unacademy

Introduction to Python Programming

Course on Data Structure and Algorithms Using Python

Ankush Saklecha • Lesson 1 • Dec 8, 2023

unacademy

Data Structure using Python

By Ankush Saklecha

unacademy

Algorithm Procedure

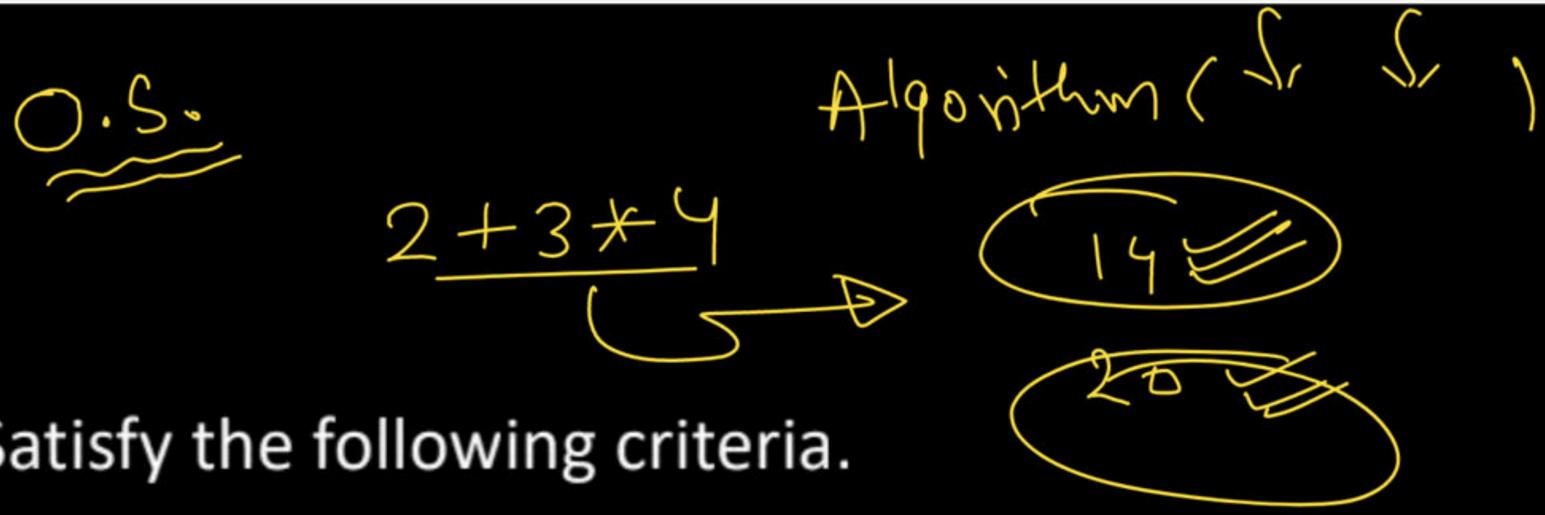
- A **data structure** is a particular way of storing and organizing data so that it can be used effectively.

- A well defined procedure to solve a specific problem is called **Algorithm.**

- **Data Structure + Algorithm = Programming**

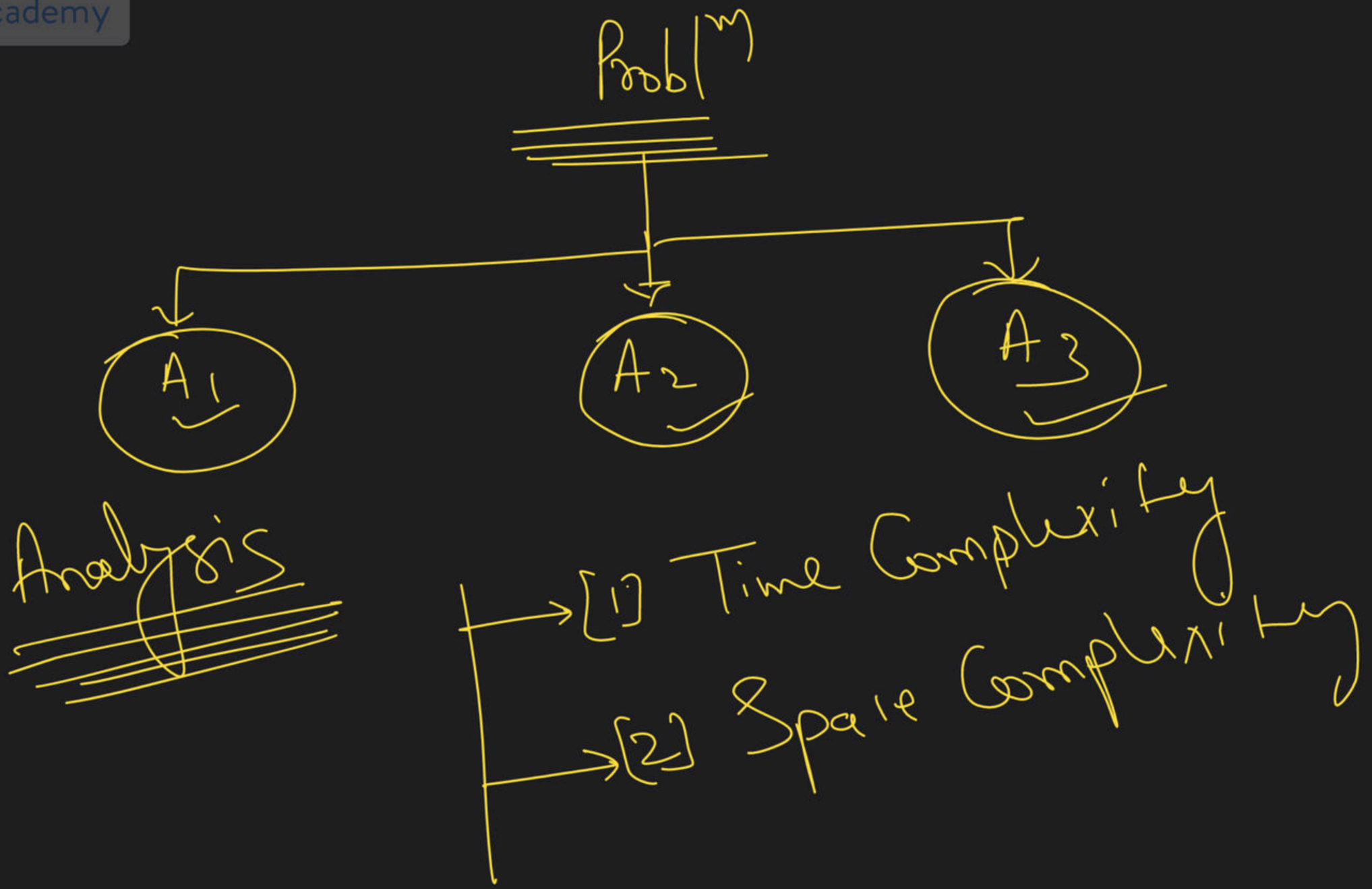
unacademy

Algorithm



An Algorithm Should Satisfy the following criteria.

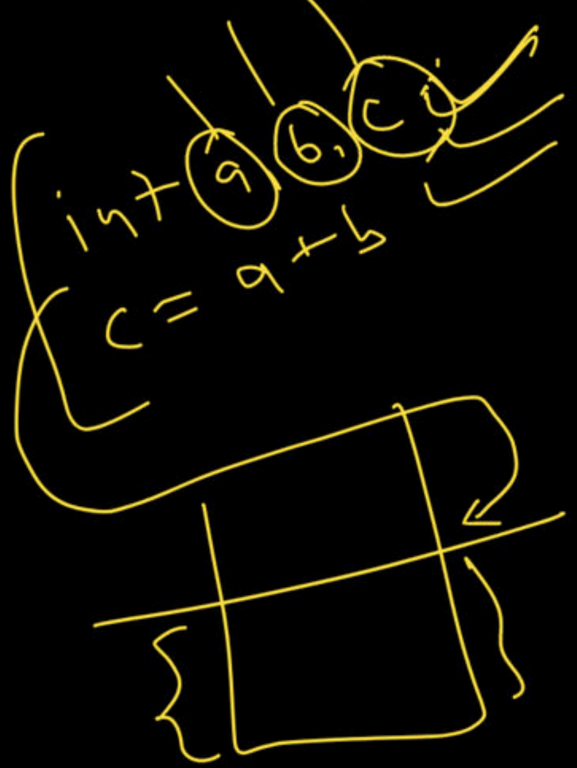
- **Input :** Zero or more quantities are externally supplied.
- **Output :** An algorithm should produce at-least one output.
- **Finiteness:** An algorithm should terminate after finite number of steps.
- **Definiteness:** Algorithm should be clear and unambiguous.
- **Effectiveness:** Algorithm must be language independent and easy to implement in any programming language.

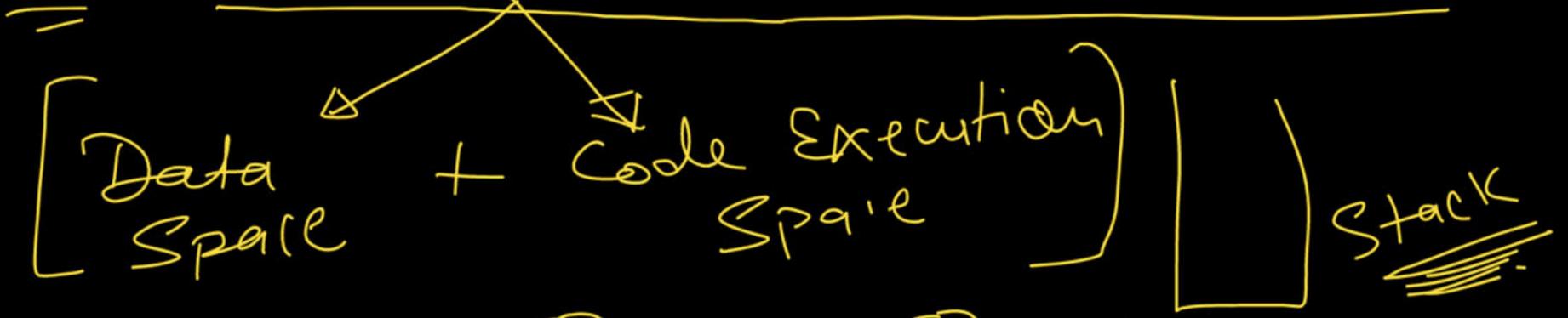

Problem
A1
A2
A3
Analysis
[1] Time Complexity
[2] Space Complexity

unacademy

Behavior of algorithm

- Behavior of an algorithm is analyze on the basis of Time and Space.
  - Time complexity : An amount of time required to run an algorithm.
  - Space complexity : An amount of space required to execute the Algorithm.





