

Control Structures (Conditional Statement)

Python Programming
CT108-3-1-PYP

Boolean Expression

- › A Boolean expression is an expression whose value is either True or False.
- › Examples:
 - $5 < 10?$
 - $X = 10?$
 - $Y \geq 5?$
 - $1355/5 = 0?$

π How to get Boolean results

- › Comparison Operator
 - To get Boolean result, we compare two things

Equal	==
Not Equal	!=
Greater than	>
Greater than or equal	>=
Less than	<
Less than or equal	<=

π How to get Boolean results

- › Boolean Operator
 - To get Boolean results, we combine one or two Boolean results

Boolean Operators	Operators
And	and
Or	or
Not	not

True and True	True	True or True	True	not True	False
True and False	False	True or False	True	not False	True
False and True	False	False or True	True		
False and False	False	False or False	False		

Indentation

- › Indentation refers to the practice of adding spaces or tabs at the beginning of a line of code to visually define its structure and scope.
- › It is commonly used to indicate which code belongs to a particular block, such as those defined **by if, for, while**, and other statements.
- › Proper indentation makes code more readable and easier to maintain.
- › When the block ends, reduce the indentation back to the level of the controlling statement (e.g., if, for, while) to signify the end of that block.
- › Blank lines and Comments in your code are ignored for indentation purposes. Python uses the concept of blocks extensively.

π Indentation

```
x = 5
if x > 2 :
    print ('Bigger than 2')
    print ('Still bigger')
print ('Done with 2')
```

GOOD

```
x = 5
if x > 2 :
print ('Bigger than 2')
    print ('Still bigger')
print ('Done with 2')
```

BAD

π pass-statement for empty blocks

- › In Python, we cannot have an empty block.

```
if height <= 140:  
    print("I'm here")
```

- › If you want a block that does nothing, put the **pass** statement inside it.

```
if height <= 140:  
    pass  
    print("I'm here")
```



Creating Conditions

All **conditional structures** have a ***condition***.



A condition is just an expression that is either true or false.



Example, the condition used in the if structure is $x < 0$.

This condition evaluates to either true or false, depending on the value of x .

If the value of x is less than zero, then the condition is true. Otherwise, the condition is false.

π Conditional Statements (Decision Making)

The basic decision statements in computer is selection structure.



The decision is described to computer as a conditional statement that can be answered **True or False**.



Python language provide the following conditional (Decision making) statements.

if statement

if...else statement

if...elif...else statement

Nested if..else statement

π If statement - One Way Decision

- › `if` statement evaluates a condition, and controls if the following statements shall be executed.
- › The statements inside the `block` will be executed when the condition is `True`.

If statement - One Way Decision Example

- › Problem 1 - MRT
 - Children whose height is not larger than or equal to 140cm can enjoy a free ride.

- › Problem 1 Answer

```
height = int(input("Enter your height: "))  
if (height <= 140):  
    print("Hi, You can enjoy a free ride")  
    price = 0
```

If-Else statement (Two-Way Decision)

- › The if...else statement is called alternative execution, in which there are two possibilities, and the condition determines which one gets executed.
- › do one thing if the condition is true, do something else if it's false.
- › If structure is good but not good enough, what if you want your program to "make a choice" based on a condition:
- › Syntax:

```
if (condition):  
    Body of if  
  
else:  
    Body of else
```

π Syntax of else

- › You create an else clause immediately following the if block with else, followed by a colon, followed by a block of statements.
- › The else statement must be in the same block as its corresponding if.
- › That is, the else and if must be indented the same amount; otherwise, your program will generate a nasty error.

