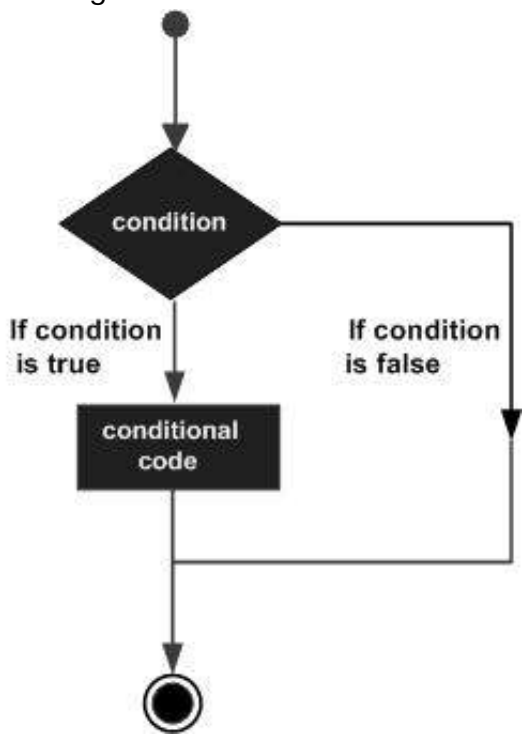
Lesson 2: Conditions and Loops

1. Conditions - Compare with "if", "elif" and "else"

1. Evaluate expressions which produce TRUE or FALSE as outcome
2. Based on the outcome of the expression a specific block of code is executed
3. The expressions could be: equality (==) , inequality (!=), less/greater than (<)(>), less/greater than or equal
   (<=) (>=), membership (in, not in)
4. Expressions can be combined using "and" and "or" and "and not"



Let us write a small code that mimics a dice by generating a random number between 1 and 6. It then asks you to enter a number between 1 and 6. If what you enter equals the dice value you win or else you loose.

In [2]:

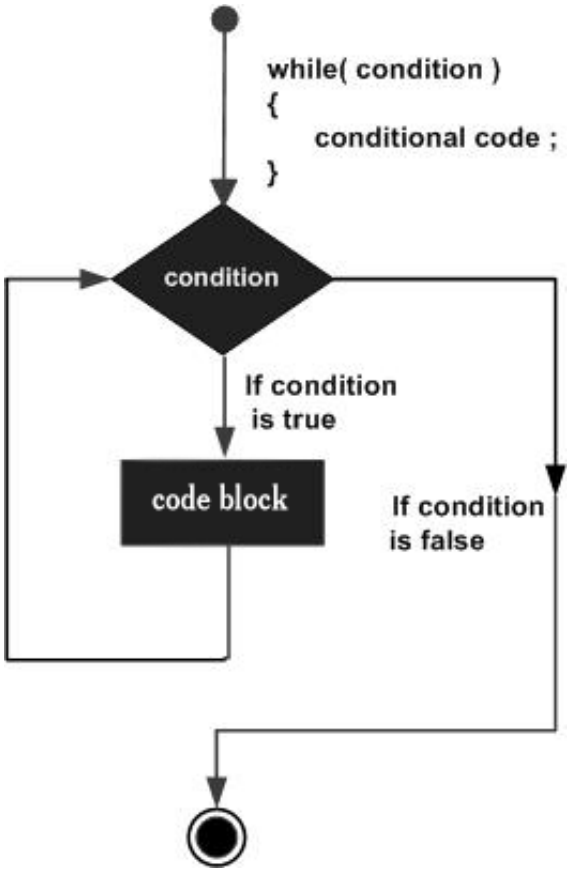
```
from random import randint #From the random library we import the randint f
unction.

def dice(user_input):
    '''This function mimics a dice. It generates a random number between 1
and 6 and sees if the value given by
        the user equals to the dice value'''

    dice_value = randint(1,6) #Generate a randome number between 1 and 6

    if user_input == dice_value: # Note the syntax ":" and indention is com
pulsory. The conditions could also be ==, !
        print('Congratulations ! The value entered is the dice value')
    else:
        print('Sorry, the dice showed: ', dice_value)

user_input = input('Input a number between 1 and 6: ')
dice(user_input) #we pass the user_input as an argument to the function.
```

```
Input a number between 1 and 6: 1
Sorry, the dice showed:  6
```

