~~Overall work~~

Python

Being one of the most powerful, and versatile programming languages, Python is easy to learn to learn and is widely used among developers. Its popularity and demand is increasing ever since its launch and still continuing to grow. Demand for Python developers has ted by 41% in last 5 years. Out of 28 million developer in the world, 8.2 million are Python develop

What is python?

Python is a high-level, general purpose and interpreted programming language used in various sectors including M.L, A.I, data analysis, web development, and many more. Python is known for its ease of us, powerful standarad library, and dynamic semantics.

History of Python

Python, first created in 1980s by Guido Van Rossum one of the most popular prog. language. During his research at the National Research Institutude for Mathema and Computer Science in the Netherlands. The first even v was released in the year 1991.

Python 1.5 released in 1997
" 2.0 " in 2000
" 3.0 in 2008 brought newer functionalities

The latest version of python, python 3.10 was released in 202

Newer - functionalities being added to python makes it's.

Python 2x	Python 3.X
← Legacy	Future →
Library	Library
ASCII	Unicode
7/2=3	7/2 = 3.5
print "Welcome to Geeksforgeeks"	print("Welcome to Geeksforg

Features of Python

- Easy to read and understan
- Interpreted language
- object-oriented programming System language (OOPs),
- Free and open-Source
- Multi-platform
- Hundreds of libraries and frameworks ...
- Flexible, supports GUI
- Dynamically typed
- Huge and active Community

~~Advantage of and Dist~~

Advantages of python

→ Easy to learn, read and understan
versatile and open source
Improves productivity
Supports libraries
Huge library
Strong community
Interpreted language

Disadvantages

Restrictions in design
Memory inefficient
Weak mobile computing
Runtime errors
Slow execution speed
less security provider

PVM - It is Power Virtual Machine which deletes unknown object and Convert machine code to Bi-Code.

Applications of Python:

Python is a highly-demanding and popular programming language:

Back and development
data analysis
data
data scientist
Auto-mation
Machine learning
Deep learning
Big Data analystics
AI
Game development
Cloud

Parameters	C	C++	JAVA	Python
~~Language Type~~ Language Type	Procedure oriented programming	object	〃	both types of language [pop & OOP]
Developer	Dennis Ritchie	Bjarne Stroustrup	James Gosling	Guido van Rossum

Compiler :-

A Comp. translates code from a high-level (programming language) (like python, Javascript or Go) into machine code before the program runs.

Interpreter :- An inter. translates code written in a high-level 〃 into machine code line-by-line as the code runs.

#1 Variable :-

- A variables are containers for storing data values.

We can say that like variable is a containers to hold the single values as well as multiple values in a single variable.

- A variable is the name given to a memory location. A value holding python variable is also known as an identifier.

~~Example :-~~

#1 Printing Single Variable

```
a = 5
```

Single line & Multiline

Single line comments are used to comment single line
Multiple line ~~~~~ multiple ll.

#1 printing Single variable

```
# a=5
print (a)
print ((a))
```

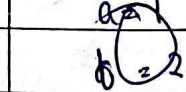
#2 Multiple

```
a=5
b=6
# printing multiple variables
print (a,b)
# Separate the variables by the comma
print (1, 2, 3, 4, 5, 6, 7, 8)
```

variable Creating on behalf of the data types.

```
name = "Devansh"
age = 20
marks = 80.50

print (name)
print (age)
print (marks)
```



```
a=1
b=2
Sum=a+b
Sub=a-b
Div=a/b
Mult=a*b
print(Sum, Sub, Div, Mult)
```

Identifiers Rule :-

Valid character:-

Must beging with a letter (a-z, A-Z) or an underscore (_). The remaining characters can be letters, underscores, or digits (0,9).

Case Sensitivity:-

Python is case sensitive. For example, myVar and myvar are different identifiers.

Reserved words :-

Avoid using python reserved words as identifiers. These are keywords that have special meanings in the language and cannot be used as variable names. Examples include if, else, while, for, class, def, return, and others.

