



# Chapter 04. Choices and Repeats

Python Programming for Bioinformatics

Robert C. Chi

# Agenda

- **Introduction**
  - for Flow Controls
- **Branching**
  - Select Which Code to Run
- **Looping**
  - Execute a Specific Code Repeatedly



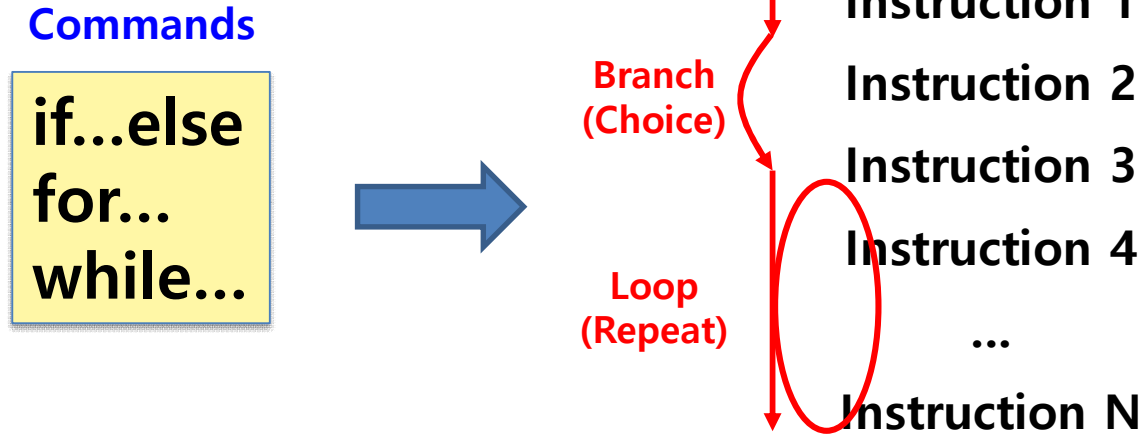


Flow Controls

# **INTRODUCTION**

# What is Flow Controls?

- **Commands** controlling the execution **order** of instructions



# Types of Flow Controls

- All programs can be built by **ONLY 3 types** of flows

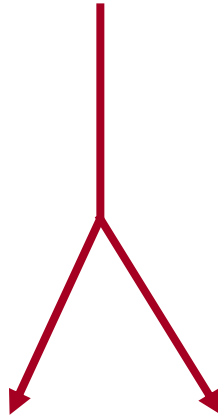
## Sequential

(No Commands  
Required)



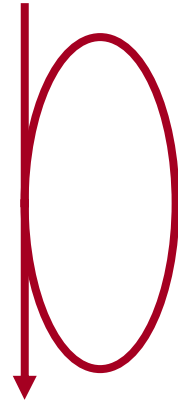
## Branching

if...elif...else...



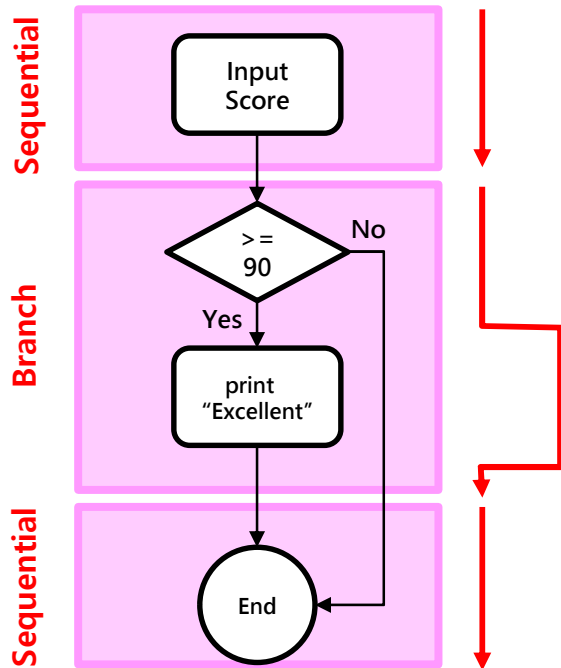
## Looping

for...  
while...

