Introduction to Programming

PYTHON

Chapter 5 – Command: If … Else statement

Presenter: **Dr. Nguyen Dinh Long**

Email: dinhlonghcmut@gmail.com

Google-site: https://sites.google.com/view/long-dinh-nguyen

Oct. 2022

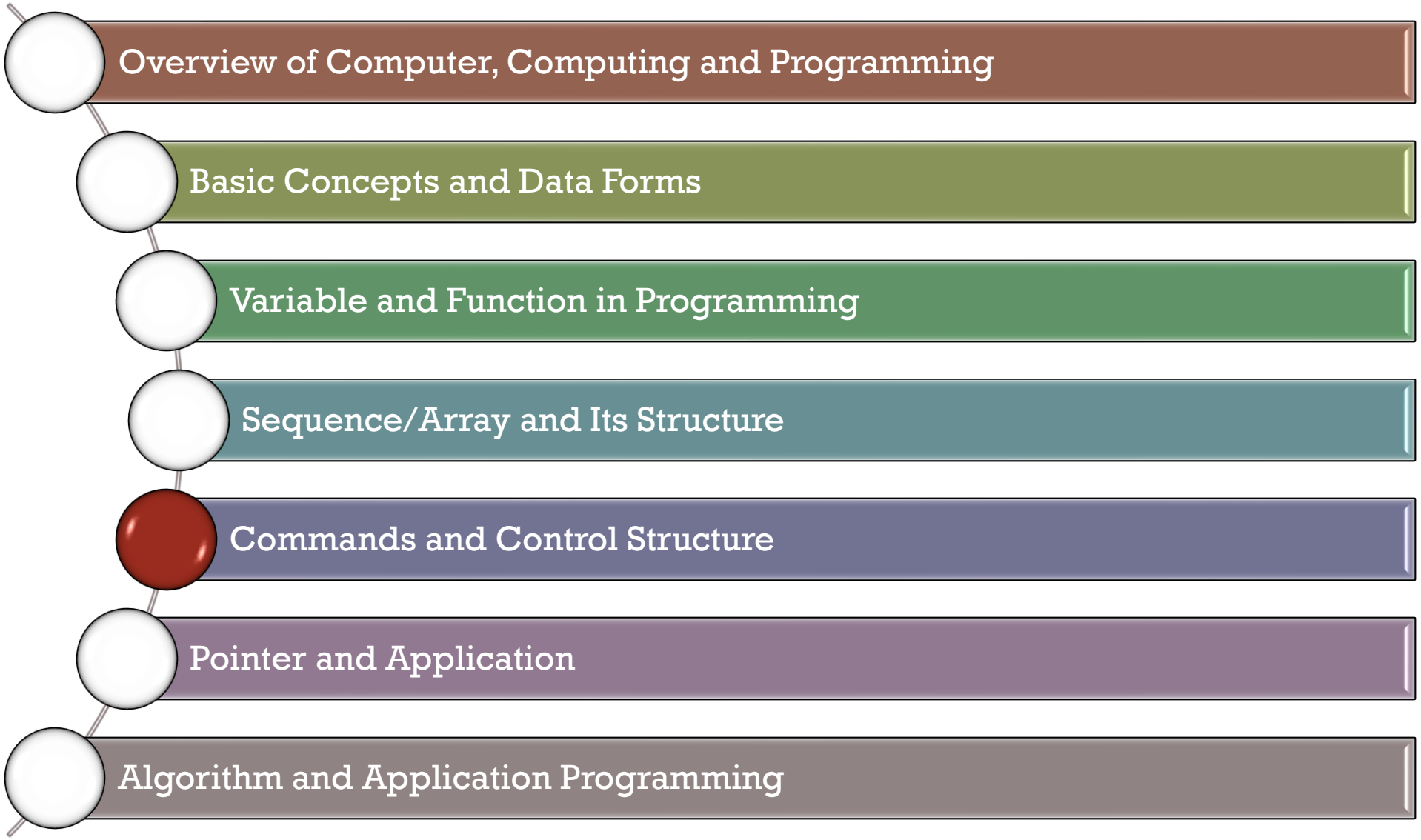
Programming

Hamilton was a self-taught programmer, working in the US in the 1960's. Owing to the success of her previous work, Hamilton was the first programmer to be hired for the Apollo project. She became the Director of Software Engineering at the MIT Instrumentation lab. Her lab developed the on-board flight software for NASA's Apollo space project, which took humankind to the moon.

The achievement was a monumental task at a time when computer technology was in its infancy: The astronauts had access to only 72 kilobytes of computer memory (a 256-gigabyte cell phone today carries almost a million times more storage space). Programmers had to use paper punch cards to feed information into room-sized computers with no screen interface.

Margaret Hamilton, NASA's lead software engineer for the Apollo, stands next to the code she wrote by hand that took humanity to the moon in 1969.

Outline


Overview of Computer, Computing and Programming
Basic Concepts and Data Forms
Variable and Function in Programming
Sequence/Array and Its Structure
Commands and Control Structure
Pointer and Application
Algorithm and Application Programming

References

Main:

- Maurizio Gabbrielli and Simone Martini, 2010. *Programming Languages: Principles and Paradigms*, Springer.
- Cao Hoàng Trụ, 2004. *Ngôn ngữ lập trình- Các nguyên lý và mô hình*, Nhà xuất bản Đại học Quốc gia Tp. Hồ Chí Minh