	A1	A2
Time complexity ⇒		
Space complexity		

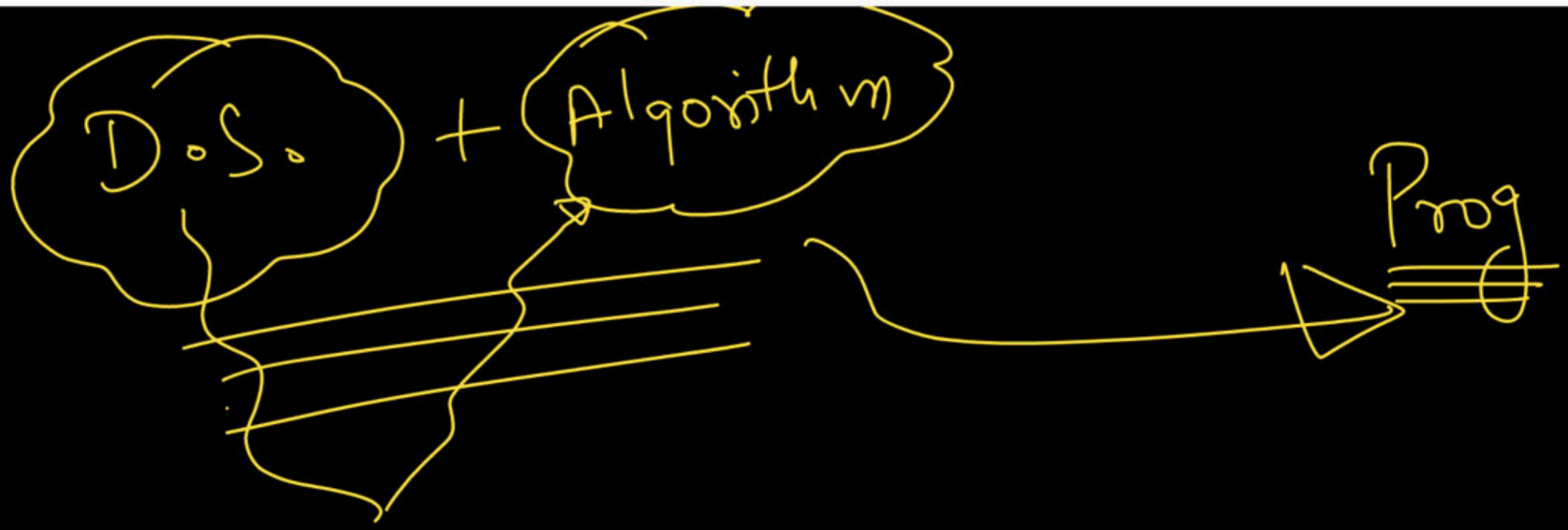
Best

Best

Which algorithm is best A1 or A2?

Time complexity	A1	A2
Best case		
Worst case		
Average Case		

Lower bound <= Running time <= upper bound



Python Programming Language

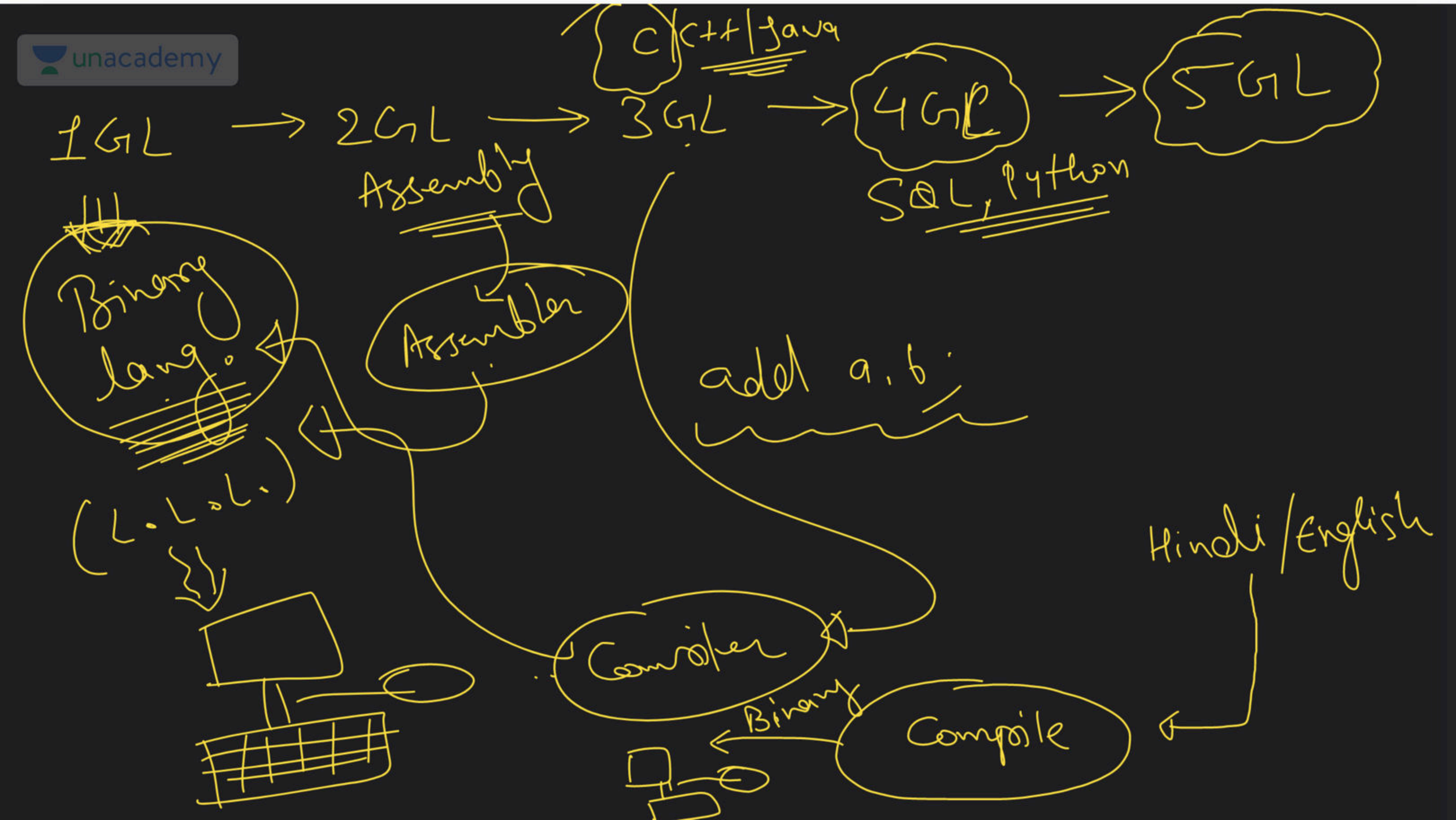
About Python

- Python is created in 1991 by Guido van Rossum.
- It is high level and interpreted language.
- It is an object oriented programming.

unacademy

Why Python?

- Python allows solving of mathematical and scientific problems.
- It has feature to deal with high volume and diverse data sets.
- It support development of artificial intelligence based software.
- It also offers implementation of neural network, deep and machine learning algorithms.
- It support development of Internet of Things (IoT) based applications.
- It offers web development.

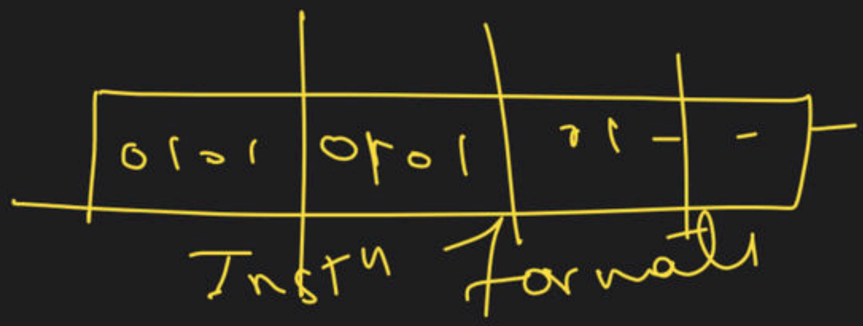

1GL
2GL
Assembly
3GL
C/C++/Java
4GL
SQL, Python
5GL
Binary lang.
(1.1.1.1)
Assembler
add a, b
Compiler
Binary
Compile
Hindi/English

