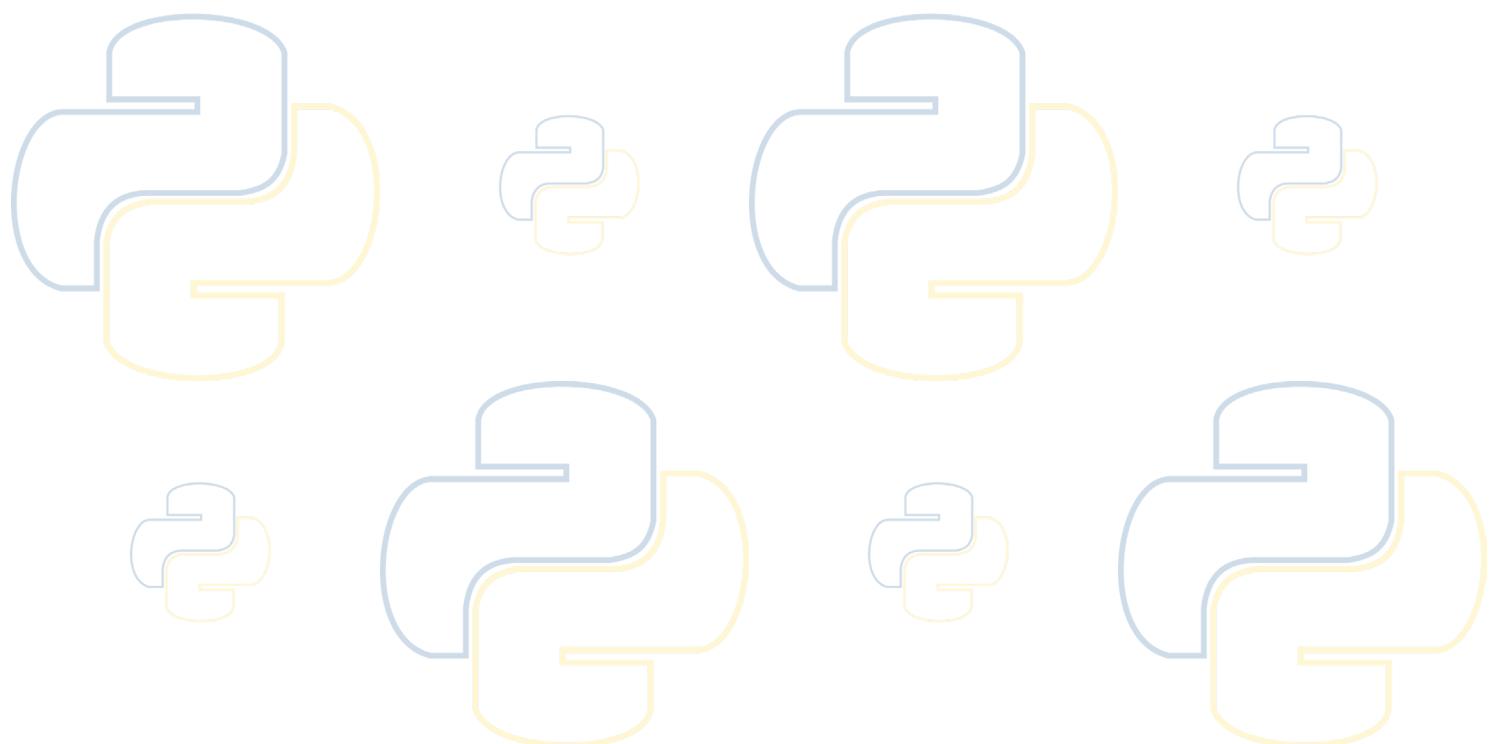
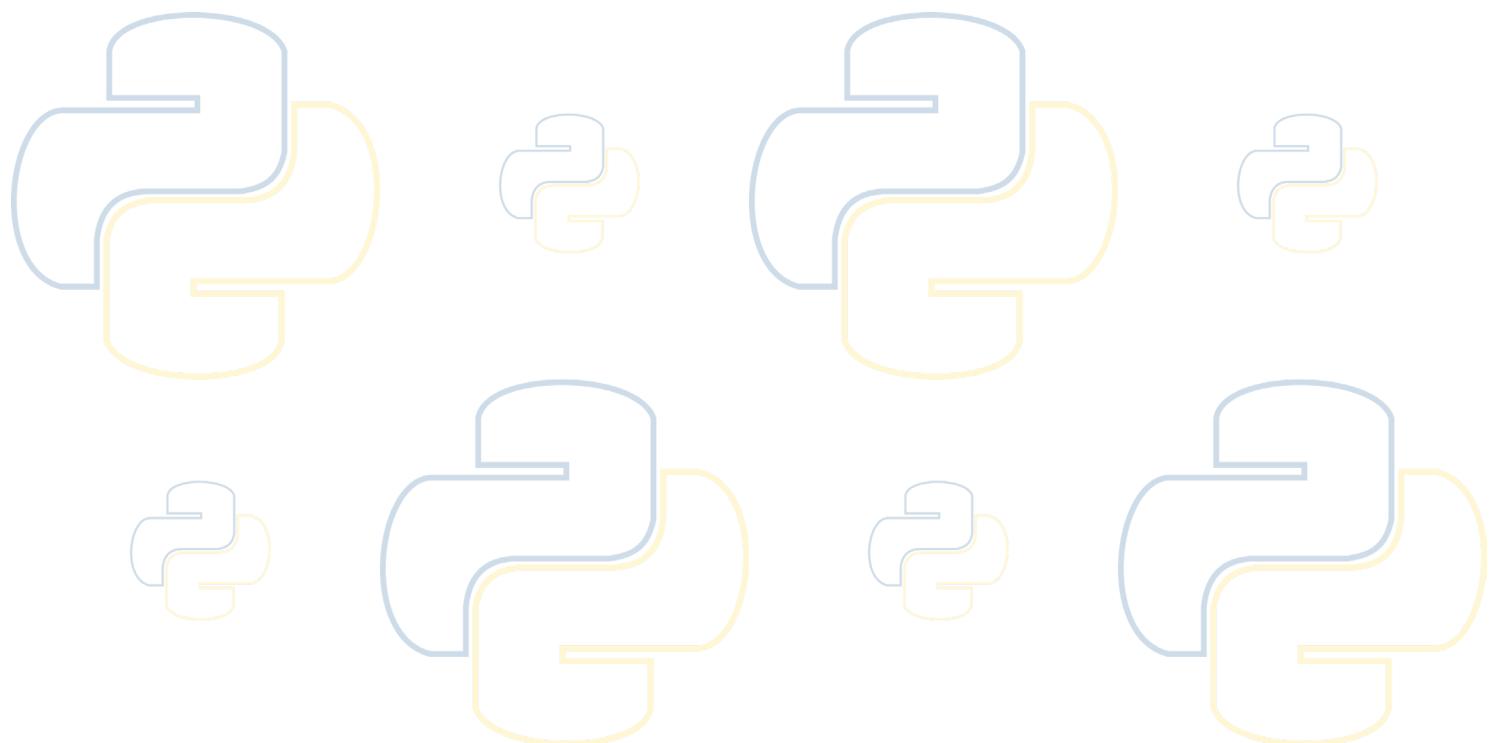
---

3

How to code to make the screen green?

Answer: \_\_\_\_\_

---



## Meeting 8

# EXAM PYTHON BASIC



## What will we learn today?

1. Review Meeting 1-7
2. Exam Python 1

# PYTHON 2

## Meeting 1

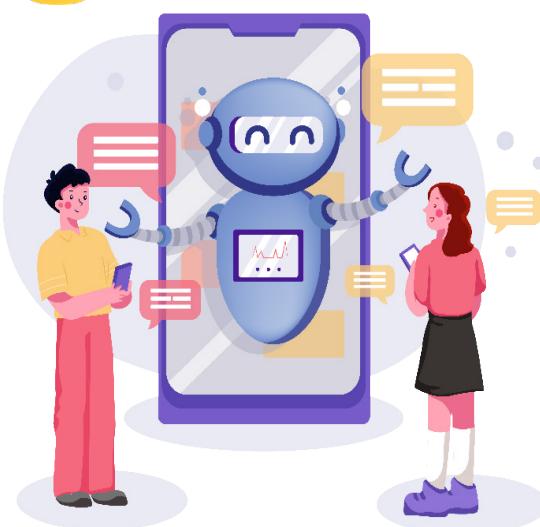
# INTRO TO PYTHON CHATBOT



## What will we learn today?

1. Introduction to Chatbot
2. Learn About Basic Python Chatbot Programming

# INTRO TO CHATBOT



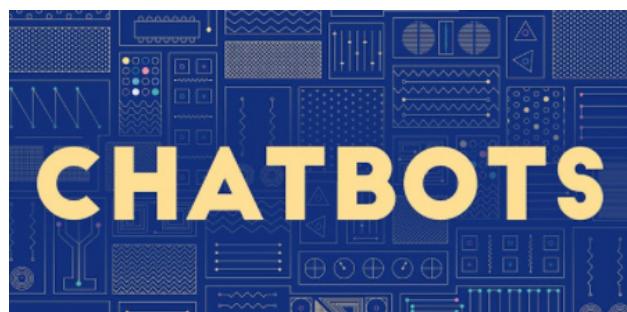
Do you know what a Chatbot is?

Have you used the Chatbot feature before?

What Chatbot features have you used?



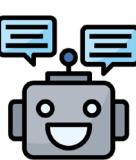
To understand more about Chatbot, let's watch the video below!



Source :  
[Youtube] What are Chatbots?



## VOCABULARY



Chatbots are robots that can communicate like humans through chat.

There are 2 types of Chatbot, namely:



### Programmed Chatbot

A chatbot that can only accept certain commands written in the program. This type of chatbot cannot respond to commands outside the program.

# INTRO TO LIBRARY



## AI Chatbot

A chatbot that uses artificial intelligence technology so that it can understand and respond to all forms of commands easily.



### Who created the world's first Chatbot?



He is Professor Joseph Weizenbaum, the person who created the world's first Chatbot. The chatbot, named Eliza, was released in 1966.

### Do you know what a library is in Python?



#### VOCABULARY



Library is a library (collection) of functions for a specific purpose that can be reused in other projects.

Actually, we have often used libraries when creating Python projects, examples of libraries that we have used are Random, String, and Pygame. Each of these libraries has different uses according to its intended use. At Python 2 level you will use the discord library as the main library in your project.

# MAKE DISCORD ACCOUNT



## Social Media to create a chatbot



Let's guess, what is the name of social media with a logo like a picture beside?  
This social media is often used by gamers!

That's right, this is **Discord**...



### VOCABULARY



Discord is a communication platform for gaming.  
Discord is used as a tool to create content, chatbot servers, to create local groups that focus on games.

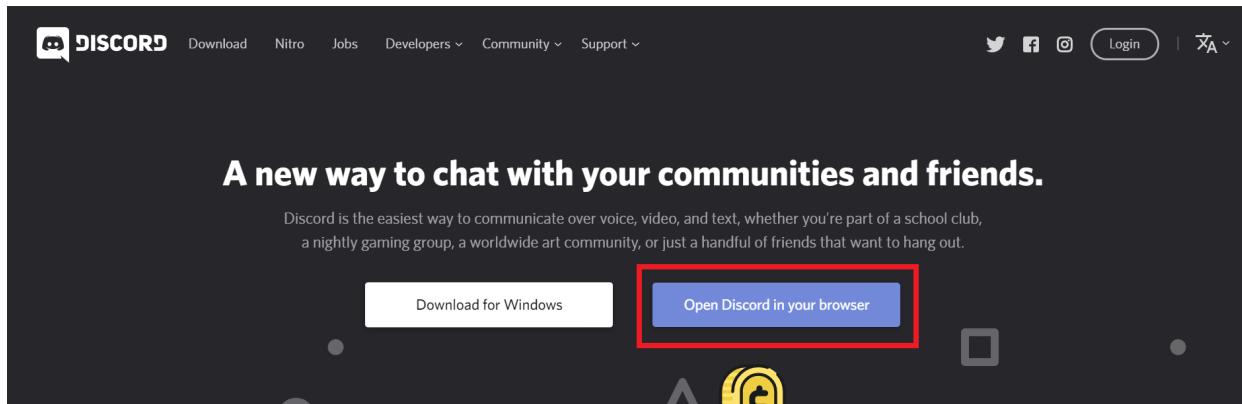
You can download the discord app on your smartphone via the app store or play store



Now it's time for us to create a Discord Account to be able to access the features on Discord.

For those of you who already have an account, please log in directly and don't need to follow the tutorial above!

1. Open link : [discordapp.com](https://discordapp.com)
2. Click Open Discord in your browser

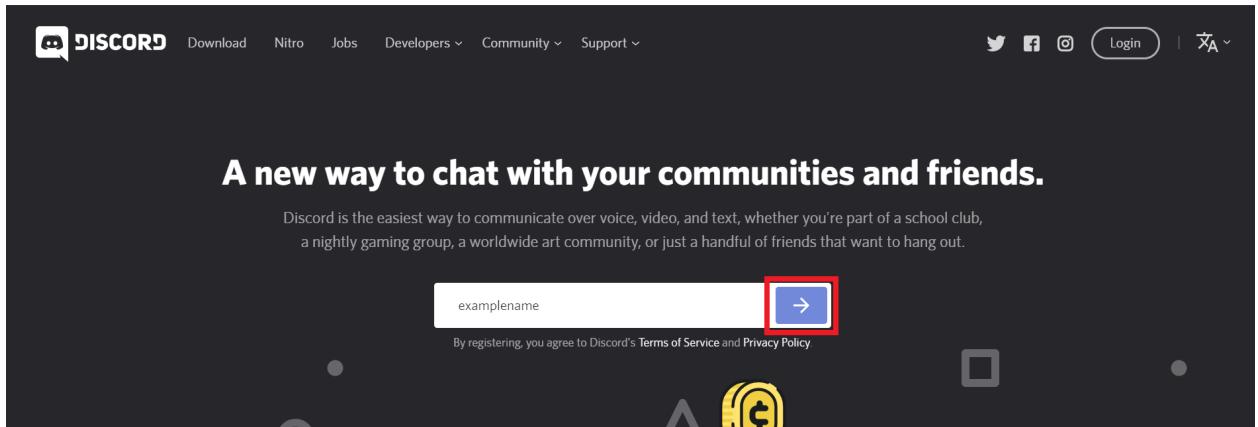


The screenshot shows the official Discord website with a dark theme. At the top, there is a navigation bar with links for "Download", "Nitro", "Jobs", "Developers", "Community", "Support", "Login", and a language selector. Below the navigation, a large heading reads "A new way to chat with your communities and friends." A descriptive paragraph follows, stating: "Discord is the easiest way to communicate over voice, video, and text, whether you're part of a school club, a nightly gaming group, a worldwide art community, or just a handful of friends that want to hang out." Two buttons are present at the bottom: "Download for Windows" and "Open Discord in your browser". The "Open Discord in your browser" button is highlighted with a red border.

