Python

by ...

Dr. Pradyumna Kumar Tripathy

Associate Professor & Head, Dept. of CSE,
Silicon Institute of Technology, Bhubaneswar

Disclaimer:

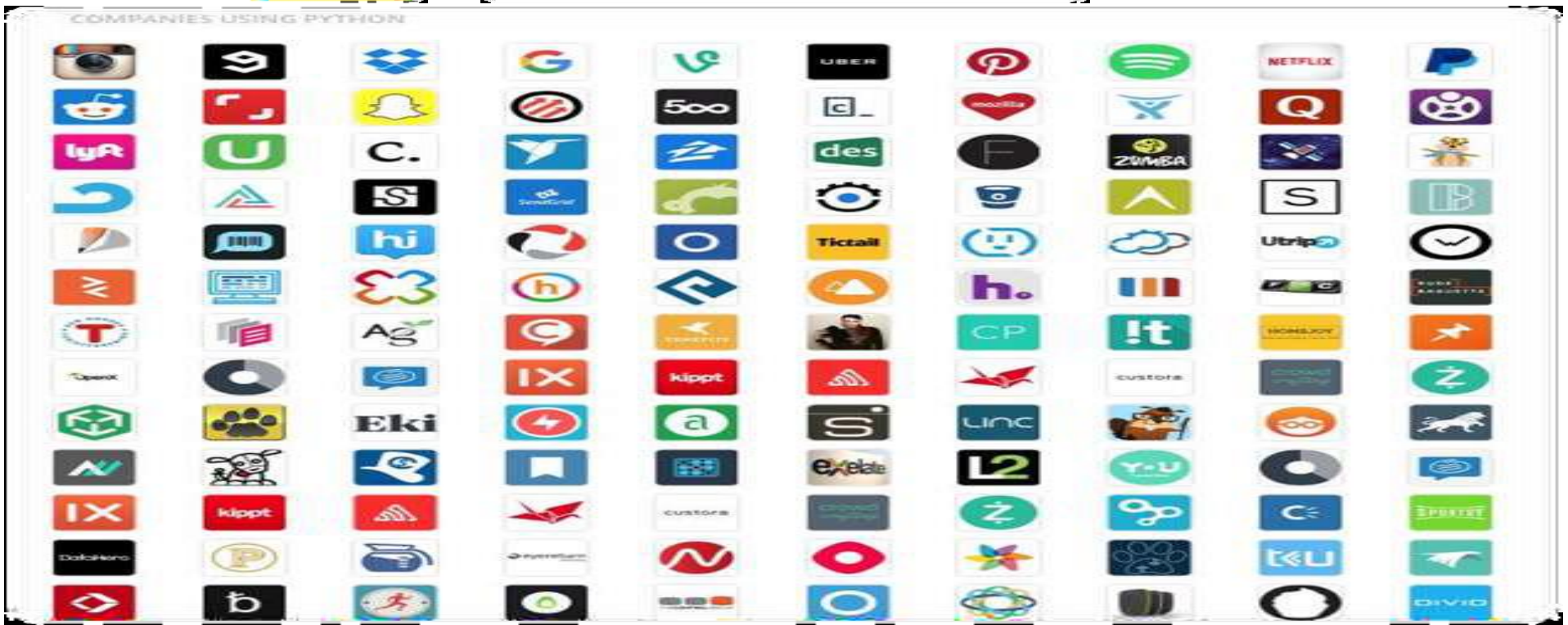
Few of the slides are prepared by collecting information from various sources, particularly from Internet. Those information are collected, compiled and presented here.

Introduction

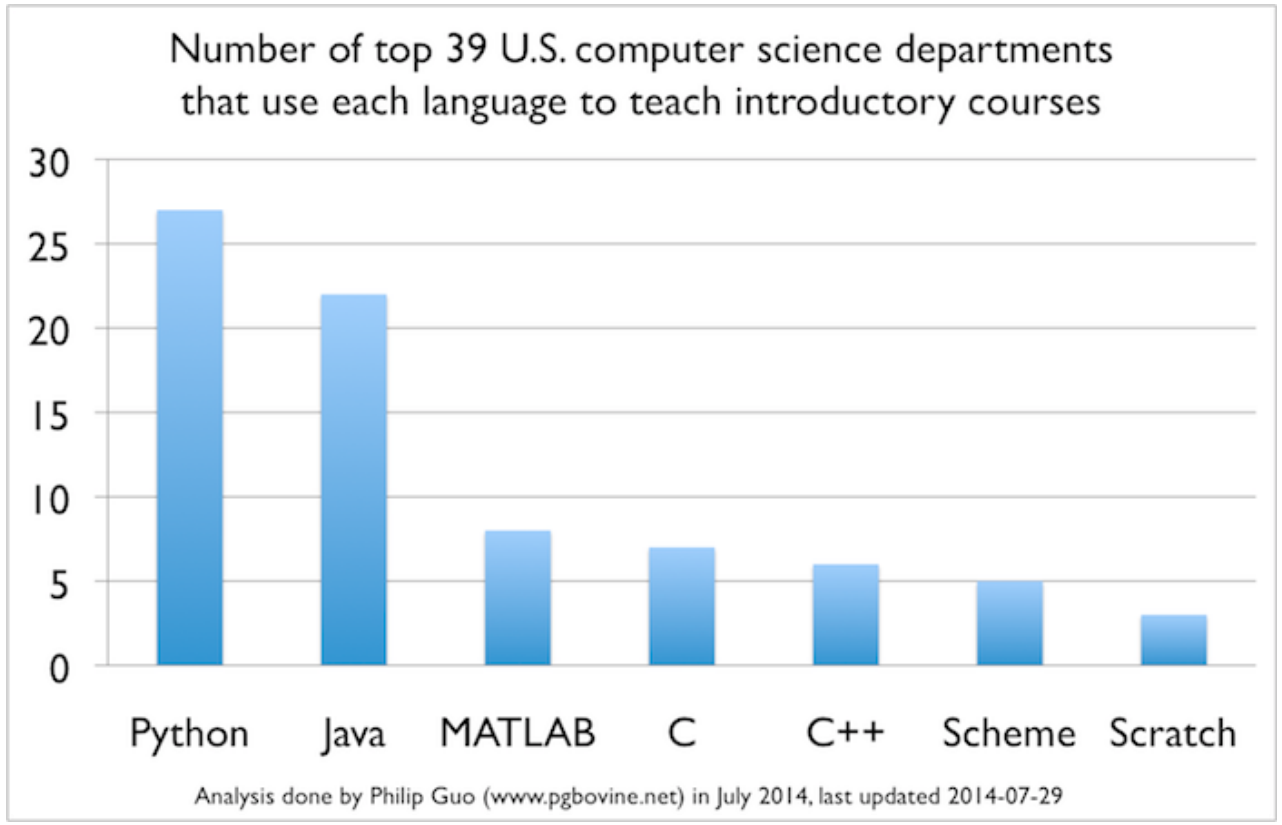
- High Level Programming Language.
- Developed by Guido Van Rossum in Feb 1989 while working at Centrum Wiskunde & Informatics (CWI). [released as **Open Source** in February 1991]
- Based on/ Influenced by two programming Languages:
  - ABC Language: A teaching Language created as a replacement of BASIC
  - Modula-3
- Easy-to-learn, powerful, object oriented programming Language

Python is named : Idea is from a British Comedy "**Monty Python's flying circus**", not the snake.

Companies Using Python



Why Python ? ??