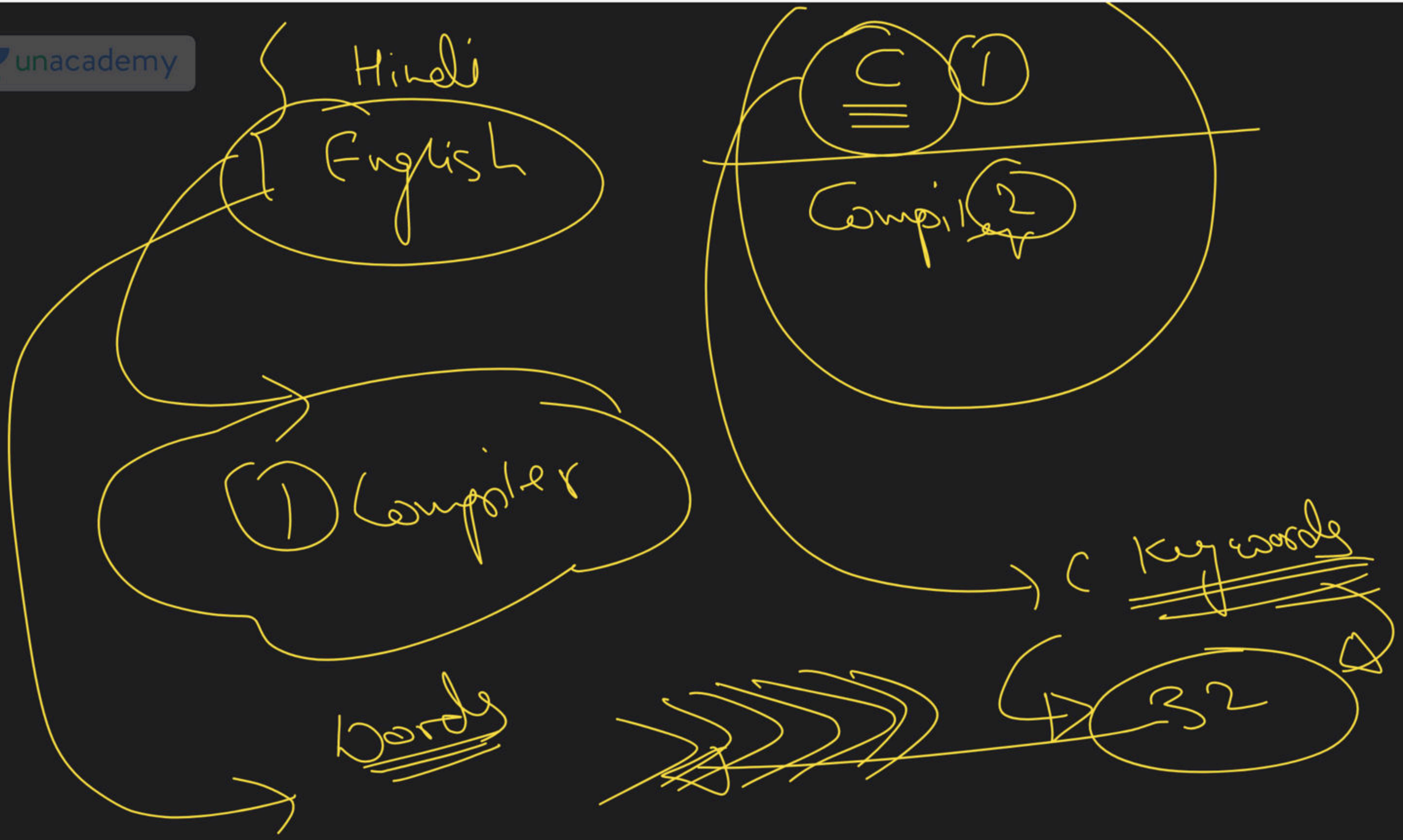
C

C

B.

byte code





Python Vs Other Programming

- Python is dynamically typed language.
- It is free and open source language.
- Python has collections of list, set, tuple and dictionary.
- Python does not support increment and decrement operator.

unacademy

Python Programming Language

- Python is a high-level, general-purpose and a very popular programming language.
- Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry.
- Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

unacademy

Facts of Python Programming

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms.
- Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- It is case sensitive.

main ( )

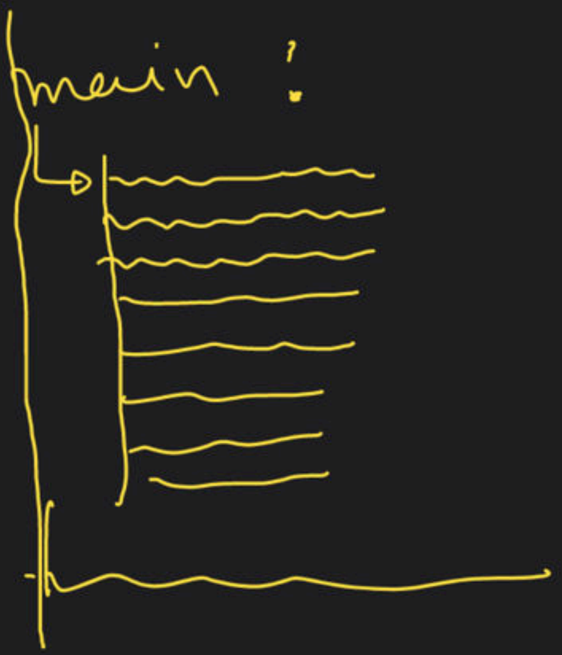
{

;

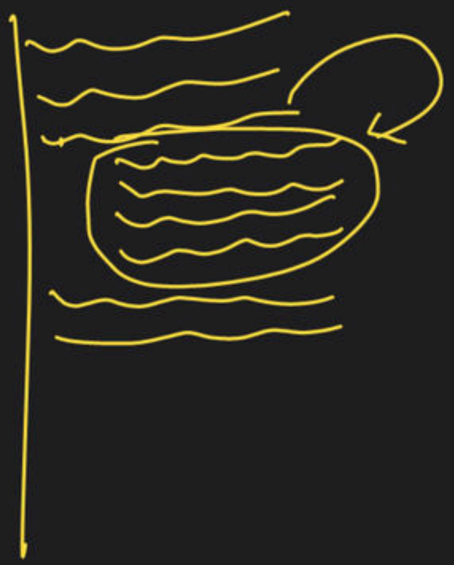
;

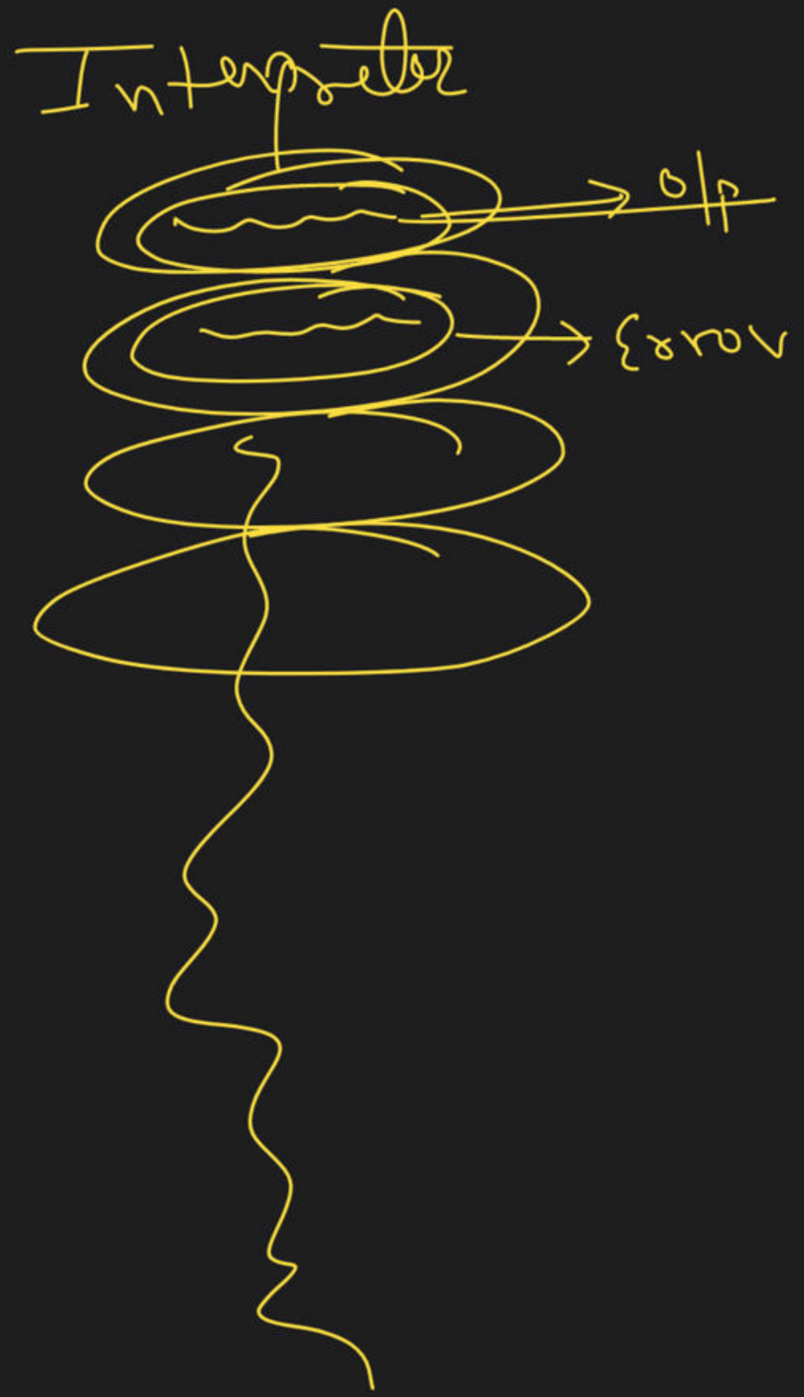
;

}



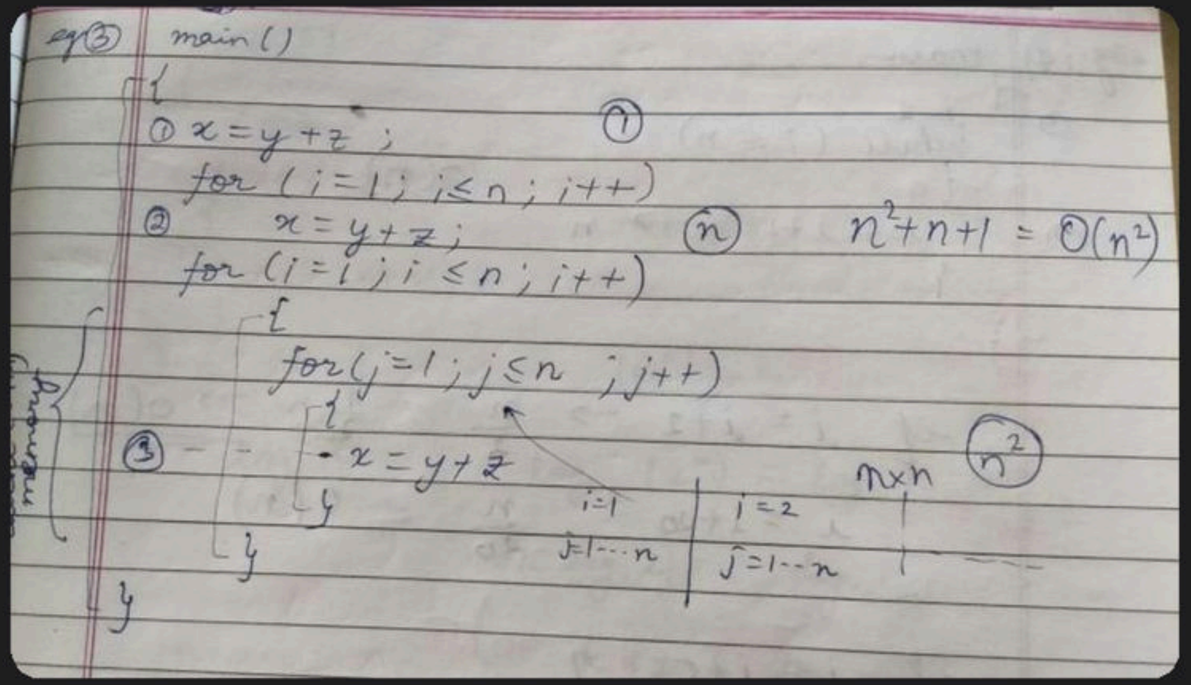
main :





