Q- What are the conditions?

A- <u>Conditions-</u> Suppose your friend is throwing a party for his birthday and you are invited. You see that mom is not in home and party is in the night. What will you do? You make a plan-

If mom returns before 8 PM, you will go to the party otherwise not. See the plan you just made is a condition. **Condition is something that can affect output or result.**

TBR- you can apply anything logical that can be conditions like-

x**<**10 , x should be less than 10

Y **>** 67, y should be great than 67

x**=**y, x must be equal to y

x**>**0 and x%2=0, x should be greater than 0 and it should be divisible by 2 (that means remainder must be 0 when divided by 2 for example 2,4,6,8 …)

Partner**=** girl and partner**>**18, partner should be girl and must be equal greater than 18 (I am not going to marry a boy, if you want to - it's your life; let it be unproductive)

If statements in python -

Syntax-

```
If (condition):
   Print ("whatever you want to print")
   Print ("You can print another line too")
   Print ("You can print as many as lines you want to print")
Print("this is outside of the loop")
```

Explanation- If your condition is true, all the statements within the loop (written in italic fonts) will be printed. And if condition is not true then the last line which is bold will be printed as it is not in the loop.

So the question arises, how to identify what is in the loop? For that indentation was introduced. You can see that

```
If (condition):
   Print ("statement 1, inside loop")
   Print ("Statement 2, inside loop")
Print("outside from loop")
```

Spaces under the "if" is known as indentation, statements are written just under the condition. It means if condition is right, next part is written just under it. Last statement is not under the condition statement. It is under "if" that means we are not in the loop. We are outside of the loop. **If condition is true, statements written under condition (in green) will be printed. If not, we will be outside of the loop and the last statement (in red) will be printed.**

USING ELSE AND ELIF CONDITION-

If there is only one condition, the example given above can work. What if there are multiple conditions-

You have to go to the party and there are multiple conditions-
Condition 1 - if mom comes home, you'll go to the party by bus
Condition 2 - if dad comes home, you'll take car to the car
Condition 3- if no one comes, plan cancel

```
If (mom = home):
    print("I'll go to party by bus")
elif (dad=home):
    print("I'll take the car to the party)
else:
    print("no party")
```

Taking input from User:

You can take input from user, all you have to do is - ask. Once a prophet, a very famous prophet, I don't know who but a very famous once said - "ask, you will be given"! How can you ask from the computer screen? Let me show you the way-

Suppose you want to ask the name of the user's girlfriend - :-)

```
>>>name_gf = raw_input("What's your gf's name- ")
0/P What's your gf's name - "here you/user has to answer"
>>>print(name_gf)
```

Explanation - raw_input takes input from user, within parenthesis you have to write whatever you want to ask to user. name_gf is a variable (remember variables from first lecture??? Jars in the kitchen, when we wanted to store something). It stores you gf name in computer memory. In the end by using print(name_gf) will print whatever user typed.

Q- What to do when you have to take multiple inputs?
A- remember the split method from string? Oh okay, let me explain it again

Suppose we have a string named my_string

>>>my_string = “you forgot what i told you”
>>>print(my_string.split())
>>>[‘you’, ‘forgot’, ‘what’, ‘i’, ‘told, ‘you’]

You can see that split() method has separated each single word, yes that's what split does. It separates string whenever it find a space. If there is a space, words going to be splitted (separated)

#taking two inputs from user

>>> x, y = raw_input(“ write two things here: “).split()
O/P = write two things here: ***first two***
(Now split method is going to separate them on the basis of spaces)
>>>print(“first thing is “ , x)
O/P = first thing is ***first***
>>>print(“second thing is” , y)
O/P = second thing is ***two***
>>>print() #printing new line

Q- Write a program that print your name with you sirname-
A- In this specific program we will be taking two inputs from user

>>>frst_name, sirname = raw_input(“Write your first name and sirname, separated by space”).split()
O/P = Write your first name and sirname, separated by space - Johnny Sinns
>>>print(“your first name is: “, frst_name)
>>>print(“your last name is: “, sirname)
>>>print()
O/P = your first name is Johnny
Your last name is Sinns (printed in new line because of print() method we wrote in the last while coding)

TBR - you’ll be thinking how we can take multiple inputs (as many as you want). We will be doing that once we read looping and type casting.

Understanding the conditions

and : both case must be true
Example - alex is my schoolmate and we are in same class (in this example, alex and me are in same class and same school)

or : any one must be true
Example - John is my schoolmate or coachingmate (in this example, either john is in my school or in my coaching, he may be in both but he must satisfy at least one condition)

TBR - in python = is being used to assign value to something and == means equal to.

%: modulo - 10%2 ==0 (remainder is 0, divisible that means)
11 % 2 == 1 (remainder is 1, non divisible that means)
X % Y == 0 (means non divisible)
X % 2 == 0 (means divisible by 2, even numbers)
x%3 != 0 (means not divisible by 2, odd numbers right?) (== means equal to and != means not equal to)

Type casting
raw_input() method always takes input in string format. What happens when you type integer in your answer? Your program will end up with an error. You'll not be able to do whatever you wanted to do. We use type casting to change one kind of data type to another kind.

Changing integer to string =
>>>x = input("type a number")
>>>y= str(x)
Explanation - raw_input("type a number") will take your input (suppose you put 10). Now you want to change it in string. str( ) method will change it into strings. First I saved your input in "x" and then I changed x into string by using str(x) and saved it in "y".

You could have done this in a single line too-
x = str(input("type a number"))

Changing string into integer-
You can't change a string into integer because ten can't be changed into 10.

Changing integer into floating value :
x=float(input("type a integer value here: "))
print(x)

Changing floating into integer value :
x=int(input("type a floating value here: ")
print(x)

TBR - raw_input() and input () methods are not the same. raw_input() takes strings as input and input() takes number (interger or float). Make sure if you want to take input as strings use raw_input() and use input() method when you want to take integer of floats as input.

EXAMPLES -

Q- Write a program that takes input from user and tests whether it is an even number or not

#first we have to take input from user, so that he can enter a number to check
#since user has to put integer as input, we can use input() method, in case if it was string we could have used raw_input() method.
#now checking if number is completely divisible by 2 (even number)
#we will be using conditions for that and the condition will be (num %2 ==0)
#If the condition is true, the first statement (“it is an even number”) will be printed, else other will be printed. Codes is given below -

```
>>>num= input("what's the number you want to check: ")
>>> if (num % 2 == 0):
        print("it is an even number")
>>>else:
        print("it is an odd number")
```

Q- Write a program that ask user about his name first name and last name

```
first_name = raw_input("whats your first name: ")
last_name= raw_input("whats your last name: ")
full_name= first_name +" " +  last_name
print(full_name)
```

Here is the O/P -

```
whats your first name: Akshansh
whats your last name: Rawat
Akshansh Rawat
```

What is that “ “ in full name? That is “ “ space, if you will not add space your first name and last name will be printed together. To add space between first name and last name we used “ “