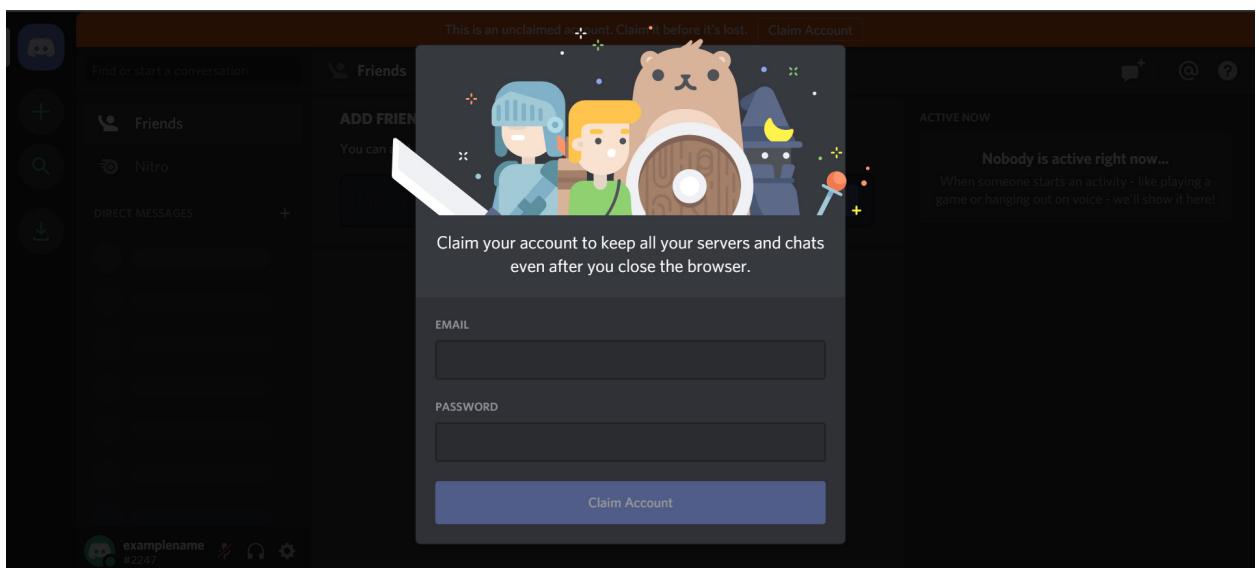
# MAKE DISCORD ACCOUNT



- Enter the username you want and click the arrow.



- Complete the Captcha according to the instructions.
- A Get Started popup will appear giving you the option to create a new server or skip to complete account creation. Click Skip.
- Enter your email and password settings. Then click Claim Account.

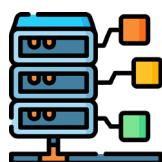


**Hooray!**  
You have successfully created a Discord account!

After this, we will learn to create a Server for Chatbot on Discord!

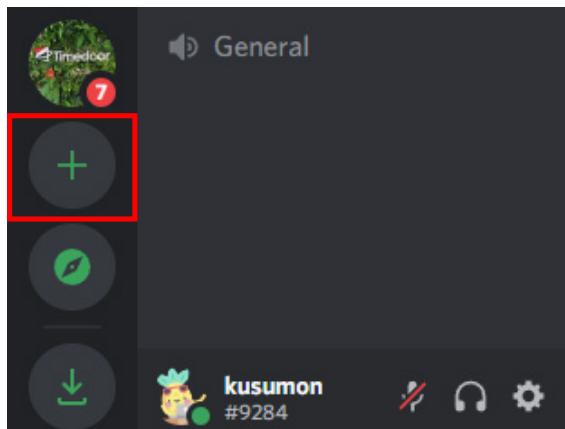


# MAKE DISCORD SERVER



Now it's time for us to create a Discord Server to be able to access the Chatbot feature on Discord.

1. In the lower-left corner of Discord click “+”



2. Click “Create My Own” -> Click “For me and my friends”

**Create a server**

Your server is where you and your friends hang out. Make yours and start talking.

Create My Own >

**Tell us more about your server**

In order to help you with your setup, is your new server for just a few friends or a larger community?

For me and my friends >

3. Name the server with **My Bot Server** and set the image as you wish, then click Create

**Customize your server**

Give your new server a personality with a name and an icon. You can always change it later.

**SERVER NAME**

By creating a server, you agree to Discord's [Community Guidelines](#).

Back
Create

Hooray!  
You have successfully created a  
Discord server!

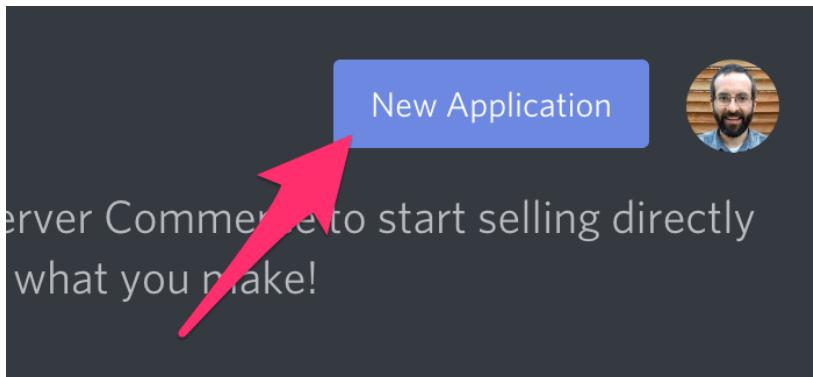


# MAKE DISCORD CHATBOT ACCOUNT

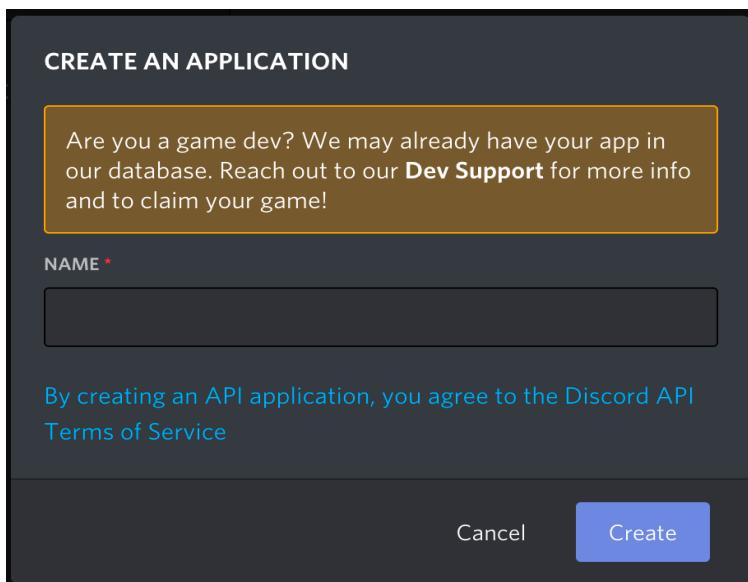


Now it's time for us to create a Discord Chatbot using Discord Developer.

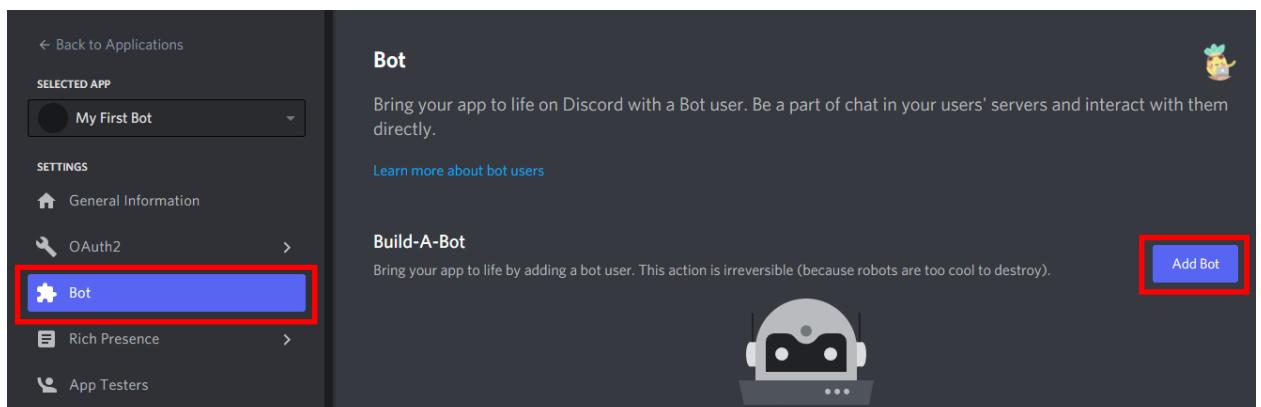
1. Open link : <https://discord.com/developers/applications>
2. Click "New Application" button in the top right corner



3. Name the Chatbot with "**My First Bot**", then click Create

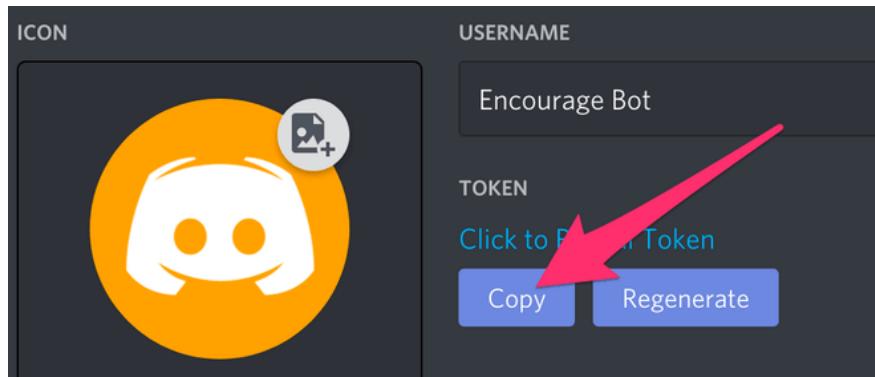


4. Click "Bot" on the tab -> click "Add Bot" -> click "Yes, do it!"

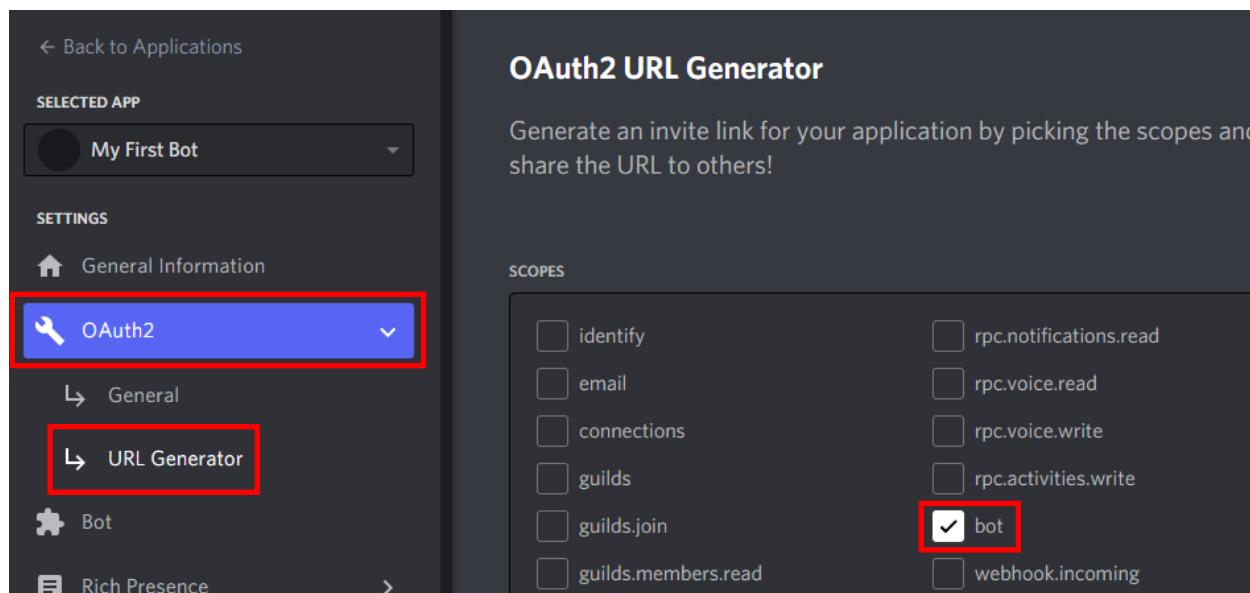


# MAKE DISCORD CHATBOT ACCOUNT

5. Your bot has been successfully created! Now copy the Token and paste it on the notepad



6. Click “OAuth2” pada tab -> click “URL Generator” -> then choose “bot” in “SCOPES”



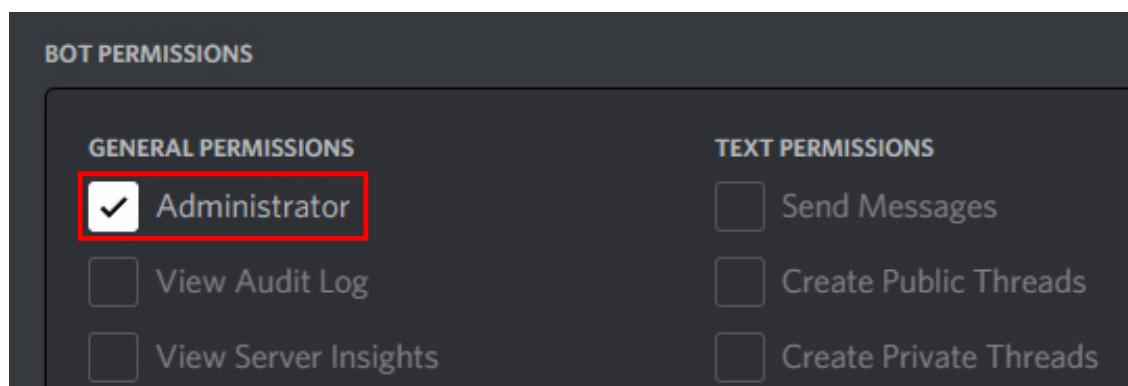
**OAuth2 URL Generator**

Generate an invite link for your application by picking the scopes and share the URL to others!