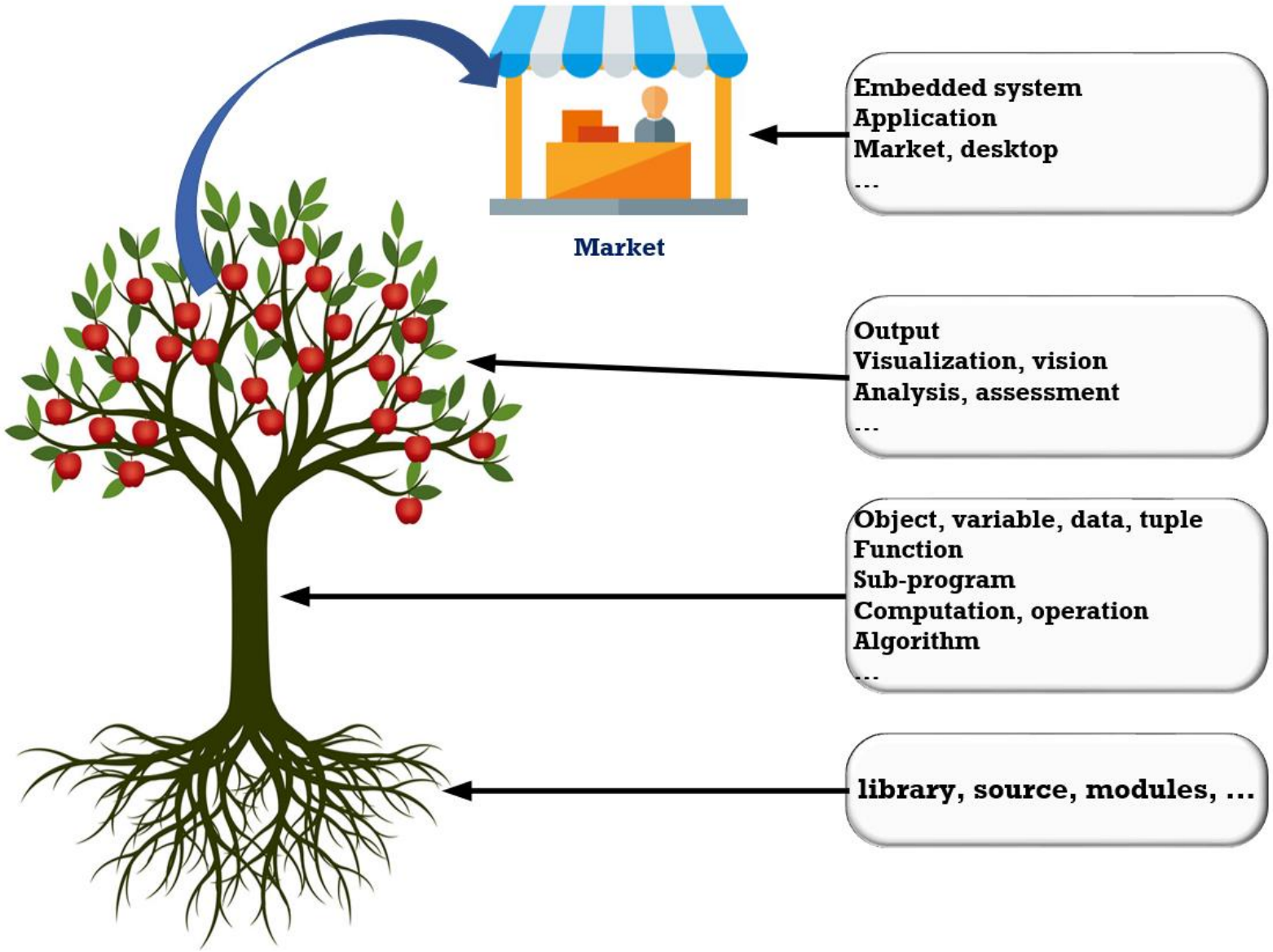
More:

- Wes McKinney, 2013. *Python for Data Analysis*, O'Reilly Media.
- Guido van Rossum, Fred L. Drake, Jr.,, 2012. *The Python Library Reference*, Release 3.2.3.

- Slides here are collected and modified from several sources in Universities and Internet.

Computer programs

- **General structure:**

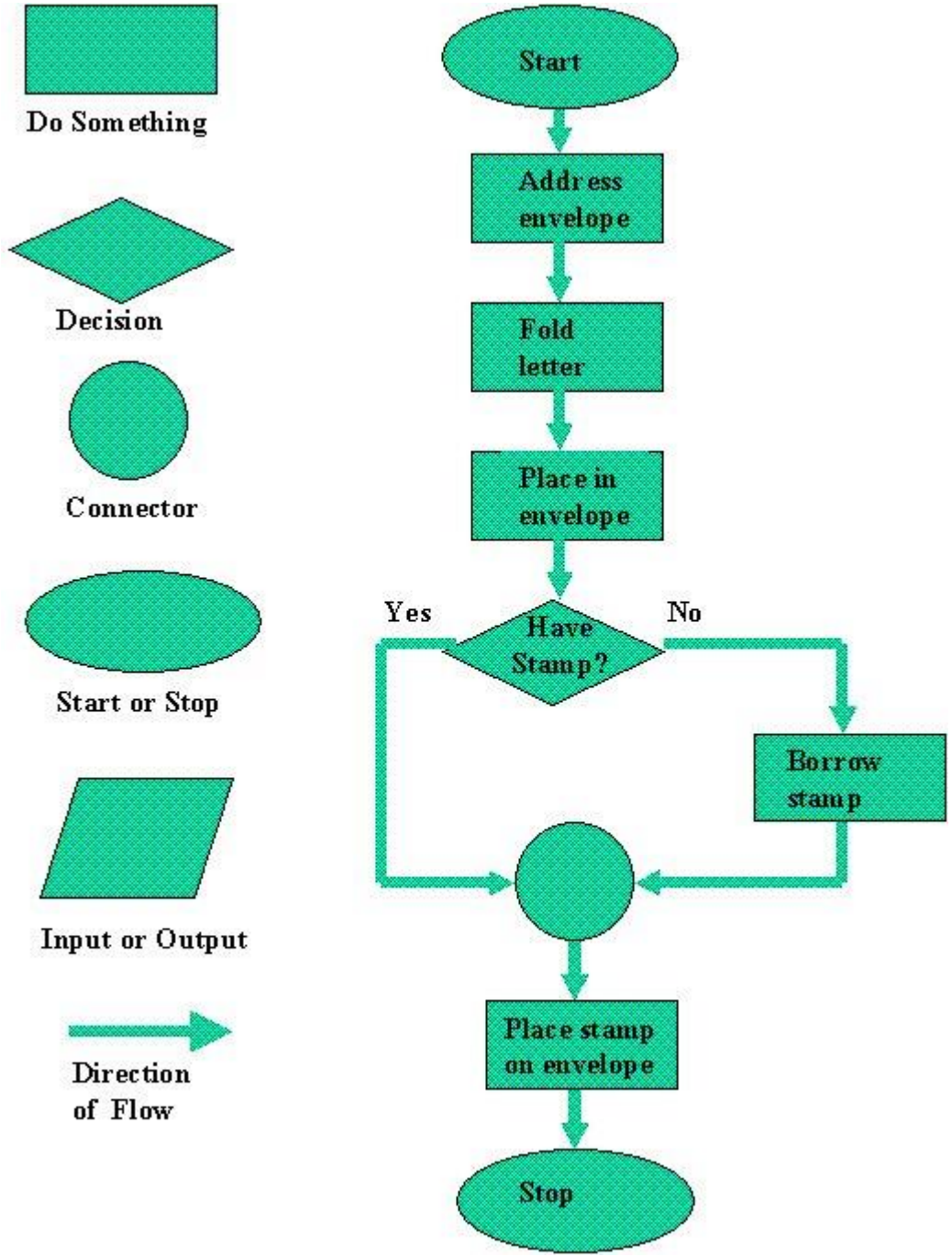


Content of Chapter 5

1. Commands in Python programming
2. Control programs: “for ...” loop and Examples (1 week)
3. Control programs: “while ... do" loop and Examples (1 week)
4. **Control programs: “if ... else" statement and Examples (1 week)**
5. Examples and Practices: Combine Python loops (1 week)