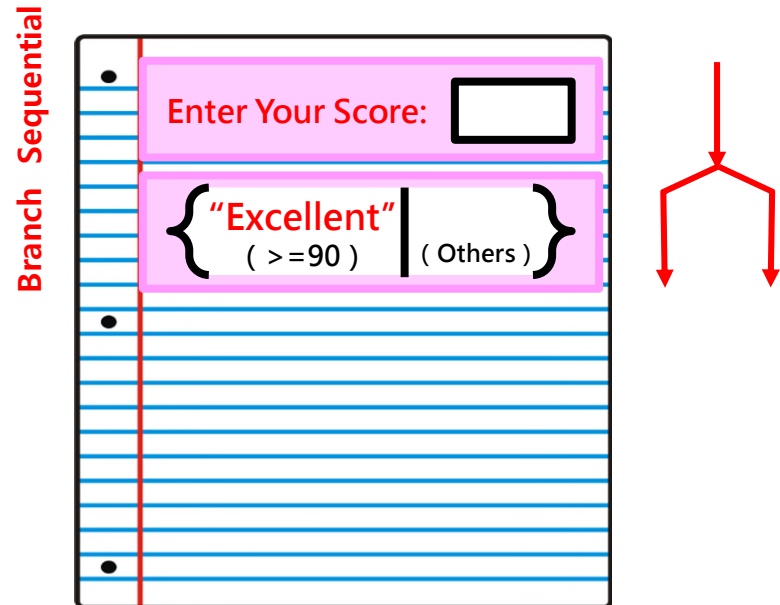


# Decide the Flow You Need

- Draw a Flow Chart



- Draw a Screen Output





# Commands for Flow Controls

- **Branching**

**if...elif...else...**

- **Looping**

Countable Loop

**for...**

Conditional Loop

**while...**



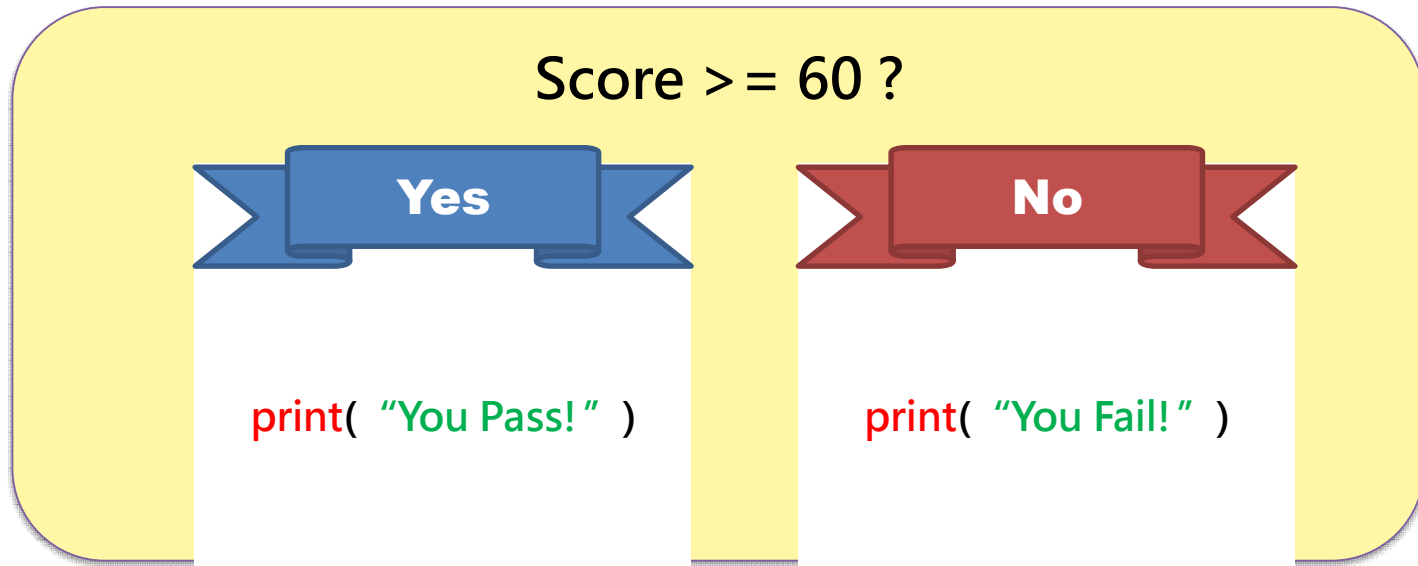
Select Which Code to Run