**SCOPES**

|  |   |
|--|---|
| <input type="checkbox"/> identify            | <input type="checkbox"/> rpc.notifications.read |
| <input type="checkbox"/> email               | <input type="checkbox"/> rpc.voice.read         |
| <input type="checkbox"/> connections         | <input type="checkbox"/> rpc.voice.write        |
| <input type="checkbox"/> guilds              | <input type="checkbox"/> rpc.activities.write   |
| <input type="checkbox"/> guilds.join         | <input checked="" type="checkbox"/> bot         |
| <input type="checkbox"/> guilds.members.read | <input type="checkbox"/> webhook.incoming       |

7. Thick on the “Administrator” in “BOT PERMISSIONS”



**BOT PERMISSIONS**

| <b>GENERAL PERMISSIONS</b>                        |  | <b>TEXT PERMISSIONS</b>                         |  |
|---|--|---|--|
| <input checked="" type="checkbox"/> Administrator |  | <input type="checkbox"/> Send Messages          |  |
| <input type="checkbox"/> View Audit Log           |  | <input type="checkbox"/> Create Public Threads  |  |
| <input type="checkbox"/> View Server Insights     |  | <input type="checkbox"/> Create Private Threads |  |

# MAKE DISCORD CHATBOT ACCOUNT



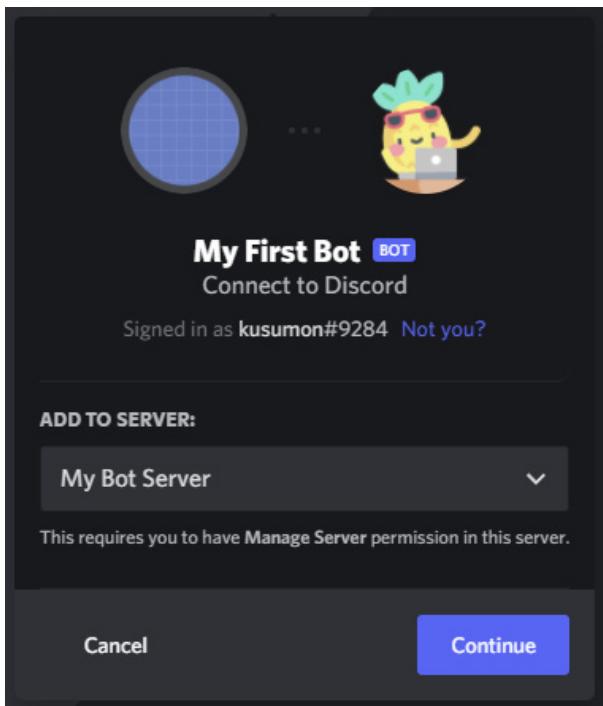
8. On the “GENERATED URL“ click “Copy” -> then paste the url on your web browser

GENERATED URL

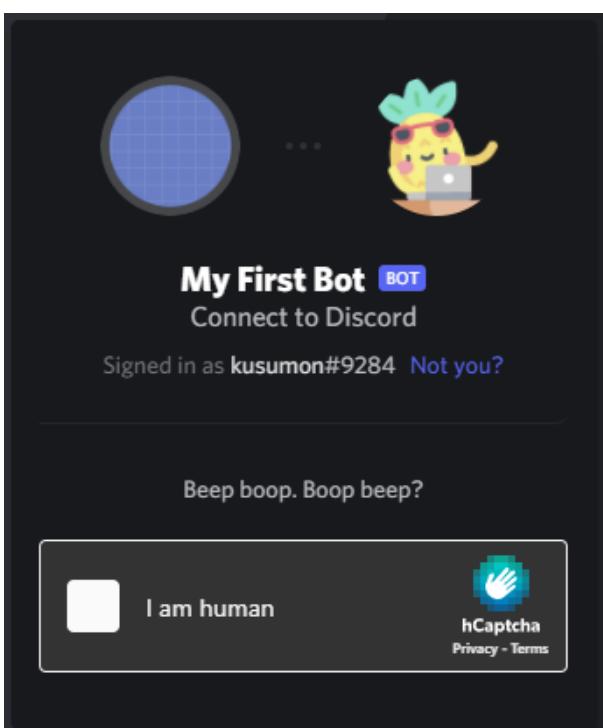
[https://discord.com/api/oauth2/authorize?client\\_id=910784160589832192&permissions=8&scope=bot](https://discord.com/api/oauth2/authorize?client_id=910784160589832192&permissions=8&scope=bot)

Copy

9. On the “ADD TO SERVER“ Choose the server that is “My Bot Server” -> click Continue -> click Authorized



10. Click “I am a human“ -> enter the captcha



Hooray! You have successfully created a Discord server!  
After this, we will learn to program a Chatbot!



# MAKE DISCORD CHATBOT



## What code editor do we use?



Do you still remember about Replit?

Now please open your web Replit and Log in with your Google account, we will get ready to create a Chatbot!



Now it's time for us to learn how to make a Simple Chatbot.

1. Click [+Create repl](#)
2. Choose Python template, add title **my first bot**, then [+Create repl](#)
3. Type the following code in the text editor to download the library Discord and connecting it to discord

```
import discord
```

```
client = discord.Client()
```

To connect to  
Discord

4. Type the following code in the text editor to check if it's a bot ready to use.

To register an event



Called when the bot is ready to use

```
@client.event
async def on_ready():
    print("We have logged in as {0.user}".format(client))
```

5. Type the following code in the text editor to check the message sent by the user and reply to the user's message

Called when the bot receives a message

```
@client.event
async def on_message(message):
    if message.content.startswith("$hello"):
        await message.channel.send("Hello!")
```

To check if Message.content (message sent by the user) starts with "\$hello".

If so, the bot will reply to the message with "Hello".

# MAKE DISCORD CHATBOT

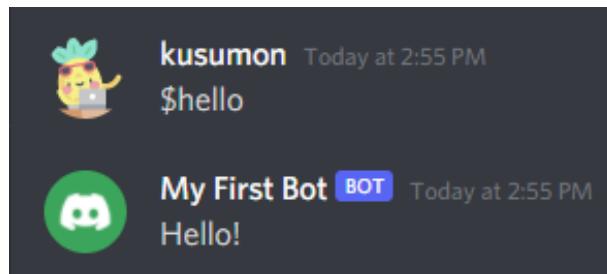


- Type the following code in the text editor to input the Token, then click Run

```
client.run("Your Token")
```

Paste the Token that you copied earlier in this section

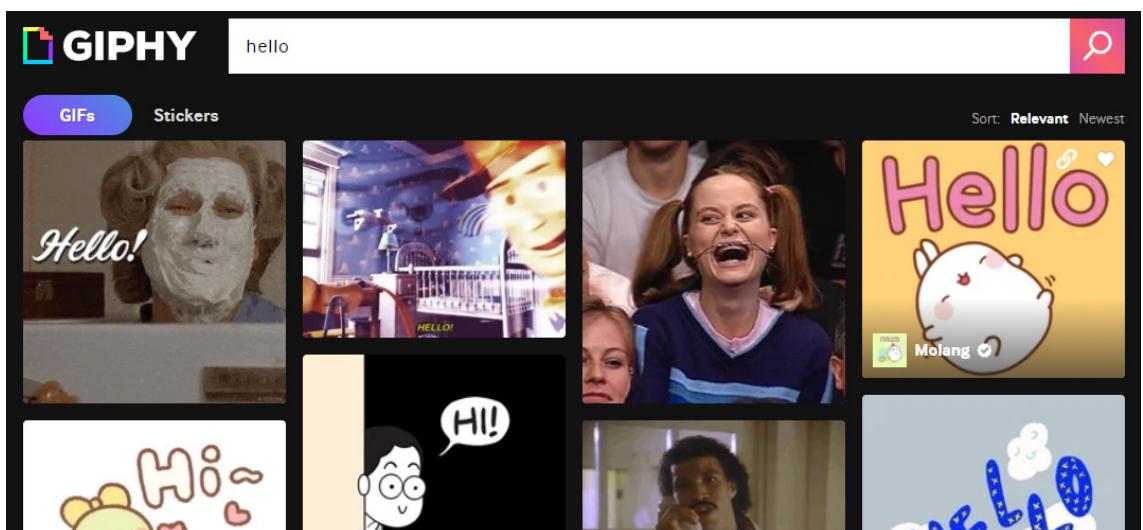
- Reopen Discord in the browser, open the My Bot Server server, then type **\$hello** and send. See what messages reply from the Chatbot that you've created!



- To add an emoji to a message reply, you can copy and paste the emoji from the website: <https://getemoji.com/> We will modify the code in **Line 12** to add paragraphs and emojis on message replies.

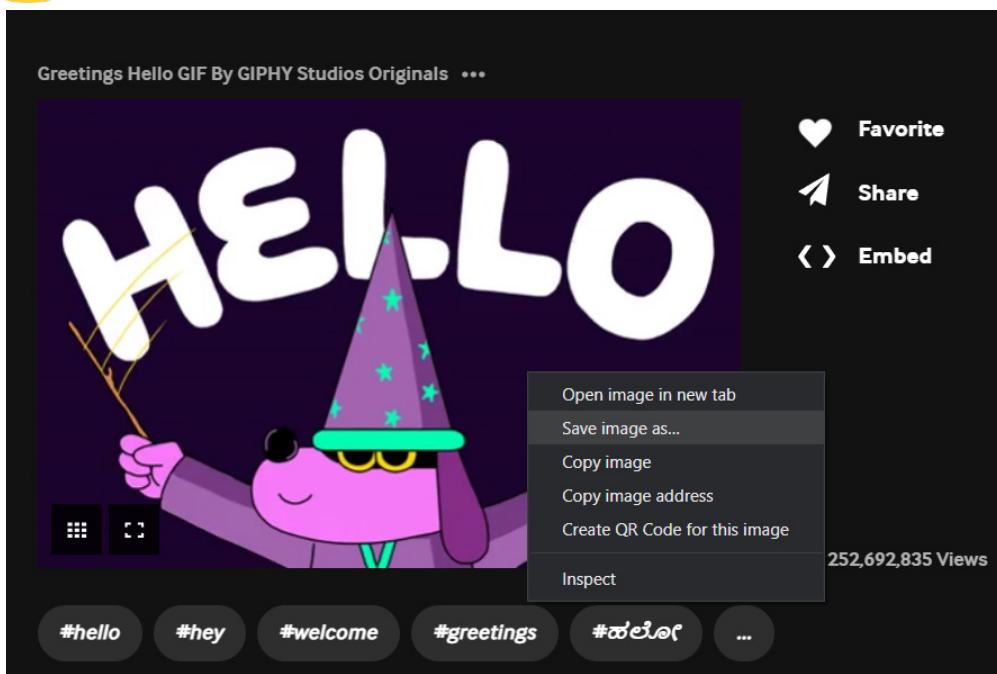
```
await message.channel.send("""Hello!
Welcome to My Bot! 😊👋""")
```

- To send an animated .gif for a message reply, you can download animation from website: <https://giphy.com/>

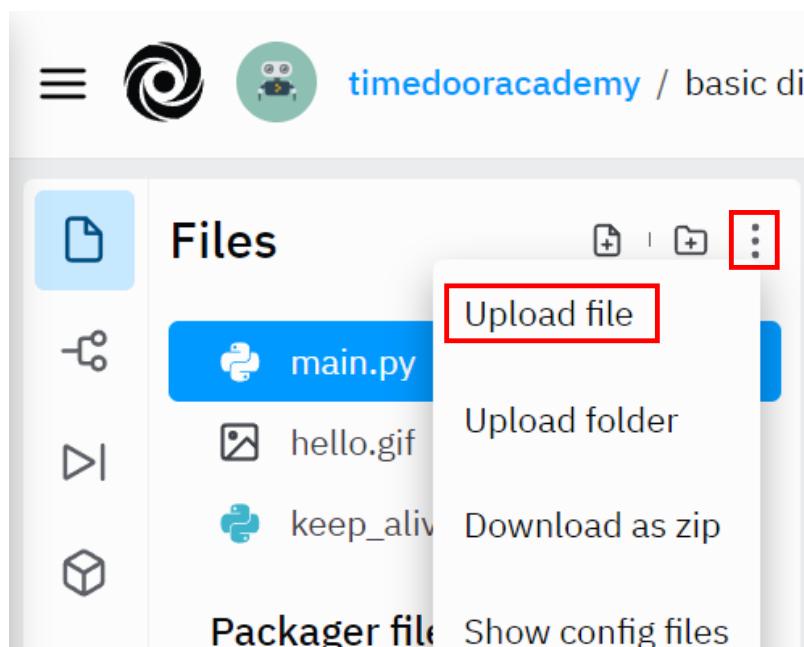


- Click the animated image that you like, then after entering the data page Right click on the animated image -> Click Save image as -> save with the name “hello” and make sure Save as type : GIF Image

# MAKE DISCORD CHATBOT



11. After downloading the animation we will upload the animation into a duplicate folder with a tutorial like the picture below.