The chart above shows how many of the top 39 departments teach either CS0 or CS1 using the seven most common languages. The bar heights add up to more than 39 since many schools offer both CS0 and CS1.

Different Versions of Python by Date

- **Python 1.0 - January 1994**
  - Python 1.5 - December 31, 1997
  - Python 1.6 - September 5, 2000
- **Python 2.0 - October 16, 2000**
  - Python 2.1 - April 17, 2001
  - Python 2.2 - December 21, 2001
  - Python 2.3 - July 29, 2003
  - Python 2.4 - November 30, 2004
  - Python 2.5 - September 19, 2006
  - Python 2.6 - October 1, 2008
  - Python 2.7 - July 3, 2010
- **Python 3.0 - December 3, 2008**
  - Python 3.1 - June 27, 2009
  - Python 3.2 - February 20, 2011
  - Python 3.3 - September 29, 2012
  - Python 3.4 - March 16, 2014
  - Python 3.5 - September 13, 2015
  - **Python 3.6 - December 23, 2016**

- Expressive Language

In C++	In Python

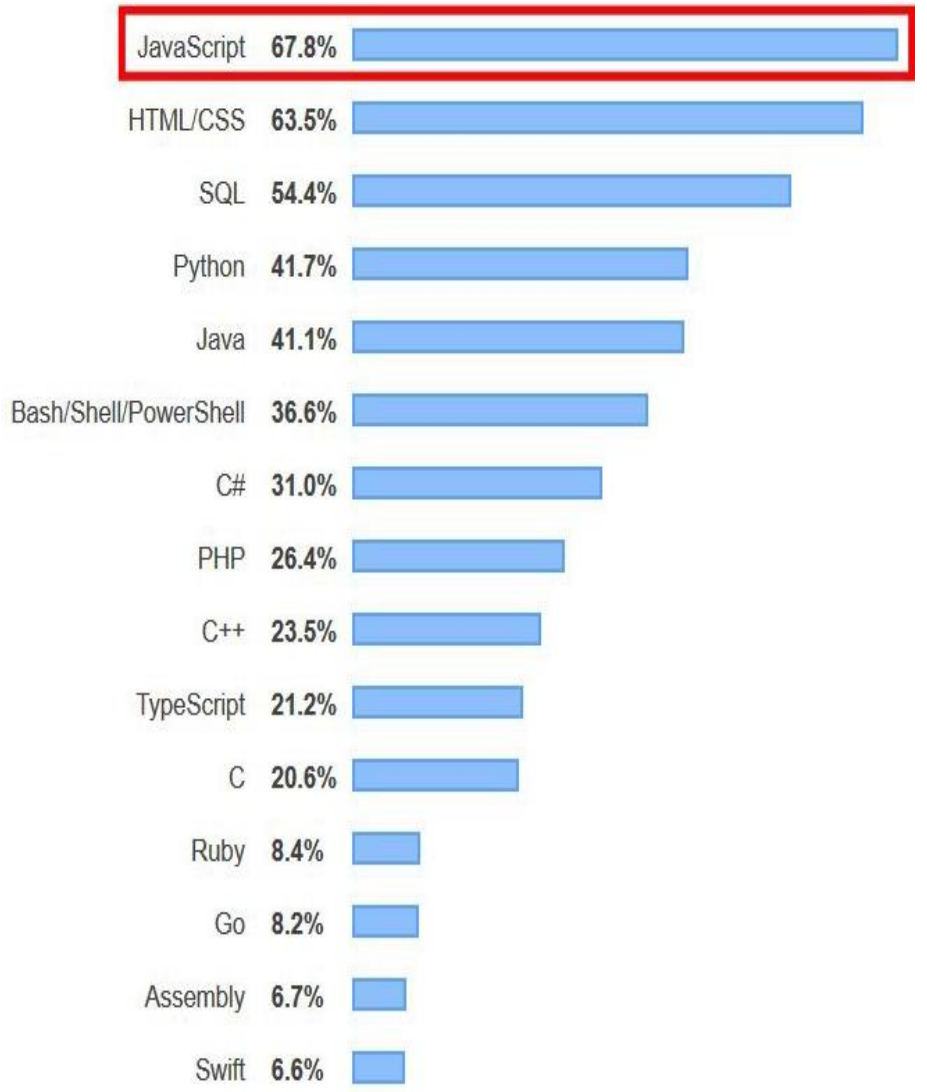
int a=2, b=3, temp;
temp=a;
a=b;
b=temp;
	a, b = 2, 3
	a, b = b, a

- Interpreted Language
- Completeness (Batteries included)
  - Emails, web-pages, databases, GUI, development, network and many more…
- Cross-Platform Language (Portable)
- Free and Open Source
- Variety of Usage/ Applications

Limitations

- **Not the fastest Language**
- Lesser Libraries than C, Java, Perl
- Not Strong on Type-binding
  (Declare int and later store a string-No worry)
- **Not Easily Convertible**
  (Not Structured. So, Translating into another programming language-difficult)

It is the **4th** most popular programming language after Java Script, HTML/CSS and SQL