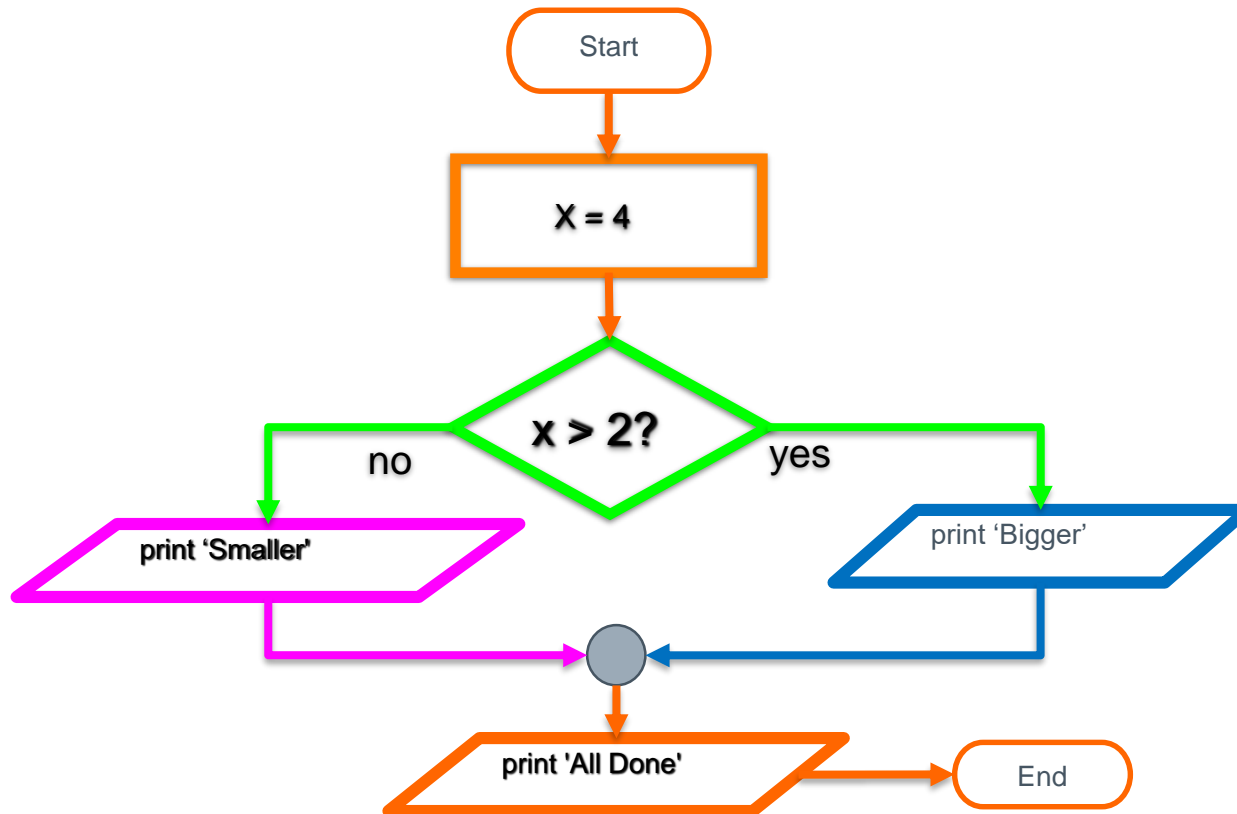
π If-Else statement (Two-Way Decision) – Example

```
if x < 0:  
    print("Negative")  
else:  
    print("Positive")
```

In an if-else structure, you're guaranteed that exactly one of the code blocks will execute.

```
if mark >= 50:  
    print("Pass")  
else:  
    print("Fail")
```

π If-Else statement (Two-Way Decision) – Example



π If-Else statement (Two-Way Decision) - Exercise

Program 1:

Write a program to check if a number is Odd or Even

```
n = int(input("Enter the number:"))
```

```
if (n % 2) == 0:
```

```
    print ("Given number is Even")
```

```
else:
```

```
    print(" Given number is Odd")
```


π If-Else statement (Two-Way Decision) - Exercise

Write a program that asks the user for their age.

- › If the age is **18 or older**, print "You are eligible to vote."
- › Otherwise, print "You are not eligible to vote."

π If-Else statement (Two-Way Decision) - Exercise

```
# Ask the user for their age
age = int(input("Enter your age: "))

# Check if the user is eligible to vote
if age >= 18:
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")
```

π Nested “if-else” statements

- › an “if” statement or “if-else” statement is present inside another if or if-else block.
- › We can write an entire if-else statement in another if-else statement called nesting, and the statement is called nested if.
- › In a nested if construct, you can have an if -elif -else construct inside an if –elif-else construct.
- › Python provides this feature as well, this in turn will help us to check multiple conditions in a given program.