# **BRANCHING**



# What is a “Branch”

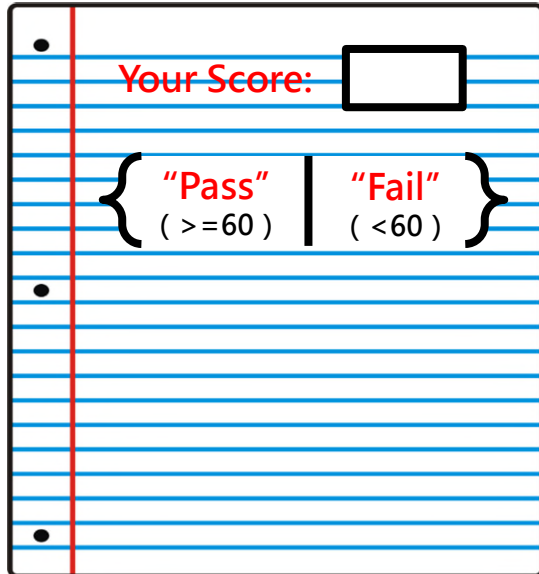
- A **Command** to Select “Which Code to be Run”



# When to Use a Branch?

- When **"One Place, Two or more Outputs Exclusively"**

Screen Outputs



Your Score:

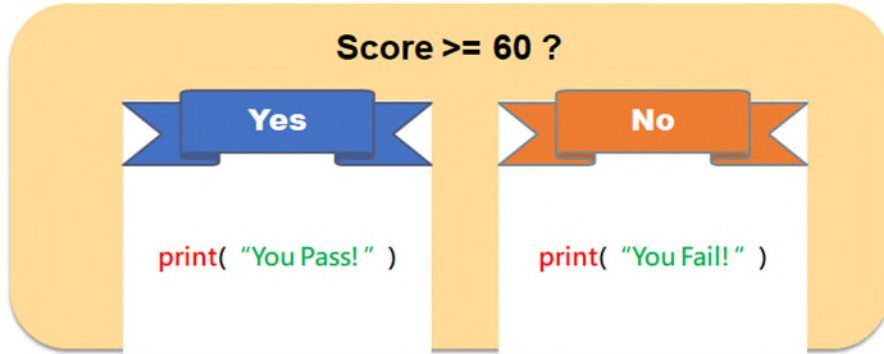
"Pass" ( >=60 )	"Fail" ( <60 )
--------------------	-------------------