Survey: 2020
Source: Internet

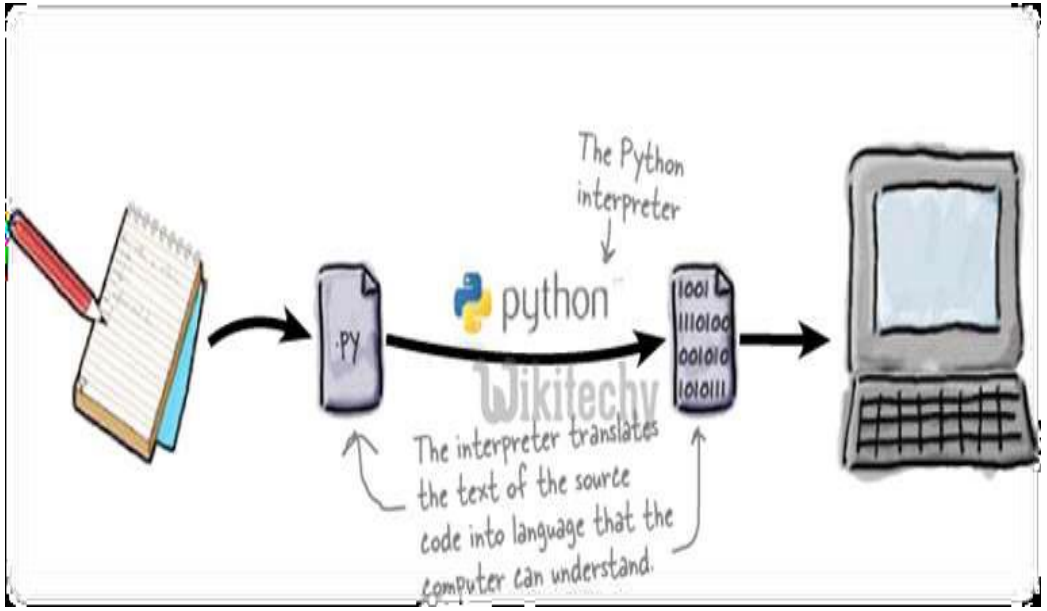
Installation

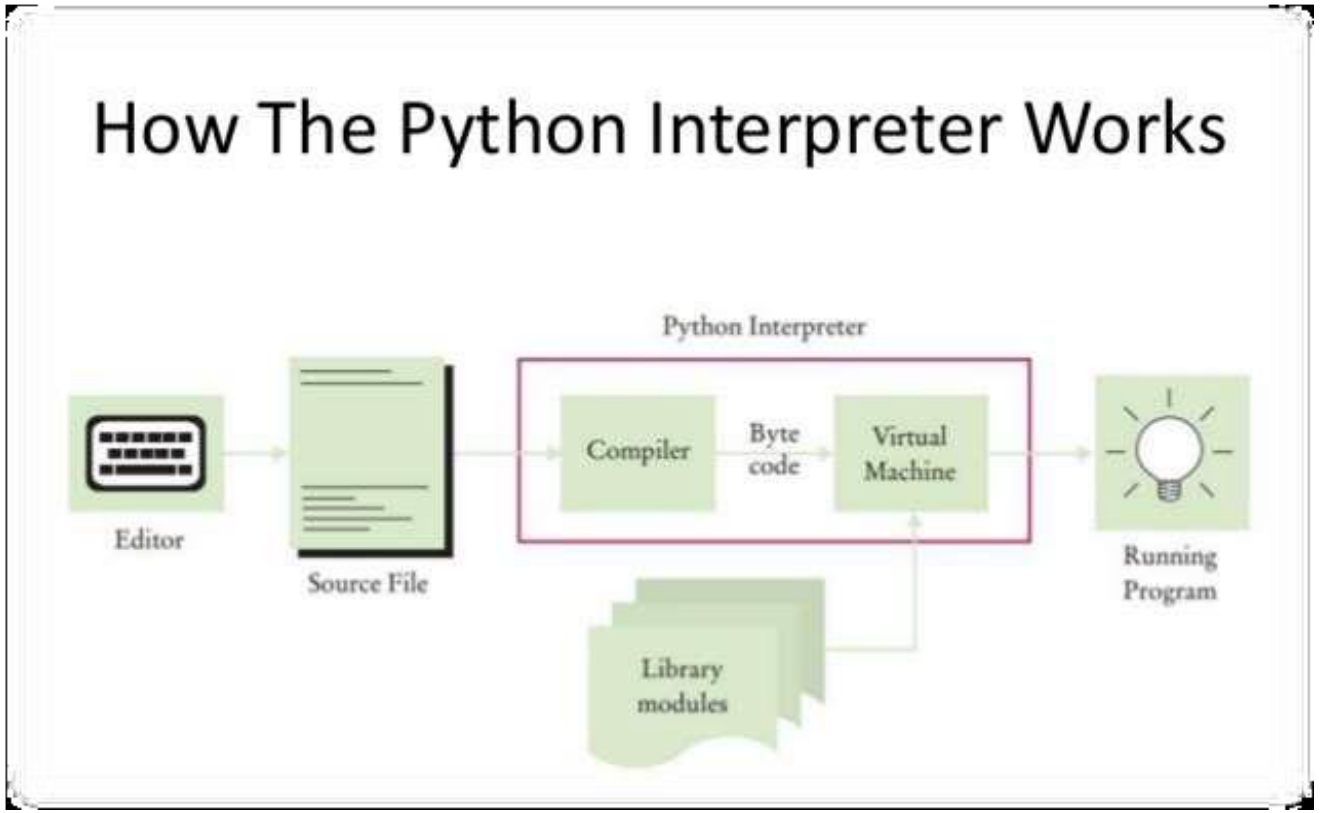
- Pre-installed in most of the version of Linux OS
- Download from
  - www.python.org/download (select the python distribution)
- Two Mode of Operation
  - Interactive mode (>>> prompt / python shell)
  - Script Mode (File extension **.py**)

Scripts are reusable.
A script is a text file containing the statements that comprise a Python program

A shell is usually an "interactive shell",
usually termed a REPL which stands for
"Read - Execute - Print - Loop"

Python uses Interpreter





Hands-on

Type the following on Interactive window and analyze result

```
>>>3 * 5
>>>print (3 * 5)
>>>print (“ hello”)

>>>a=20
>>>print (“my age is “ , a)

>>>3 ** 3
>>>6 + 2 * 4
>>>(6 + 2) * 4
>>>5 – 3 – 3
>>>k = 5 – (3 – 3)
```

Hands-on Contd…

- Open **idle editor**
- Create a program file (name it as myfile.py)
- Type the following :

```
print ('hello Ashok', 'hello Pooja')
Var1 = 'hello Ashok'
Var2 = 'hello Pooja'
Name1 = 'Ashok'
Name2 = 'Pooja'
print (Var1, Var2)
print ('hello', Name1, ',' , Name2)
Name1 = 'Bikram'
Name2 = 'Annu'
print ('hello', Name1, ',' , Name2)
```

(*save with* ***.py*** and run it with ***python filename*** command)

Hands-on Contd…

```
>>> a = 7
>>>type(a)
>>>type(99.9)
>>>type('ABC')
>>>5
>>>015
>>>9 / 5
>>>9 // 5
>>>a //= 3
>>>5 / 0
>>>a *= 2
>>> 9 % 5
>>>divmod(9,5)
```

Hands-on Contd…

```
>>>int(True)
>>>int(False)
>>>int(98.6)
>>>int(1.0e4)
>>>int('99')
>>>int('-45')
>>>int('+12')
>>>True + 3
>>>False + 5.0
>>> googol=10 ** 100
>>>googol * googol
```

(called googol)

Python Character Set

- Letters: A-Z, a-z
- Digits : 0-9
- Special Symbols: space + - * / ** \ () [ ] // = != == <> . ‘ “ ”’ , ; : % ! & # <= >= @ >>> << >> _
- Whitespaces : Blank space, tab, carriage return, newline, form-feed
- Other Characters: Python can process all ASCII and unicode characters

Tokens

The smallest individual units of a program is called token or lexical unit.

- Keywords
- Identifiers
- Literals
- Operators
- Punctuators

Keywords

```
>>>import keyword
 >>> print(keyword.kwlist)
```

```
False      def        if         raise
None       del        import     return
True       elif       in         try
and        else       is         while
as         except     lambda     with
assert     finally    nonlocal   yield
break      for        not
class      from       or
continue   global     pass
```

```
>>> keyword.iskeyword(s)
```

Identifier

- Arbitrary long sequence of letters and digits
- First character is a letter (_ is counts as a letter)
- All Characters are significant
- The digits can be a part of identifier except the first character
- Must not be a python keyword
- No special character except _

Case Sensitive

Literals

- String (‘Silicon’, “Silicon”, ”Silicon’, “’Silicon”’)
- Numeric
- Boolean
- Special (None)
- Literal collection

Escape Sequences

Escape Sequence	Meaning
\newline	Ignored
\\	Backslash (\)
\'	Single quote (')
\"	Double quote (")
\a	ASCII Bell (BEL)
\b	ASCII Backspace (BS)
\f	ASCII Formfeed (FF)
\n	ASCII Linefeed (LF)
\N{name}	Character named *name* in the Unicode database (Unicode only)
\r	ASCII Carriage Return (CR)
\t	ASCII Horizontal Tab (TAB)
\uxxxx	Character with 16-bit hex value *xxxx* (Unicode only)
\Uxxxxxxxx	Character with 32-bit hex value *xxxxxxxx* (Unicode only)
\v	ASCII Vertical Tab (VT)
\ooo	Character with octal value *ooo*
\xhh	Character with hex value *hh*

