Python Programming Language

Prepared by: Mohamed Ayman

Algorithm Engineer at Valeo

Deep Learning Researcher and Teaching Assistant

at The American University in Cairo (AUC)

spring 2020

sw.eng.MohamedAyman@gmail.com

facebook.com/cs.MohamedAyman

linkedin.com/in/cs-MohamedAyman

github.com/cs-MohamedAyman

codeforces.com/profile/Mohamed_Ayman

Lecture 4

Conditions

Course Roadmap

Part 1: Python Basics and Functions

Lecture 1: Python Overview

Lecture 2: Variable Types

Lecture 3: Basic Operations

Lecture 4: Conditions

Lecture 5: Loops

Lecture 6: Functions

Lecture Agenda

We will discuss in this lecture the following topics

1- Decision making Definition

2- IF Statement

3- IF and ELSE Statements

4- IF, ELIF and ELSE Statements

5- Nested IF Statements

6- Single Statement Suites


Let's
START UP

Lecture Agenda

Section 1: Decision making Definition

Section 2: IF Statement

Section 3: IF and ELSE Statements

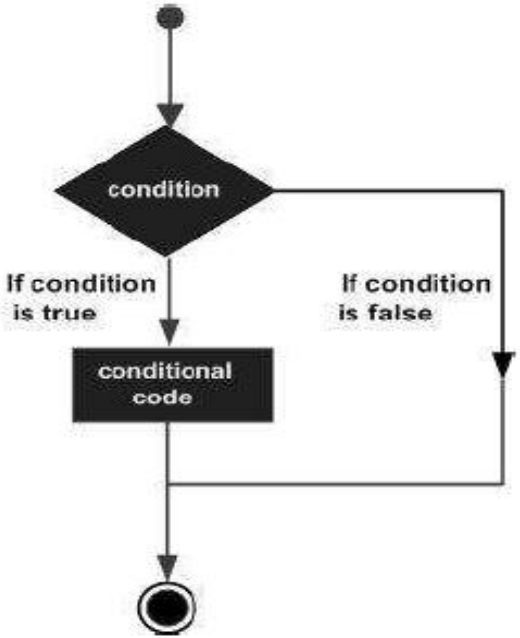
Section 4: IF, ELIF and ELSE Statements

Section 5: Nested IF Statements

Section 6: Single Statement Suites

Decision making Definition

- Decision making is required when we want to execute a code only if a certain condition is satisfied.
- Decision-making is the anticipation of conditions occurring during the execution of a program and specified actions taken according to the conditions.
- Decision making structure evaluate multiple expressions, which produce TRUE or FALSE as the outcome. You need to determine which action to take and which statement to execute if the outcome is TRUE or FALSE otherwise.
- The general form of a typical decision making structure found in most of the programming language.



Decision making Definition

- Python programming language assume any non-zero and non-null values as TRUE, and any zero or null values as FALSE values.
- Python programming language provides the following types of decision making statement

Statement	Description
if statements	An if statement consists of a Boolean expression followed by one or more statements.
if...else statements	An if statement can be followed by an optional else statement, which executes when the boolean expression is FALSE.
nested if statements	You can use one if or else if statement inside another if or else if statement(s).

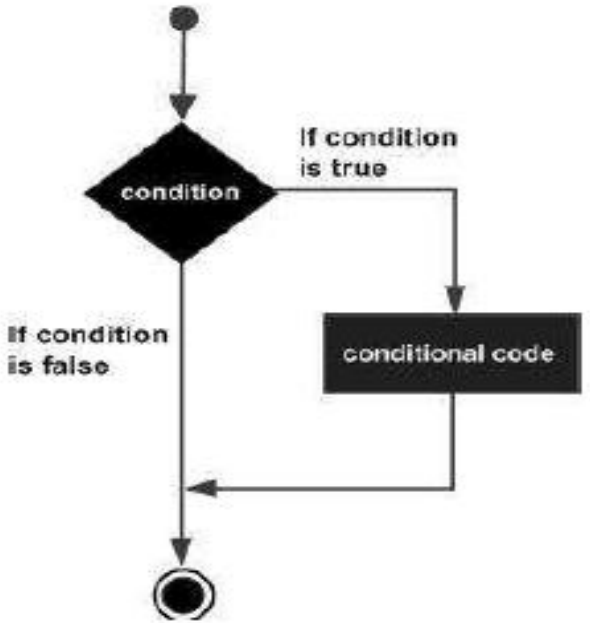
Lecture Agenda

- ✔ Section 1: Decision making Definition
- **Section 2: IF Statement**
- Section 3: IF and ELSE Statements
- Section 4: IF, ELIF and ELSE Statements
- Section 5: Nested IF Statements
- Section 6: Single Statement Suites

