IT Department

Selection Statements (if-else)

Soma Solaiman Zadeh

Programming I (IT 117)

Fall 2024 - 2025

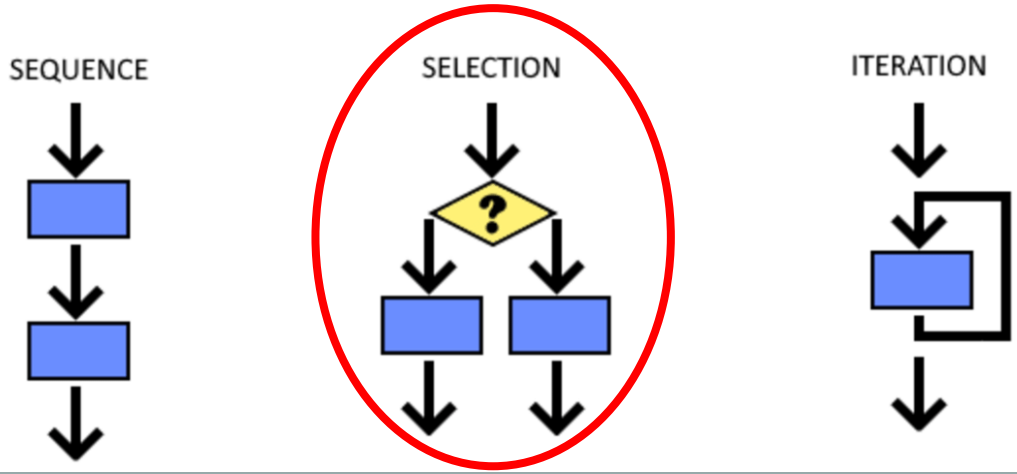
Week 5

January 15, 2025

Outline

- **Selection** Control Structures
- **IF** Statement
- **Nested IF**
- **Logical** Operators

SELECTION Control Structure



SELECTION Statements

- **SELECTION** statements in Python:
  - **IF** Statement
    - **if** → One-way decision
    - **if-else** → Two-way decisions
    - **if-elif-…-else** → Multi-way decisions
- **Switch** control structure is another way to implement a selection process in programming languages. Python unlike other programming languages doesn't have a built-in **SWITCH** control structure, but user-defined structures can implement it.

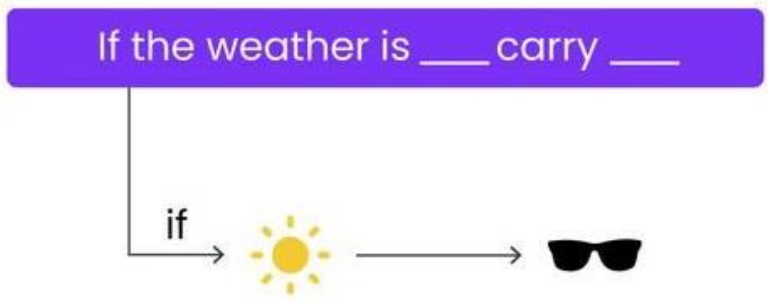
IF Statement

- IF statement is used to **make a decision**.
- **IF** statement: If a condition (or conditions) is True, it executes a set of commands (The If block statements).
  - Otherwise, the set of commands is skipped.

```
if  condition :
        execute statements
```

Indentation

IF Statement



IF Statement

```
temperature = -10

if (temperature<0) :
    print("It is below freezing point!")
```

Output

It is below freezing point!

Indentation

- **Indentation** is whitespaces at the beginning of a line in a Python code.
- **Indentation** defines scope in code. For example, in **if** statement, we use indentation to define the scope of **if** statement.
- Other programming languages use curly-brackets for this purpose.

```
temperature = -10

if (temperature<0) :
    print("It is below freezing point!")
```

```
temperature = -10

if (temperature < 0) :
print("It is below freezing point!")
```

IF Statement

```
temperature = -10

if (temperature<0) :
    print("It is below freezing point!")
    print("Wear a coat and hat")
    print("It is", temperature,"degrees!")
```

Output

```
It is below freezing point!
Wear a coat and hat
It is -10 degrees!
```

Multiple IF Statements

```
temperature = 45
if (temperature>40) :
    print("Temperature is > 40")

if (temperature>20) :
    print("Temperature is > 20")

if (temperature>0) :
    print("Temperature is above freezing point!")
```

Output

```
Temperature is >40
Temperature is >20
Temperature is above freezing point!
```