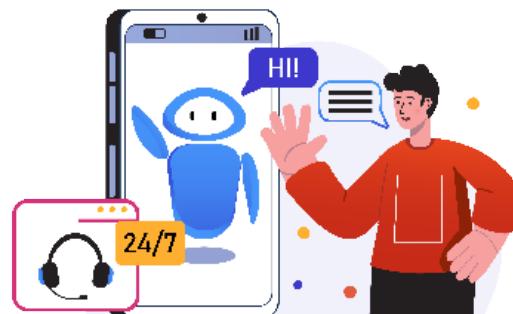
12. The last one, we will add coding to the [line 14](#) to send animated .gif/images to the user. Then click Run and try sending a message **\$hello** on the discord!

```
await message.channel.send(file=discord.File('hello.gif'))
```

# CHALLENGE!



**CHALLENGE**



Try to make your own version of the chatbot!

You are free to use any command and reply message for your chatbot (greeting message, emoji, animation)!

# HOW WAS TODAY'S LEARNING ?



**1** What is Chatbot?

Answer: \_\_\_\_\_

**2** Please mention the types of Chatbot!

Answer: \_\_\_\_\_

**3** Who created the World's first Chatbot? What is the name of the chatbot?

Answer: \_\_\_\_\_

**4** How to code to reply to a user's message with the sentence "Welcome to my server!"?

Answer: \_\_\_\_\_

## Meeting 2

# API & QUOTES CHATBOT



## What will we learn today?

1. Introduction to API
2. Learn to Make a Quotes Chatbot



# INTRO TO API & JSON



Have you ever imagined how a system that can send and receive data works?

Do you know what an API system is?



To understand more about the API, let's watch the video below!



Source :  
[\[Youtube\] What is an API?](#)



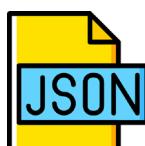
## VOCABULARY



API is a messenger on the website that can receive data requests from users and notify the system, then provide a response according to the request.



## VOCABULARY



JSON is a format with a concise language for storing and transferring human-readable data.

# INTRO TO API & JSON



## The Structure of JSON



API is like a waiter who records food orders and provides a list of food orders to the chef. JSON is like a menu list and a food order list. The system is like a chef who cooks food according to customer orders. Then the waiter gives the food according to the order to the customer.



### Who was created the first API?



They are Maurice Wilkes and David Wheeler, British computer scientists created the world's first API in the 1940s.

# REQUESTING QUOTES DATA

Now it's time for us to make coding to request data quotes in JSON form from an API website.

1. Open link : <https://zenquotes.io/api/random>
2. We will request data quotes and authors in JSON to display on the terminal

**q = quote**

**a = author (the quote's author)**

```
[ {"q": "Honesty is the first chapter in the book of wisdom.", "a": "Thomas Jefferson", "h": "<blockquote>&ldquo;Honesty is the first chapter in the book of wisdom.&rdquo; &mdash; <footer>Thomas Jefferson</footer></blockquote>"} ]
```

3. Open the Replit website and log in with your Google account
4. Click [+Create repl](#)
5. Choose Python template, add title **get\_quotes**, then [+Create repl](#)
6. Type the following code in the text editor to download the library requests and JSON

```
import requests
import json
```

7. Type the following code in a text editor to create a function that can request quotes

To request data from API website

```
def get_quote():
    response = requests.get("https://zenquotes.io/api/random")
    json_data = json.loads(response.text)
    quote = json_data[0]['q'] + " - " + json_data[0]['a']
    print(quote)
```

To display quote and author      To load quote data      To load author data      To load JSON data

8. Type the following code in the text editor to call the function, then click Run!

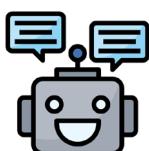
```
get_quote()
```

# QUOTES CHATBOT



Console Shell

```
Sometimes you will never know the value of something, until it becomes a memory. -Dr. Seuss
```



Now it's time for us to learn to make a version of the Chatbot Quotes.

1. Create a Chatbot Discord Account with the name Quotes Bot use the same method as [on the page 7-9](#), after that, join the Chatbot on the same server, with name "My Bot Server"
2. Reopen the Replit website, then Click [+Create repl](#)
3. Choose Python template, add title **quotes\_chatbot**, then [+Create repl](#)
4. Type the following code in the text editor to download the library request, JSON, and discord also for connecting to discord

```
import discord
import requests
import json

client = discord.Client()
```

5. Copy and paste the coding that we made in step 7 earlier on line 7, and change the coding on the last line of "print(quote)" into "return(quote)"
6. Type the following code in the text editor to check if it's a bot ready to use.

```
@client.event
async def on_ready():
    print('We have logged in as {0.user}'.format(client))
```



# PRINT VS RETURN

7. Type the following code in the text editor to check the message sent by the user and reply to the user's message with a quote

```
@client.event
async def on_message(message):
    if message.content.startswith('$inspire'):
        quote = get_quote()
        await message.channel.send(quote)
```

→ To store the result of a function into a variable

8. Type the following code in the text editor to input the token, then click Run and try sending a **\$inspire** message on discord!

```
client.run("Your Token")
```



**Do you know what is the difference between coding print and return in function?**



## print

The function is to display data or variables, but the data or variables cannot be used in functions and other coding parts (for local use only).

**VS**



## return

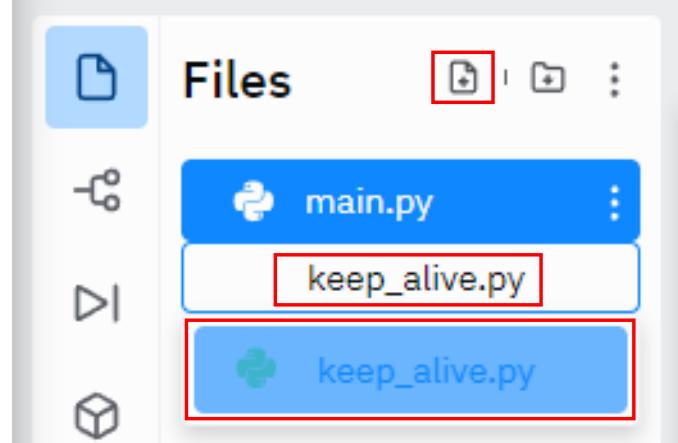
The function is to make data or variables that can be called/used many times in functions and other coding parts (to be used globally).

# SET THE BOT TO RUN CONTINUOUSLY



Now is the time for us to learn how to make the Chatbot run continuously even though the replica website is closed

1. We will add a new python file to the project.  
In the upper left corner of the Replit website, click the "+" icon, then name the file "**keep\_alive.py**" then click.



2. Type the following code in the text editor file **keep\_alive.py** to make a web server that can check the chatbot is still running continuously.

```
from flask import Flask
from threading import Thread

app = Flask('')

@app.route('/')
def home():
    return "Hello. I am alive!"

def run():
    app.run(host='0.0.0.0', port=8080)

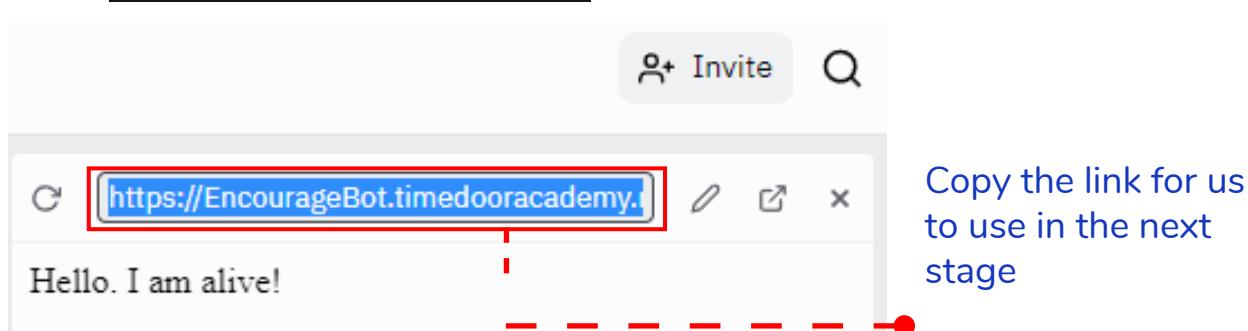
def keep_alive():
    t = Thread(target=run)
    t.start()
```

# SET THE BOT TO RUN CONTINUOUSLY

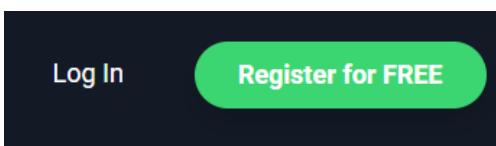
3. Type the following code in the main.py file text editor to connect the web server to the chatbot, then click Run.

```
import discord
import requests
import json
from keep_alive import keep_alive
```

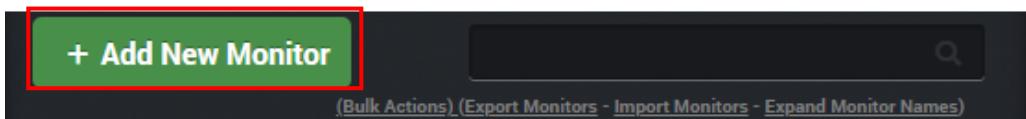
```
keep_alive()
client.run("
```



4. To keep the bot active, use Uptimerobot on <https://uptimerobot.com/>. Robot Uptime can be set to ping the bot's web server on a loop every 5 minutes. With constant ping, the bot will never enter the sleep stage and will just keep going.
5. Open Uptime Robot link -> Click Register for FREE -> register your account by enter your name, email address, and password -> Click I'm not a robot -> Click Register now



6. After successfully logging into your account, click "Add New Monitor".



# SET THE BOT TO RUN CONTINUOUSLY



7. In Monitor Type select HTTP(s)

New Monitor

### Monitor Information

|  |   |
|--|---|
| Monitor Type                               | Please Select<br>Please Select<br><b>HTTP(s)</b>  |
| <b>Select "Alert Contact"</b>              | Keyword<br>Ping<br>Port   |
| Selected: 0 of 1 ( <a href="#">hide</a> ). | <input type="text" value="Search"/>  |

8. Fill in the form according to the data in the red box, then click “Create Monitor”. Try close replica and see the result on Discord !

New Monitor

### Monitor Information

|                     |   |
|---------------------|---|
| Monitor Type        | HTTP(s)   |
| Friendly Name       | Quotes Chatbot  |
| URL (or IP)         | <a href="https://quotes_chatbot.timedooringacademy.repl.co">https://quotes_chatbot.timedooringacademy.repl.co</a> |
| Monitoring Interval | every 5 minutes   |
| Monitor Timeout     | in 30 seconds   |

**Select "Alert Contacts To Notify"**

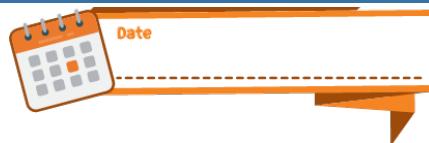
Selected: 0 of 1 ([hide](#)).

| Type                     | Alert Contact                   |
|--------------------------|---------------------------------|
| <input type="checkbox"/> | timedooringacademy.adv@gmail... |

New alert contacts can be defined from the "My Settings" page.

**Create Monitor**

# HOW WAS TODAY'S LEARNING ?



1

What is API?

Answer: \_\_\_\_\_

2 What is JSON?

Answer: \_\_\_\_\_

3

Please mention the analogy of API, JSON, and System!

Answer: \_\_\_\_\_

4

What is the difference between print and return coding?

Answer: \_\_\_\_\_

## Meeting 3

# TRANSLATOR CHATBOT



## What will we learn today?

1. Learn to Make a Chatbot Translator

# INTRO TO GOOGLETRANS LIBRARY



Have you ever used Google Translate to translate words?

What language do you often translate?

Did you know that Python has libraries that can help you translate any words?



## VOCABULARY



Googletrans library is a library that can help us in translating words into many languages.

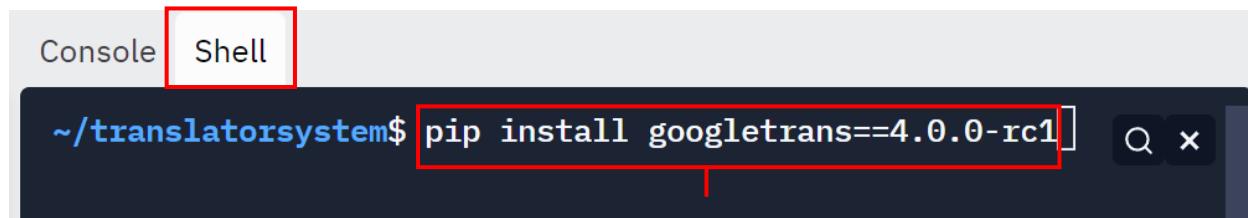


Now is the time for us to learn to use the Googletrans library to create a translator system in Replit!