▲ 1 • Asked by Siddharth

Sir can you plz share demo of conversion of this C program into Python



unacademy

Facts of Python Programming

- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
- The biggest strength of Python is huge collection of standard library which can be used for the following:
  - Machine Learning
  - GUI Applications (like Kivy, Tkinter, PyQt etc.)
  - Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  - Image processing (like OpenCV, Pillow)
  - Web scraping (like Scrapy, BeautifulSoup, Selenium)
  - Test frameworks
  - Multimedia
  - Scientific computing
  - Text processing and many more..

unacademy

unacademy

Importance of Python

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

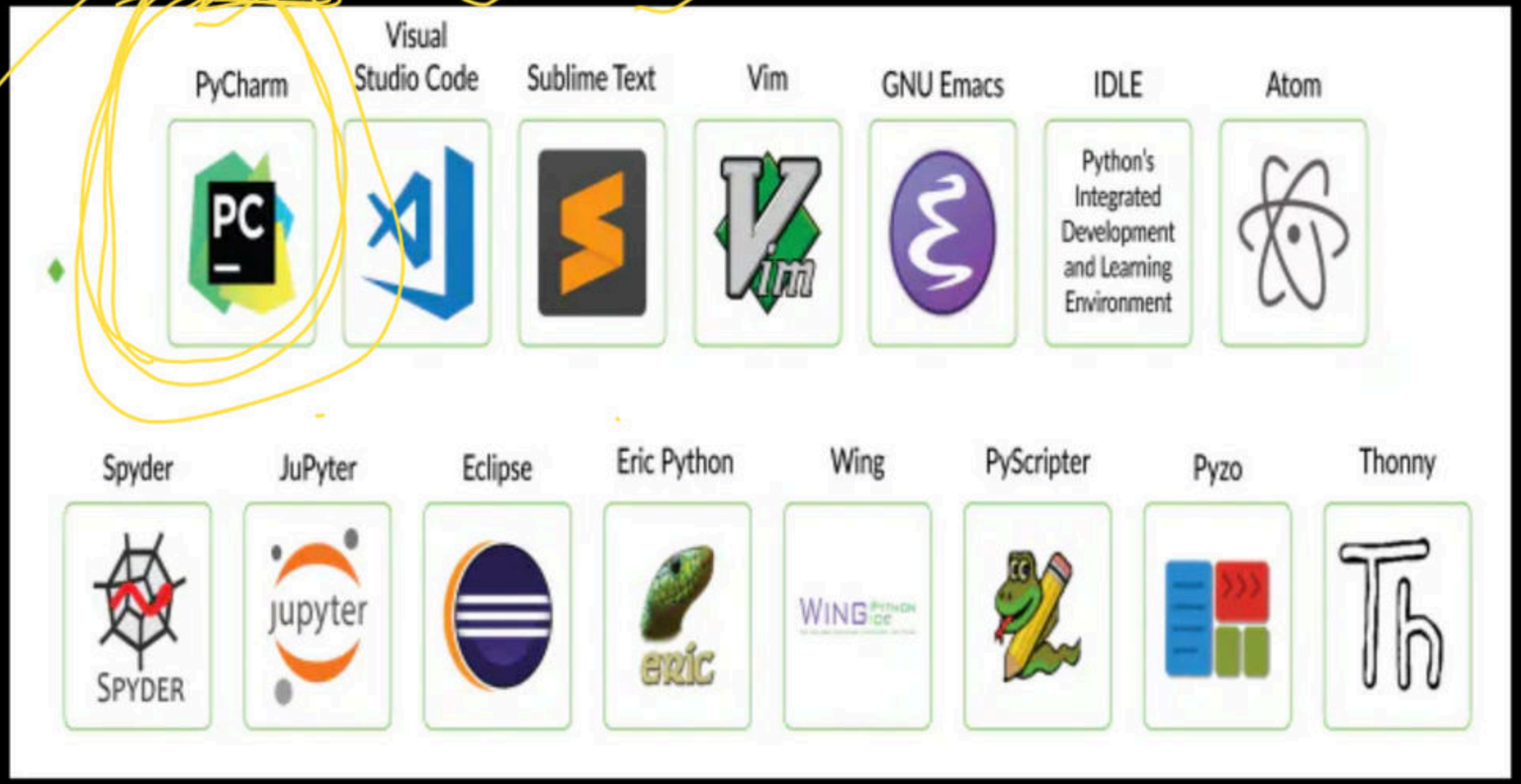
unacademy

Importance of Python

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python IDEs

- Offline IDEs



- Online IDEs

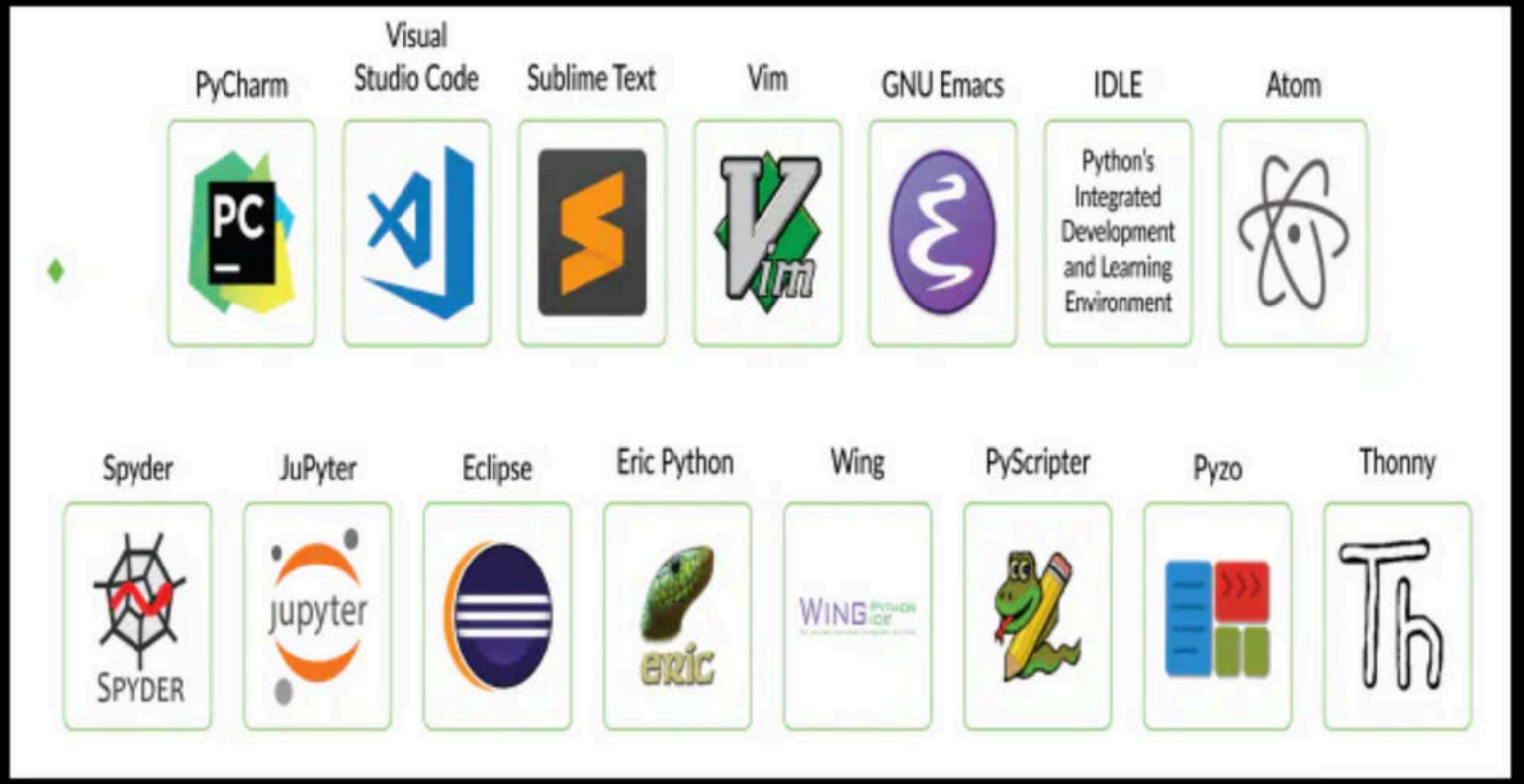
Google colab

replit

programiz

Python IDEs

- Offline IDEs



- Online IDEs

Google colab

replit

programiz

Program Structure

Python Program Structure
Comment Section(Optional)
Functions Definition Section (Optional)
General Statement Section

Python Program Example
First Python Program print('Welcome to Python')

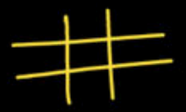
print('Welcome to Python')

C Program Structure
Comment Section(Optional)
Function Declaration Section (Optional)
Preprocessor Section (Optional)
Main Function Section (Mandatory)
Functions Definition Section (Optional)

C Program Example
// First C Program #include<stdio.h> void main(){ printf("Welcome to C"); }

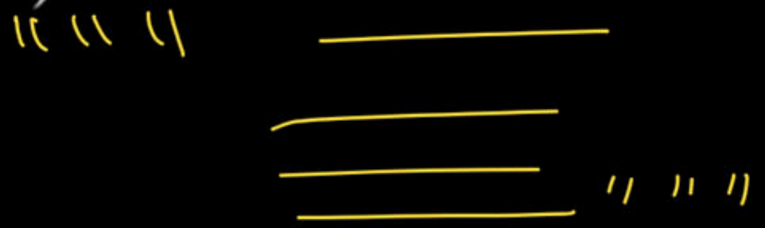
Python Comment

A comment is useful for programmer which help to understand and debug the code.