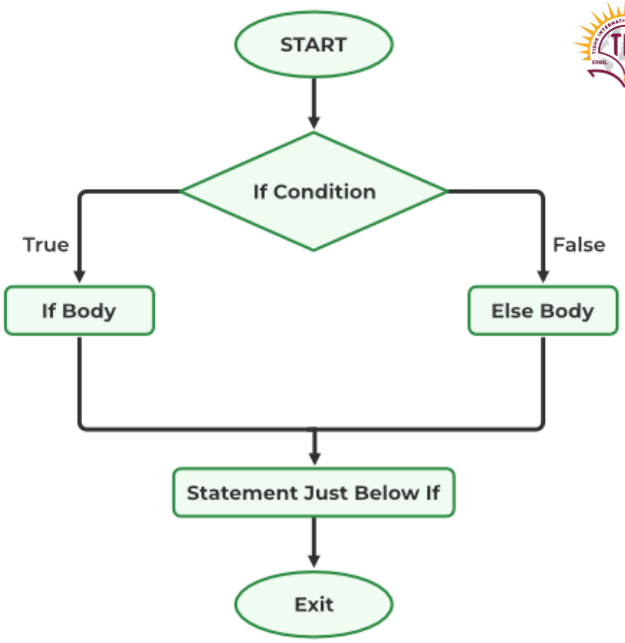
Forming Simple Conditions

- Relational expressions using relational operators are used as conditions in **IF** statement.
  - 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

```
a = 7
b = 10
if a < b :
    print('a is less than b')
```

IF Flowchart

(Two-Way Decisions)



if-else Statement (Two Way Decisions)

- A two-way decision can be implemented by attaching an **else** clause to an **if** clause.
- The **else** keyword is **optional**.
- The **else** keyword is used to decide what to do if the condition is **False**.

```
if   condition :
        execute statements
else :
        execute statements
```

if-else Statement (Two Way Decisions)

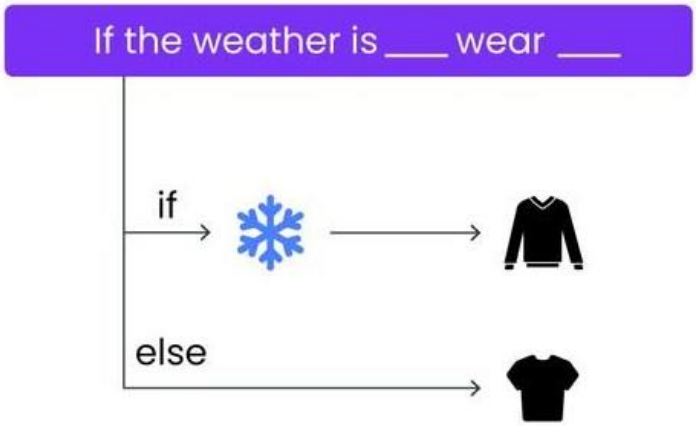
- If the condition is **True**, the **if block is executed**, and the else block is skipped.
- If the condition is **False**, the **else block is executed**, and the if block is skipped.

```
if   condition :
        statement1
        statement2
        ...
else :
        statement1
        statement2
        ...
```

if Block

else Block

if-else Statement



if-else Statement

```
a = 100
b = 40

if (b > a) :
    print("b is greater than a!")
else:
    print("b is not greater than a!")
```

Output

b is not greater than a!

if-else Statement

```
mark = int(input('Enter your mark in Programming 1: '))

if (mark >= 80) :
    print("Well Done!")
else:
    print("Practice More!")
```

Output if the user enters 60 as their mark

```
Practice More!
```

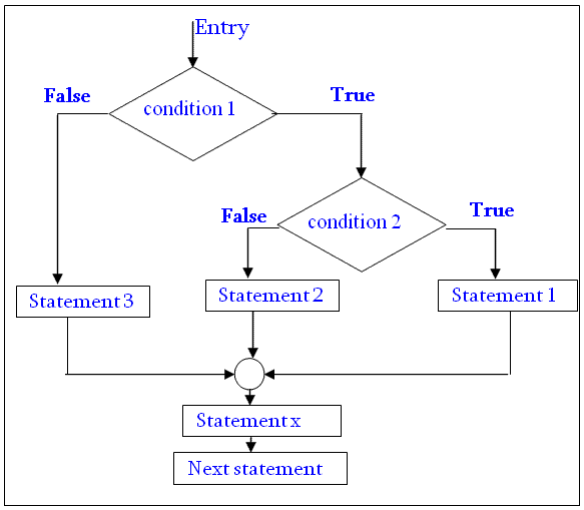
Nested IF Statement

- You can have **IF** statement inside another **IF** statement, which is called **Nested IF** statement.

```
if condition :
    if condition :
        statement1
    else:
        statement2
else :
    statement3
```