Structure of Computer programs

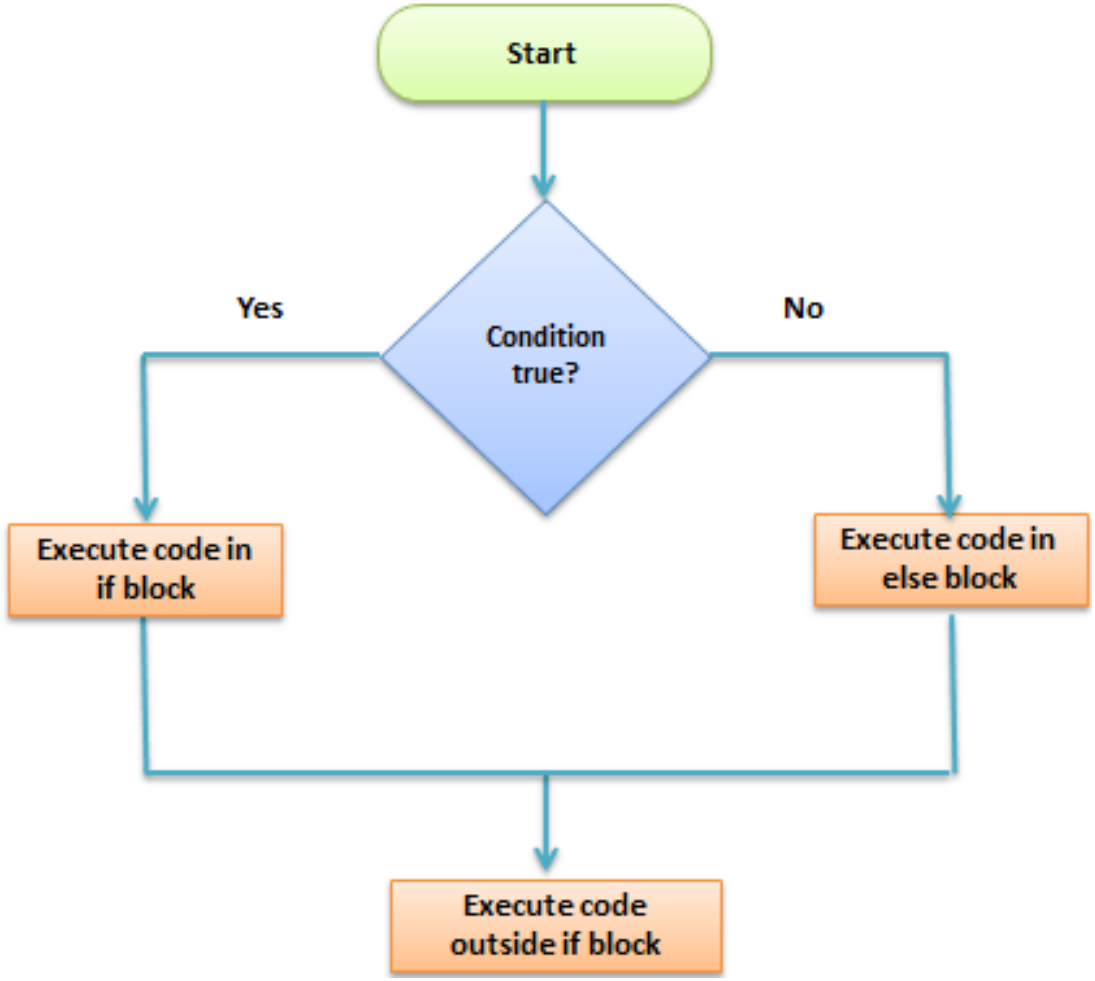
- **Computer programming:**
  - Objects
  - Types
  - Variables
  - Methods
  - Sequences/Arrays



If ... Else statement

- **Difinition:**

Decision making is required when we want to execute a code only if a certain condition is satisfied.



If ... Else statement

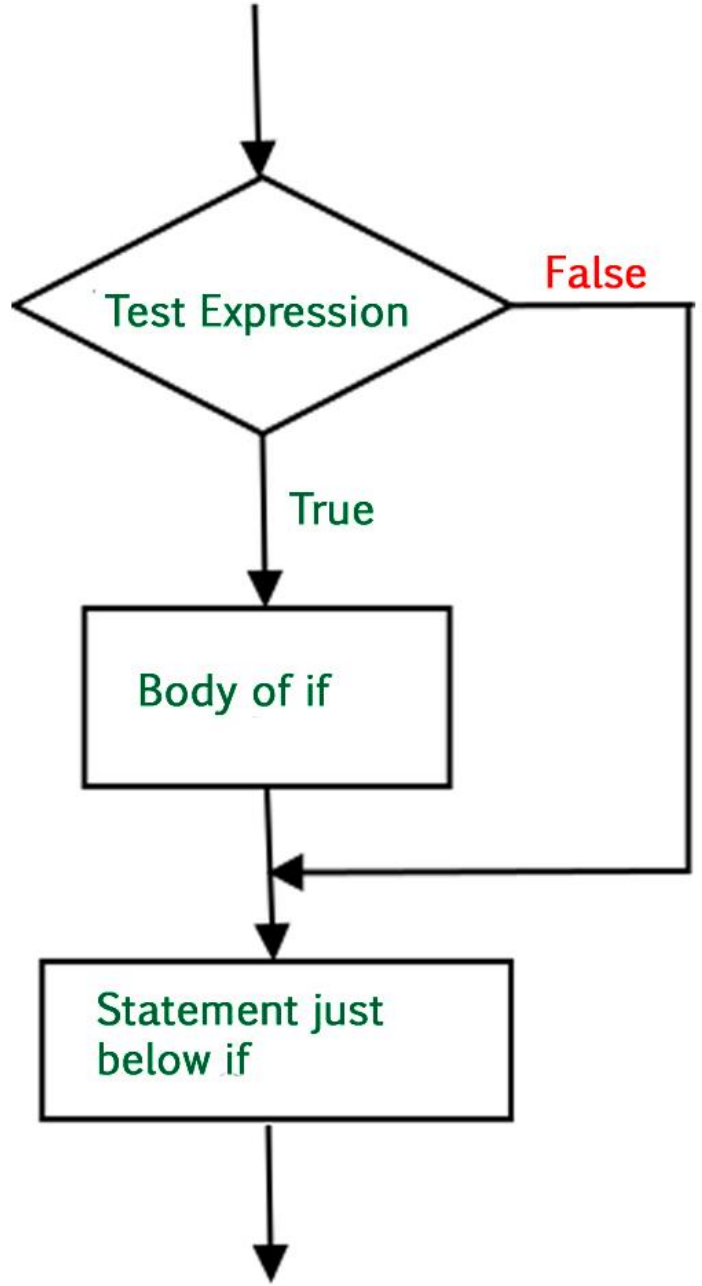
❑ Difinition - Python if Statement:

Decision making is required when we want to
execute a code only if a certain condition is satisfied.

The if… statement is used in Python for decision making.

Python if Statement Syntax

```
if test expression:
    statement(s)
```



Here, the program evaluates the test expression and
will execute statement(s) only if the test expression is True.

If the test expression is False, the statement(s) is not executed.

In Python, the body of the if statement is indicated by the indentation.
The body starts with an indentation and the first unindented line marks the end.

- Python interprets non-zero values as True.
- None and 0 are interpreted as False.

If ... Else statement

❑ Examples - Python if Statement:

```
# If the number is positive, we print an appropriate message

num = 3
if num > 0:
    print(num, "is a positive number.")
print("This is always printed.")

num = -1
if num > 0:
    print(num, "is a positive number.")
print("This is also always printed.")
```

In the above example, num > 0 is the test expression.

- The body of if is executed only if this evaluates to True.
- When the variable num is equal to 3, test expression is true and statements inside the body of if are executed.
- If the variable num is equal to -1, test expression is false and statements inside the body of if are skipped.
- The print() statement falls outside of the if block (unindented).

Hence, it is executed regardless of the test expression.

```
3 is a positive number
This is always printed
This is also always printed.
```

If ... Else statement

- **Difinition - Python if...else Statement:**

Decision making is required when we want to
execute a code only if a certain condition is satisfied.