IF Statement

- The IF statement is similar to that of other language. The if statement contains a logical expression using which the data is compared and a decision is made based on the result of the comparison.
- In Python, the body of the if statement is indicated by the indentation. Body starts with an indentation and the first un-indented line marks the end.
- If Syntax:

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



IF Statement

Example:

```
x = 3
if x > 0:
    print("it's a positive number")
x = -3
if x > 0:
    print("it's a positive number")
```

Output:

```
it's a positive number
```

- In the above example, n › 0 is the test expression.

The body of if is executed only if this evaluates to True.

When variable n is greater than 0, test expression is true and body inside body of if is executed.

If variable n is less than 0, test expression is false and body inside body of if is skipped.

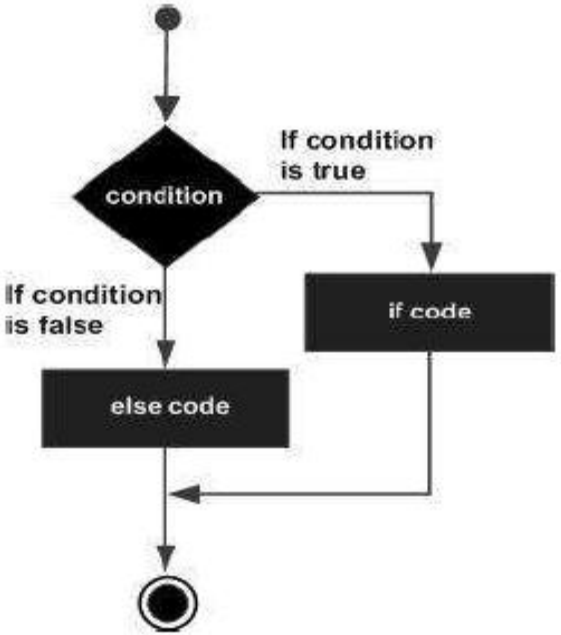
Lecture Agenda

- ✔ Section 1: Decision making Definition
- ✔ Section 2: IF Statement
- **Section 3: IF and ELSE Statements**
- Section 4: IF, ELIF and ELSE Statements
- Section 5: Nested IF Statements
- Section 6: Single Statement Suites

IF and ELSE Statements

- An else statement can be combined with an if statement. An else statement contains a blocks of code that executes if the conditional expression in the if statement resolve to 0 or a FALSE value.
- The else statement is an optional statement and there could be at the most only one else statement following if.
- If and else syntax:

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



IF and ELSE Statements

Example:

```
n = 3
if n % 2 == 0:
    print("it's an even number")
else:
    print("it's an odd number")
```

Output:

```
it's an odd number
```

In the above example,

- when n is even, the test expression is true and body of if is executed and body of else is skipped,
- when n is odd, the test expression is false and body of else is executed and body of if is skipped.

Quiz

- Which of the following programs will print True, according to python conditions?

☐
```
if 0:
    print('True')
else:
    print('False')
```

☐
```
if '':
    print('True')
else:
    print('False')
```

☐
```
if None:
    print('True')
else:
    print('False')
```

☐
```
if -3:
    print('True')
else:
    print('False')
```

☐
```
if 3:
    print('True')
else:
    print('False')
```

☐
```
if []:
    print('True')
else:
    print('False')
```

Quiz Solution

- Which of the following programs will print True, according to python conditions?