Source Codes

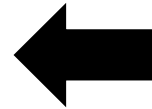
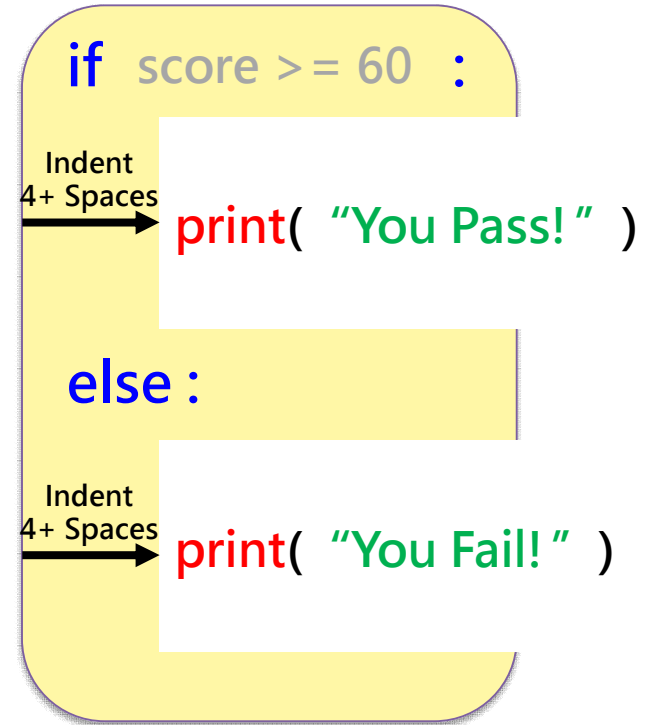
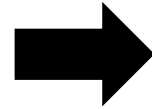
```
score = eval(input( "Enter Your Score: " ))  
  
print( "Pass!" ) / print( "Fail!" )
```

How to make choice of these two?

# Syntax of Branch



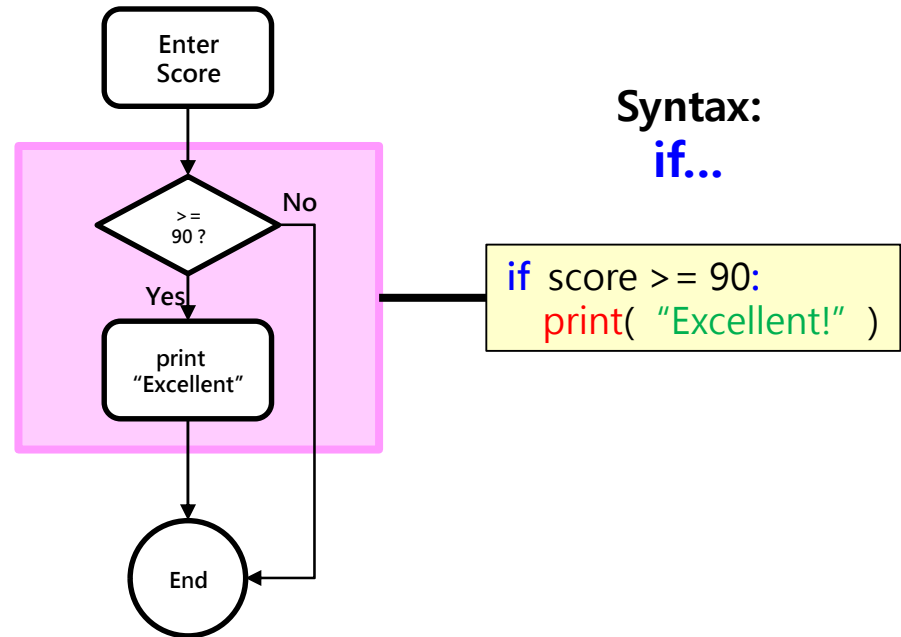
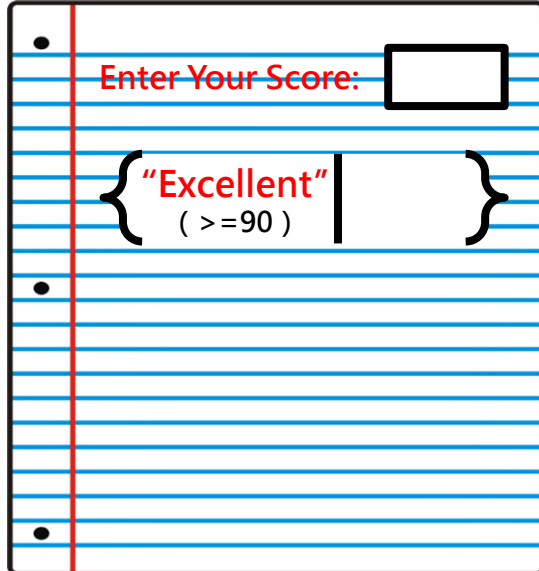
```
if score >= 60 :  
    print( "You Pass!" )  
else:  
    print( "You Fail!" )
```