The if…else statement is used in Python for decision making.

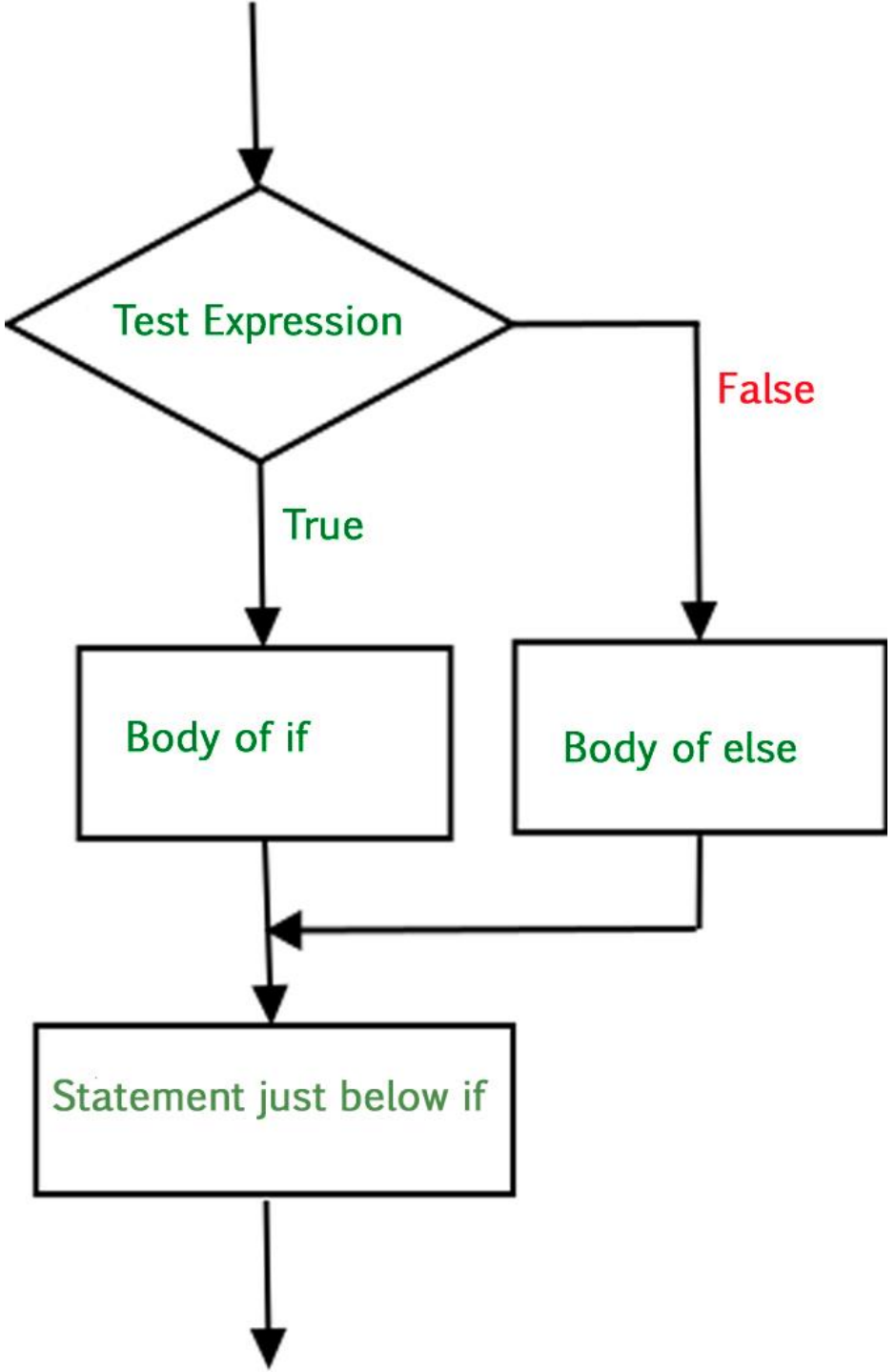
Syntax of if...else

```
if test expression:
    Body of if
else:
    Body of else
```

The if..else statement evaluates test expression and
will execute the body of if only when the test condition is True.

If the condition is False, the body of else is executed.

Indentation is used to separate the blocks.



If ... Else statement

- **Examples - Python if...else Statement :**

```
# Program checks if the number is positive or negative
# And displays an appropriate message

num = 3

# Try these two variations as well.
# num = -5
# num = 0

if num >= 0:
    print("Positive or Zero")
else:
    print("Negative number")
```

In this example,

- when num is equal to 3, the test expression is true and the body of if is executed and the body of else is skipped.
- If num is equal to -5, the test expression is false and the body of else is executed and the body of if is skipped.
- If num is equal to 0, the test expression is true and body of if is executed and body of else is skipped.

Output

```
Positive or Zero
```

If ... Else statement

❑ Difinition - Python if...elif...else Statement:

Decision making is required when we want to
execute a code only if a certain condition is satisfied.

The if…elif…else statement is used in Python for decision making.

Syntax of if...elif...else

```
if test expression:
    Body of if
elif test expression:
    Body of elif
else:
    Body of else
```

- The elif is short for else if. It allows us to check for multiple expressions.
- If the condition for if is False, it checks the condition of the next elif block and so on.
- If all the conditions are False, the body of else is executed.
- Only one block among the several if...elif...else blocks is executed according to the condition.
- The if block can have only one else block. But it can have multiple elif blocks.