Length :-

There is no maximum length for an identifier but it is recommend to keep them short and descriptive.
Special characters are not allowed as identifiers. like &, ##, etc.

How to Assigning or Re-Assigning the Variable

Python is both a strongly typed and a dynamically typed language. because it provide the such types of features.

Example

```
my = 2
my = ["srk", "abc"]
```

```
a = 5
at = a
print (a)
```

```
a = 5
a* = a*a
print (a)
```

output
25

```
a = 5
a* = a
print(a)
```

output
25

```
a = 5
a* a*a
print(a)
```

Variable design

There are two types of design the variable.

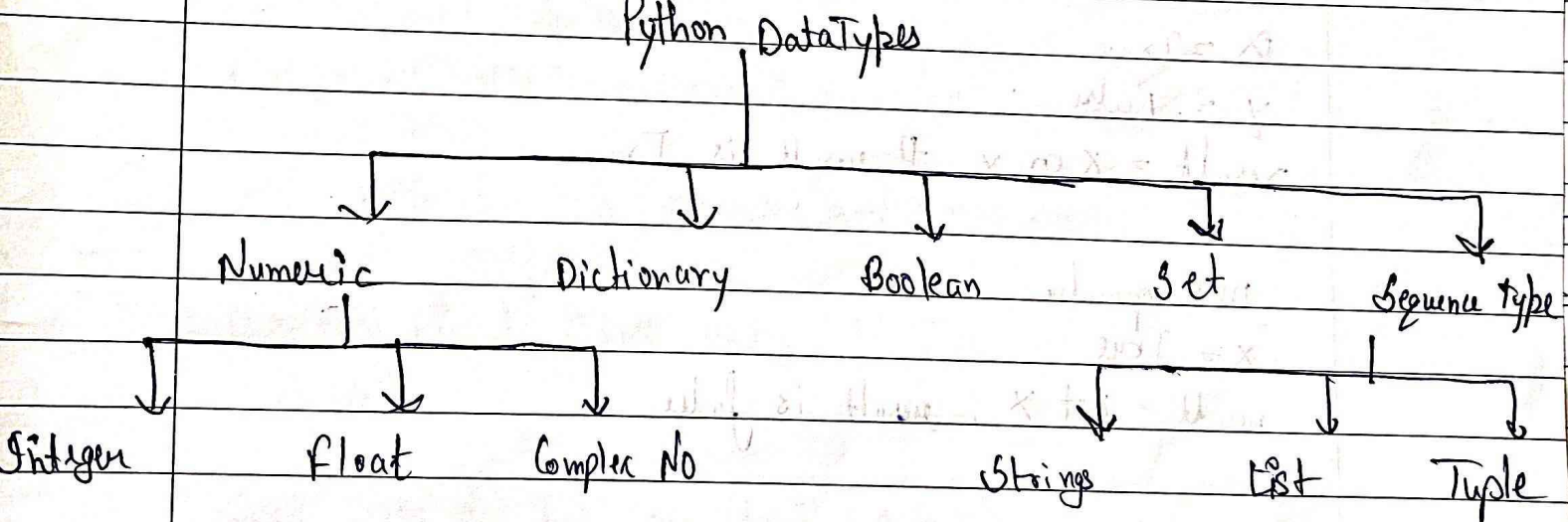
1.) Assigning single value to multiple variables
2.) Assigning multiple value to multiple variables.

```
a=b=c=5
print(a,b,c)
```
Single value to multiple variables

```
a,b,c = 2,3,4
print(a,b,c)
```
Multiple value to multiple variables

```
a=5
print(a)
del(a)
```
Output = 5

```
a=5
del(a)
print(a)
```
Output = ~~not defined~~ Error



Operators :

→ In Python, an operator is a special symbol that represents a computation or operation on variables or values. Python supports a variety of operators, which can be classified into different types based on their functionality