# (1) Simple Branch

- **Has / Has no Output**

Screen Output



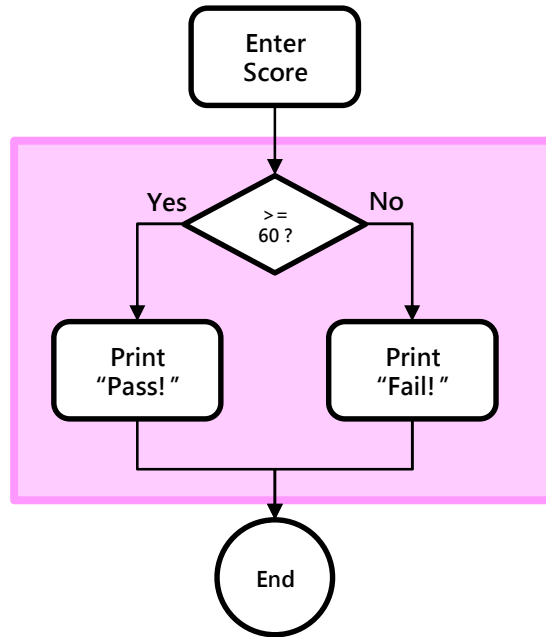
## (2) Double Branch

- Option A / Option B

Screen Output

Enter Your Score:

{ "Pass!" ( >=60 ) | "Fail!" ( <60 ) }



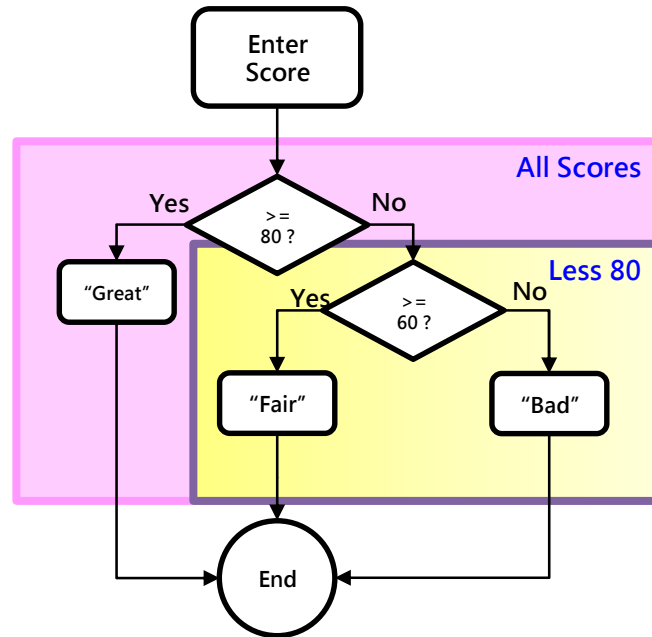
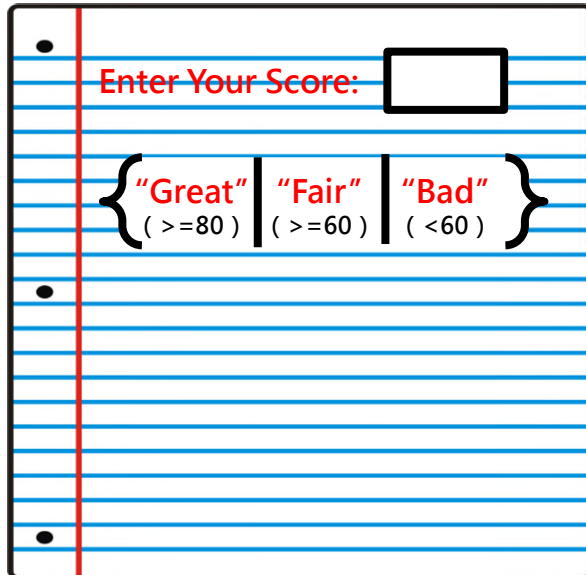
Syntax:  
if...else...

```
if score >= 60:  
    print( "You Pass!" )  
else:  
    print( "You Fail!" )
```