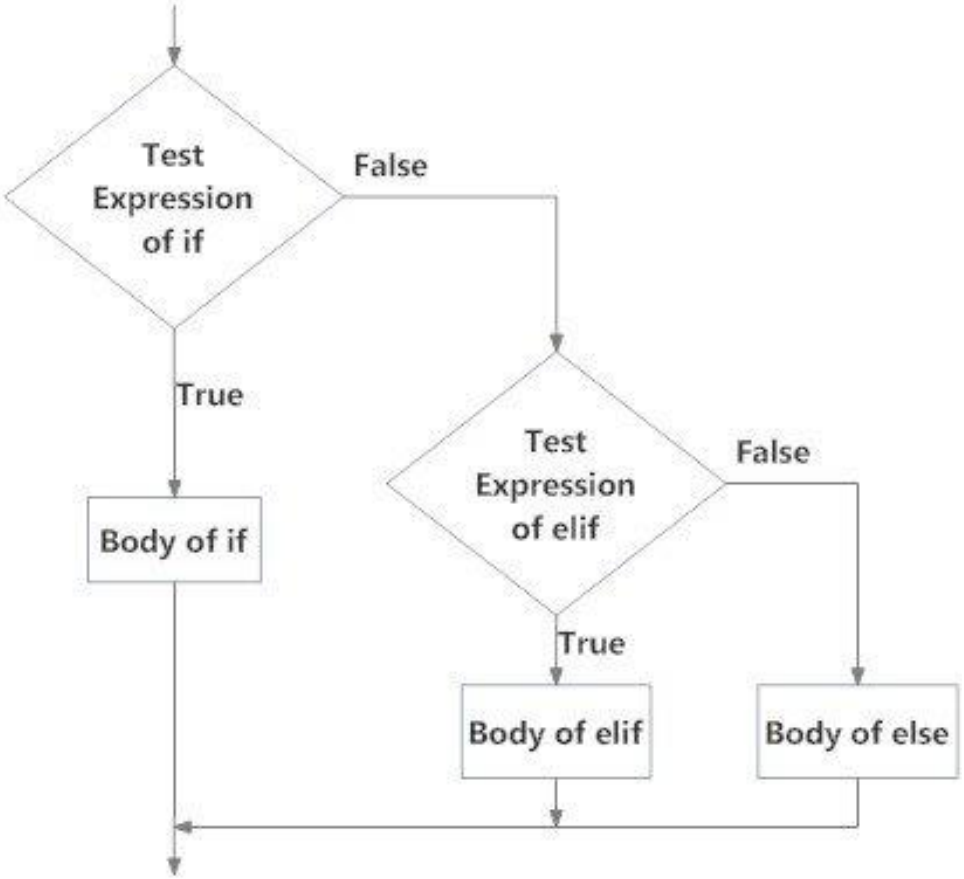


Fig: Operation of if...elif...else statement

If ... Else statement

- **Difinition - Python if...elif...else Statement:**

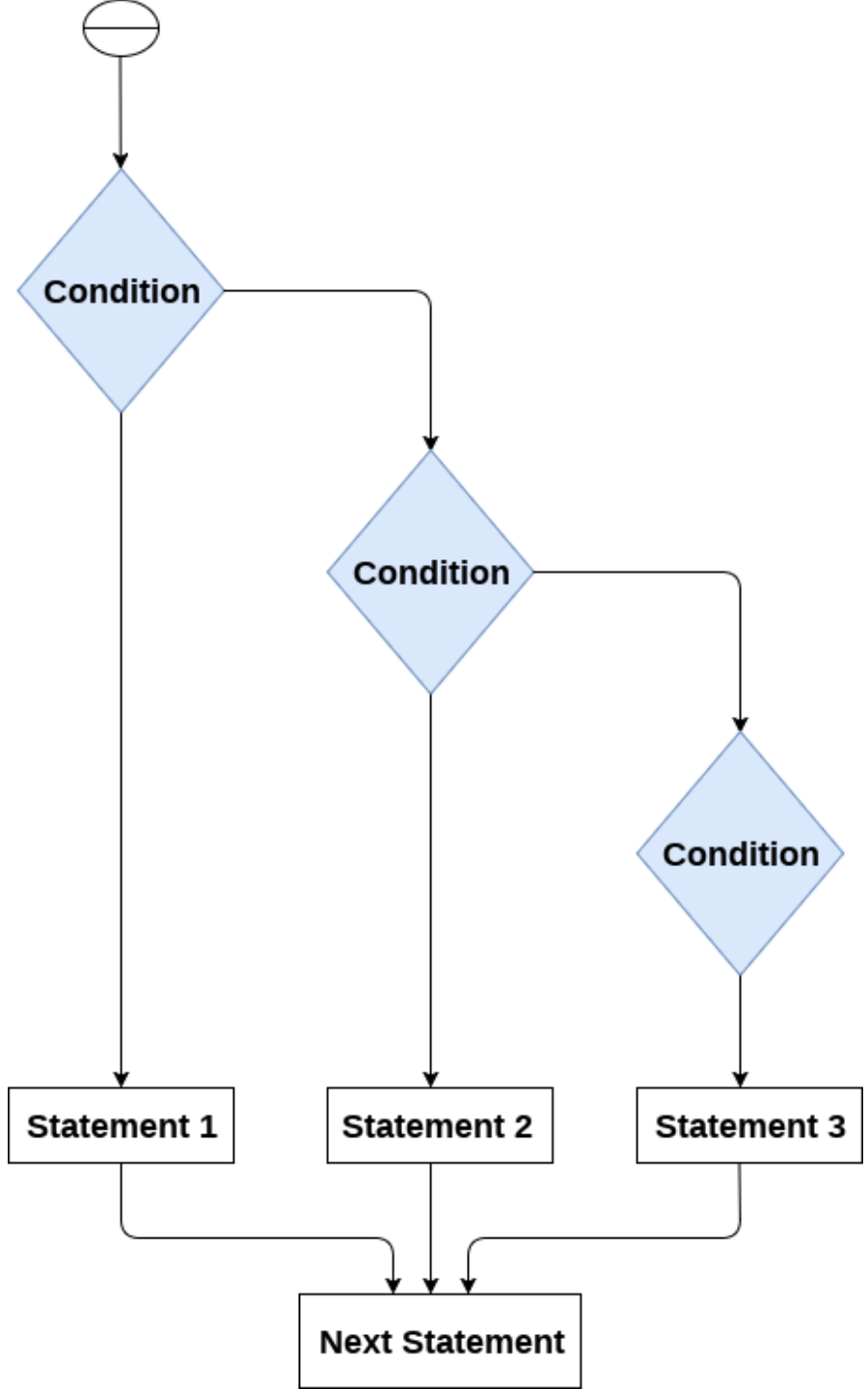
The if…elif…else statement is used in Python for decision making.

```
if expression 1:
    # block of statements

elif expression 2:
    # block of statements

elif expression 3:
    # block of statements

else:
    # block of statements
```



If ... Else statement

- **Examples - Python if...elif...else Statement:**

```
'''In this program,
we check if the number is positive or
negative or zero and
display an appropriate message'''

num = 3.4

# Try these two variations as well:
# num = 0
# num = -4.5

if num > 0:
    print("Positive number")
elif num == 0:
    print("Zero")
else:
    print("Negative number")
```

- When variable num is positive, Positive number is printed.
- If num is equal to 0, Zero is printed.
- If num is negative, Negative number is printed.

If ... Else statement

❑ Python Nested if statements:

We can have a if...elif...else statement inside another if...elif...else statement. This is called nesting in computer programming.

```
'''In this program, we input a number
check if the number is positive or
negative or zero and display
an appropriate message
This time we use nested if statement'''

num = float(input("Enter a number: "))
if num >= 0:
    if num == 0:
        print("Zero")
    else:
        print("Positive number")
else:
    print("Negative number")
```

Output 1

```
Enter a number: 5
Positive number
```

Output 2

```
Enter a number: -1
Negative number
```

Output 3

```
Enter a number: 0
Zero
```

If ... Else statement

❑ Python Nested if statements:

We can have a if...elif...else statement inside another if...elif...else statement.