1. Open the Replit website and Log in with your Google account
2. Click [+Create repl](#)
3. Choose Python template, add title **translator\_system**, then Click [+Create repl](#)
4. Type the following code in the text editor to download Translator from the Googletrans library to translate the word, then click Run

```
from googletrans import Translator
```

5. Because the version of the Googletrans library in the replica is old, then we manually update to the new version of the library via shell with the following code, then press Enter.



```
~/translatorsystem$ pip install googletrans==4.0.0-rc1
```

**pip install googletrans==4.0.0-rc1**

# TRANSLATOR SYSTEM PROJECT



6. Type the following code in the text editor to translate the word, then click Run

```

word = "Bonjour"
translator = Translator()
translate = translator.translate(word, dest="en")
result = translate.text
print(result)

```

To save the translated text into the result variable

The word you want to translate is stored in the word variable

Coding to translate

Translated language code  
en = English



## CHALLENGE

Try to make your own version of the translator system with the language code choices below:

'ar': 'arabic'  
 'en': 'english'  
 'tl': 'filipino'  
 'fr': 'french'  
 'de': 'german'  
 'id': 'indonesian'

'it': 'italian'  
 'ja': 'japanese'  
 'ko': 'korean'  
 'ms': 'malay'  
 'ru': 'russian'

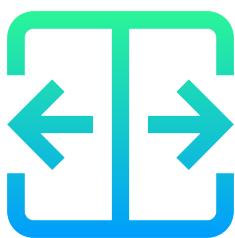


## History of Google Translate



Google Translate is a translation service developed by Google in April 2006. It translates various forms of text and media such as words, phrases, and web pages. Initially, the input text must be translated into English before being translated into the selected language.

# INTRO TO FUNCTION SPLIT()

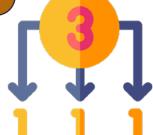


Have you ever split a sentence into word fragments to get information?

We're going to learn to use the `split()` function to split a sentence into fractions of a word!



## VOCABULARY



The `split()` function is a function that can separate or delete the same word or symbol in a sentence and then convert it into a list form.



Now it's time for us to learn to use the `split()` function to separate sentences into word fractions!

1. Click [+Create repl](#)
2. Choose Python template, add title **Split**, then Click [+Create repl](#)
3. Type the following code in a text editor to separate words into form a list using commas and spaces, then click Run

```
txt = "hello, my name is Roy, I am 11 years old"
x = txt.split(", ")
print(x)
```

To separate words

Console    Shell

```
['hello', 'my name is Roy', 'I am 11 years old']
```

4. Type the following code in a text editor to separate words into form a list using the “\$” symbol, then change the form of a list into a string, then click Run

# TRANSLATOR CHATBOT



```
message = "$ Hello my name is Cobe"
text = message.split("$ ")
ListToStr = "" .join(map(str, text))
print(ListToStr)
```

To convert a list to a string

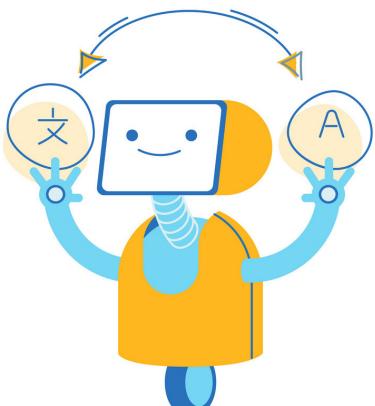
Hello my name is Cobe



**Coding**

**split( )** : Function to separate words or convert strings into list form.

**join( )** : Function to group words in a list or convert a list into strings



Have you ever chatted with people from other countries?

Of course, it's difficult isn't it to understand the contents of the chat because you don't understand the language?

We will learn to make a Chatbot Translator that can translate sentences via chat to make it easier for you to communicate with people from other countries!

1. Create a Chatbot Discord Account with the name Translator Bot using the same method as on pages 7-9, after that join the Chatbot on the same server, with the name "**My Bot Server**"
2. Reopen the Replit website, then Click **+Create repl**
3. Choose template Python, add title **translator\_chatbot**, then **+Create repl**
4. Type the following code in the text editor to download the library discord, googletrans, and textblob. Then click Run



# TRANSLATOR CHATBOT

```
import discord
from googletrans import Translator
from textblob import TextBlob
```

To process textual data

5. Because the version of the Googletrans library in the Replit is old, then we manually update to the new version of the library via shell with coding `pip install googletrans==4.0.0-rc1`, then press Enter button.
6. Type the following code in the text editor to connect to the discord, translate text messages into English, and check if the bot is ready to use.

```
client = discord.Client()

def translate(message):
    translator = Translator()
    translation = translator.translate(message, dest="en")
    return (translation.text)

@client.event
async def on_ready():
    print("We have logged in as {0.user}".format(client))
```

Function to translate message text into English

7. Type the following code in the text editor to check the message sent by the user starting with an exclamation mark, after that separate exclamation marks from words/sentences and translates them by calling the translate function, then replying to the message user with English translation

```
@client.event
async def on_message(message):
    m = TextBlob(message.content)
    print(message.content)
```

To save  
message  
text

To display the message content on the console

# TRANSLATOR CHATBOT



```

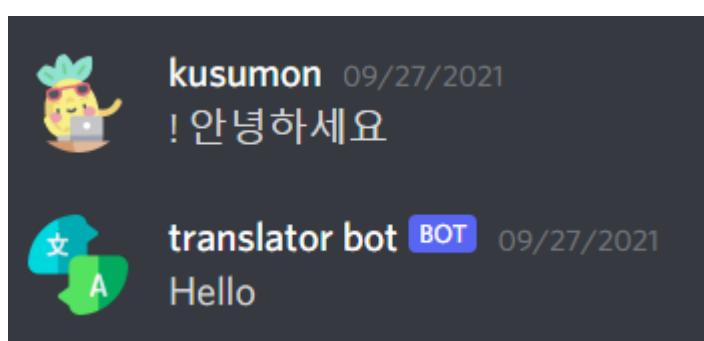
→ if message.content.startswith("!"):
→   text = message.content.split("! ")
→   listToStr = "".join(map(str, text))
→   t = translate(listToStr)
→   await message.channel.send(t)
→ elif m.detect_language() == "en":
→   return
  
```

call the translate function to translate the text

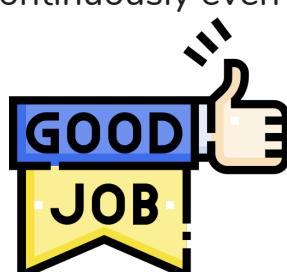
To check whether the message sent by the user is in English. If yes, then the message will reply it based on the word sent because it is already in English, so there is no need to translate it again.

- Type the following code in the text editor to input the Token, then click Run and try to send a message with the format **“! [spasi][the text you want to translate into English]”** in discord!

```
client.run("Your Token")
```



- Make the Chatbot run continuously even though the website Replit is closed!



# TRANSLATOR CHATBOT



Are you tired of making Translator Chatbot?

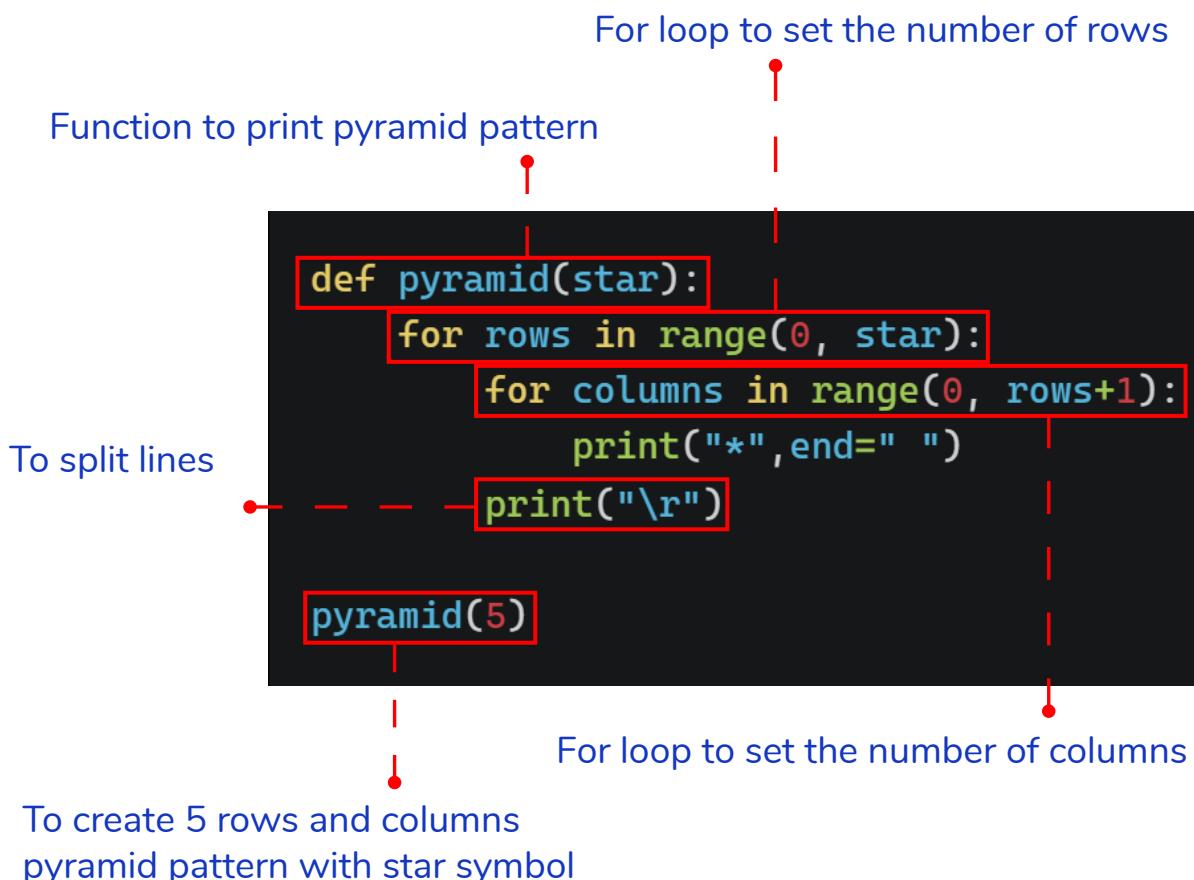
Do you still remember the For Loop concept? What's the coding like?

Let's refresh for a moment by making a Pyramid Pattern using the For Loop concept!



Pyramid Pattern can display a pattern with a pyramid shape using symbols, numbers, and letters.

1. Click +Create repl
2. Choose Python template, add title **Pyramid Patern**, then Click +Create repl
3. Type the following code in the text editor to form a pyramid pattern, then click Run



# TRANSLATOR CHATBOT



Console    Shell

```
* 
* *
* * *
* * * *
* * * * *
> []
```

4. Try to make 20 rows and columns of a pyramid pattern with the symbol “+” !



Console    Shell

```
* * * * *
* * * *
* * *
* *
*
```

How do you get the pyramid to face down?  
Let's try with the following code!

1. Type the following code in a text editor to separate words into form a list using commas and spaces, then click Run

```
def pyramid_down(star):
    for rows in range(star+1, 0, -1):
        for columns in range(0, rows-1):
            print("*", end=" ")
    print("\r")

pyramid_left(5)
```



## CHALLENGE



Try to make a pyramid pattern like the picture below!

# HOW WAS TODAY'S LEARNING ?



1

What is the function of Googletrans library?

Answer: \_\_\_\_\_

\_\_\_\_\_

2

How to code to download Translator from Googletrans library?

Answer: \_\_\_\_\_

\_\_\_\_\_

3

What is the use of Function split() ?

Answer: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

4

```
translate = translator.translate(word, dest="en")
result = translate.text
print(result)
```

What is the function of the code above?

Answer: \_\_\_\_\_

\_\_\_\_\_

## Meeting 4

# DAILY HOROSCOPE CHATBOT



## What will we learn today?

1. Learn to Make a Daily Horoscope Chatbot



# INTRO TO PYAZTRO LIBRARY



Have you read the zodiac predictions?

Do you know what your zodiac sign is?

Did you know Python has a library that can provide daily zodiac forecasts?



## VOCABULARY



**Pyaztro library** is a library that can give us daily horoscope information such as zodiac predictions, Lucky colors, etc.

**The Zodiac** is a symbol of the constellations based on the time of our birth

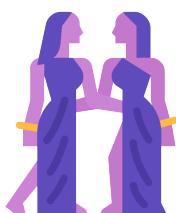
## Zodiac Signs



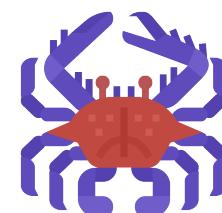
Aries (Ram)  
March 21–April 19



Taurus (Bull)  
April 20–May 20



Gemini (Twins)  
May 21–June 21



Cancer (Crab)  
June 22–July 22



Leo (Lion)  
July 23–August 22



Virgo (Virgin)  
August 23–September 22

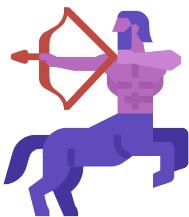


Libra (Balance)  
September 23–October 23



Scorpius (Scorpion)  
October 24–November 21

# DAILY HOROSCOPE CHATBOT



Sagittarius (Archer)  
November 22–  
December 21

Capricornus (Goat)  
December 22–  
January 19



Aquarius (Water Bearer)  
January 20–  
February 18

Pisces (Fish)  
February 19–  
March 20



Now it's time for us to learn to use the Pyaztro library to create a Horoscope chatbot!

1. Create a Chatbot Discord Account with the name **Horoscope Bot** the way is the same as the way on [page number 7-9](#), after that, join the chatbot on the same server that is "**My Bot Server**"
2. Reopen the Replit website, then Click [+Create repl](#)
3. Choose Python template, add title **horoscope\_chatbot**, then [+Create repl](#)
4. Type the following code in the text editor to download the library discord and Pyaztro. Then click Run

```
1 import discord
2 import pyaztro
```

5. Type the following code in the text editor to connect to the discord and check if the bot is ready to use

```
4 client = discord.Client()
5
6 @client.event
7 async def on_ready():
8     print("We have logged in as {0.user}".format(client))
```

6. Type the following code in the text editor to check the zodiac on the message sent by the user, if the message reads Aquarius then the chatbot will reply to the user's message with the date on that day, zodiac prediction description, and zodiac lucky color of the Aquarius.