- Single line Comment- It is start with # (hash)

  Example- # Python Program



- Multiline Comment- It start and end with three double quotes “””.

  Example- “””Python is high language……

  Python is script language……”””

Python Variable and Constant

- A variable is a memory location where any type of data is stored. Address may be change.
- Variable may be declared using a combination of numbers, letters, and the underscore character.
- An example:

#define Pi = 3.14

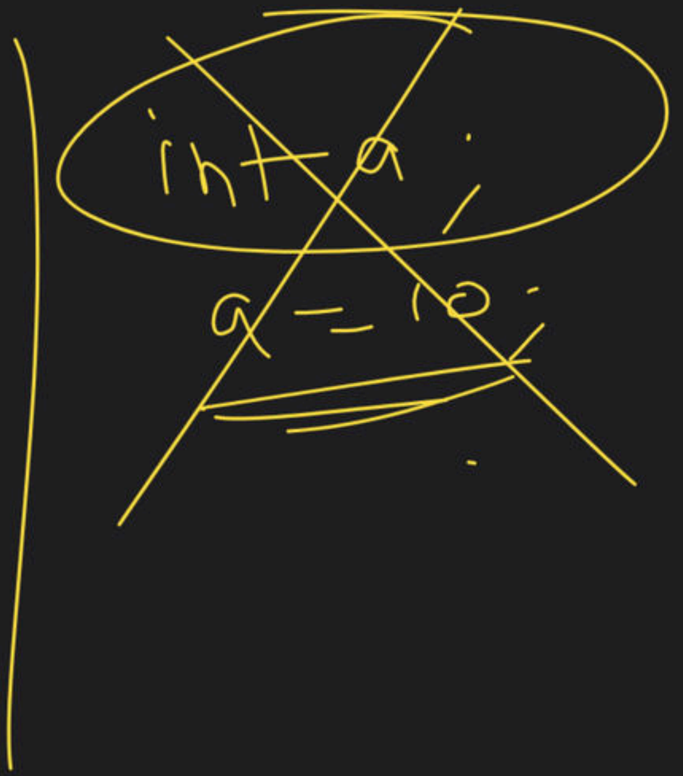
```
pi=3.14
name='Jhon'
```

unacademy

a = 10

b = 10.5

c = "Hello"



Python Variable

An example of C variable:

int a=10;

a= 20;

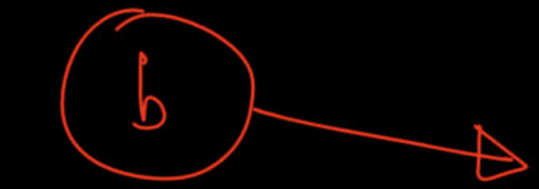
a	
Value	20
Location Address	0x4e5

a = 10
↳ address of a
a = 20
↳ address of a

An example of Python variable:

a=10

a=20

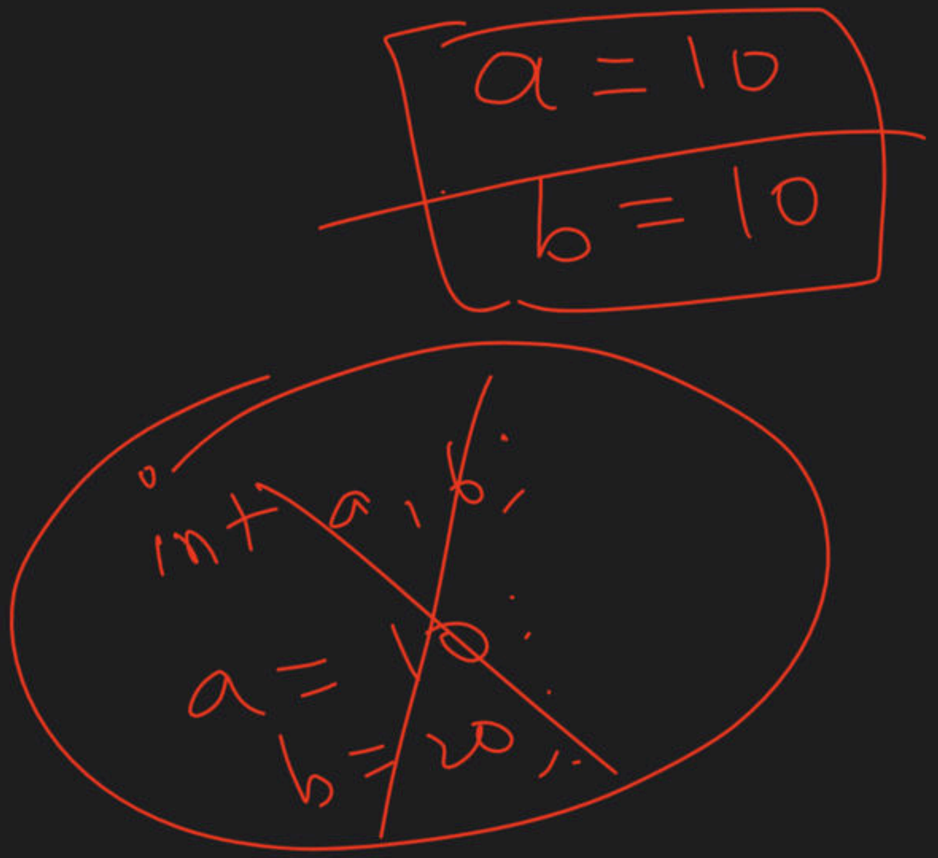


Python Name
a

Pyobject	
Type	Integer
value	10
Address	139653862310416
Refcount	0

Pyobject	
Type	Integer
value	20
Address	139653862324578
Refcount	0


a = 10
b = 10
Py Object



unacademy

Python Variables, Data Types, Input Method

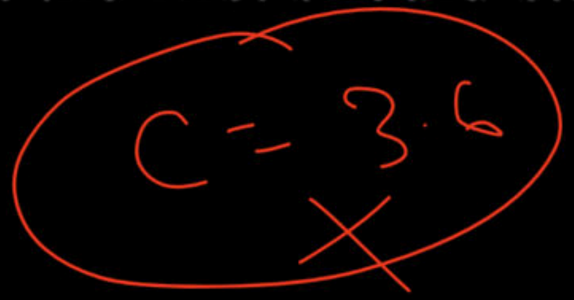
Rules for Variable

- The name of a variable cannot start with a number. It should start with either an alphabet or the underscore character.
- Variable names are always case sensitive and can contain alphanumeric characters and the underscore character.
- Reserved words cannot be used as variable names.
- Python Variables are always assigned using the equal to sign followed by the value of the variable.

Case for Variable Name

- Camel Case- In camel case each word except the first should start with a capital letter.

  Example- circleRaduis=3.6

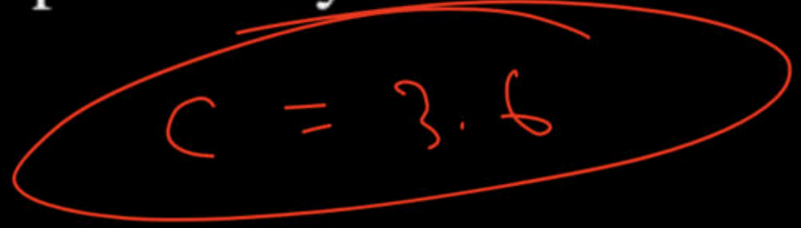


- Pascal Case- In this case each word should start with a capital letter.

  Example- CircleRaduis=3.6

- Snake Case- In this case each word should separated by underscore.

  Example- circle_raduis=3.6



Python Variables

- In Python, variables are created when you assign a value to it:

- Variables in Python:
  - x = 10
  - y = "Hello, Dear“

- Python has no command for declaring a variable.

Python Variables

- In Python, variables are created when you assign a value to it:
- Variables in Python:
  - x = 10
  - y = "Hello, Dear“
- Python has no command for declaring a variable.
- Variables are containers for storing data values.
- Variable names are case-sensitive.

Let's Try…

Guess Valid/Invalid Variable Name	
1. PI=3.14	Valid
2. _radius=5.5	Valid
3. Student_Name='KUSH'	Valid
4. @Marks=85	Invalid
5. X%=6	Invalid
6. a/b=8	Invalid
7. a/=6	Invalid

unacademy

Example

```
x = 10          # x is now of type int
y = "Kush"  # y is now of type str
print(x)
print(y)
```

unacademy

x = 3

- If you want to specify the data type of a variable, this can be done with casting.

```
x = str(3)    # x will be '3'
y = int(3)    # y will be 3
z = float(3)  # z will be 3.0
```

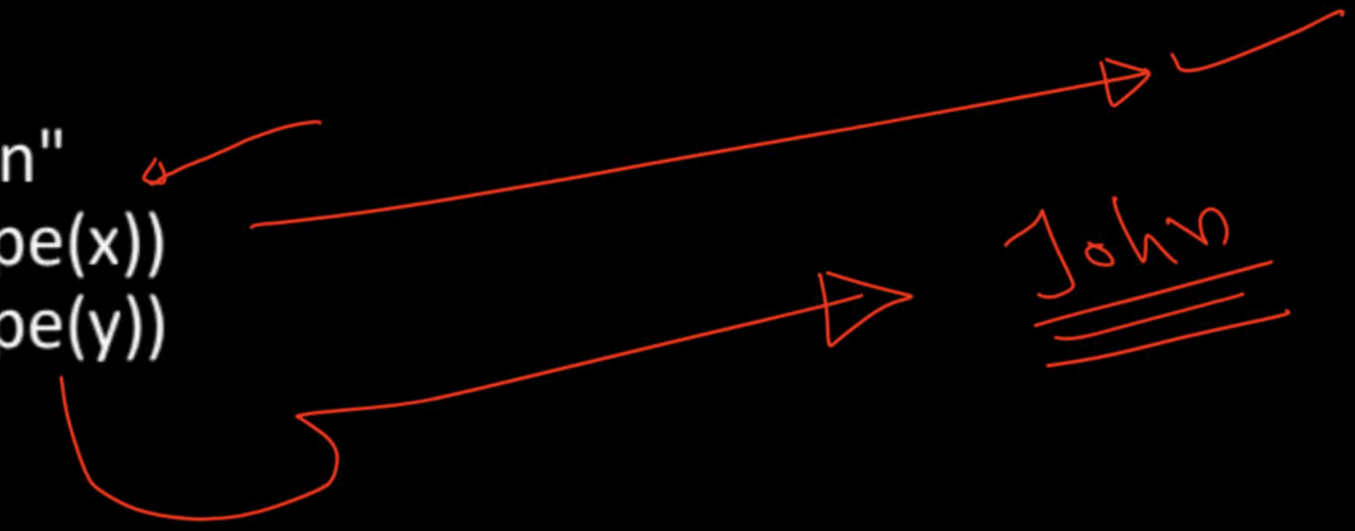
x = str(3)

'3'

unacademy

- We can get the data type of a variable with the type() function.

