

LECTURE

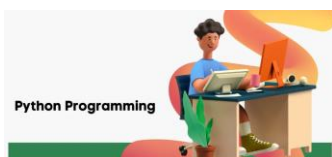
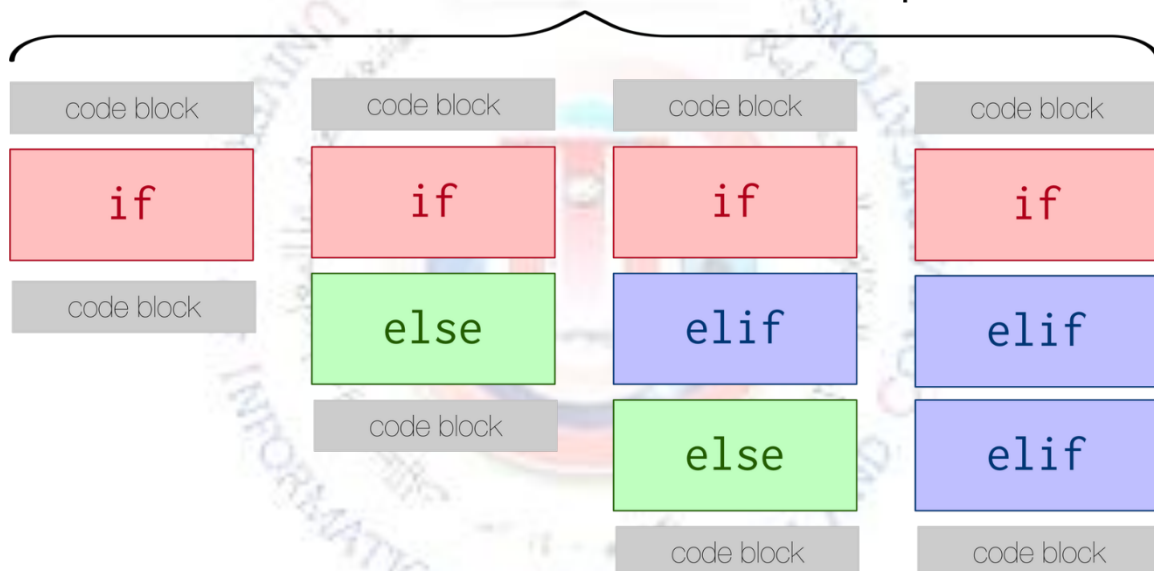
EIGHT&NINE



CONDITIONAL STATEMENTS

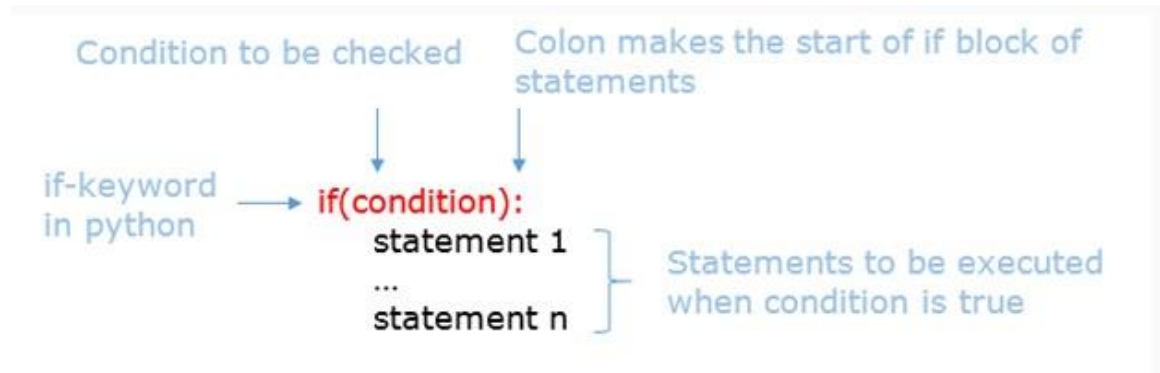
Decision making is required when we want to execute a code only if a certain condition is satisfied. Implementing decision making gives the power to incorporate branching in a program. As stated earlier, a program is a set of instructions given to a computer. The instructions are given to accomplish a task and any task requires making decisions. So, conditional statements form an integral part of programming. The syntax of the construct is as follows:

valid if/elif/else order examples



Python if Statement Syntax

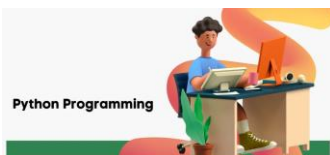
The general form of the **if** statement is:



- The reserved word **if** begins a if statement.
- The condition is a Boolean expression that determines whether or not the body will be executed. A colon (:) must follow the condition.
- The block is a block of one or more statements to be executed if the condition is true. The statements within the block must all be indented the same number of spaces from the left.