String Types

- **Single line Strings**
- **Multi line Strings**

a) Single line Strings

Enclosed in ‘ ‘ or “ “

>>>text1=‘ hello

there’ ERROR

b) Multi line Strings

>>>text1 = ‘ hello\

there’ OK

OR

>>>text1=‘ ’ ’hello

there’ ’ ’ OK

>>>text1=“ ” ”hello

there” “ “ OK

Size of a String

```
>>>len('\\')                    1
>>>len('abc')                   3
>>>len("\ab")                   2
>>>len("Seema\'s pen")          11
>>>len("Amys")                  4

>>>str3 = ' ' ' a               5
b
c ' ' '

>>>str4 = 'a\                   3
b\
c'
```

Hands-on Contd…

```
>>> ‘ ‘
>>>” “
>>>’ ‘ ‘ ‘ ‘ ‘
>>>” ” “ “ “ “
>>> bottle = 99
>>>base = ‘ ‘
>>>base += ‘current inventory:’
>>>base += str(bottles)
>>>base
```

Hands-on Contd…

```
>>>start = ‘Na ‘ * 4 + ‘\n’
>>>middle = ‘Hey ‘ * 3 + ‘\n’
>>>end = ‘Goodbye.’
>>>print ( start + middle + end)
```


```
>>>letters = ‘abcdefghijklmnopqrstuvwxyz’
>>>letters[0]
>>>letters[1]
>>>letters[-1]
>>>letters[-2]
>>>letters[100]

>>>name = ‘Henny’
>>>name[0] = ‘P’
>>>name.replace(‘H’, ‘P’)
>>>’P’ + name[1:]
```

ERROR

Hands-on Contd…

[start: end: step]

```
>>>letters = ‘abcdefghijklmnopqrstuvwxyz’

>>>letters[:]
>>>letters[20:]
>>>letters[12:15]
>>>letters[-3:]
>>>letters[18:-3]
>>>letters[-6:-2]
>>>letters[::7]
>>>letters[4:20:3]
>>>letters[19::4]
>>>letters[:21:5]
>>>letters[-1::-1]
>>>letters[-51:-50]
```

Hands-on Contd…

```
>>>t = 'get gloves, get mask, give cat vitamins, call ambulance'
>>>t.split(',')
>>>t.split( )
```

-Store a paragraph of text in a variable **t**

-extract first 13 characters (t[:13])

-find total no of characters (len(t))

-does it starts with letters 'All' (t.startswith('All'))

-does it ends with letters 'Bye' (t.endswith('Bye'))

-find the offset of the first occurrence of aword (t.find('word'))

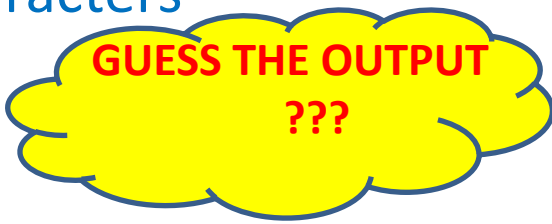
-find the offset of the last occurrence of aword (t.rfind('word'))

-How many times the word 'the' occurs in the text (t.count('the'))

Hands-on Contd…

Try out the following function on a string variable:

strip( )	#t.strip() Removes leading and trailing characters
capitalize( )	#t.capitalize() Convert First Character in Upper case
title( )	#t.title() Convert First Character of each word in Upper case
upper( )	#t.upper() Convert All Characters in Upper case
lower( )	#t.lower() Convert All Characters in Lower case
swapcase( )	#t.swapcase() Swap case of each Characters
center(n)	#t.center(30) Place the text ‘t’ at the center of 30 characters
ljust( )	#t.ljust(30) Place the text ‘t’ left justified among 30 characters
rjust( )	#t.rjust(30) Place the text ‘t’ right justified among 30 characters



Numeric Literals

- int (signed integers)
  - Decimal
  - Binary
  - Octal
  - Hexa

```
>>>10
>>>0b10
>>>0o10
>>>0x10
```

- long (long integers)
- float (floating point real values)
  - Fractional Form (2.0, -13.7, -0.0076)
  - Exponent Form (172E05, 1.7E07, 0.152E08, -0.007E-3)
- Complex (complex numbers)

Boolean Literal

- True
- False

Special Literal

- None

None Keyword

```
>>> None == 0                    False

>>> None == [ ]                  False

>>> None == False                False

 >>> x = None

 >>> y = None

>>> x == y                       True
```

Operators

- Unary Operators
  - +
  - -
  - ~ (bitwise complement)
  - not (logical negation)
- Arithmetic Operators + - * / % ** //
- Bitwise Operators & ^ | << >> is is not
- Relational Operators < > <= >= == != <>

Operators Contd…

- Assignment Operators = /= += *= %= -= **= //=
- Logical Operators and or
- Membership Operators in not in

Punctuators

(Symbols used to organize sentence structure and indicate the rhythm and emphasis of expressions, statements, and program structure)

‘ “ # \ ( ) [ ] { } @ , : . ` = ;

Barebones of a Python Program

```
#This program shows a program's components
#Definition of seeyou( ) follows
def seeyou():
        print ('Time to say good bye !!')
#main program code follows now
a = 15
b = a – 10
print (a+3)
if b > 5:
    print ('value of 'a' is more than 5')
else:
    print ('value of 'a' is less than 5')
seeyou( )     #Call of Function
```

Function Definition

Input - Output

Name = raw_input('what is your name')
Age = raw_input('what is your age')
-It always returns a value of string type

>>>Age+5 ???? ERROR

Age = int(raw_input('what is your age'))

Amount = float (raw_input('what is your salary'))

Or

Age = (raw_input('what is your age'))

Age = int(Age)

raw_input ( ) is used in Python 2.7.X

What happen if you input your age as Sixteen ???

Input - Output

```
>>>Age = input('Enter your age')
>>>Retire_age = input('Enter age:')
>>>Enter age: 40 + 20
```

input ( ) is used in Python 3.7.X onwards

-input is not a stable function on all installation

Print statement without any argument prints a blank line

Sample Program

Problem: **Compute Area of a Circle**

Solution:

```
pi = 3.14159
radius = 2.2
# area of circle equation <- this is a comment
area = pi*(radius**2)
print(area)

# change values of radius
# use comments to help others understand what you are doing in code
radius = radius + 1
print(area)     # area doesn't change
area = pi*(radius**2)
print(area)
```

Sample Program

Problem: Input a filename and print the extension of that

Solution:

```
filename = input('Input the Filename:' )
file_extns = filename.split('.')
print ('The extension of the file is :' + (file_extns[-1]))
```

Problem: Input an integer (n) and computes the value of n + nn + nnn

Solution:

```
a = int(input('Input an integer :'))
n1 = a*10+a
n2 = n1*10+a
print (a+n1+n2)
```

Sample Program

Problem: Write a Python program to calculate number of days between two dates.
Sample dates : (2014, 7, 2), (2014, 7, 11)
Expected output : 9 days

Solution:

```
from datetime import date
f_date = date(2014, 7, 2)
l_date = date(2014, 7, 11)
delta = l_date - f_date
print(delta.days)
```

Problem: Print the calendar of a given month and year

Solution:

```
import calendar
y = int(input('Input the year :'))
m = int(input('Input the month :'))
print(calendar.month(y, m))
```

Sample Program

Problem: Get the Python version and Display the current date and time

Solution:

```
import sys
import datetime
print('Python version«', sys.version)
print('Version info.«', sys.version_info)
print ('Current date and time :',datetime.datetime.now())
```

Sample Program

Problem: Compute the distance between two points

Solution:

```
import math
p1 = [4, 0]
p2 = [6, 6]
distance = math.sqrt( ((p1[0]-p2[0])**2)+((p1[1]-p2[1])**2) )
print(distance)
```

Sample Program

Problem: Sort three inputted numbers in ascending order without using any conditional or looping statements.