☐
```
if 0:
    print('True')
else:
    print('False')
```

☐
```
if '':
    print('True')
else:
    print('False')
```

☐
```
if None:
    print('True')
else:
    print('False')
```

☑
```
if -3:
    print('True')
else:
    print('False')
```

☑
```
if 3:
    print('True')
else:
    print('False')
```

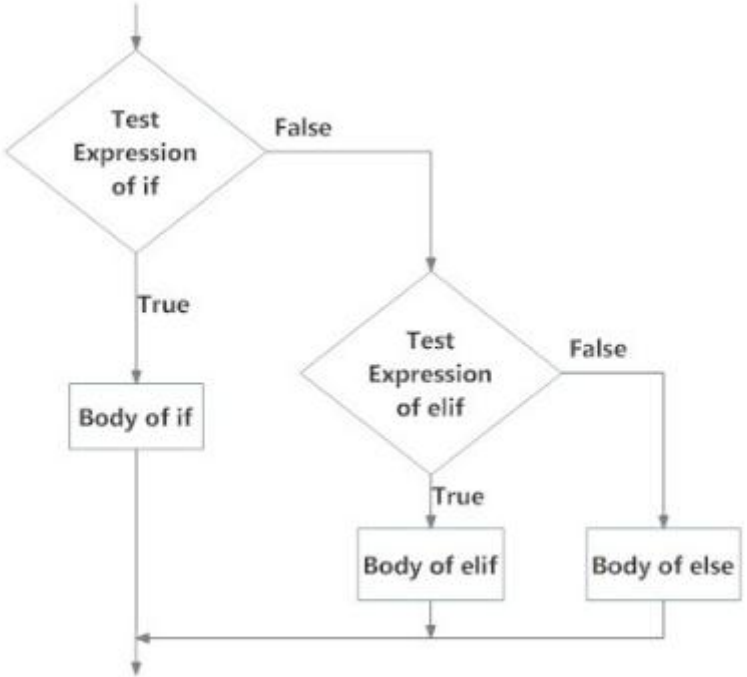
☐
```
if []:
    print('True')
else:
    print('False')
```

Lecture Agenda

- ✔ Section 1: Decision making Definition
- ✔ Section 2: IF Statement
- ✔ Section 3: IF and ELSE Statements
- **Section 4: IF, ELIF and ELSE Statements**
- Section 5: Nested IF Statements
- Section 6: Single Statement Suites

IF, ELIF and ELSE Statements

- 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, 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.



IF, ELIF and ELSE Statements

- 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, 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.

```
if expression:
    statement(s)
elif expression:
    statement(s)
elif expression:
    statement(s)
.
.
.
else:
    statement(s)
```

IF, ELIF and ELSE Statements

Example:

```
n = 3
if n > 0:
    print("It's a positive number")
elif n < 0:
    print("It's a negative number")
else:
    print("It's zero")
```

Output:

```
It's a poitive number
```

- When variable n is positive, positive statement is printed.
- When variable n is negative, Negative statement is printed
- When variable n is equal to 0, Zero statement is printed.

Summary

- At any condition block we have if statement 1 time, elif statement between 0 and infinite times and else statement 0 or 1 time.

```
if expression:          1 time
    statement(s)
elif expression':       [0, oo) times
    statement(s)
else:                   [0, 1] time
    statement(s)
```

Quiz

- What is the maximum number of lines will be printed, according to python conditions?

```
if expression:
    print('Line')
elif expression':
    print('Line')
elif expression'':
    print('Line')
else:
    print('Line')
```

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3

```
if expression:
    print('Line')
else:
    print('Line')
```

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3

```
if expression:
    print('Line')
elif expression':
    print('Line')
elif expression'':
    print('Line')
else:
    print('Line')

if expression:
    print('Line')
else:
    print('Line')
```

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3

Quiz Solution

- What is the maximum number of lines will be printed, according to python conditions?

```
if expression:
    print('Line')
elif expression':
    print('Line')
elif expression'':
    print('Line')
else:
    print('Line')
```

- [ ] 0
- [ ] 2
- [x] 1
- [ ] 3

```
if expression:
    print('Line')
else:
    print('Line')
```

- [ ] 0
- [ ] 2
- [x] 1
- [ ] 3

```
if expression:
    print('Line')
elif expression':
    print('Line')
elif expression'':
    print('Line')
else:
    print('Line')

if expression:
    print('Line')
else:
    print('Line')
```