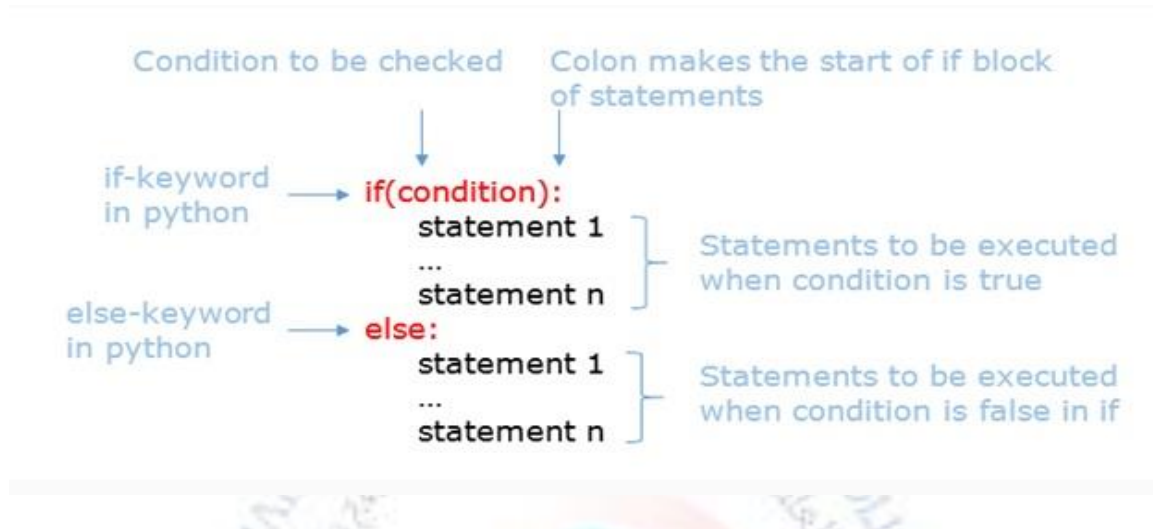
Example (1): Write Python Program to read any two number and print the largest value between them.

```
x= int(input('Enter a number : '))  
y= int(input('Enter a number : '))  
if x >=y:  
    print("largest ",x)  
if y >x:  
    print("largest ",y)
```



Python if/else Statement Syntax

The general form of an **if/else** statement is

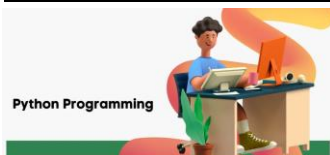


- The reserved word **if** begins the **if/else** statement.
- The *condition* is a Boolean expression that determines whether or not the **if** block or the **else** block will be executed. A colon (:) must follow the condition.
- The *if-block* is a block of one or more statements to be executed if the condition is true. As with all blocks, it must be indented one level deeper than the **if** line. This part of the **if** statement is sometimes called the body of the **if**.
- The reserved word **else** begins the second part of the **if/else** statement. A colon (:) must follow the **else**.
- The *else-block* is a block of one or more statements to be executed if the condition is false. It must be indented one level deeper than the line with the **else**. This part of the **if/else** statement is sometimes called the body of the **else**.

Example (2): Write Python Program to read degree of student and check if it's pass or fail.

```

d= int(input('Enter a number : '))
if d >=50:
    print(" pass ")
else:
    print(" fail ")
  
```



Example (3): Write Python Program to read number and check if it's even or odd.