Solution:

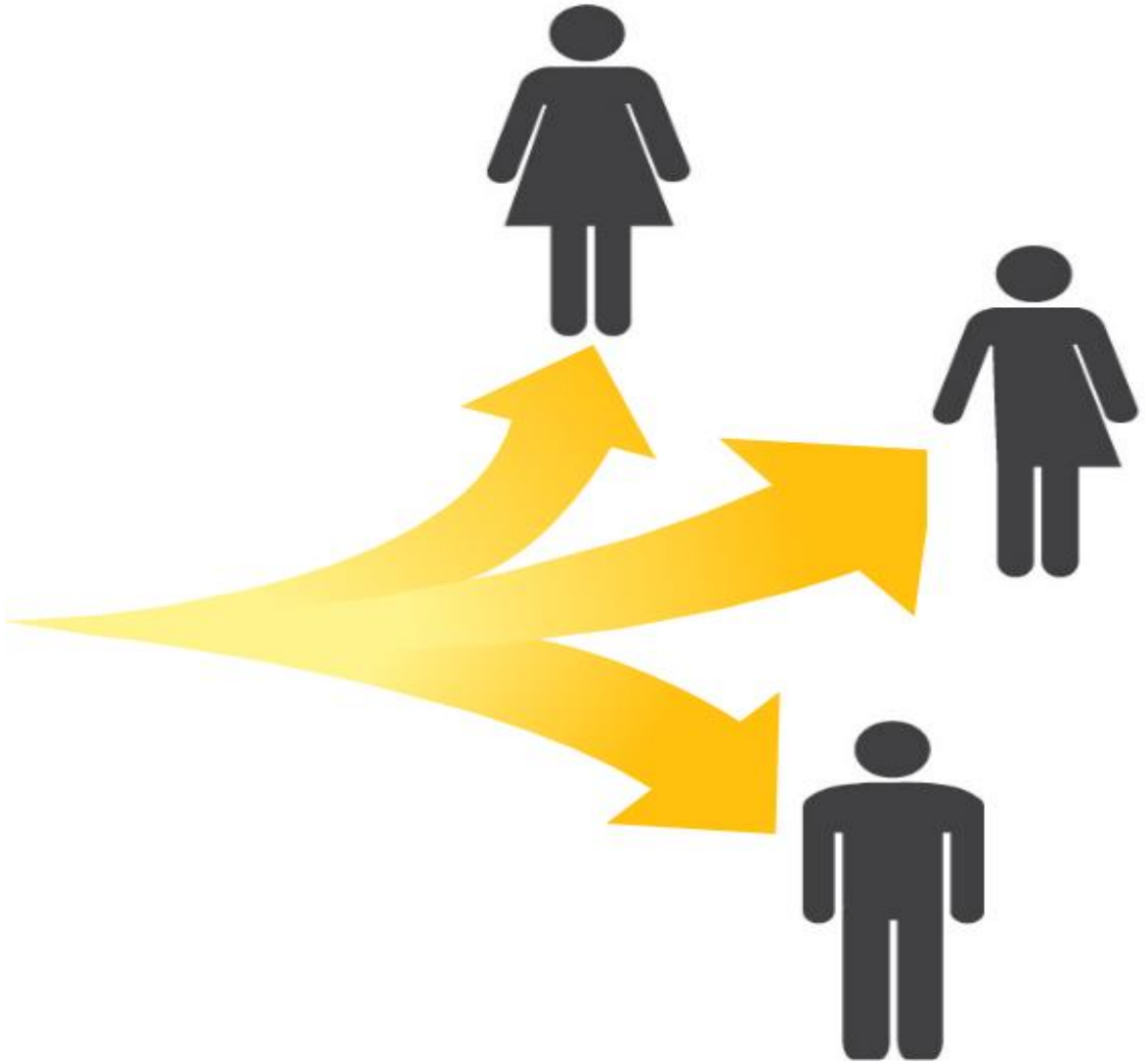
```
x = int(input("Input first number: "))
y = int(input("Input second number: "))
z = int(input("Input third number: "))
a1 = min(x, y, z)
a3 = max(x, y, z)
a2 = (x + y + z) - a1 - a3
print("Numbers in sorted order: ", a1, a2, a3)
```

Sample Program

Problem: Determine the largest and smallest integers, longs, floats

Solution:

```
x = 'true'
x = int(x == 'true')
print(x)
x = 'abcd'
x = int(x == 'true')
print(x)
```



CONDITIONAL STATEMENTS

Dr. Pradyumna Kumar Tripathy, Silicon Institute of Technology, Bhubaneswar

TechnoZeal

If Statement

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

```
num = 3
if num > 0:
        print(num, "is a positivenumber.")
print("This is always printed.")
```

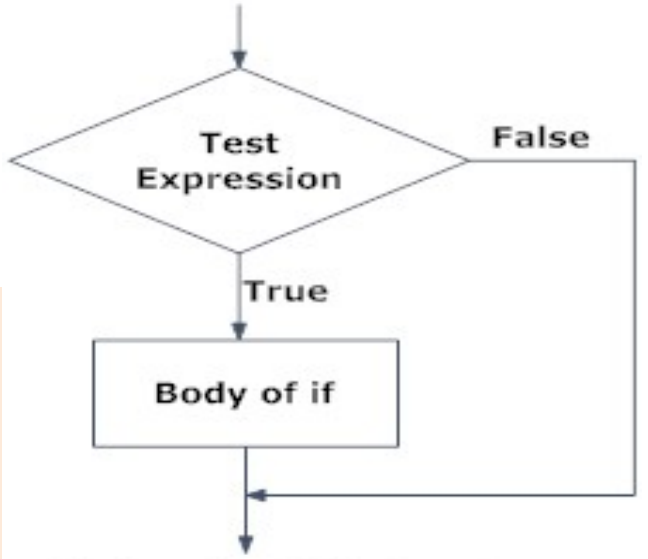


Fig: Operation of if statement

```
num = -1
if num > 0:
        print(num, "is a positive number.")
print("This is also always printed.")
```

If-else statement

```
if test expression:
    Body of if
else:
    Body of else
```

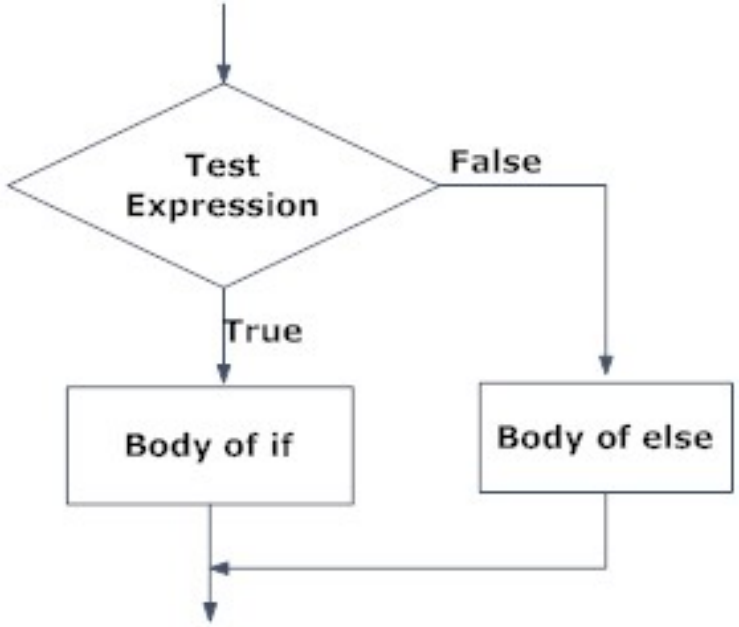


Fig: Operation of if...else statement

```
num = 3
if num >= 0:
        print("Positive or Zero")
else:
        print("Negative number")
```