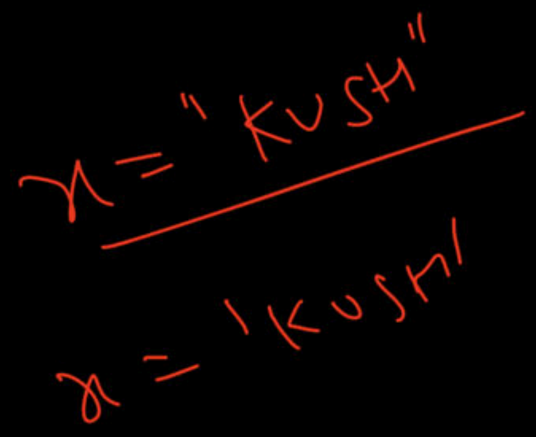
```
x = 5
y = "John"
print(type(x))
print(type(y))
```



unacademy

- String variables can be declared either by using single or double quotes:

```
x = “Kush"
# is the same as
x = ‘Kush'
```



- Python allows you to assign values to multiple variables in one line:

```
x, y, z = “Luv", “Kush", “Atharva"
print(x)
print(y)
print(z)
```

- We can assign the *same* value to multiple variables in one line:

```
x = y = z = “Kush”
print(x)
print(y)
print(z)
```

fruits = ["apple", "banana", "cherry"]

→ Name of List

→ List

- If you have a collection of values in a list, tuple etc. Python allows you to extract the values into variables. This is called *unpacking*.

```
fruits = ["apple", "banana", "cherry"]
x, y, z = fruits
print(x)
print(y)
print(z)
```

- In the print() function, you output multiple variables, separated by a comma :

```
x = "Python"
y = "is"
z = "awesome"
print(x, y, z)
```

unacademy

Example :

```
x = "Python "
y = "is "
z = "awesome"
print(x + y + z)
```

print (x, y, z)

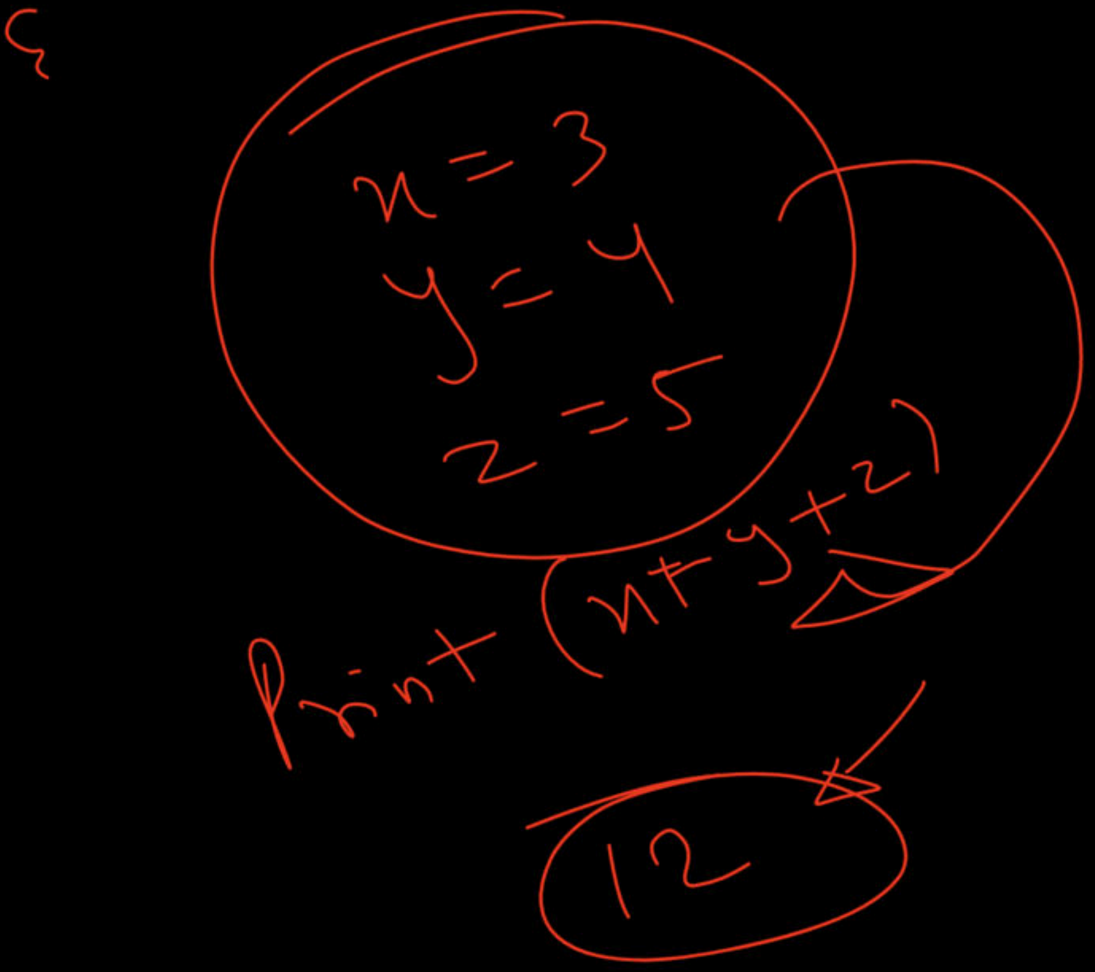
- Output : Python is awesome

unacademy

Example :

```
x = "Python "
y = 5
z = "awesome"
print(x + y + z)
```

- Output : Error



print(x, y, z)