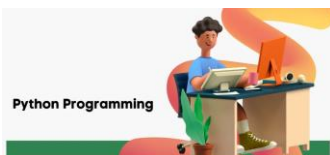
```
x= int(input('Enter a number : '))
if x %2==0:
    print(" even ")
else:
    print ("odd")
```

Example (4): Write Python Program to read number and check if it's positive or negative.

```
x= int(input('Enter a number : '))
if x >=0:
    print("positive ")
else:
    print ("negative ")
```

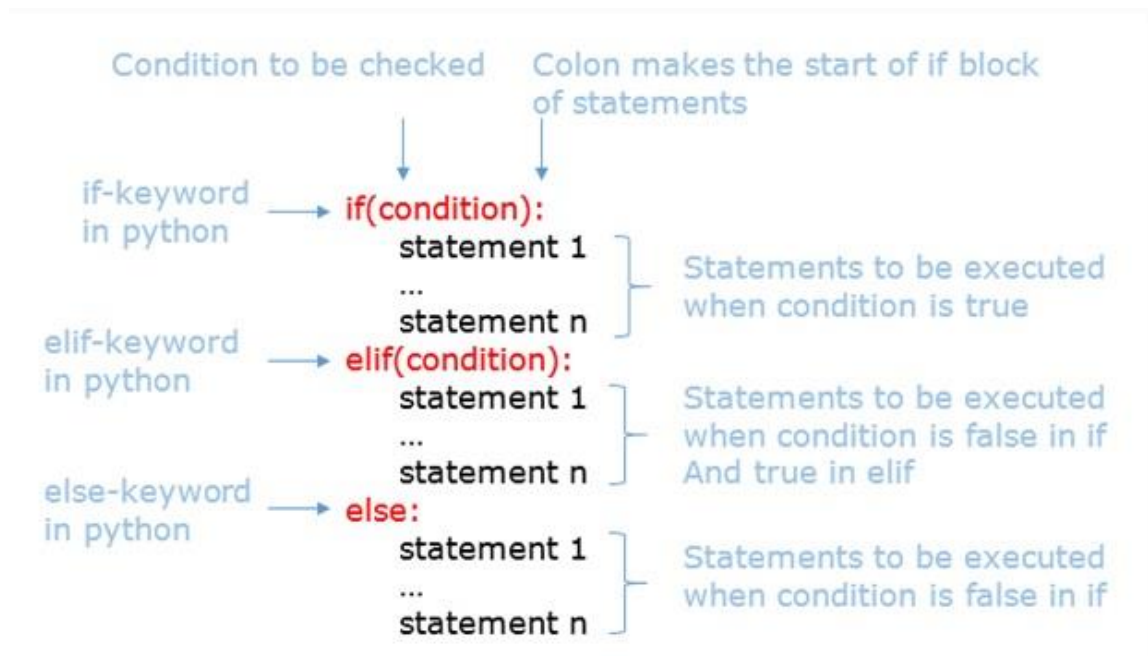
Example (5): Write Python Program to find the greater number between two numbers by if/else

```
x= int(input('Enter a number : '))
y= int(input('Enter a number : '))
if x >=y:
    print(x)
else:
    print (y)
```



Python if/elif/else Statement Syntax (multi-way conditional)

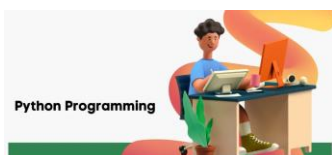
Python provides a multi-way conditional construct called **if/elif/else** that permits a more manageable textual structure for programs that must check many conditions.



Example (6): Write Python Program to read three sides of a triangle and check a triangle is equilateral, isosceles or scalene..

```
x = int(input("x: "))
y = int(input("y: "))
z = int(input("z: "))

if x == y == z:
    print("Equilateral triangle")
elif x==y or y==z or z==x:
    print("isosceles triangle")
else:
    print("Scalene triangle")
```



Example (7): Write Python Program to read number and convert to writing number

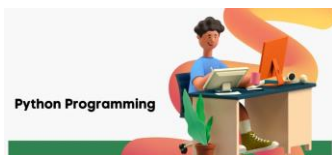
```
value = int(input("Please enter value: "))
if value < 0:
    print("Too small")
elif value == 0:
    print("zero")
elif value == 1:
    print("one")
elif value == 2:
    print("two")
elif value == 3:
    print("three")
elif value == 4:
    print("four")
elif value == 5:
    print("five")
else:
    print("Too large")
```

Nested Conditionals

Nested if statements to develop arbitrarily complex program logic.