GUESS THE OUTPUT ???

if...elif...else Statement

```
if test expression:
    Body of if
elif test expression:
    Body of elif
else:
    Body of else
```

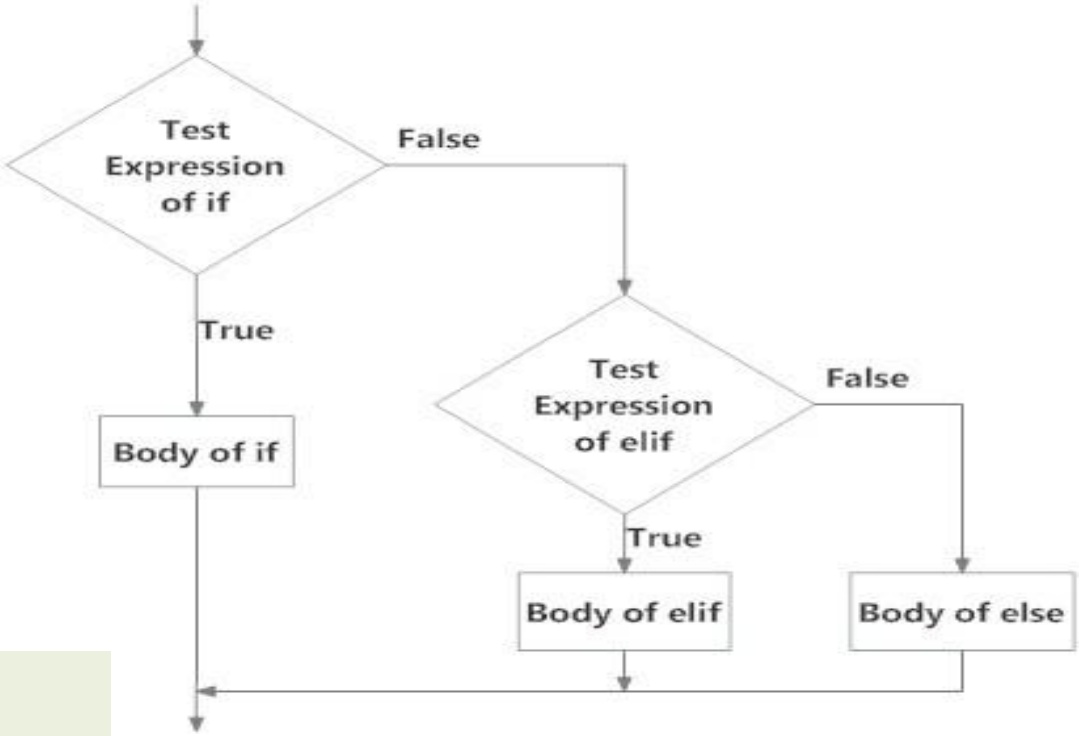


Fig: Operation of if...elif...else statement

```
num = 3.4
if num > 0:
        print("Positive number")
elif num == 0:
        print("Zero")
else:
        print("Negative number")
print('Thank you')
```

GUESS THE OUTPUT ???

Nested if else

```
num = float(input("Enter a number: "))
if num >= 0:
    if num == 0:
        print("Zero")
    else:
        print("Positive number")
else:
    print("Negative number")
```

Comparison Operators

```
x = 5
if x == 5 :
    print('Equals 5')
if x > 4 :
   print('Greater than 4')
if  x >= 5 :
    print('Greater than or Equals 5')
if x < 6 :
    print('Less than 6')
if x <= 5 :
    print('Less than or Equals 5')
if x != 6 :
    print('Not equal 6')
```