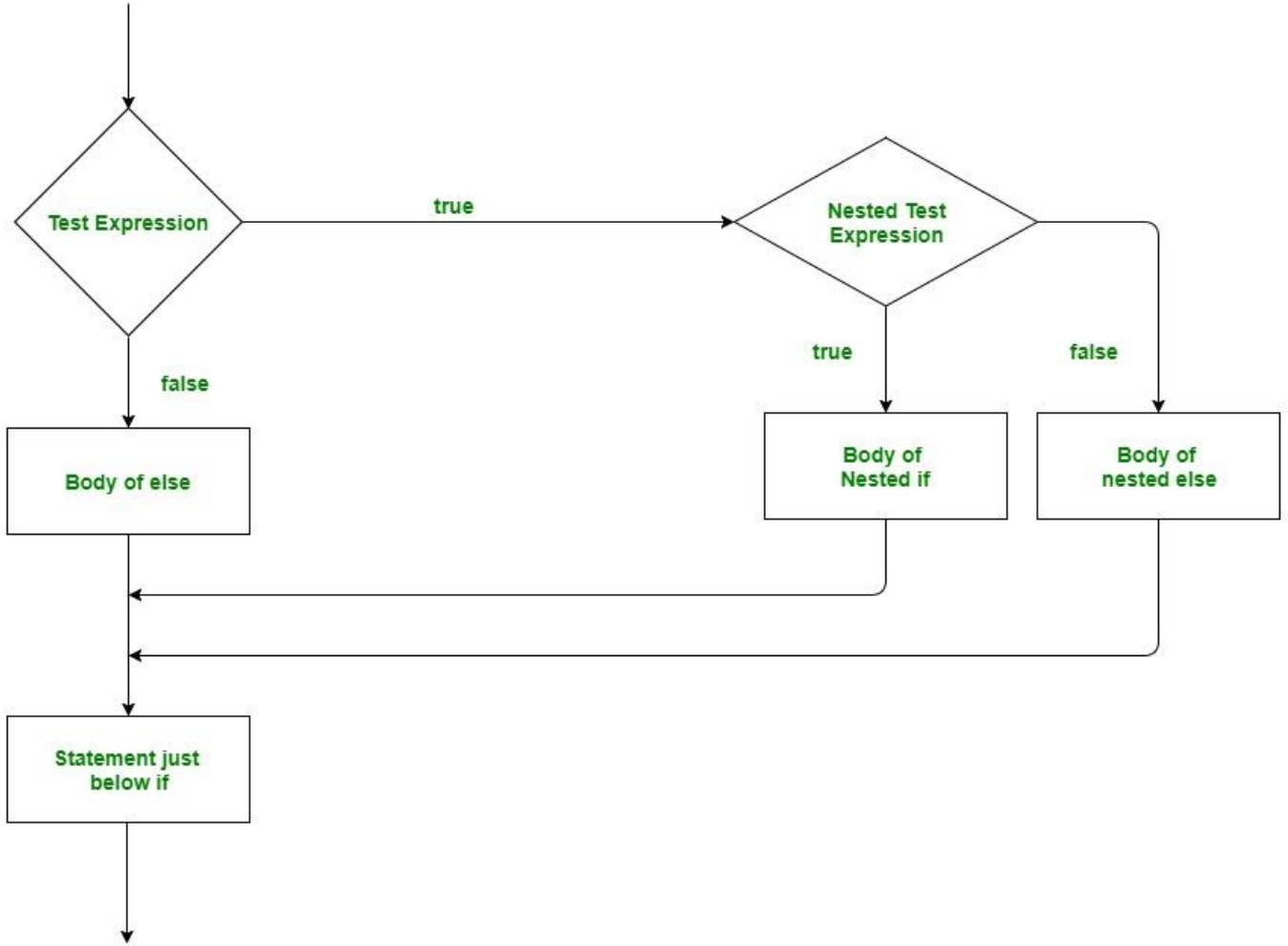
This is called nesting in computer programming.

Python3

```
# python program to illustrate nested If statement
#!/usr/bin/python
i = 10
if (i == 10):

    #  First if statement
    if (i < 15):
        print("i is smaller than 15")

    # Nested - if statement
    # Will only be executed if statement above
    # it is true
    if (i < 12):
        print("i is smaller than 12 too")
    else:
        print("i is greater than 15")
```



If ... Else statement

- **Short Hand if ... else statement:**

.

Syntax:

```
if condition: statement
```

Syntax:

```
statement_when_True if condition else statement_when_False
```

Python3

```
# Python program to illustrate short hand if-else
i = 10
print(True) if i < 15 else print(False)
```

If ... Else statement

- **Python pass Statement:**

.

Python

```
if True:

print('foo')
```

```
C:\Users\john\Documents\Python\doc>python foo.py
  File "foo.py", line 3
    print('foo')
        ^
IndentationError: expected an indented block
```

Python

```
if True:
    pass

print('foo')
```

If ... Else statement

- **Python break Statement:**

.

```
for i in range(6,10):
    print(i)
    if i == 5:
        break
else:
    print("This code will only execute if the for loop completes without hitting a break st

# this will output:
6
7
8
9
This code will only execute if the for loop completes without hitting a break statement.
```

If ... Else statement

❑ Difinition - Switch Case Statement in Python:

A switch statement is a multiway branch statement that compares the value of a variable to the values specified in case statements.

Python language doesn’t have a switch statement.

Python uses dictionary mapping to implement Switch Case in Python.

```
function(argument){
    switch(argument) {
        case 0:
            return "This is Case Zero";
        case 1:
            return " This is Case One";
        case 2:
            return " This is Case Two ";
        default:
            return "nothing";
    };
};
```

For the above Switch case in Python

```
def SwitchExample(argument):
    switcher = {
        0: " This is Case Zero ",
        1: " This is Case One ",
        2: " This is Case Two ",
    }
    return switcher.get(argument, "nothing")


if __name__ == "__main__":
    argument = 1
    print (SwitchExample(argument))
```

If ... Else statement

- **Use the in operator in an if statement:**

```
#create a string
s = 'jQuery'
#create a list
l = ['JavaScript', 'jQuery', 'ZinoUI']

# in operator is used to replace various expressions that use the or operator
if s in l:
    print(s + ' Tutorial')

#Alternate if statement with or operator

if s == 'JavaScript' or s == 'jQuery' or s == 'ZinoUI':
     print(s + ' Tutorial')
```

If ... Else statement

❑ Conditional Expressions:

Syntax of the conditional expression is as follows

Python

```
<expr1> if <conditional_expr> else <expr2>
```

Equals: `a == b`
Not Equals: `a != b`
Less than: `a < b`
Less than or equal to: `a <= b`
Greater than: `a > b`
Greater than or equal to: `a >= b`

Python

```
>>> raining = False
>>> print("Let's go to the", 'beach' if not raining else 'library')
Let's go to the beach
>>> raining = True
>>> print("Let's go to the", 'beach' if not raining else 'library')
Let's go to the library

>>> age = 12
>>> s = 'minor' if age < 21 else 'adult'
>>> s
'minor'

>>> 'yes' if ('qux' in ['foo', 'bar', 'baz']) else 'no'
'no'
```

If ... Else statement

❑ Python Conditions and If statements:

The and keyword is a logical operator, and is used to combine conditional statements

Test if `a` is greater than `b`, AND if `c` is greater than `a`:

```
a = 200
b = 33
c = 500
if a > b and c > a:
  print("Both conditions are True")
```

The or keyword is a logical operator, and is used to combine conditional statements

Test if `a` is greater than `b`, OR if `a` is greater than `c`:

```
a = 200
b = 33
c = 500
if a > b or a > c:
  print("At least one of the conditions is True")
```

Python If ... Else

Examples

If ... Else statement - Examples

- **if-else – is the number even or odd:**

.

```
#Take user input
inp_num = input("Enter a number: ")

#Convert string to int
inp_num = int(inp_num)

if inp_num == 0:
  print(inp_num, "is Even")
elif inp_num%2==0:
  print(inp_num, "is Even")
else:
  print(inp_num, "is Odd")
```

If ... Else statement - Examples

- **Program to print the largest of the three numbers:**

.

```
a = int(input("Enter a? "));
b = int(input("Enter b? "));
c = int(input("Enter c? "));
if a>b and a>c:
    print("a is largest");
if b>a and b>c:
    print("b is largest");
if c>a and c>b:
    print("c is largest");
```