# DAILY HOROSCOPE CHATBOT

```

10 @client.event
11 async def on_message(message):
12     if message.content.startswith("Aquarius"):
13         horoscope = pyatzro.Aztro(sign="aquarius")
14         quote = "Current date: " + str(horoscope.current_date) + "\n" +
15             "Horoscope: " + horoscope.description + "\n" + "Lucky color: " +
16             horoscope.color
17         await message.channel.send(quote)

```

7. Type the following code in the text editor to input the token, then click Run and try to send a message with the format “**Aquarius**” on the Discord!

```
17 client.run("Copy and Paste Your Token Here")
```



**What if you had to code for the 12 zodiacs?  
Of course, the coding will be very long and tiring right?**

**Do you remember what concepts can be used to shorten coding and can be called many times?**

8. Type the following code in the text editor to create a function detail zodiac forecast

```

10 def detail(astrology):
11     horoscope = pyatzro.Aztro(sign=astrology)
12     quote = "Current date: " + str(horoscope.current_date) + "\n" +
13         "Horoscope: " + horoscope.description + "\n" + "Lucky color: " +
14         horoscope.color
15     return(quote)

```

# DAILY HOROSCOPE CHATBOT



9. To call function detail, we will do a modification on the [line 18](#) as the code below

```
15 @client.event
16 async def on_message(message):
17     if message.content.startswith("Aquarius"):
18         sign = detail("Aquarius")
19         await message.channel.send(sign)
```

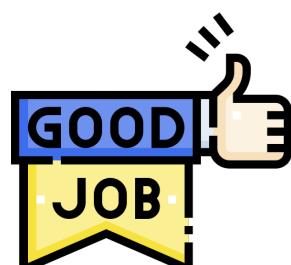
10. Add the code on the [line 21](#) to check the zodiac in the message sent by the user and the chatbot will reply to the user's message with the date on that day, a description of the zodiac forecast, and the lucky color of the zodiac.

```
22 elif message.content.startswith("Taurus"):
23     sign = detail("Taurus")
24     await message.channel.send(sign)
```

11. Add the same coding for the other 10 zodiacs:  
Pisces, Aries, Gemini, Cancer, Leo, Virgo, Libra, Scorpio, Sagittarius, and Capricorn  
12. Add coding like [on page 12 step 12](#) to display the image for each zodiac. You can download the zodiac image file at the following link: <https://bit.ly/31GvyEV>

```
17 if message.content.startswith("Aquarius"):
18     sign = detail('Aquarius')
19     await message.channel.send(sign)
20     await message.channel.send(file=discord.File("Aquarius.png"))
```

13. Click Run and try to send the message of [zodiac](#) on the discord!  
14. Make the Chatbot run continuously even though the Replit website is closed!





Do you remember the Turtle library?

We will play again with the turtle to refresh for a moment.

We will draw stars non-stop!

Open Python IDLE apps then click File -> New File. Type the following code in the text editor, then press the F5 key and see the results!

```
import turtle as t
from random import randint, random

def draw_star(points, size, col, x, y):
    t.penup()
    t.goto(x, y)
    t.pendown()
    angle = 180 - (180 / points)
    t.color(col)
    t.begin_fill()
    for i in range(points):
        t.forward(size)
        t.right(angle)
    t.end_fill()

# Main code
t.Screen().bgcolor('dark blue')
while True:
    ranPts = randint(2, 5) * 2 + 1
    ranSize = randint(10, 50)
    ranCol = (random(), random(), random())
    ranX = randint(-350, 300)
    ranY = randint(-250, 250)
    draw_star(ranPts, ranSize, ranCol, ranX, ranY)
```



# HOW WAS TODAY'S LEARNING ?



1

What is the function of Pyaztro library?

Answer: \_\_\_\_\_

How to code to download the Pyaztro library?

Answer: \_\_\_\_\_

3

horoscope.color

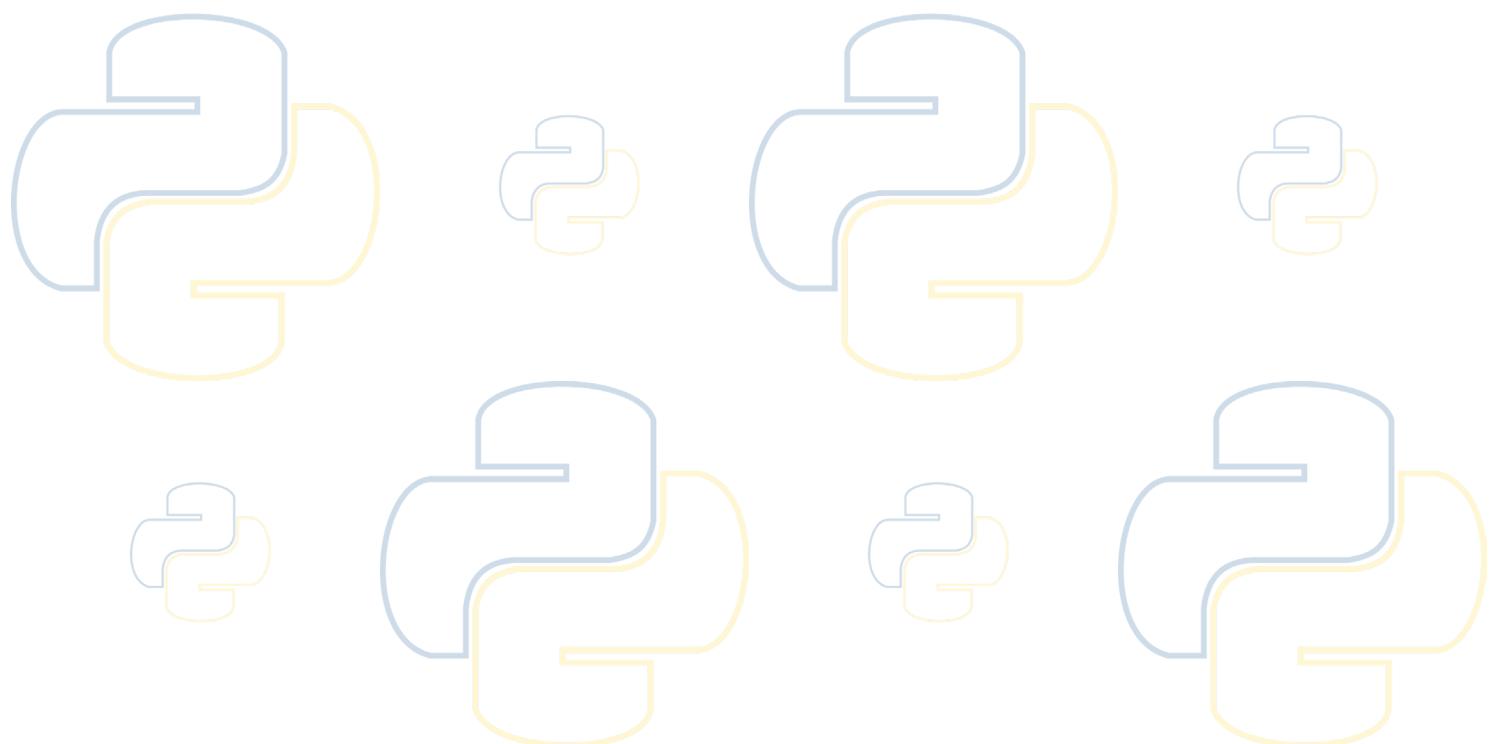
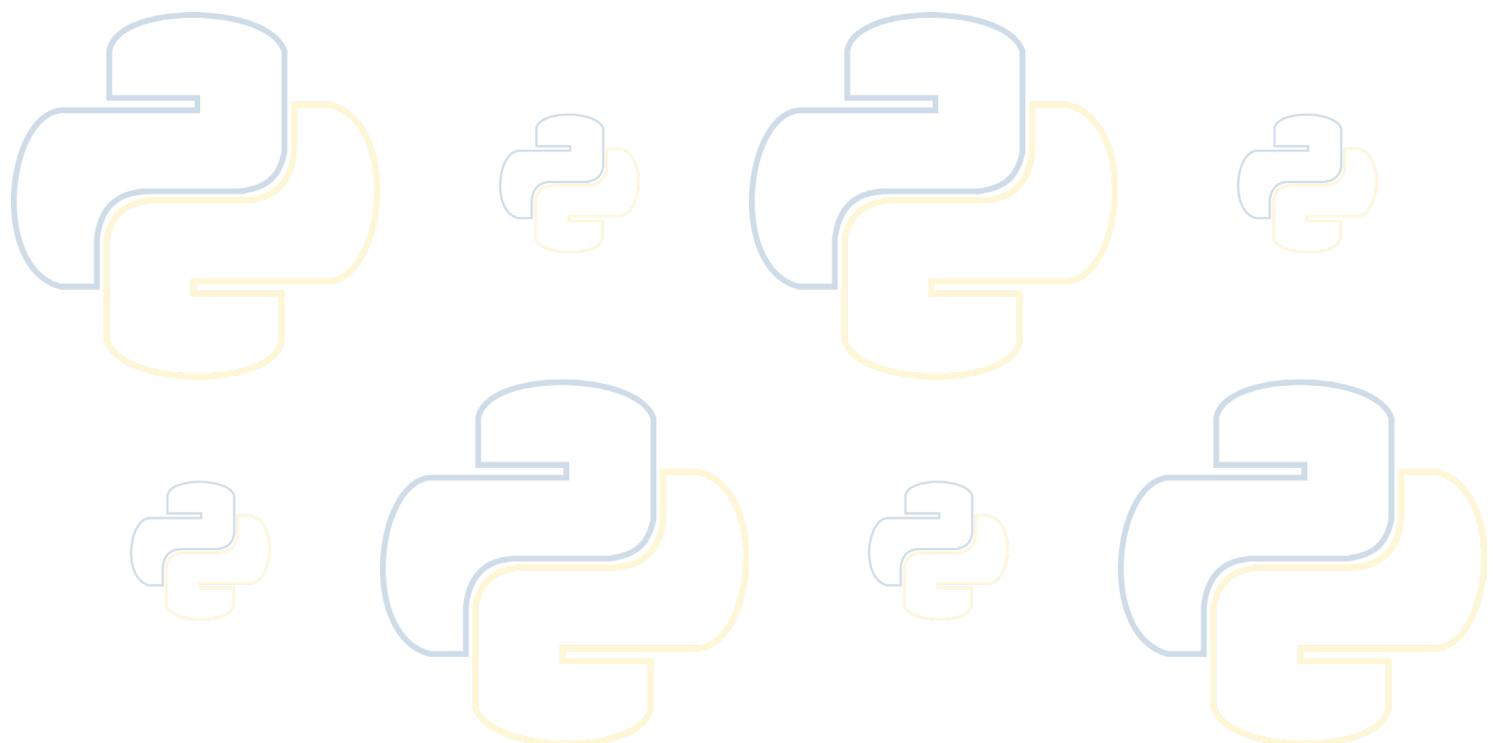
What can the code above display?

Answer: \_\_\_\_\_

3

What is the code to call the function detail of zodiac forecast?

Answer: \_\_\_\_\_



## Meeting 5

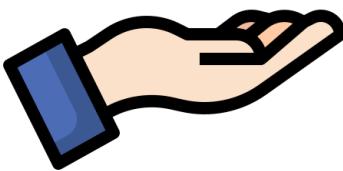
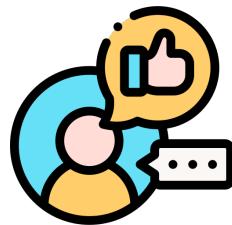
# RECOMMENDER SYSTEM CHATBOT



## What will we learn today?

1. Learn to Make a Recommender System Chatbot

# RECOMMENDER SYSTEM CHATBOT



Do you know what a recommender system is?  
A recommender system is a system used to offer items to users and provide helpful information.

We can create a Recommender Chatbot with just the Discord library!



Now it's time for us to learn to make a Recommender System Chatbot!

1. Create a new server in Discord with the name “Recommender”
2. Create Chatbot Discord Account with the name “Food Recommender Bot” using the same way as [page number 7-9](#), after that, join the chatbot to the same server that is “Recommender”
3. Reopen the Replit website, then Click [+Create repl](#)
4. Choose Python template, add title `food_recommender_chatbot`, then [+Create repl](#)
5. Type the following code in the text editor to download the library discord, connect to discord, and check if Bot is ready to use. Then click Run

```

1 import discord
2
3 client = discord.Client()
4
5 @client.event
6 async def on_ready():
7     print("We have logged in as {}".format(client))

```

6. Type the following code in the text editor to check the message sent by the user, if the message reads “\$hello” then the chatbot will reply to the user’s message with a welcome message containing the tutorial use of chatbots.