π

Nested “if-else” statements - Syntax

Non-linear

```
if expression1:
    if expression2:
        statement(s)
    elif expression3:
        statement(s)
else:
    statement(s) .
```

Linear

```
if condition:
    statement

else:
    if condition:
        statement
    else:
        statement
```

π Nested “if-else” statements - Example

```
num = int(input("Enter the number:"))  
  
if (num<=15):  
    if(num == 10):  
        print ("play cricket")  
    else:  
        print("play PUBG")  
  
else:  
    print("Don't play game")
```

π Nested “if-else” statements - Example

Write a program that checks a person's eligibility for a job based on their **age** and **experience**.

The criteria are as follows:

- › If the person is **18 or older**:
 - If they have **2 or more years of experience**, they are eligible for the job.
 - Otherwise, they are not eligible.
- › If the person is **younger than 18**, they are **not eligible** for the job.

π Nested “if-else” statements - Example

```
age = int(input("Enter your age: "))
experience = int(input("Enter your years of experience: "))

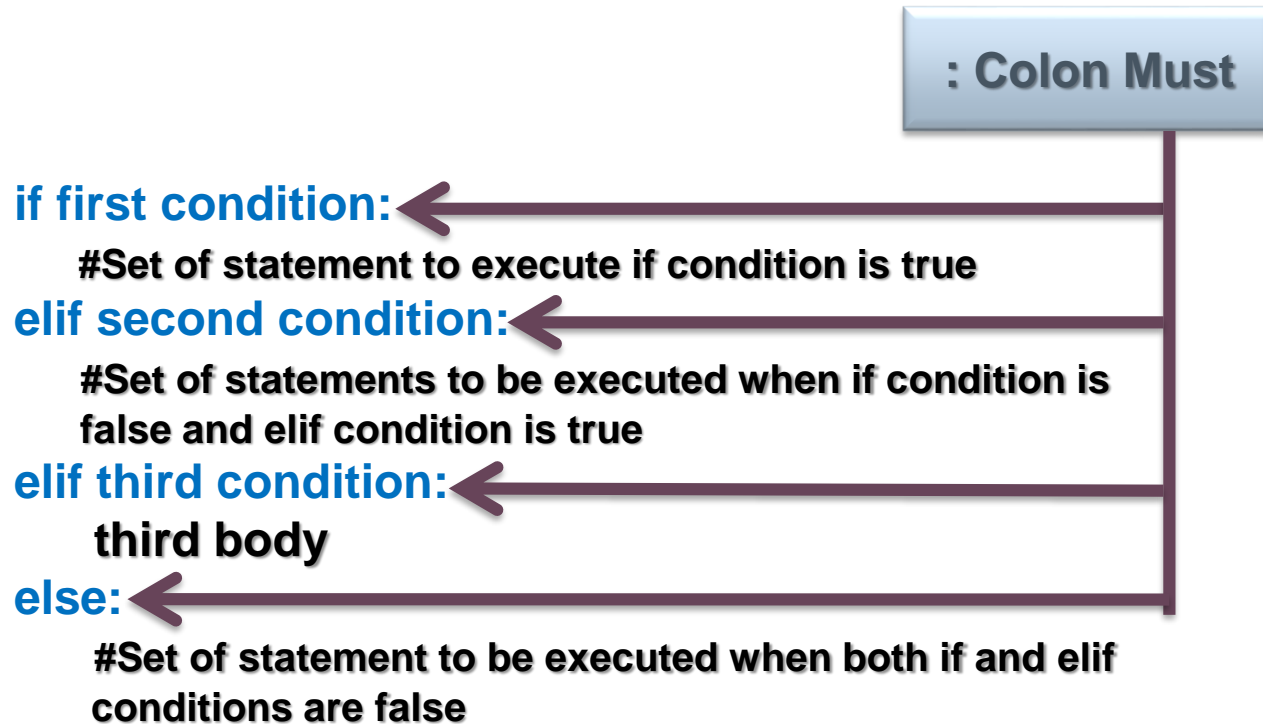
if age >= 18: # Check if the person is 18 or older
    if experience >= 2: # Check if they have 2 or more years of experience
        print("Eligible for the job.")
    else:
        print("Not eligible for the job.")
else:
    print("Not eligible for the job.")
```

π

if-elif-else statement (Multi-way Decision)

- › “elif” statement is used to check multiple conditions only if the given condition is false.
- › Similar to an “if-else” statement and the only difference is that in “else” we will not check the condition but in “elif” we will check the condition and evaluate multiple conditions.