3 4 5

unacademy

```
x = "3"
y = "4"
z = "5"
print (x + y + z)
```

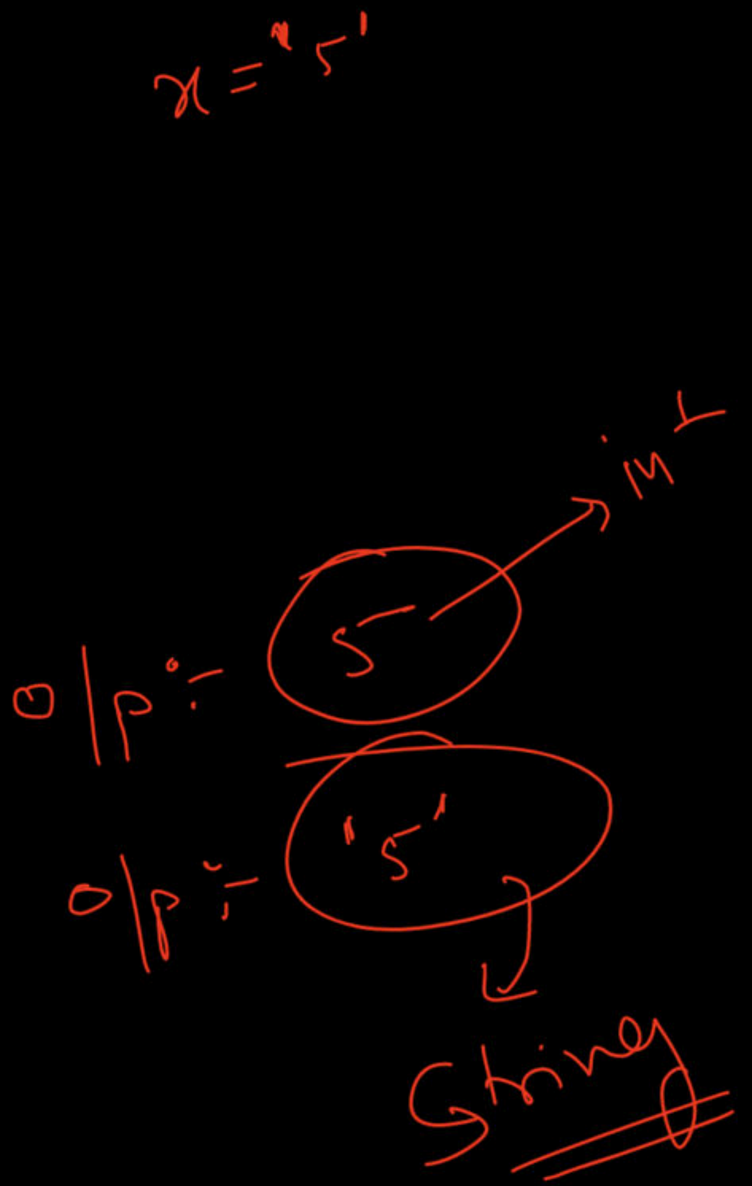
```
3 4 5
```

unacademy

Example :

```
x = "Python "
y = 5
z = "awesome"
print(x , y , z)
```

- Output : Python 5 awesome



unacademy

Example :

```
x = 10
y = 5
z = 20
print(x + y + z)
```

- Output : 35

Global Variables

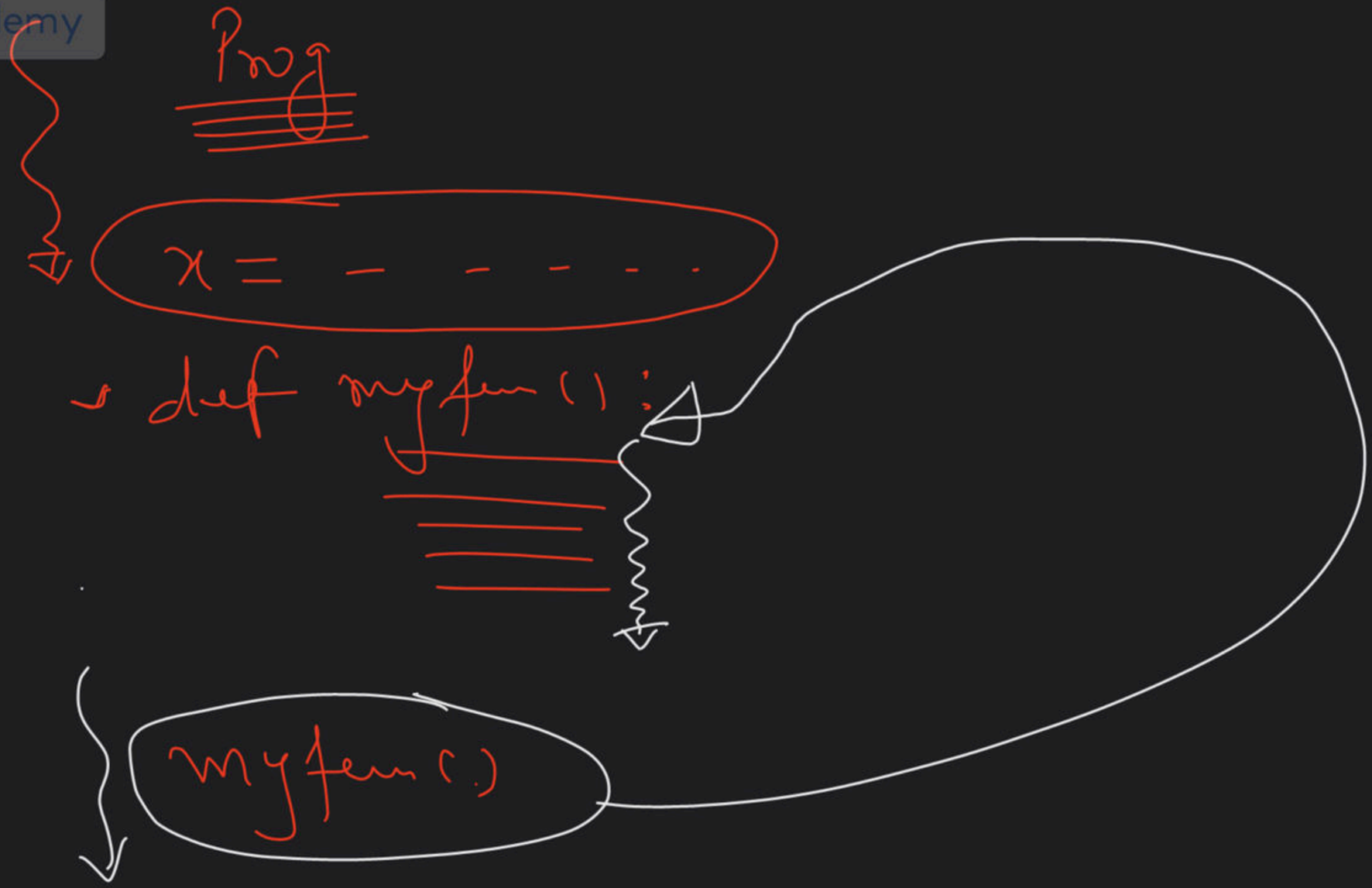
- Variables that are created outside of a function (as in all of the examples above) are known as global variables.
- Global variables can be used by everyone, both inside of functions and outside.

```
x = "animal"

def myfunc():
  print("Lion is " + x)

myfunc()
```

```
def myfunc():
  Print("Lion is" + x)
```



Example

```
x = "awesome"

def myfunc():
  x = "fantastic"
  print("Python is " + x)

myfunc()

print("Python is " + x)
```

Output : Python is fantastic

Python is awesome

unacademy

- To create a global variable inside a function, you can use the global keyword.

```
def myfunc():
  global x
  x = "fantastic"

myfunc()

print("Python is " + x)
```

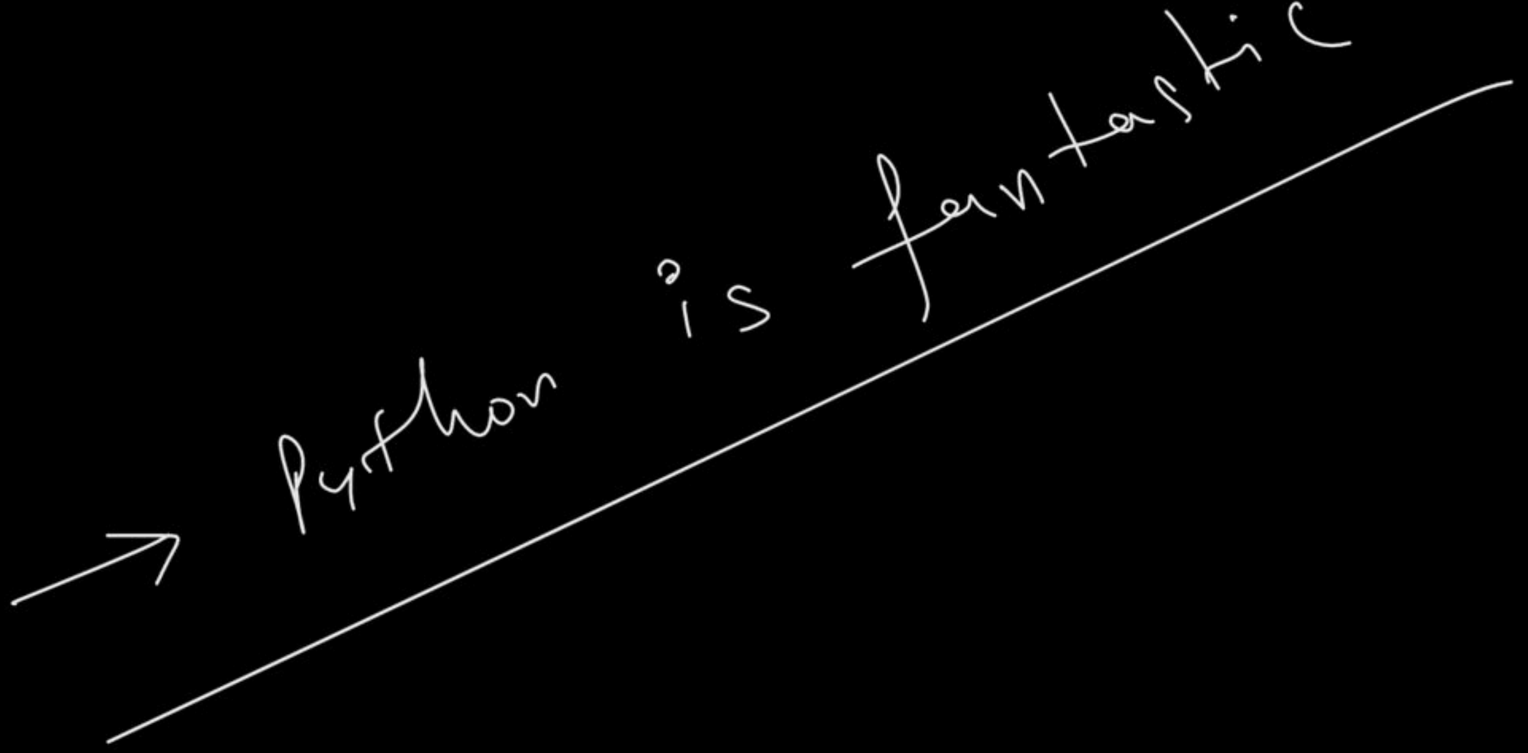
unacademy

- To change the value of a global variable inside a function, refer to the variable by using the global keyword:
- x = "awesome"

```
def myfunc():
  global x
  x = "fantastic"

myfunc()

print("Python is " + x)
```



Python Data types

- Integer: Integer allows numeric value only. It is immutable.(This means that once you create an integer variable and assign it a value, you cannot change the value of that variable. If you want to update the value of an integer variable, you have to create a new integer object with the desired value and assign it to the variable.)
- Float: It support real number. It is also immutable.
- String- It represent single or group of characters. Immutable
- Boolean- It gives two value either true or false.

a = 10 20

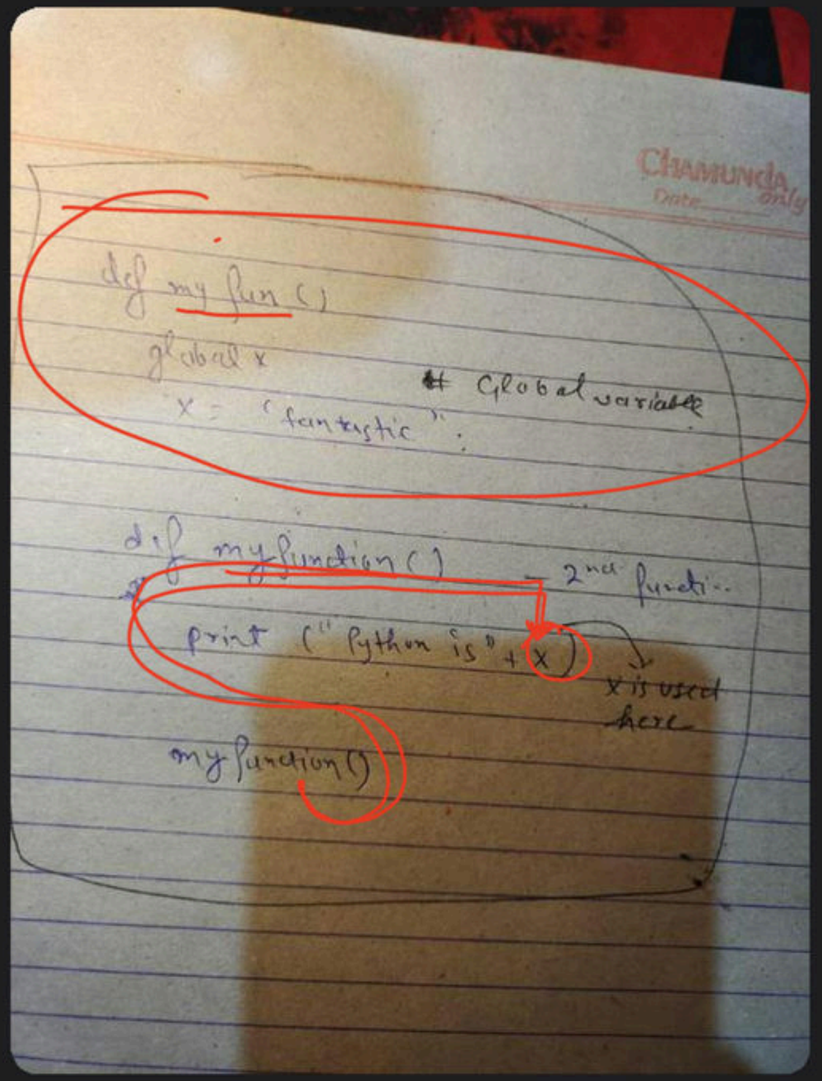
a = 10

a = 20

unacademy

▲ 1 • Asked by Ankit

Sirji ism error aayega kya?