Example (8): Write Python Program to read number and determines if a number is between 0 and 10.

```
value = int(input("enter value "))
if value >= 0:    # First check
    if value <= 10: # Second check
        print("In range")
```



Compound Boolean Expressions

We can combine simple Boolean expressions, each involving one relational operator, into more complex Boolean expressions using the logical operators **and** , **or** , and **not** . A combination of two or more Boolean expressions using logical operators is called a compound Boolean expression.

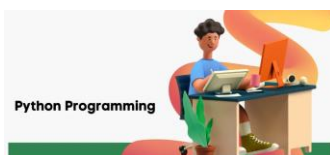
Example (9): Write Python Program to read an integer value and check if value in the range 0 to 10.

```
value = int(input("enter value "))
if value >= 0 and value <= 10:
    print("In range")
```

Multi-way Versus Sequential Conditionals

<pre># Use a multi-way conditional statement value = int(input()) if value == 0: print("zero") elif value == 1: print("one") elif value == 2: print("two") elif value == 3: print("three") elif value == 4: print("four") elif value == 5: print("five") print("Done")</pre> <p>Check #1 points to <code>if value == 0:</code> Check #2 points to <code>elif value == 1:</code> Text to print points to <code>print("one")</code> A red bracket groups the remaining <code>elif</code> statements with the text "Other checks skipped".</p>	<pre># Use sequential if statements value = int(input()) if value == 0: print("zero") if value == 1: print("one") if value == 2: print("two") if value == 3: print("three") if value == 4: print("four") if value == 5: print("five") print("Done")</pre> <p>Check #1 points to <code>if value == 0:</code> Check #2 points to <code>if value == 1:</code> Check #3 points to <code>if value == 2:</code> Check #4 points to <code>if value == 3:</code> Check #5 points to <code>if value == 4:</code> Check #6 points to <code>if value == 5:</code> Text to print points to <code>print("one")</code></p>
--	--

Note: input value is 1



Conditional Expressions

Python provides an alternative to the **if/else** construct called a conditional expression. A conditional expression evaluates to one of two values depending on a Boolean condition.

expression-1 **if** *condition* **else** *expression-2*



finxter

Python One Line If-Then-Else

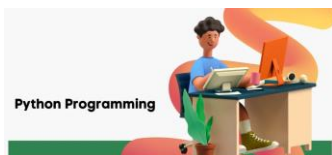
```
if pw == '123':  
    x = 'yes'  
else:  
    x = 'no'
```



```
x = 'yes' if pw == '123' else 'no'
```

Where:

- **expression-1** is the overall value of the conditional expression if condition is true.
- **condition** is a normal Boolean expression that might appear in an if statement.
- **expression-2** is the overall value of the conditional expression if condition is false.



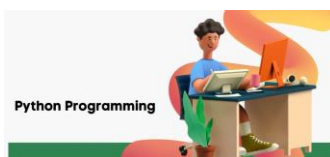
Example: Conditional Expressions

```
if a != b:  
    c = d  
else:  
    c = e
```

```
c = d if a != b else e
```

Errors in Conditional Statements

- Confusing logical **and** and logical **or** is a common logical error.
- compound Boolean expressions that are always false.



WORK SHEET (4)

Homework write a Python program to find the result of equation:

$$f(x) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Homework Write Python Program to find the greater number among three numbers

Homework Write a Python program that requests an integer value from the user. If the value is between 1 and 100, print "OK;" otherwise, do not print anything.

Homework Write a Python program that requests an integer value from the user. If the value is between 1 and 100 inclusive, print "OK;" otherwise, print "Out of range."

Homework Write a Python program that allows a user to type number and convert to English day of the week (Sunday, Monday, etc.) .

Hint 1 → Sunday , 2 → Monday , etc.

Homework Write a Python program to read a city of Iraq as number and print the estimation to refer it.

Hint: 1 → Baghdad, 2 → Basrah 3 → Mosul, 4 → Irbil



 **Homework** Consider the following Python code fragment:

```
# i, j, and k are numbers
if i < j:
    if j < k:
        i = j
    else:
        j = k
else:
    if j > k:
        j = i
    else:
        i = k
print("i =", i, " j =", j, " k =", k)
```

What will the code print if the variables i, j, and k have the following values?

- (a) i is 3, j is 5, and k is 7
- (b) i is 3, j is 7, and k is 5
- (c) i is 5, j is 3, and k is 7
- (d) i is 5, j is 7, and k is 3
- (e) i is 7, j is 3, and k is 5
- (f) i is 7, j is 5, and k is 3