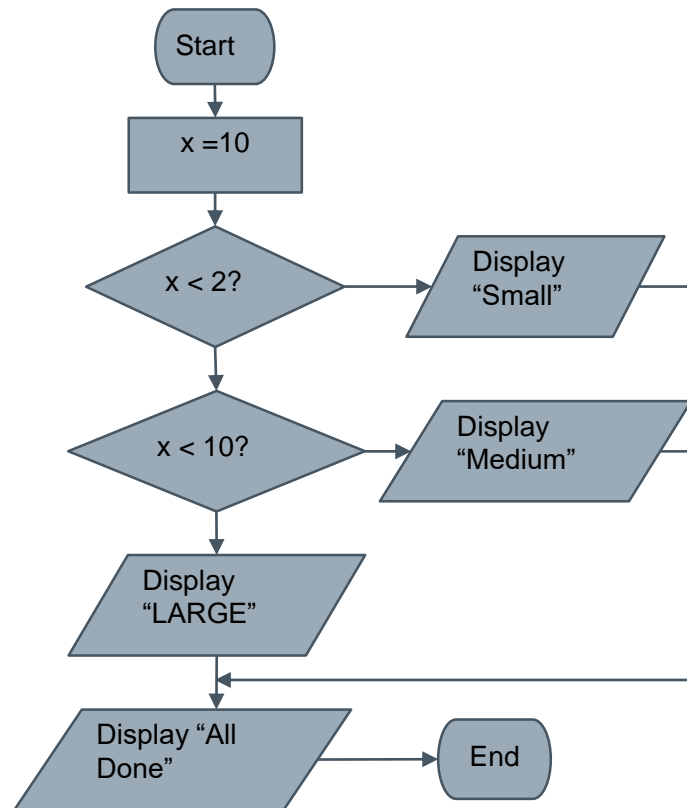
π if-elif-else statement(Multi-way Decision) - Syntax



π

if-elif-else statement (Multi-way Decision) - Example

```
x = 10  
if x < 2 :  
    print ('Small')  
elif x < 10 :  
    print ('Medium')  
else :  
    print ('LARGE')  
print ('All Done')
```



if-elif-else statement (Multi-way Decision) - Example

Program 2: Check if the number enter is a positive number, negative number or zero.

```
num = 10
```

```
if (num == 0):
```

```
    print("Number is Zero")
```

```
elif (num > 0):
```

```
    print("Number is Positive")
```

```
else:
```

```
    print("Number is Negative")
```

if-elif-else statement (Multi-way Decision) - Example

Write a program that checks whether a number is **positive**, **negative**, or **zero** and also identifies if the number is **even** or **odd**.

- If the number is **positive** and **even**, print "Positive Even".
- If the number is **positive** and **odd**, print "Positive Odd".
- If the number is **negative** and **even**, print "Negative Even".
- If the number is **negative** and **odd**, print "Negative Odd".
- If the number is **zero**, print "Zero".

π if-elif-else statement (Multi-way Decision) -

```
number = int(input("Enter a number: "))
```

```
if number == 0:
```

```
    print("Zero")
```

```
elif number > 0:
```

```
    if number % 2 == 0:
```

```
        print("Positive Even")
```

```
    else:
```

```
        print("Positive Odd")
```

```
else:
```

```
    if number % 2 == 0:
```

```
        print("Negative Even")
```

```
    else:
```

```
        print("Negative Odd")
```

π

if-elif-else statement (Multi-way Decision)

– Example: No Else

```
x = 5
```

```
if x < 2 :
```

```
    print ('Small')
```

```
elif x < 10 :
```

```
    print ('Medium')
```

```
print ('All Done')
```