# (3) Nested Branch

- **Three or More Outputs**

Screen Output



Syntax:  
**if...elif...else...**

```
if score >= 80:  
    print( "Great" )  
elif score >= 60:  
    print( "Fair" )  
else:  
    print( "Bad" )
```

# Complete Syntax of Branch

```
if condition1 :  
    Instruction1  
elif condition2 :  
    Instruction2-1  
    Instruction2-2  
elif condition3 :  
    Instruction3  
else :  
    Instruction4
```

## Notice:

1. if, elif, else must have a **colon (:)** at the end.
2. "**Instruction**" belongs to a specific **condition** must be **indented** with **Tab** or **4 blanks**.



Execute a Specific Code Repeatedly

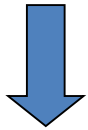
# **LOOPING**



# What is a Loop?

- **Commands** to execute **instructions repeatedly**

```
print "-----" 5 times
```



```
print("-----")  
print("-----")  
print("-----")  
print("-----")  
print("-----")
```

What about  
1000 times?

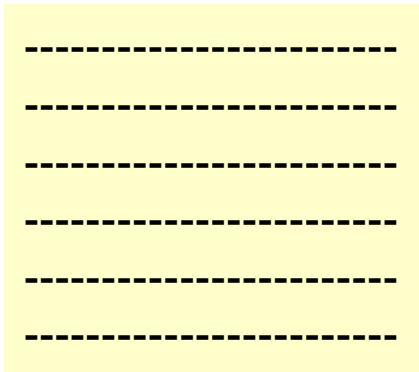


repeat 1000 times

```
print("-----")
```

# When to Use a Loop?

- When **some things** in output appear **repeatedly**...



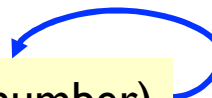
x 6



```
i = 1  
i = 2  
i = 3  
i = 4  
i = 5  
i = 6
```

x 6, number = 1, 2, 3, ...

```
i = (a number)
```



# Types of a Loop

- **Countable Loops**

- Repeat **specific times**

```
for i in range(1, 11, 1):  
    Instruction 1;  
    Instruction 2;  
    ...
```

- **Conditional Loops**

- Repeat when the **condition is still true**

```
while x <= 10:  
    Instruction 1;  
    Instruction 2;  
    ...
```

# Syntax of **for** Loop

for i in **range(100)** :  
    Repeated Instructions

100 times {  
    i = 0  
    i = 1  
    i = 2  
    ...  
    i = 99

for i in **range(10, 100)** :  
    Repeated Instructions

90 times {  
    i = 10  
    i = 11  
    i = 12  
    ...  
    i = 99

for i in **range(10, 100, 2)** :  
    Repeated Instructions

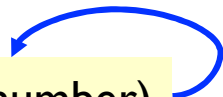
45 times {  
    i = 10  
    i = 12  
    i = 14  
    ...  
    i = 98

# Example of **for** Loop

```
i = 1  
i = 2  
i = 3  
i = 4  
i = 5  
i = 6  
i = 7  
i = 8  
i = 9  
i = 10
```



i = (a number)



x 10, number = 1, 2, 3, ...

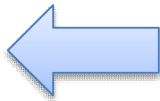


```
for i in range(1, 11):  
    print("i = {}".format(i))
```

# Nested Loop

- **What is a Nested Loop?**
  - A **Loop** inside another **Loop**.