- [ ] 0
- [x] 2
- [ ] 1
- [ ] 3

Practice

Problem Statement

- Find the largest among three numbers using python conditions, such that the input number data type will be integer and positive.
- Test Cases:

Test Case 1	Test Case 2	Test Case 3
30 10 20	15 15 5	40 40 40
30	15	40

Problem Solution

- Find the largest among three numbers using python conditions, such that the input number data type will be integer and positive.
- Test Cases:

Test Case 1	Test Case 2	Test Case 3
30 10 20	15 15 5	40 40 40
30	15	40

```
x, y, z = map(int, input().split())
if x >= y and x >= z:
    print(x)
elif y >= x and y >= z:
    print(y)
else:
    print(z)
```

Problem Statement

- Find the category of the courses total for a student using python conditions, such that the input number data type will be integer and positive between [0 and 100].
- Category Excellent between [85 - 100], Category Very Good between [75 - 84], Category Good between [65 - 74], Category Pass between [50 - 64] and less than 50 is Fail.

Test Case 1	Test Case 2	Test Case 3	Test Case 4
88	77	66	55
Excellent	Very Good	Good	Pass

Problem Solution

- Find the category of the courses total for a student using python conditions, such that the input number data type will be integer and positive between [0 and 100].
- Category Excellent between [85 - 100], Category Very Good between [75 - 84], Category Good between [65 - 74], Category Pass between [50 - 64] and less than 50 is Fail.

Test Case 1	Test Case 2	Test Case 3	Test Case 4
88	77	66	55
Excellent	Very Good	Good	Pass

```
x = int(input())
if x >= 85: print('Excellent')
elif x >= 75: print('Very Good')
elif x >= 65: print('Good')
elif x >= 50: print('Pass')
else: print('Fail')
```

Lecture Agenda

- ✔ Section 1: Decision making Definition
- ✔ Section 2: IF Statement
- ✔ Section 3: IF and ELSE Statements
- ✔ Section 4: IF, ELIF and ELSE Statements
- **Section 5: Nested IF Statements**
- Section 6: Single Statement Suites

Nested IF Statements

- There may be a situation when you want to check for another condition after a condition resolve to true. In such a situation, you can use the nested if construct.
- In a nested if construct, you can have an if ... elif ... else construct inside another if ... elif ... else construct.
- We can have a if...elif...else statement inside another if...elif...else statement. This is called nesting in computer programming.
- Any number of these statements can be nested inside one another. Indentation is the only way to figure out the level of nesting. This can get confusing, so must be avoided if we can.

```
if expression:
    statement(s)
elif expression:
    statement(s)
    if expression:
        statement(s)
    elif expression:
        statement(s)
    else:
        statement(s)
    statement(s)
elif expression:
    statement(s)
    if expression:
        statement(s)
    statement(s)
else:
    statement(s)
```

Nested IF Statements

- There may be a situation when you want to check for another condition after a condition resolve to true. In such a situation, you can use the nested if construct.
- In a nested if construct, you can have an if ... elif ... else construct inside another if ... elif ... else construct.
- We can have a if...elif...else statement inside another if...elif...else statement. This is called nesting in computer programming.
- Any number of these statements can be nested inside one another. Indentation is the only way to figure out the level of nesting. This can get confusing, so must be avoided if we can.

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

Nested IF Statements

Example:

```
n = 6
if n % 2 == 0:
    if n % 3 == 0:
        print('this number divisible by 2 and 3')
    else:
        print('this number divisible by 2')
else:
    if n % 3 == 0:
        print('this number divisible by 3')
    else:
        print('this number not divisible by 2 nor 3')
```

Output:

```
this number divisible by 2 and 3
```

- This program, check for a number which is divisible by 2 or 3 or both or none of them.

Equivalent to Nested IF Statements

Example:

```
n = 6
if n % 2 == 0 and n % 3 == 0:
    print('this number divisible by 2 and 3')
elif n % 2 == 0:
    print('this number divisible by 2')
elif n % 3 == 0:
    print('this number divisible by 3')
else:
    print('this number not divisible by 2 nor 3')
```

Output:

```
this number divisible by 2 and 3
```

- This program, check for a number which is divisible by 2 or 3 or both or none of them.