Data types Examples

```
# Python program to demonstrate Data types
a=10      #integer datatype
pi=3.14     #float datatype
str='A'    #string datatype
print(a)
print(pi)
print(str)
print(type(a))
print(type(pi))
print(type(str))
```

10
3.14
A
int
fl
str

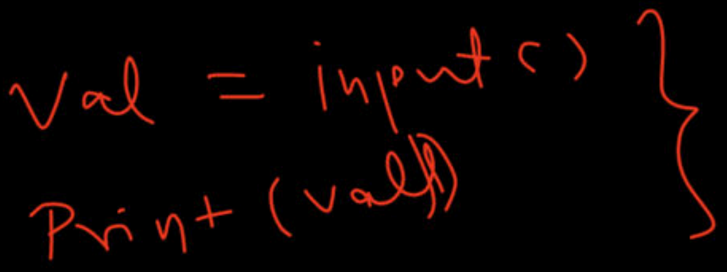
unacademy

Taking input in Python

- **input ():** This function first takes the input from the user and converts it into a string.
- The type of the returned object always will be <type 'str'>.
- It does not evaluate the expression it just returns the complete statement as String.
- For example, Python provides a built-in function called input which takes the input from the user.
- When the input function is called it stops the program and waits for the user's input.
- When the user presses enter, the program resumes and returns what the user typed.

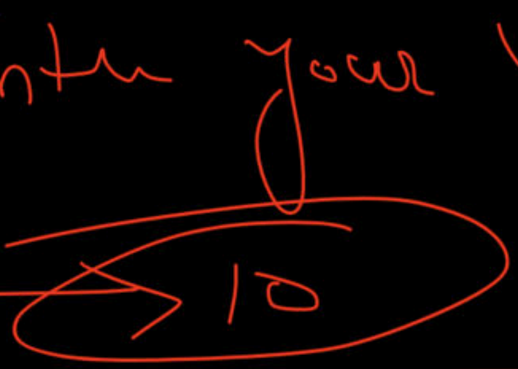
unacademy

Example



- # Python program showing a use of input()

```
val = input("Enter your value: ")
print(val)
```



- **raw_input():** This function works in older version (like Python 2.x).

Input and Output: Reading Input

There are many ways to assign value or read input for the python variables.

Method-1: Assign value using assignment operator
Example:
diameter=5.8 vehicle='car' pi, a, name=3.14,55, 'David' x=y=z=25

unacademy

Input and Output: Reading Input

Method-2: Assign value using input() function
Example:
diameter=float(input('Enter value of diameter')) vehicle=input('Enter name of vehicle') a, b=input('Enter two values').split() x = list(map(int, input("Enter multiple values: ").split()))

10 20 30 40 50

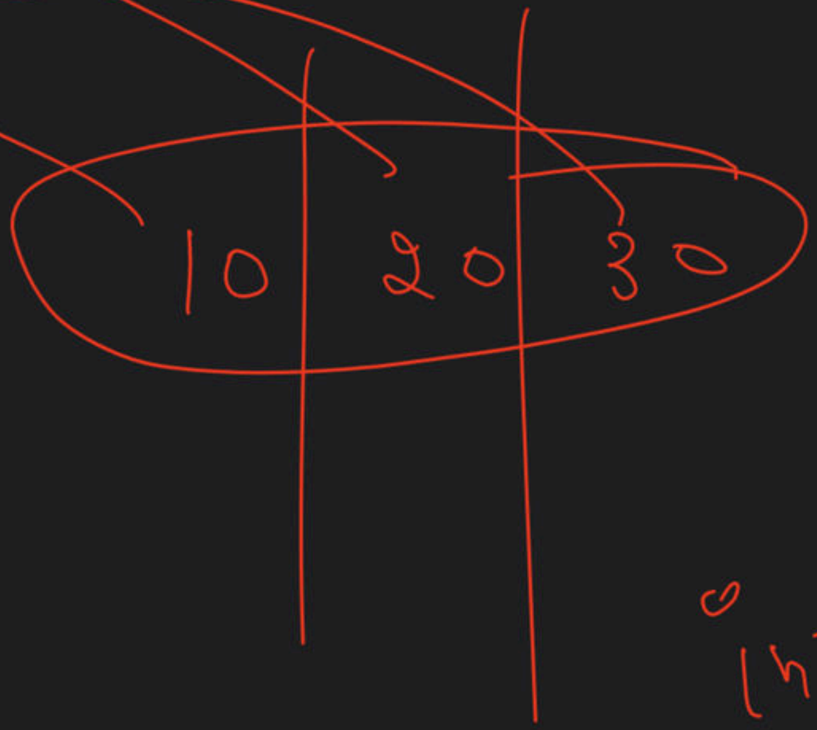
x = [10, 20, 30, 40, 50]

```
x = list(input("Enter multiple values ").split())
print(x)
```

Enter multiple values 10 20 30 40 50

['10', '20', '30', '40', '50']

a, b, c = map(int, input("Enter three value").split())



int( )

float( )

Let's Try…

Guess Output?

Code-1:
```
x, y = input("Enter number of boys and girls: ").split('.')
print("Number of boys: ", x)
print("Number of girls: ", y)
```

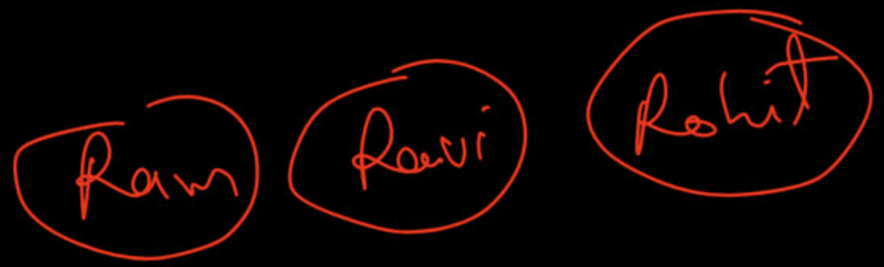
Input- 6 8

Output- **Error**

Code-2:
```
x = list(map(int, input("Enter multiple values: ").split()))
print(x)
```

Input- 6.5 2.4 4.8

Output- **Error**



Let's Try…

Guess Output?

Code-3:
```
x = list(map(int, input("Enter multiple values: ").split()))
print(x+x)
```

Input- 4 8 12

Output- [4, 8, 12, 4, 8, 12]

Code-4:
```
name = "Jony" * 3 * 4
print(name)
```

Output- JonyJonyJonyJonyJonyJonyJonyJonyJonyJonyJonyJony

Input and Output: Displaying Output

In python, print() function is uses to print or display a string or value of variables. A syntax for print() function given below.

Displaying Output: Method-1
Syntax:
print('string',var1,var2....)
Example:
name = "Jenny" print('Name of student is: ', name)

Input and Output: Displaying Output

Displaying Output: Examples

```
print('Welcome \n User')
```

Welcome
User

```
print('Welcome',end=' ')
print('User')
```

Welcome User

```
print("Hello", "how are you?", sep='--')
```

Hello -- how are you?

```
x = float('14.34')
print('%f, %e, %F, %E' % (x, x, x, x))
```

When you run this code, it will format and print the value of x according to the specified format specifiers:
%f: Formats x as a decimal floating-point number.
%e: Formats x in scientific notation (lowercase 'e').
%F: Formats x as a decimal floating-point number with uppercase 'F'.
%E: Formats x in scientific notation with uppercase 'E'.

Output :

14.340000, 1.434000e+01, 14.340000, 1.434000E+01

Let's Try…

Guess Output?

```
x = float('NaN')
print('%f, %e, %F, %E' % (x, x, x, x))
```

nan, nan, NAN, NAN

```
print('%x, %X' % (15, 15))
```

f, F

```
print('%c' % 69)
```

E

```
print('%d %d %.2f' % (11, '22', 11.22))
```

Error : use %s for '22'

```
name = "Jony" * -3  * 4
print(name)
```

Nothing

Input and Output: Output Formatting

With the format () method, print() function display in formatted form. A syntax for print() function with format method() given below.

Displaying Output: Method-2
Syntax:
print(‘UserString{} '.format(‘Otherstring'))
Example:
print('Welcome {}'.format('User')) print('{0} and {1}'.format(‘Hello', ‘World'))

Let's Try…

Guess Output?
print('Best College is {0}, {1}, {other} ' .format('IISC', 'For', other ='M.Tech'))
Best College is IISC, For, M.Tech
print(" No of Circles: {1:d}, Area of each Circle: {0:7.2f}". format(167.34, 6))
No of Circles: 6, Area of each Circle: 167.34
print('@@', '@@', '@@', sep='@@')
@@@@@@@@@@@

Sample Problems

Problem-1:

Write a program that reads sequence of inputs from the user provided on single row and separated by the character ':‘ and display each number.

Example :

Sample Input: 3:12:21:8:4:7

Sample Output: [‘3’, ’12’, ’21’, ‘8’, ‘4’, ‘7’]

Problem-2:

Write a python code to read three heterogeneous data items from the user and display them.

Example

Sample Input- 6 8.9 abc

Sample Output- [‘6’, ‘8.9’,’abc’]

unacademy

#Program 1 :

```
x = input("enter multiple value").split(":")
print (x)
```

#Program 2 :

```
x = input("enter multiple value").split()
print (x)
```

Thank You