Like a Nest  
(with 90° Rotation)



```
for i in range(1, 4):  
    for j in range(1, 6):  
        ....
```

# Example of Nested Loop

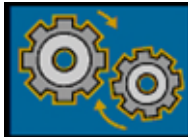
- Please print out the following **result**:

```
for i in range(1, 4):  
    for j in range(1, 6):  
        ...
```

Inner Loop: from 1 to 5 (like a minute hand)

```
(1,1) (1,2) (1,3) (1,4) (1,5)  
(2,1) (2,2) (2,3) (2,4) (2,5)  
(3,1) (3,2) (3,3) (3,4) (3,5)
```

Outer Loop: 1 outer x 5 inner (like an hour hand)



# Example of Nested Loop

- **Source Code**

```
for i in range(1, 4):  
    for j in range(1, 6):  
        print("{} {}".format(i, j), end=" ")  
    print()
```



```
(1, 1) (1, 2) (1, 3) (1, 4) (1, 5)  
(2, 1) (2, 2) (2, 3) (2, 4) (2, 5)  
(3, 1) (3, 2) (3, 3) (3, 4) (3, 5)
```

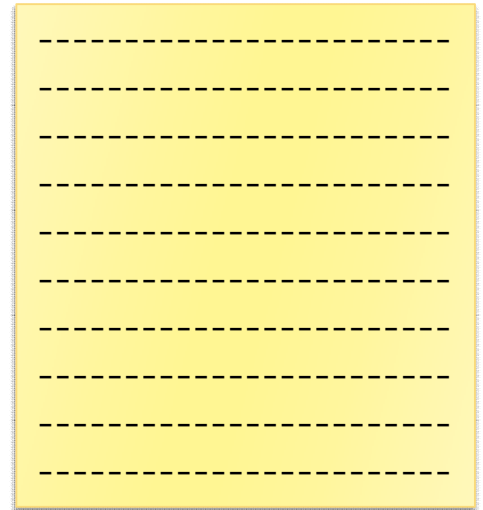


# Definition of **While** Loop

- **while** loop → loop when **condition is true**  
(**Good** when you **don't** know **how many times** to loop)

```
condition Initialization
while condition:
    instructions
    ...
    condition update
else:
    wind-up instructions
```

```
x = 1
while x <= 10 :
    print( "-----" )
    x += 1
```



Optional

# Example: Guess a Number

- Generate a **random number** from **1~10**
- Ask users to **guess** the **number**:
  - if `guess < answer` → print “Larger!”
  - if `guess > answer` → print “Smaller!”
  - if `guess == answer` → print “You got it!”

(Generate a Number)  
Have a guess (1~10): 5  
Smaller!  
Have a guess (1~10): 3  
Smaller!  
Have a guess (1~10): 1  
Larger!  
Have a guess (1~10): 2  
You got it!

( Generate a Number )  
Ask a number from users

Print “Smaller” / Print “Larger”  
Ask a number from users

Print “You got it”

As `guess ≠ answer`  
→ continue

# Example: Guess a Number

- **Source Code**

(Generate a Number)  
Ask a number from users

As guess  $\neq$  answer  
→ continue

Print "Smaller" / Print "Larger"  
Ask a number from users

Print "You got it"

```
import random
answer = random.randint(1, 10)
guess = eval(input("Please guess a number (1~10): "))

while guess != answer:
    if guess < answer:
        print("Larger!")
    else:
        print("Smaller!")
    guess = eval(input("Pleaes guess a number (1~10): "))
else:
    print("You got it!")
```

# break & continue

- **break**: A command to **terminate** a loop

```
for i in range(1, 11) :  
    if i == 7:  
        break  
    print("i = {}".format(i))
```

```
i = 1  
i = 2  
i = 3  
i = 4  
i = 5  
i = 6  
X i = 7  
X i = 8  
X i = 9  
X i = 10
```