2. Loops - Repeat with "while" and Iterate with "for"

While loop

1. A loop statement allows us to execute a statement or group of statements multiple times
2. While loop is the simplest loop which executes a block of code until and expression is True
3. A 'break' command can be used to exit any loop prematurely
4. A 'continue' command can be used to skip ahead to the next iterations without exiting the loop



Now lets improve the previous function "dice()" such that the user has 5 attempts at entering the number and getting the right dice value.

In [3]:

```
from random import randint

def dice(user_input):
    '''This function mimics a dice. It generates a random number between 1 and 6 and sees if the value given by
        the user equals to the dice value'''

    dice_value = randint(1,6) #Generate a randome number between 1 and 6

    if user_input == dice_value: # Note the syntax ":" and indention is compulsory. The conditions could also be ==, !
        print('Congratulations ! The value entered is the dice value')
    else:
        print('Sorry, the dice showed: ', dice_value)

iterate = 0
while iterate < 5:
    user_input = int(input('Input a number between 1 and 6: ')) # Input by default assumes that the value you enter is a string so we need to conver it
    dice(user_input)
    iterate = iterate  + 1
```

```
Input a number between 1 and 6: 4
Sorry, the dice showed:  2
Input a number between 1 and 6: 4
Sorry, the dice showed:  6
Input a number between 1 and 6: 4
Sorry, the dice showed:  3
Input a number between 1 and 6: 4
Congratulations ! The value entered is the dice value
Input a number between 1 and 6: 4
Congratulations ! The value entered is the dice value
```

Excercise : Can you edit the the previous code so that it stops asking the user to enter a number when the value entered matches the dice value ? Hint: You will need to use the "break" command within the while loop. Also remember that you can use the "return" command to pass values back to the calling function.

In [6]:

```
from random import randint

def dice(user_input):
    '''This function mimics a dice. It generates a random number between 1
and 6 and sees if the value given by
        the user equals to the dice value'''

    dice_value = randint(1,6) #Generate a randome number between 1 and 6

    if user_input == dice_value: # Note the syntax ":" and indention is com
pulsory. The conditions could also be ==, !
        print('Congratulations ! The value entered is the dice value')
        return 1
    else:
        print('Sorry, the dice showed: ', dice_value)
        return 0

iterate = 0
while iterate < 5:
    user_input = int(input('Input a number between 1 and 6: ')) # Input by
default assumes that the value you enter is a string so we need to conver i
t
    returned_value = dice(user_input)
    if returned_value == 1:
        break
    iterate = iterate  + 1
```

```
Input a number between 1 and 6: 3
Sorry, the dice showed:  6
Input a number between 1 and 6: 3
Congratulations ! The value entered is the dice value
```

For loop

1. For loops are used to iterate over data
2. It makes it possible for you to traverse data structures without knowing how large they are
3. You can even iterate over datastreams in real time

Let us write a program that asks the user to enter a text sentence and counts the number of times a particular character occurs in the sentence.

In [8]:

```
def count(sentence, char):
    '''This function counts the number of times a particular charactors occ
urs in a given
    sentence'''

    count = 0
    for x_char in sentence:
        if x_char == char:
            count += 1
    print("Number of times the character '",char, "' occurs is: ", count) #
Careful with the quotes !

sentence = input('Input your sentence: ')
sentence = sentence.lower() #the lower() comand converts the sentence to lo
wer case
char = input('Input the character that needs to be counted: ').lower()
count(sentence, char)
```

```
Input your sentence: My name is poonacha
Input the character that needs to be counted: o
Number of times the character ' o ' occurs is:  2
```

Can you use the for loop so that it counts the number of times a given word occurs in a sentence ? Hint: Use the split() command to split the sentence into a list of words and then use the for loop to traverse through the list.

In [ ]:

```

```