Equals 5

Greater than 4

Greater than or Equals 5

Less than 6

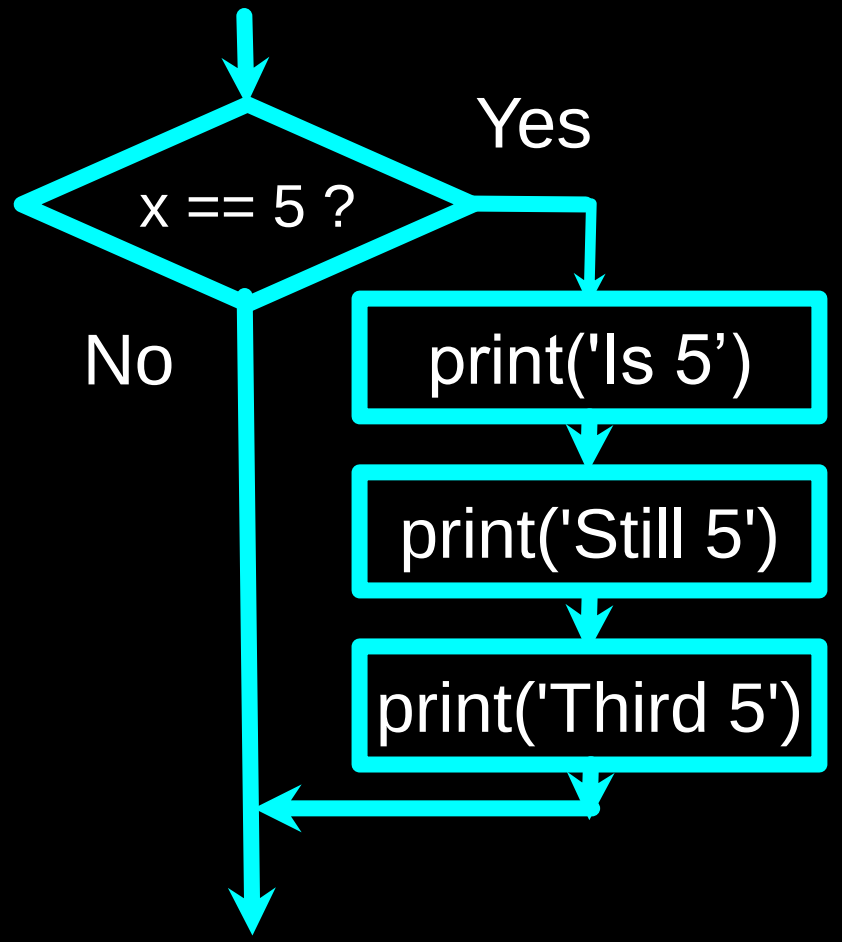
Less than or Equals 5

Not equal 6

One-Way Decisions

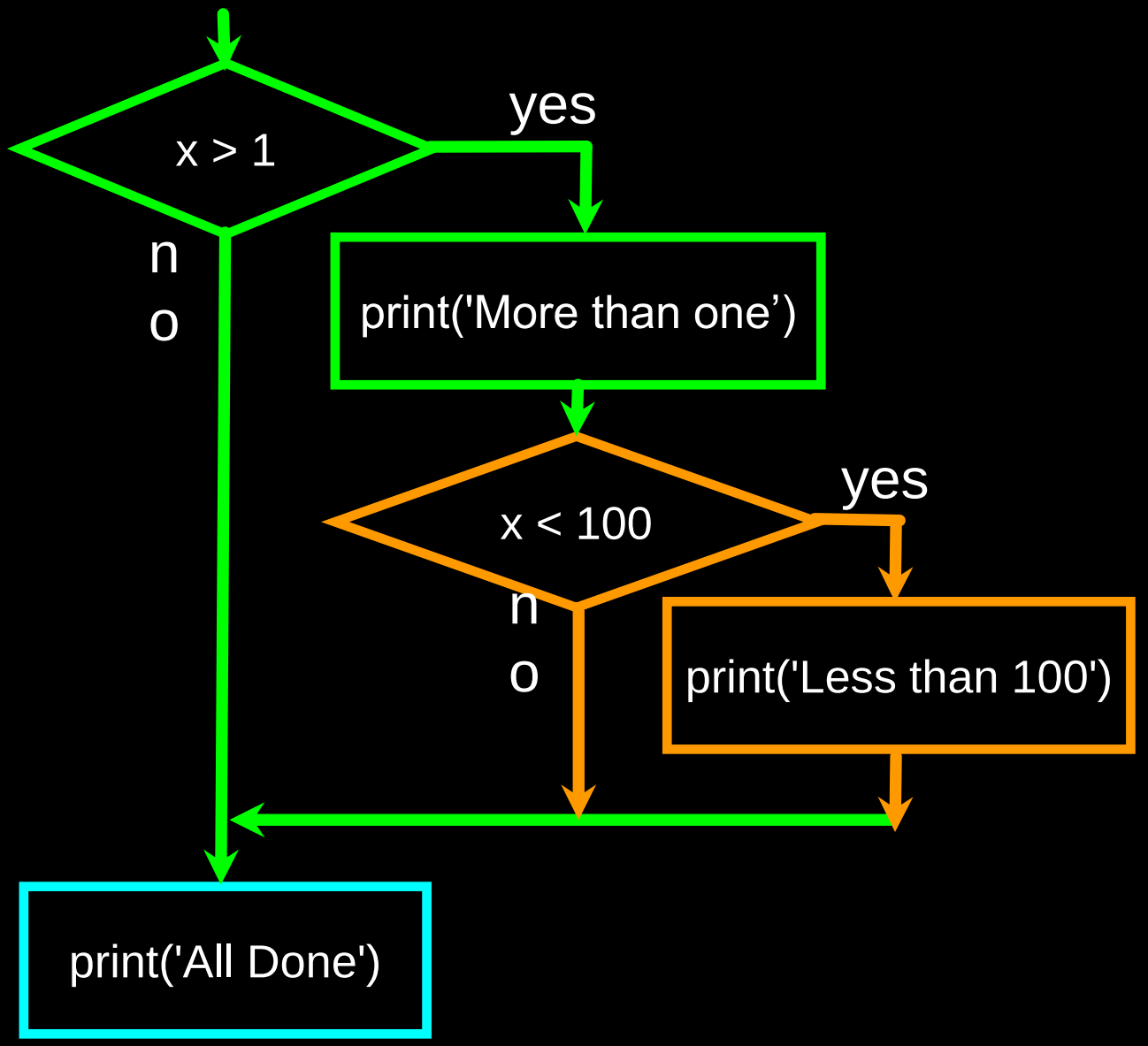
```
x = 5
print('Before 5')
if  x == 5 :
    print('Is 5')
    print('Is Still 5')
    print('Third 5')

print('Afterwards 5')
print('Before 6')
if x == 6 :
    print('Is 6')
    print('Is Still 6')
    print('Third 6')
print('Afterwards 6')
```

```
Before 5

Is 5
Is Still 5
Third 5
Afterwards 5
Before 6

Afterwards 6
```



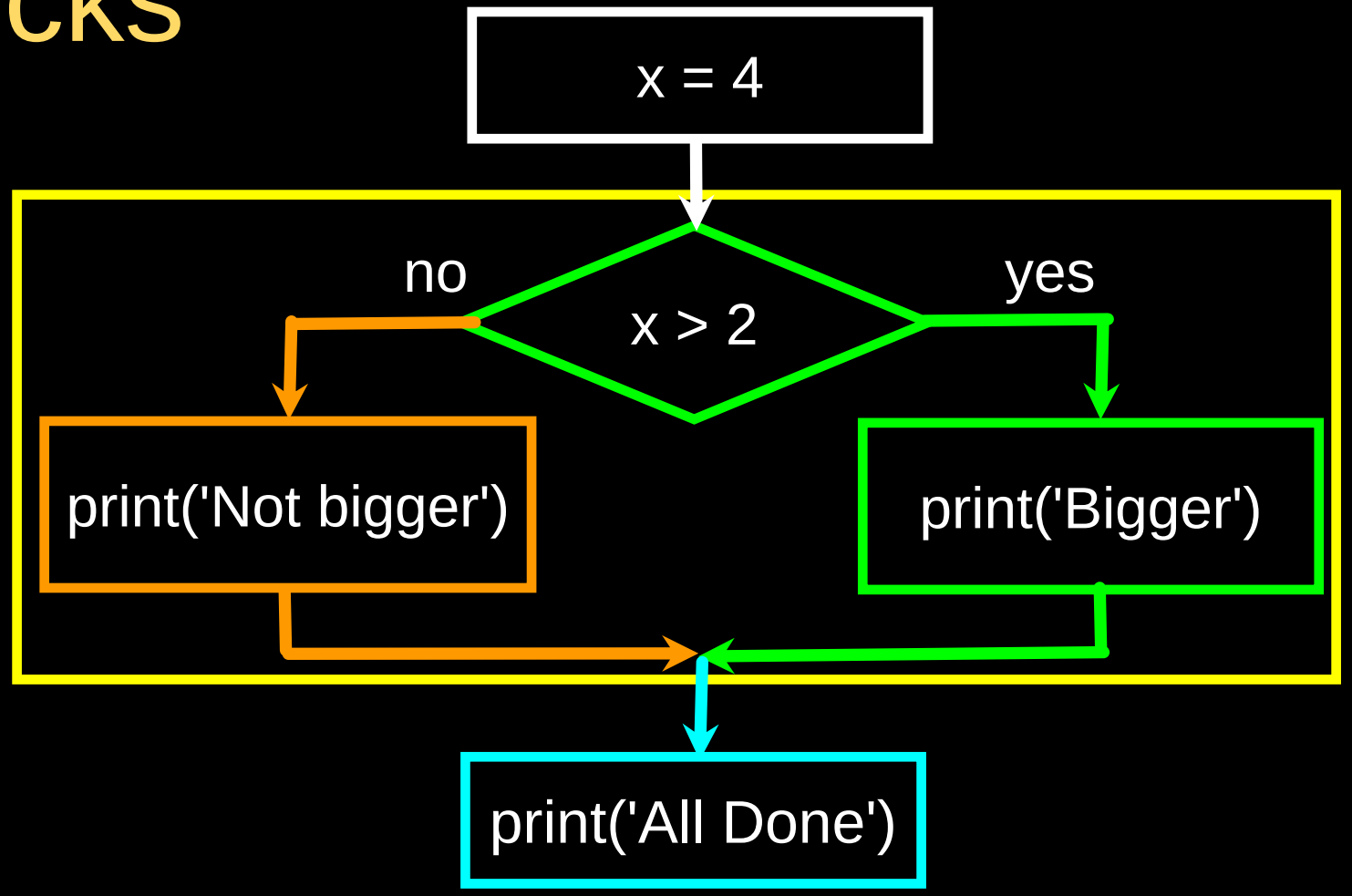
Nested Decisions

```
x = 42
if x > 1 :
    print('More than one')
    if x < 100 :
        print('Less than 100')
print('All done')
```