# RECOMMENDER SYSTEM CHATBOT

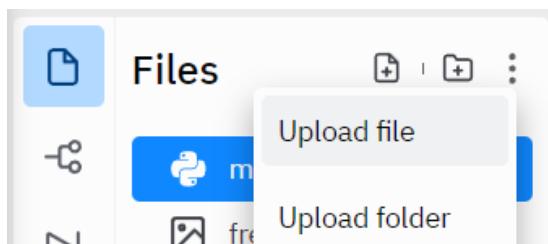


```

9 @client.event
10 v async def on_message(message):
11 v     if message.content.startswith("$hello"):
12         await message.channel.send("""Hello!
13 Welcome to Meal Recommender System
14 $breakfast / $lunch / $dinner
15 1: food / 2: snack
16 Command Example : $breakfast 2""")

```

7. For the image of the food, you could download the file with this link : <https://bit.ly/3lRAdeb> , then upload the food images file in Replit



8. Type the following code in the text editor to check the message sent by the user, if the message reads “**\$breakfast 1**” or “**\$breakfast 2**” then the chatbot will send recommendations food with pictures.

```

18 v     elif message.content.startswith("$breakfast"):
19         text = message.content.split(" $breakfast ")
20         listToStr = " ".join(map(str, text))
21 v         if int(listToStr)==1:
22             choice = ""
23             await message.channel.send(file=discord.File
24                 ('sandwich.png'))
24 v         elif int(listToStr)==2:
25             choice = 'salad'
26             await message.channel.send(file=discord.File
27                 ('salad.png'))
28             recommendation = 'Food recommendation: ' + choice
29             await message.channel.send(recommendation)

```



# RECOMMENDER SYSTEM CHATBOT

9. Type the following code in the text editor to input the Token, then click Run and try to send a message with the format “\$breakfast 1” or “\$breakfast 2” on the discord!

```
30 client.run("Copy and Paste Your Token Here")
```

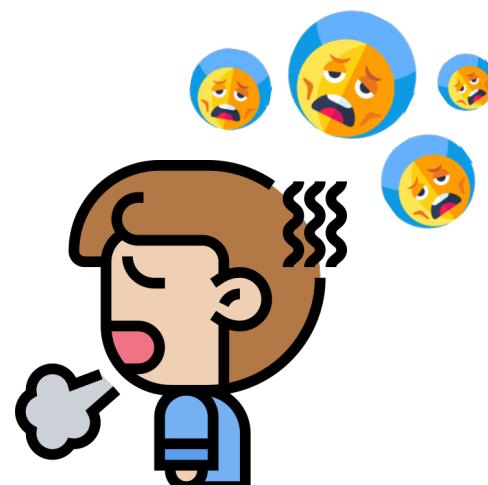
10. Type the code for the lunch and dinner section as in the format in step 8 on line 30, with the food choices as below:

- \$lunch 1 : spaghetti
- \$lunch 2 : tacos
- \$dinner 1 : sushiv
- \$dinner 2 : french fries

```
30 v    elif message.content.startswith("$lunch"):
31         text = message.content.split("$lunch ")
32         listToStr = " ".join(map(str, text))
33 v         if int(listToStr)==1:
34             choice = "spaghetti"
35             await message.channel.send(file=discord.File
36 ("spaghetti.png"))
37         elif int(listToStr)==2:
38             choice = "tacos"
39             await message.channel.send(file=discord.File
40 ("tacos.png"))
41         recommendation = "Food recommendation: " + choice
42         await message.channel.send(recommendation)
```

The coding is very long and tiring to type, isn't it?

Do you have any ideas for shortening the code?



# CHALLENGE



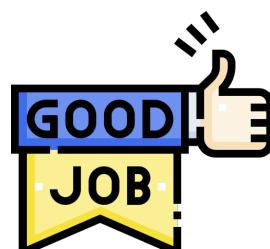
7. To shorten the code you can modify the code has been made like the following format in a text editor!

```

18 v elif message.content.startswith("$breakfast"):
19     text = message.content.split("$breakfast ")
20     listToStr = " ".join(map(str, text))
21 v     if int(listToStr)==1:
22         choice = "sandwich"
23 v     if int(listToStr)==2:
24         choice = "salad"
25     recommendation = "Food recommendation: "+ choice
26     await message.channel.send(recommendation)
27     await message.channel.send(file=discord.File(choice+".png"))

```

9. Click Run and try to send a message based on the format to test your chatbot is on discord!  
 10. Make the Chatbot run continuously even though the Replit website is closed!



## CHALLENGE

Try making your own version of the recommender chatbot with the theme choices below:

- Food
- Games
- Music
- Movies
- Books / Comics / Novels
- Tourist attraction



Don't forget to input images or animations in chatbot message replies!

# HOW WAS TODAY'S LEARNING ?



1 What coding concepts are used the most when creating a Recommender System Chatbot?

Answer: \_\_\_\_\_

\_\_\_\_\_

2 How the code to shorten message replies in the form of images?

Answer: \_\_\_\_\_

\_\_\_\_\_

3 What theme of the Chatbot did you create in your individual project?

What problems did you face during project creation?

Answer: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## Meeting 6

# FOOD ORDERING CHATBOT



## What will we learn today?

1. Learn to Make a Food Ordering Chatbot

# FOOD ORDERING CHATBOT



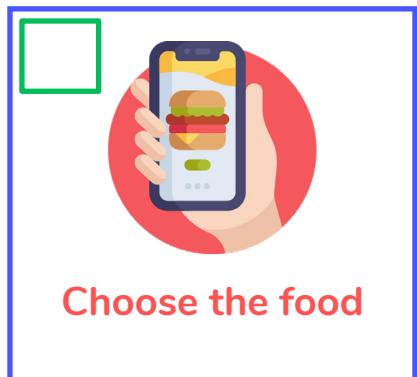
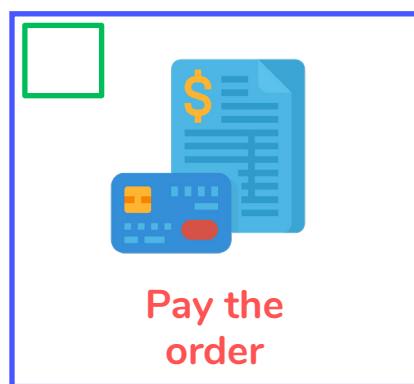
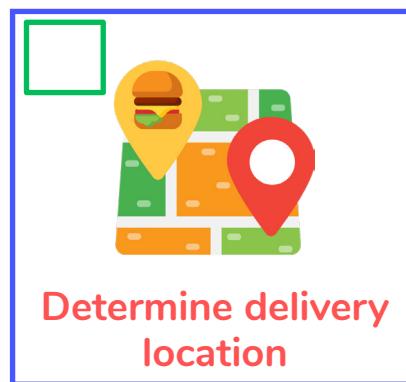
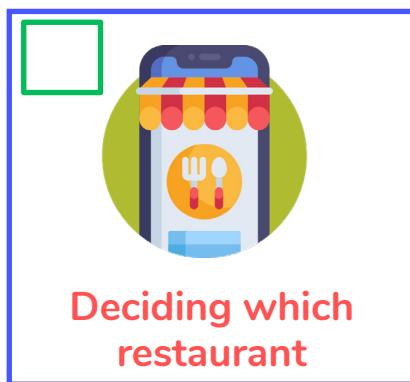
Have you ever ordered food online?

We can create a Chatbot that can serve food orders with only the Discord library!

What are the steps for ordering food online via the Grab / Gojek apps?



Try numbering the green squares in order of how to order food online!



# FOOD ORDERING CHATBOT



Now it's time for us to learn how to make a Food Ordering Chatbot!

1. Create a new server on the Discord with the name **Order**
2. Create the account of Chatbot Discord with the name **Food Ordering Bot** as same as the way on [page number 7-9](#) after that, join the chatbot with the same server that is "**Order**"
3. Reopen the Replit website, then Click [+Create repl](#)
4. Choose Python template, add title **food\_ordering\_chatbot**, then [+Create repl](#)
5. Type the following code in the text editor to download the library discord, connect to discord, and check if the bot is ready to use. Then click Run

```

1 import discord
2
3 client = discord.Client()
4
5 @client.event
6 async def on_ready():
7     print("We have logged in as {}".format(client))

```

6. Type the following code in the text editor to declare global variables.

```

9 food1 = "Pizza"
10 food2 = "Carbonara Pasta"
11 food3 = "Aglio olio Pasta"
12 choice = ""
13 quant = ""

```

7. Type the following code in the text editor to check the message sent by the user, if the message reads "**\$hello**" then chatbot will reply to the user's message with a welcome message.



# FOOD ORDERING CHATBOT

To make the Chatbot does not respond to messages sent by itself

```

15 @client.event
16 v async def on_message(message):
17 v     if message.author == client.user:
18         return
19
20 v     if message.content.startswith("$hello"):
21         await message.channel.send("""Hello!
22 Welcome to Pizza Hub - Delicious Italian food
23 What's your name?
24 Type $name [Your name]""")

```

- Type the following code in the text editor to check the message sent by the user, if the message reads “\$name [namamu]” then the chatbot will reply to the user’s message with a food menu and tutorials on using chatbots.

```

26 v     elif message.content.startswith("$name"):
27         name = message.content.split("$name ")
28         listToStr = " ".join(map(str, name))
29         menu = "Hello "+listToStr+"", What do you want to eat ?
30             1. Pizza           $15
31             2. Carbonara Pasta $10
32             3. Aglio olio Pasta $8
33             Type exp : $food 2"""
34         await message.channel.send(menu)

```

- Type the following code in the text editor to check the message sent by the user, if the message reads “\$food 1” or “\$food 2” or “\$food 3” then the chatbot will send the name of the food selected and ask for the quantity ordered.

# FOOD ORDERING CHATBOT



```

36 v elif message.content.startswith("$food"):
37     global choice
38     food = message.content.split("$food ")
39     text = " ".join(map(str, food))
40 v     if int(text)==1:
41         choice = food1
42 v     if int(text)==2:
43         choice = food2
44 v     if int(text)==3:
45         choice = food3
46     orders = "How many "+choice+"" " you want to order?
47         Type exp : $qty 5"""
48     await message.channel.send(orders)

```

10. Type the following code in the text editor to respond to the message sent by the user, if the message reads “**\$qty [jumlah pesanan]**” then the chatbot will send an order confirmation to the user.

```

50 v elif message.content.startswith("$qty"):
51     global quant
52     qty = message.content.split("$qty ")
53     quant = " ".join(map(str, qty))
54     final_orders = "You want to buy "+quant+" "+choice+"?"
      Are you sure ? (y/n)"
55     await message.channel.send(final_orders)

```

11. Type the following code in the text editor to respond to the message sent by the user, if the message reads “**Y**” then the chatbot will calculate the total purchase send it to the user.

```

57 v elif message.content.startswith("Y"):
58 v     if choice == food1:
59         price = 15
60 v     if choice == food2:
61         price = 10
62 v     if choice == food3:
63         price = 8

```



# FOOD ORDERING CHATBOT

```

64     qt = int(quant)
65     total = price * qt
66     await message.channel.send("Your total bill is
$"+str(total)+" Thank you for shopping!")

```

- Type the following code in the text editor to respond to the message sent by the user, if the message reads “N” then the chatbot will send a goodbye message to the user. If the message sent by the user cannot be understood by the chatbot then the chatbot will send a message that the chatbot does not understand the message.

```

68 v   elif message.content.startswith("N"):
69     await message.channel.send("Ok good bye")
70
71 v   else:
72     await message.channel.send("Sorry, I don't
understand")

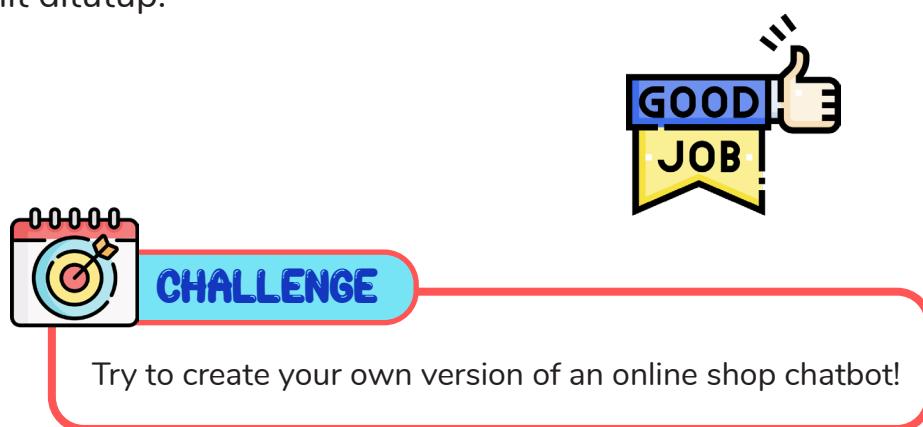
```

- Type the following code in the text editor to input the Token, then click Run and try sending a message on discord!

```
74 client.run("Copy and Paste Your Token Here")
```

\$hello -> \$name [Your name]  
-> \$food [choice] -> \$qty [amount] -> Y

- Buat agar Chatbot dapat berjalan terus menerus meskipun website Replit ditutup!



# HOW WAS TODAY'S LEARNING ?



1

What coding concepts are used when creating a Food Ordering Chatbot?

Answer: \_\_\_\_\_

\_\_\_\_\_

2

What variables do we use in creating the Food Ordering Chatbot?

Answer: \_\_\_\_\_

\_\_\_\_\_

3

What is the order in ordering food using the command on discord?