Nested **if-else** statement

Flowchart of Nested IF




```
if condition1 :
        if condition2 :
                Statement1
        else :
                Statement2
else :
        Statement3
```

Nested IF Statement

```
num = 50

if (num > 10) :
    print("Above ten, ")

    if (num > 30) :
        print("and also above 30!")
    else:
        print("but not above 30!")
```

Output

```
Above ten,
and also above 30!
```

Let's Try it more!

```
quiz1 = int(input('Enter mark of first quiz: '))
quiz2 = int(input('Enter mark of second quiz: '))
quiz3 = int(input('Enter mark of third quiz: '))

average = (quiz1 + quiz2 + quiz3) / 3

if (average < 50 ):
    print('Failed!')
else:
    if (average < 80):
        print ('Nice! but you need to practice more!')
    else:
        print ('Well Done!')
```

```
a = 5
b = 10

if(a >= 5):
    if(b != 10):
        print ("Option A")
    else:
        print ("Option B")

else:
    if(b < 11):
        print ("Option C")
    else:
        print ("Option D")
```

Output

Option B

- First **if** statement is evaluated → **if (a >=5)**
- The condition **(a >=5)** is **True**, so the <u>if block is executed</u> and the else block is skipped.
- Now the second **if** statement is evaluated → **if (b != 10)**
- The condition **(b != 10)** is **False**, so <u>the else block is executed</u> and the if block is skipped.

```
a = 4
b = 10

if(a >= 5):
    if(b != 10):
        print ("Option A")
    else:
        print ("Option B")

else:
    if(b < 11):
        print ("Option C")
    else:
        print ("Option D")
```

Output → **Option C**

- First **if** statement is evaluated → **if (a >=5)**
- The condition **(a >=5)** is **False**, so the else block is executed and the if block is skipped.
- Now the second **if** statement is evaluated → **if (b < 11)**
- The condition **(b < 11)** is **True**, so the if block is executed and the else block is skipped.

```
a = 4
b = 13

if(a >= 5):
    if(b != 10):
        print ("Option A")
    else:
        print ("Option B")

else:
    if(b < 11):
        print ("Option C")
    else:
        print ("Option D")
```

Output → **Option D**

```
a = "R"
b = ""

if(a == "S"):
    if(b != "T"):
        print ("Option A")
    else:
        print ("Option B")

else:
    if(b == "T"):
        print ("Option C")
    else:
        print ("Option D")
```

Output

Option D

Logical Operators

- Logical operators connect two or more conditions (relational expressions) into one or reverse the logic of a condition.

Operator	Description	Example	Example's Result
and	Returns **True** if both statements are **True**, otherwise it returns **False**.	**2 <= 5 and 9 > 3**	**True**
		10 > 20 and 20 < 30	**False**
or	Returns **True** if at least one of the statements is **True**. It returns **False** only if both statements are **False**.	**2 < 5 or 6 >= 9**	**True**
		8 <= 5 or 7 > 9	**False**
not	Reverse the result. It returns **False** if the statement is **True**, and returns **True** if the statement is **False**.	**not (10 > = 5)**	**False**
		not (10 < 6)	**True**

Let's Try it!

- What is the result value of each expression?

(5 < 2) **and** (5 > 3)	→	**False**
(5 < 9) **and** (5 > 3)	→	**True**
(5 < 2) **or** (5 >= 3)	→	**True**
(5 < 2) **or** (5 > 10)	→	**False**
not (5 > 10)	→	**True**
not (5 <= 10)	→	**False**

IF Statement Using Logical Operators

```
a = 100
b = 50
c = 400

if (a > b) and (c > a):
    print('Both conditions are True!')
else:
    print("At least one of the conditions is False!")
```

Output

```
Both conditions are True!
```

IF Statement Using Logical Operators

```
a = 100
b = 50
c = 400

if (a > b) or (a > c):
    print('At least one of the conditions is True!')
else:
    print("Both conditions are False!")
```

Output

At least one of the conditions is True!

IF Statement Using Logical Operators

```
a = 100
b = 50
c = 400

if (a < b) or not(c >= b):
    print('At least one of the conditions is True!')
else:
    print("Both conditions are False!")
```

Output

Both conditions are False!