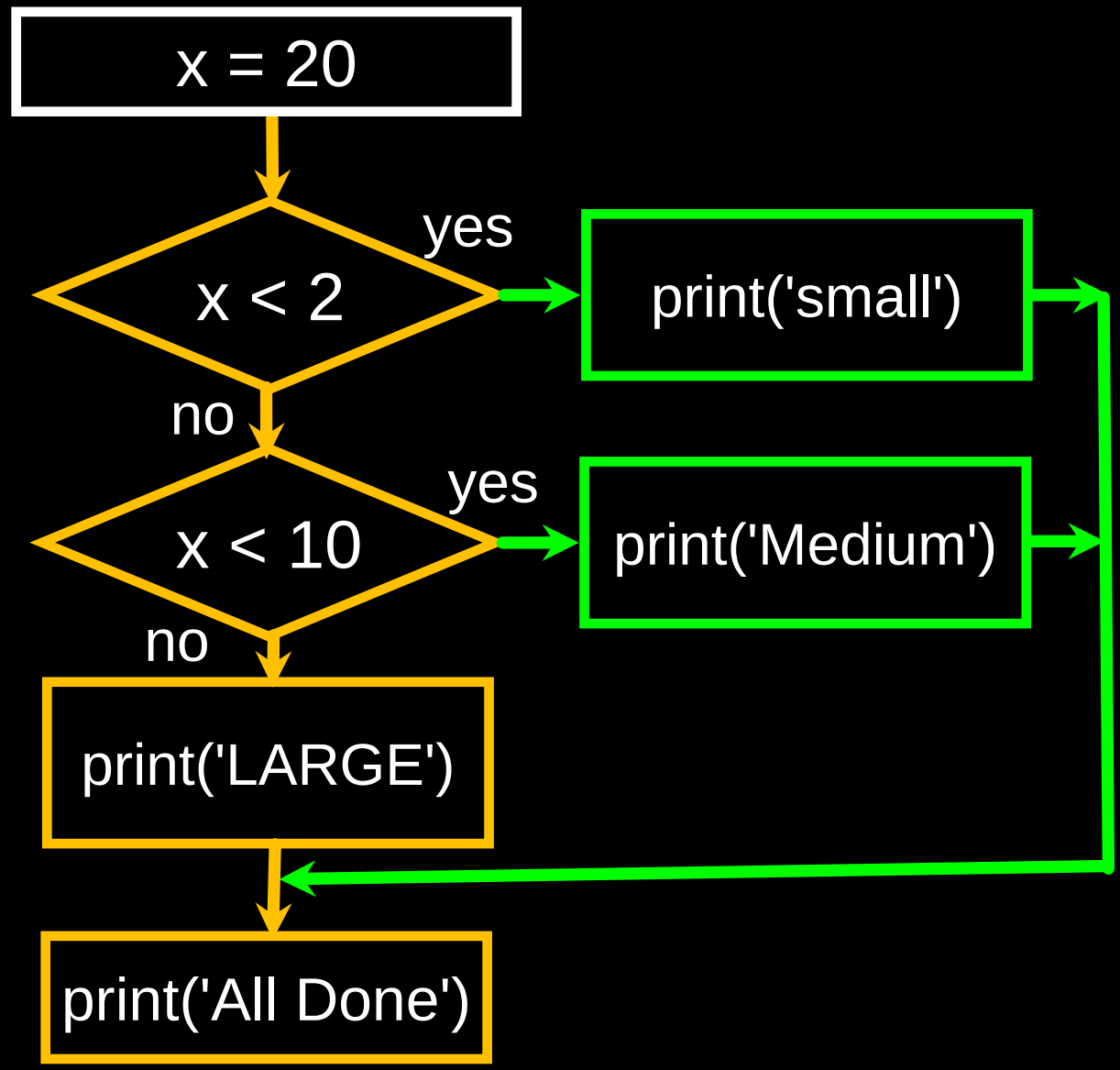
Visualize Blocks

```
x = 4

if x > 2 :
    print('Bigger')
else :
    print('Smaller')

print('All done')
```



Multi-way

```
x = 20
if x < 2 :
    print('small')
elif x < 10 :
    print('Medium')
else :
    print('LARGE')
print('All done')
```



Multi-way

```
# No Else
x = 5
if x < 2 :
    print('Small')
elif x < 10 :
    print('Medium')

print('All done')
```

```
if x < 2 :
    print('Small')
elif x < 10 :
    print('Medium')
elif x < 20 :
    print('Big')
elif x < 40 :
    print('Large')
elif x < 100:
    print('Huge')
else :
    print('Ginormous')
```

Sample Program

Problem: WAP to find largest among 3 numbers.

Solution:

```
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
num3 = int(input("Enter third number: "))

if num1 > num2 and num1 > num3:
   print(num1,"is largest number")
elif num2 > num1 and num2 > num3:
   print(num2,"is largest number")
else:
   print(num3,"is largest number")
```

Sample Program

Problem: Calculate the electric bill using different conditions. For first 100 units Rs1.50 per unit. For next 100 units Rs3.75 per unit. For next 100 units Rs4.25. For rest units Rs5.65 per unit.

Solution:

```
current = int(input("Enter current meter reading: "))
previous = int(input("Enter previous meter reading: "))
total = current - previous

if total <= 100:
    bill = total * 1.5
elif total > 100 and total <=200:
    bill = (100*1.5) + ((total - 100) * 3.75)
elif total > 200 and total <=300:
    bill = (100*1.5) + (100*3.75) + ((total - 200) * 4.25)
else:
    bill = (100*1.5) + (100*3.75) + (total - 200*4.25) + ((total - 300) * 5.65)

print(bill," is total amount")
```

Sample Program

Problem: Get a new string from a given string where 'Is' has been added to the front. If the given string already begins with 'Is' then return the string unchanged

Solution:

```
str=input(“Enter a String”)
if len(str) >= 2 and str[:2] == "Is":
        print(str)
else:
        print("Is" + str)
```

Sample Program

Problem: Get a the height and weight of a person, compute the BMI and based on BMI print the appropriate message

Solution:

```
height = float(input("What is your height? "))
weight = float(input("What is your weight? "))
bmi = weight / height ** 2
print(bmi)
if bmi < 15:
    print("Very severely underweight")
elif bmi < 16:
    print("Severely underweight")
elif bmi < 18.5:
    print("Underweight")
elif bmi < 25:
    print("Normal (healthy weight)")
elif bmi < 30:
    print("Overweight")
elif bmi < 35:
    print("Obese Class I (Moderately obese)")
elif bmi < 40:
    print("Obese Class II (Severely obese)")
else:
    print("Obese Class III (Very severely obese)")
```

Switch Statement

- A switch statement is a multiway branch statement that compares the value of a variable to the values specified in case statements.
- **Python language doesn't have a switch statement**.
- **Python uses dictionary mapping to implement switch statement in Python**

Contact Me

Dr. Pradyumna Kumar Tripathy

Associate Professor & Head, Dept. of CSE,
Silicon Institute of Technology, Bhubaneswar

STP: Python Programming

Mobile: +91- 9437141874

Email: ptripathy@silicon.ac.in


Thank
You