Answer: \_\_\_\_\_

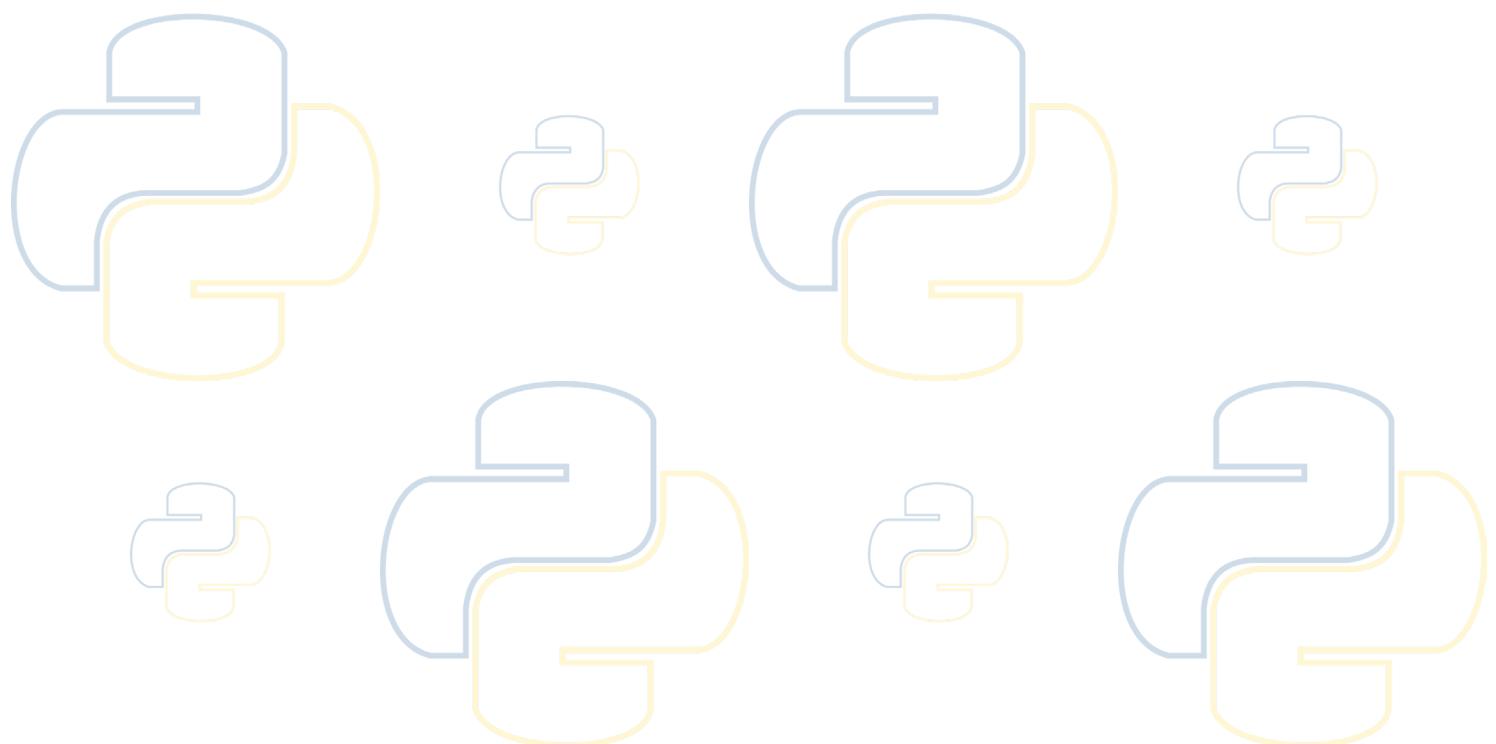
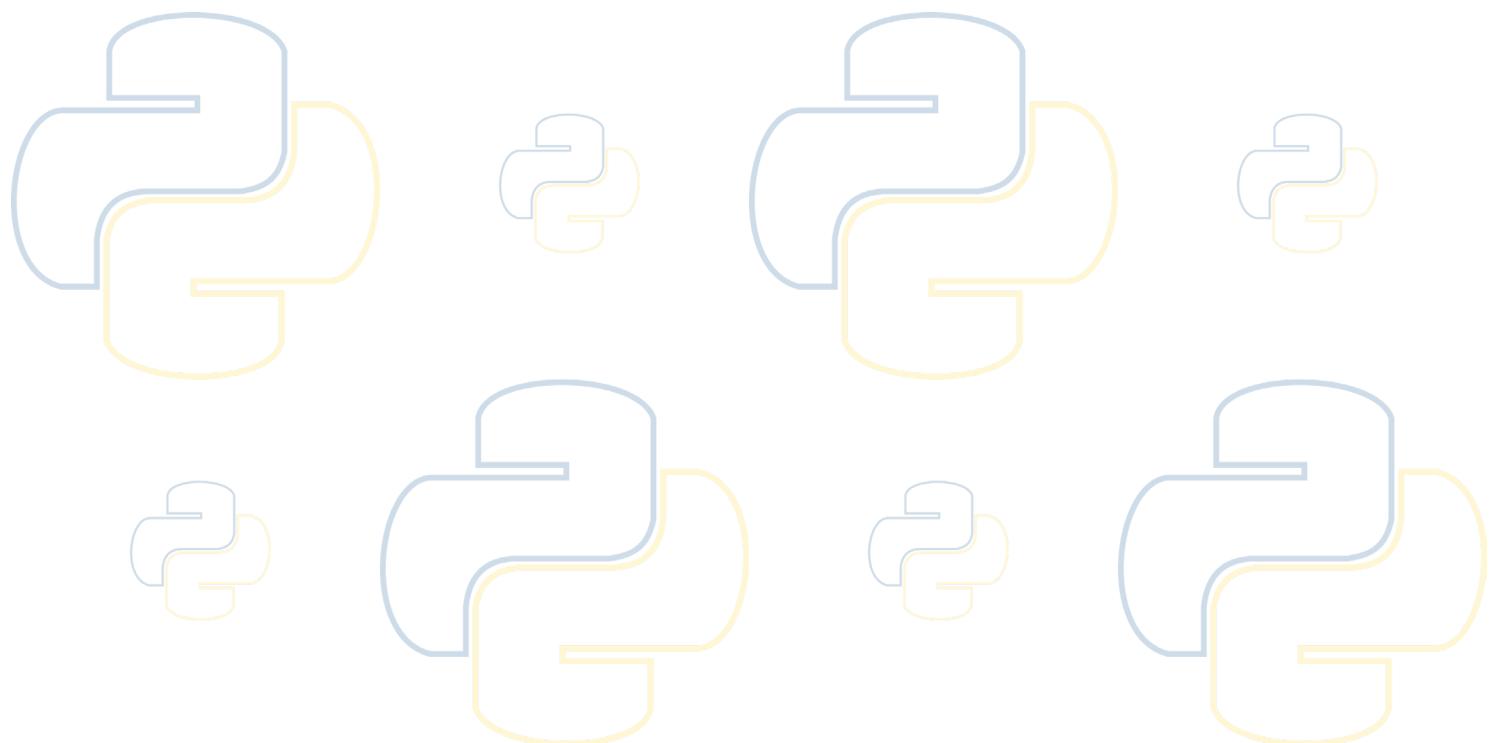
\_\_\_\_\_



## Announcement

At the next 7th and 8th meetings, you will create an exam project in the form of your own ChatBot based on your creation!

Please, start looking for ideas to make your exam project. Your exam project must have the theme of a recommendation system or an online shop (ordering system). So prepare yourself well for the project!



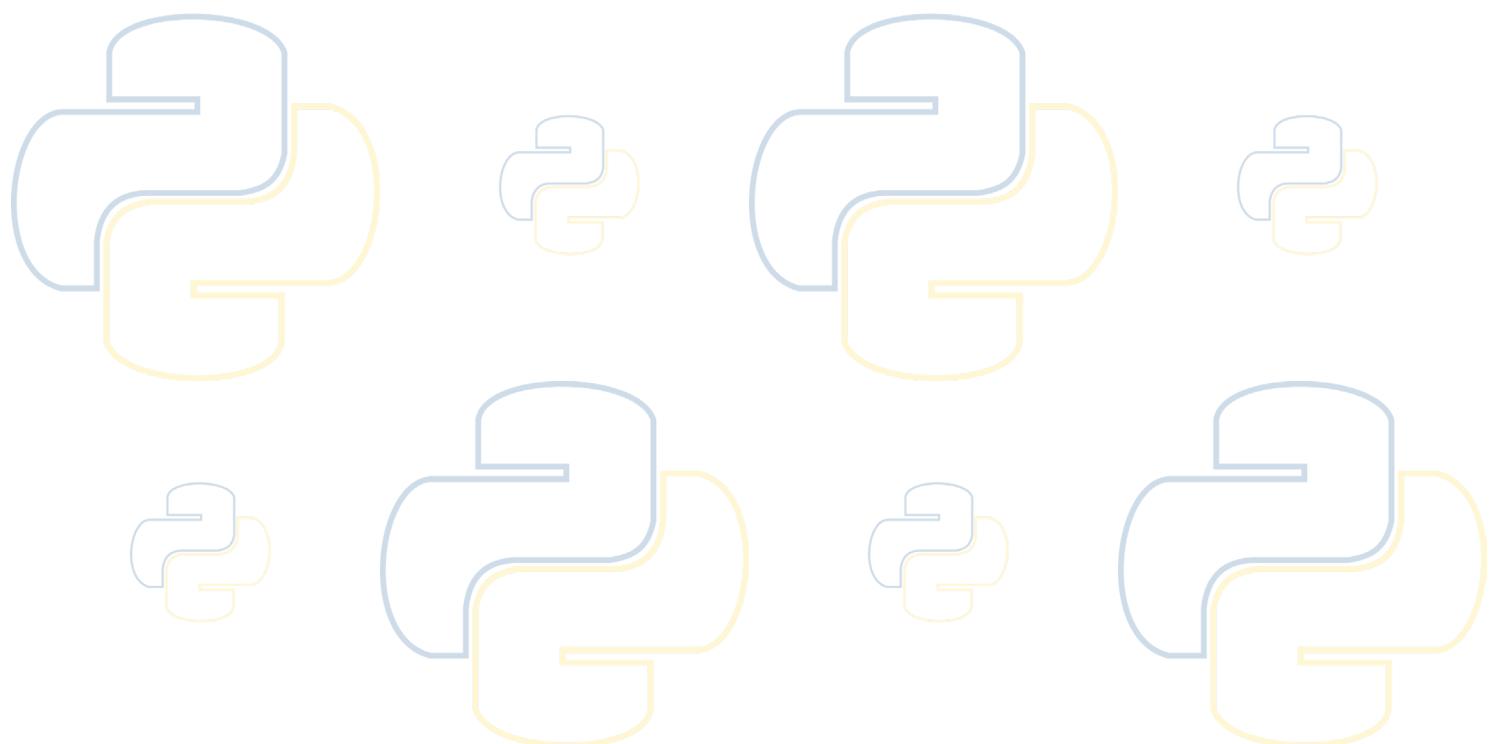
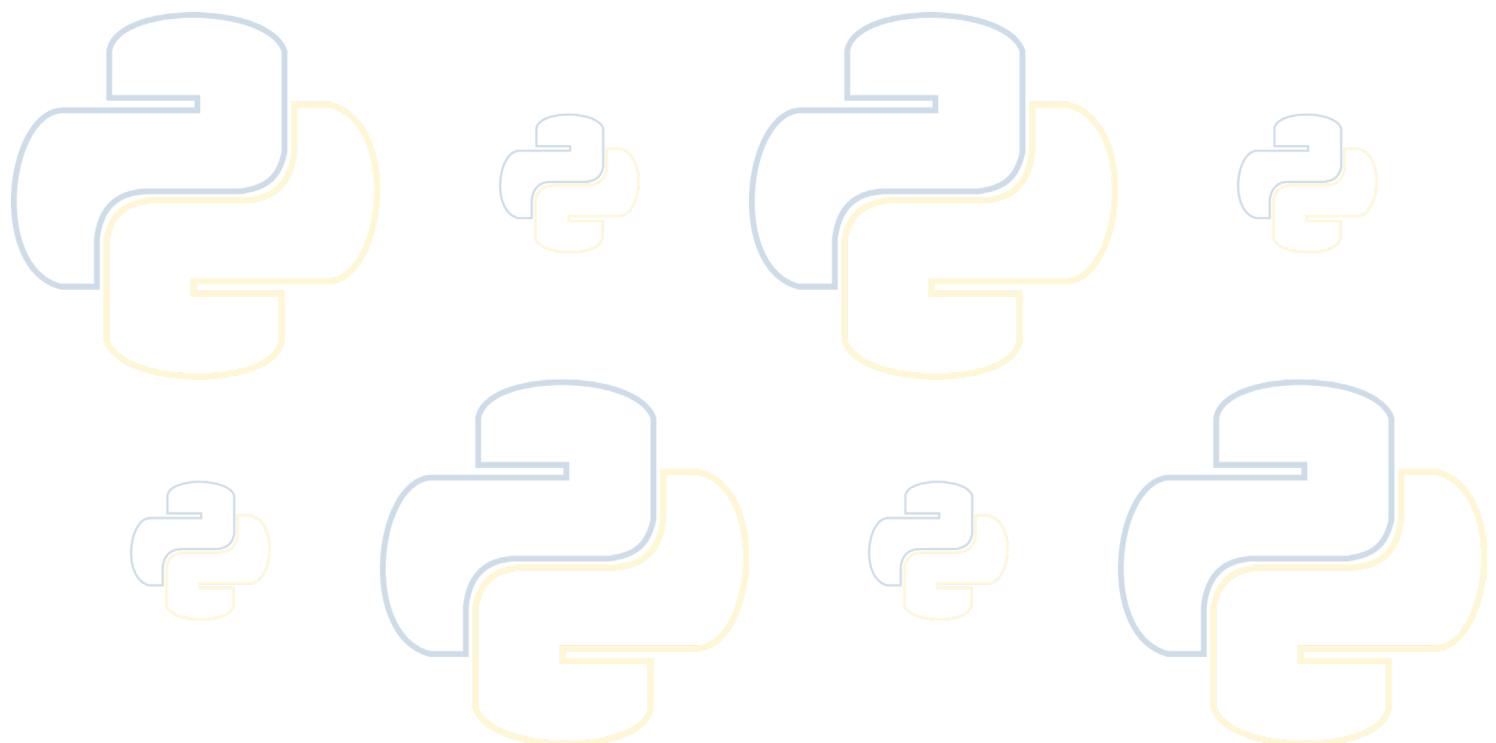
## Meeting 7

# EXAM PYTHON CHATBOT



**What will we learn today?**

1. EXAM



## Meeting 8

# EXAM PYTHON CHATBOT



**What will we learn today?**

1. EXAM