# break & continue

- **continue**: A command to **skip** the **current iteration**

```
for i in range(1, 11) :  
    if i == 7:  
        continue  
    print("i = {}".format(i))
```

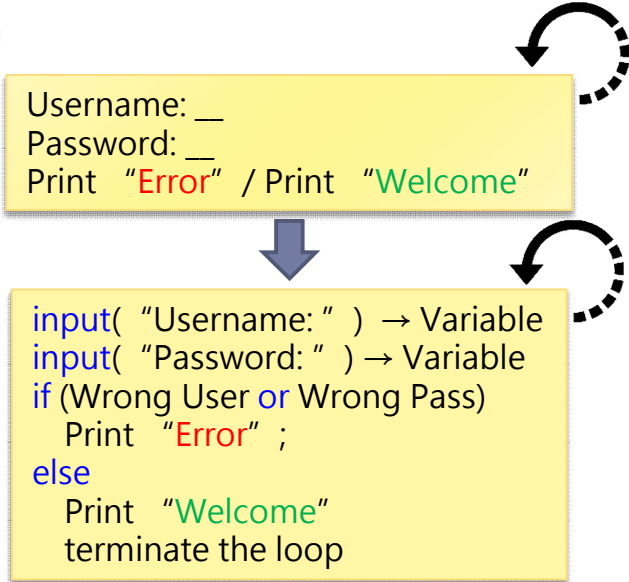
```
i = 1  
i = 2  
i = 3  
i = 4  
i = 5  
i = 6  
X i = 7  
i = 8  
i = 9  
i = 10
```

# Example of **break**

- **Verify Username and Password**
  - Correct: Print “Welcome!”
  - Failed: **Error message** + **Input again**

```
Username: aaa
Password: 111
Wrong Username or Password! Please try again!
Username: bbb
Password: 222
Wrong Username or Password! Please try again!
Username: abc
Password: 123
Welcome!
```

```
Username: __
Password: __
Print "Error" / Print "Welcome"
```



```
input( "Username:" ) → Variable
input( "Password:" ) → Variable
if (Wrong User or Wrong Pass)
    Print "Error" ;
else
    Print "Welcome"
    terminate the loop
```

# Example of **break**

- **Source Code**

```
while True:
    account = input("Username: ")
    password = input("Password: ")
    if (account != "abc") or (password != "123"):
        print("Wrong Username or Password! Please try again!")
    else:
        print("Welcome!")
        break
```

# Example of **continue**

- **Superstition to Skip the 13<sup>th</sup> Floor**
  - There is a **practice** to skip the 13<sup>th</sup> floor of a building to avoid getting bad luck.
  - Please write a script to show **all floors except 13<sup>th</sup>** based on user's input.

Floors < 13

```
How many floors of this building:5
1 Floor
2 Floor
3 Floor
4 Floor
5 Floor
```

Floors  $\geq$  13

```
How many floors of this building:15
1 Floor
2 Floor
3 Floor
4 Floor
5 Floor
6 Floor
7 Floor
8 Floor
9 Floor
10 Floor
11 Floor
12 Floor
14 Floor
15 Floor
16 Floor
```



enter total floors → variable `num`

{ `num` < 13 | `num` + 1 }  
{ (`num` < 13) | (`num`  $\geq$  13) }

if `num` == 13  
skip

show a floor



# Example of **continue**

- **Source Code**

enter total floors → variable `num`

$\left\{ \begin{array}{l} \text{num} \\ (\text{num} < 13) \end{array} \mid \begin{array}{l} \text{num} + 1 \\ (\text{num} \geq 13) \end{array} \right\}$

if `num == 13`  
skip

show a floor

```
floors = eval(input("How many floors of this building: "))  
if floors >= 13:  
    floors += 1  
  
for i in range(1, floors+1):  
    if i == 13:  
        continue  
    print("{} Floor".format(i))
```