Quiz

- What is the maximum number of lines will be printed, according to python conditions?

```
if expression:
    if expression':
        print('Line')
    else:
        print('Line')
else:
    if expression':
        print('Line')
    else:
        print('Line')
```

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3

```
if expression:
    if expression':
        print('Line')
    else:
        print('Line')
elif expression'':
    if expression':
        print('Line')
    else:
        print('Line')
else:
    if expression':
        print('Line')
    else:
        print('Line')
```

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3

Quiz Solution

- What is the maximum number of lines will be printed, according to python conditions?

```
if expression:
    if expression':
        print('Line')
    else:
        print('Line')
else:
    if expression':
        print('Line')
    else:
        print('Line')
```

☐ 0 ☐ 2

☑ 1 ☐ 3

```
if expression:
    if expression':
        print('Line')
    else:
        print('Line')
elif expression'':
    if expression':
        print('Line')
    else:
        print('Line')
else:
    if expression':
        print('Line')
    else:
        print('Line')
```

☐ 0 ☐ 2

☑ 1 ☐ 3

Practice

Problem Statement

- Find the category and subcategory of the courses total for a student using python conditions, such that the input number data type will be integer and positive between [0 and 100].
- Category Excellent between [85 - 90 - 100], Category Very Good between [75 - 80 - 84], Category Good between [65 - 70 -74], Category Pass between [50 - 60- 64] and less than 50 is Fail.

Test Case 1	Test Case 2	Test Case 3	Test Case 4
88	77	66	55
Excellent A	Very Good B	Good C	Pass D

Problem Solution

- Category Excellent between [85 - 90 - 100],
- Category Very Good between [75 - 80 - 84],
- Category Good between [65 - 70 -74],
- Category Pass between [50 - 60- 64]

and less than 50 is Fail.

```
x = int(input())
if x >= 85:
    print('Excellent')
    if x >= 90: print('A+')
    else: print('A')
elif x >= 75:
    print('Very Good')
    if x >= 80: print('B+')
    else: print('B')
elif x >= 65:
    print('Good')
    if x >= 70: print('C+')
    else: print('C')
elif x >= 50:
    print('Pass')
    if x >= 60: print('D+')
    else: print('D')
else:
    print('Fail')
```

Lecture Agenda

- ✔ Section 1: Decision making Definition
- ✔ Section 2: IF Statement
- ✔ Section 3: IF and ELSE Statements
- ✔ Section 4: IF, ELIF and ELSE Statements
- ✔ Section 5: Nested IF Statements

Section 6: Single Statement Suites

Single Statement Suites

- You can write complete if statement as an expression in single line like that, but it should have else statement.

```
[statement if expression else] [statement if expression else] ... statement
```

Example:

```
n = 3
print('positive') if n > 0 else print('negative') if n < 0 else print('zero')
print('positive' if n > 0 else 'negative' if n < 0 else 'zero')

n = 3
print('even') if n % 2 == 0 else print('odd')
print('even' if n % 2 == 0 else 'odd')
```

Output:

```
positive
positive
odd
odd
```

Problem Solution

- Find the category and subcategory of the courses total for a student using python conditions, such that the input number data type will be integer and positive between [0 and 100].
- Category Excellent between [85 - 90 - 100], Category Very Good between [75 - 80 - 84], Category Good between [65 - 70 -74], Category Pass between [50 - 60- 64] and less than 50 is Fail.

Test Case 1	Test Case 2	Test Case 3	Test Case 4
88	77	66	55
Excellent A	Very Good B	Good C	Pass D

```
x = int(input())
if x >= 85: print('Excellent ' + ('A+' if x >= 90 else 'A'))
elif x >= 75: print('Very Good ' + ('B+' if x >= 80 else 'B'))
elif x >= 65: print('Good ' + ('C+' if x >= 70 else 'C'))
elif x >= 50: print('Pass ' + ('D+' if x >= 60 else 'D'))
else: print('Fail')
```

Lecture Agenda

- ✓ Section 1: Decision making Definition
- ✓ Section 2: IF Statement
- ✓ Section 3: IF and ELSE Statements
- ✓ Section 4: IF, ELIF and ELSE Statements
- ✓ Section 5: Nested IF Statements
- ✓ Section 6: Single Statement Suites


DO
MORE.