Output:

```
Enter a? 100
Enter b? 120
Enter c? 130
c is largest
```

If ... Else statement - Examples

- **Program to assessment your grades:**

.

```
marks = int(input("Enter the marks? "))
f marks > 85 and marks <= 100:
  print("Congrats ! you scored grade A ...")
lif marks > 60 and marks <= 85:
  print("You scored grade B + ...")
lif marks > 40 and marks <= 60:
  print("You scored grade B ...")
lif (marks > 30 and marks <= 40):
  print("You scored grade C ...")
lse:
  print("Sorry you are fail ?")
```

If ... Else statement - Examples

- **Applied if, series of elif and else to get the type of a variable:**

.

```
var1 = 1+2j
if (type(var1) == int):
    print("Type of the variable is Integer")
elif (type(var1) == float):
    print("Type of the variable is Float")
elif (type(var1) == complex):
    print("Type of the variable is Complex")
elif (type(var1) == bool):
    print("Type of the variable is Bool")
elif (type(var1) == str):
    print("Type of the variable is String")
elif (type(var1) == tuple):
    print("Type of the variable is Tuple")
elif (type(var1) == dict):
    print("Type of the variable is Dictionaries")
elif (type(var1) == list):
    print("Type of the variable is List")
else:
    print("Type of the variable is Unknown")
```

If ... Else statement - Examples

❑ Making change:

The program prompts the user to enter an amount in pennies to make change for. The program then uses an algorithm to compute how many quarters, dimes, nickels and pennies need to be counted out to make change for that amount

The program uses the following logic to compute the amount of each coin to use

```
quarters = cents // 25
cents = cents % 25
dimes = cents // 10
cents = cents % 10
nickels = cents // 5
pennies = cents % 5
```

```
# Get input
cents = int(input('How many cents do you need to give out? '))

# Do computations
quarters = cents // 25
cents = cents % 25
dimes = cents // 10
cents = cents % 10
nickels = cents // 5
pennies = cents % 5

# Print results
print('Your change is')
if quarters > 1:
  print(str(quarters)+' quarters')
elif quarters == 1:
  print('1 quarter')

if dimes > 1:
  print(str(dimes)+' dimes')
elif dimes == 1:
  print('1 dime')

if nickels == 1:
  print('1 nickel')

if pennies > 1:
  print(str(pennies)+' pennies.')
elif pennies == 1:
  print('1 penny.')
```

If ... Else statement - Examples

- **Python if else in list comprehension:**

.

Python3

```
# Explicit function
def digitSum(n):
    dsum = 0
    for ele in str(n):
        dsum += int(ele)
    return dsum


# Initializing list
List = [367, 111, 562, 945, 6726, 873]

# Using the function on odd elements of the list
newList = [digitSum(i) for i in List if i & 1]

# Displaying new list
print(newList)
```

If ... Else statement - Examples

- **What happen when “if condition” does not meet:**

.

```
#
# Example file for working with conditional statement
#
def main():
    x, y = 8, 4

    if (x < y):
        st = "x is less than y"
    print(st)

if __name__ == "__main__":
    main()
```

It shows the error because it does not match our "if condition" (i.e x<y)

USE “else condition”

```
#
# Example file for working with con
#
def main():
    x, y = 8, 4

    if (x < y):
        st = "x is less than y"
    else:
        st = "x is greater than y"
    print(st)
```

If ... Else statement - Examples

- **When "if … else condition" does not work:**

```
#
# Example file for working with conditional statement
#
def main():
    x, y = 8, 8

    if (x < y):
        st = "x is less than y"
    else:
        st = "x is greater than y"
    print(st)


if __name__ == "__main__":
    main()
```

Oops ! .... Now both #numbers over here are same, still it prints out "x is greater than y"

USE "elif" condition

```
#
def main():
    x, y = 8, 8

    if (x < y):
        st = "x is less than y"

    elif (x == y):
        st = "x is same as y"

    else:
        st = "x is greater than y"
    print(st)
```

If ... Else statement

A story of a Game Theory ...

Prisoner's dilemma

Prisoner's dilemma

The Prisoner's Dilemma is an example of a game analyzed in game theory.

It is also a thought experiment that challenges two completely rational agents to a dilemma: cooperate with Police and disclose, or not cooperate and remain silent.

Cooperation, disclosing to police, entails betraying one's partner in crime; whereas not cooperating and remaining silent, entails they, equally, serve one year in jail.

If one talks, they, the betraying partner, will go free. The other will serve three years in jail.

These choices as visually represented in the matrix to the right of the page and set out in dot point form below.

Standard prisoner's dilemma payoff matrix

A \ B	B stays silent	B betrays
A stays silent	-1 / -1	0 / -3
A betrays	-3 / 0	-2 / -2

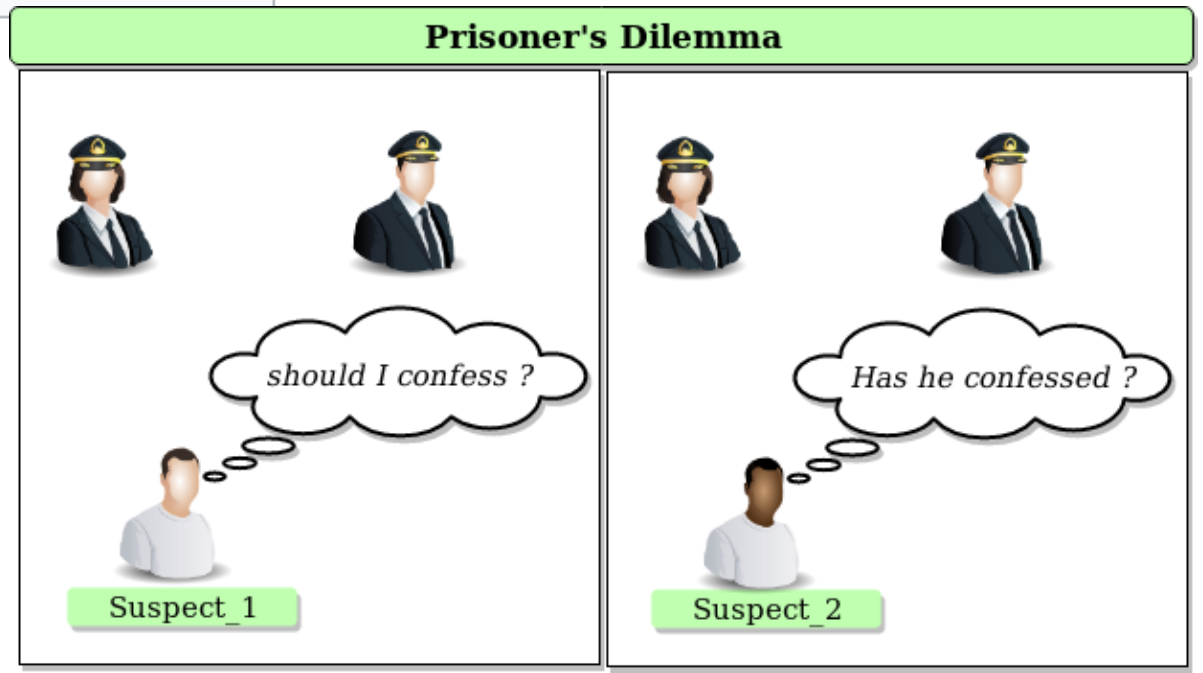
The possible outcomes are:

- A: If A and B each betray the other, each of them serves two years in prison
- B: If A betrays B but B remains silent, A will be set free and B will serve three years in prison
- C: If A remains silent but B betrays A, A will serve three years in prison and B will be set free
- D: If A and B both remain silent, both of them will serve one year in prison (on the lesser charge).

