

# Programming Tutorial

---

## 03 Control Flow

---

### Decision

if you want the program to take some decisions and do different things depending on different situations

1. boolean condition  
e.g. if n is odd, then ... ; if n is even, then ...
2. Hash condition  
e.g. Key-Value Pair; Dictionary

```
number = 23
guess = int(input('Enter an integer : '))
if guess == number:
    # New block starts here
    print('Congratulations, you guessed it.')
    print('(but you do not win any prizes!)')
    # New block ends here
elif guess < number:
    # Another block
    print('No, it is a little higher than that')
    # You can do whatever you want in a block ...
else:
    print('No, it is a little lower than that')
    # you must have guessed > number to reach here
print('Done')
# This last statement is always executed,
# after the if statement is executed.number = 23
guess = int (input('Enter an integer:'))
if guess == number:
    print('Congratulations!')
else:
    print('No')
```

### Repeat

if you wanna repeatedly execute a block of statements:

1. explicit counter

e.g. prompt user input 5 numbers and calculate the sum.

```
for i in range(1, 5):  
    print("helloworld")
```

2. as long as a condition is true

e.g. when the user enters invalid value, the system does not quit, but prompts again and waits for a valide input.

```
number = 23  
running = True  
  
while running:  
    guess = int (input('Enter an integer:'))  
    if guess == number:  
        print('Congratulations!')  
        running = False  
    elif guess < number:  
        print('Higher')  
    else:  
        print('Lower')  
else:  
    print('Done')
```

3. recursive

Some languages does not provide repeat statement but recursive statement instead. e.g. Elixir.

4. nested loop

Multiplication Table

## Use the counter to accumulate

```
sum = 0  
#calculation 1,2, 3, 4 . . . +100  
for i in range(1,101):  
    sum+=i;  
else:  
    print(sum)
```

## break & continue

The break statement is used to *break* out of a loop statement i.e. stop the execution of a

looping statement, even if the loop condition has not become False or the sequence of items has not been completely iterated over.

An important note is that if you *break* out of a for or while loop, any corresponding loop else block is **not** executed.

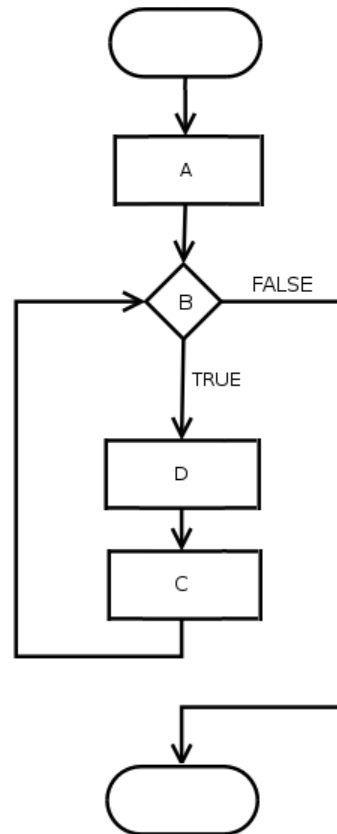
The continue statement is used to skip the rest of the statements in the current loop block and to *continue* to the next iteration of the loop.

## Flow Chart

Flow chart is a diagram that represents a workflow or process. It is a representation of an algorithm.

Shape	Name	Description
Line	Flow Line (Arrowhead)	Shows the process's order of operation.
Rounded Rectangle	Terminal	Indicates the beginning and ending of a program or sub-process.
Rectangel	Process	Represents a set of operations that changes value, form, or location of data.
Diamond	Decision	Shows a conditional operation that determines which one of the two paths the program will take.
Parallelogram	Input/Output	Indicates the process of inputting and outputting data, as in entering data or displaying results.

for(A;B;C)  
D;



```
//int i=0,sum =0; => Block A
//i<=100 => Block B
//i++ => Block C
for(int i=0,sum = 0;i<=100;i++)
    sum+=i; // sum+=i; => Block D
```

## Data Processing

Data process is typical application for our computer. The main flow for data processing is input, process and output.

```
#Bad Example for Data Processing

#Currency Exchange
euro=float(input("How many euros are you exchanging?"))
rate=float(input("What is the exchange rate?"))
print('{0} euro at an exchange rate of {1} is {2:.3f}
U.S.dollars.'.format(euro,rate,rate*euro/100))
```

Comments and extra blank lines are utilized to divided your code into threes parts: Input, Process, Output.

Although one more variable "dollars" is used, it is much more easy to read.

```
#Better Example for Data Processing

#Currency Exchange
#1. input
euro = float(input("How many eruos are you exchanging?"))
rate = float(input("What is the exchange rate?"))

#2. process
dollars = euro * 100 /rate;

#3. output
print('{0} euro at an exchange rate of {1} is {2:.2f} U.S.
dollars.'.format(euro, rate, dollars))
```