Prisoner's dilemma

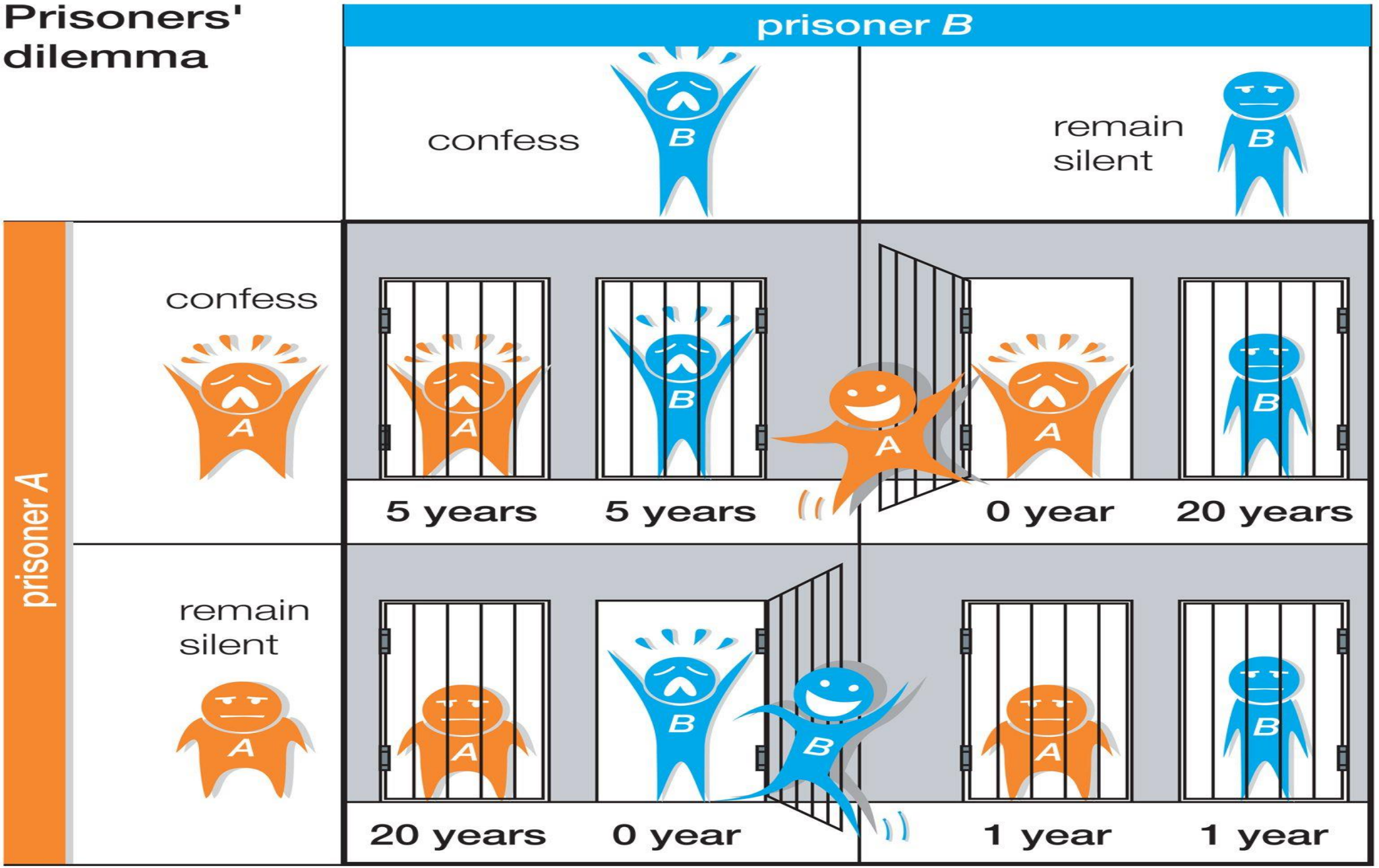
The Prisoner's Dilemma is an example of a game analyzed in game theory.

Prisoner B / Prisoner A	**Prisoner B stays silent (*cooperates*)**	**Prisoner B betrays (*defects*)**
Prisoner A stays silent (*cooperates*)	Each serves 1 year	Prisoner A: 3 years Prisoner B: goes free
Prisoner A betrays (*defects*)	Prisoner A: goes free Prisoner B: 3 years	Each serves 2 years

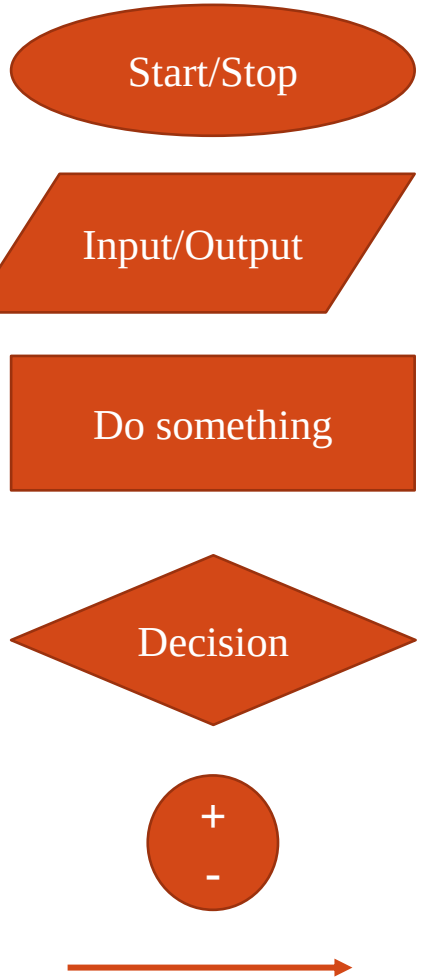


https://en.wikipedia.org/wiki/Prisoner%27s_dilemma

Prisoner's dilemma



Practices



Hai đồng phạm bị bắt và sẽ bị kết án tù. Mỗi tù nhân bị biệt giam không có phương tiện liên lạc với người kia.

Các công tố viên thiếu bằng chứng đầy đủ để kết tội cặp đôi với tội danh chính. Bản án tốt thiểu có thể buộc tội là 3 năm tù giam chỗ mỗi người, và tối đa là 20 năm đối với người chủ mưu.
Các công tố viên thực hiện trò đánh đố với tội phạm.

Mỗi tù nhân sẽ có cơ hội: phản bội người kia bằng cách làm chứng rằng người kia đã phạm tội, hoặc hợp tác người kia bằng cách giữ im lặng.

Án sẽ được kết nếu:

- If A and B each betray the other, each of them serves 5 years in prison
- If A betrays B but B remains silent, A will be set free and B will serve 20 years in prison (and vice versa)
- If A and B both remain silent, both of them will only serve 3 year in prison (on the lesser charge)