6/50

x = 2

__/__/____

Some of the main types of operators

1. Arithmetic operators :
2. Comparison operators :
3. Logical operators :
4. Assignment operators :
5. Identity Operators :
6. Membership operators :
7. Bitwise operators :

and operator

x = True
y = False
result = x and y . # result is false

or

x = True
y = False
result = x or y . # result is True

not operator
x = True
result = not x . result is false

x = 2
pi = 3.14
area of circle = pi * x * r

Base = 1
Height = 2

area of triangle = $\frac{1}{2}$ * Base * Height

a, b = 2, 4
c = a*b
area of rectangle = c: print (c)

How to take input to user

a = int (input ("Enter the first no: "))
b = int (input ("Enter the 2nd no."))
print ("The sum of 2no is :" a+b)

~~Write to~~

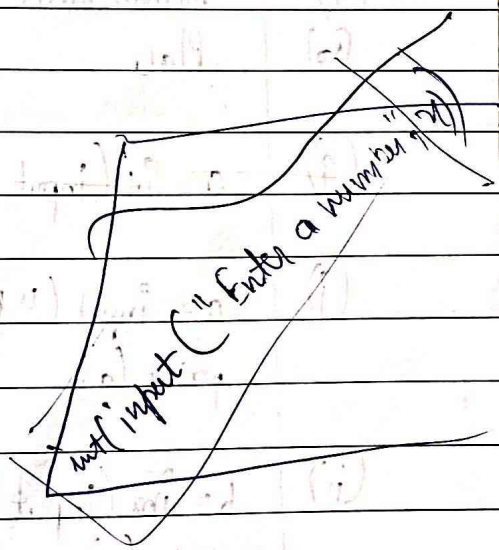
a = int (input ("Enter the mobile no."))
print print ("Enter the mobile no.")

② eval (use for both character and integer)

a = eval (input ("Enter the msg:"))
print (a)

③ String (use for both character and integer)

a = str (input ("Enter the msg:"))
print (a)



__/__/____

④ Input (They use for symbol or non)

```
a = input ("Enter the msg: ")
print (a)
```

Output

Enter the msg: % mohit

mohit@123

Ques ~~user~~ to write a program to print user information

(1) Candidate name
(2) " Roll no
(3) " Age
(4) " ~~Form~~ Course
(5) " Clg name
(6) Father name
(7) Mother name
(8) PAN Card
(9) Adhar Card
(10) Place

(i) ~~a = int(input~~

(i) a = input ("Enter the Candidate name!")
print (a)

(ii) b = int (input ("Enter the Roll No: ")
print (b)

(iii) c = int (input ("Enter the age")
print (c)

(iv) d = input ("Enter the Courses name:")
print(d)

(v) e = input ("Enter the Father name:")
print(e)

(vi) f = input ("Enter the Mother name:")
print(f)

(vii) g = input ("Enter the PAN Cards:")
print(g)

(viii) h = int (input ("Enter the Adhar Card"))
print(h)

(ix) i = input ("Enter the Place")
print(i)

Qu Write a Program to Calculate some mathematical formulas

① Area of Square
② " " Cuboid
③ " " Rectangle
④ " " Circle
⑤ " " Cone
⑥ " " Cyclinder
⑦ " " Triangle
⑧ " " Hemisphere
⑨ " " Sphere
⑩ " " Trapezium
⑪ Semicircle

__/__/____

Area of trapezium

```
a=5
b=1
h=2
print ((a+b)/2*h)
```

```
c=a+b
D=c/2
print(D*h)
```

① Area of Square

```
S=2
T=S*S
print(T)
```

```
S=int(input("enter the no:"))
print("The n
```

② Area of Cuboid

$2(lb+bh+lh)$

```
l=1
b=2
h=
```

Python statement:

Flow Control

- Conditional Statement
  1. if
  2. if-else
  3. if-elif-else
  4. nested else
- Transfer Statement
  1) break
  2) Continue
  3) pass
  4) exit
- Iterative Statements
  1) for
  2) while

Conditional Statements:

Conditional statements are programming constructs allow you to control the flow of a program based on certain conditions. They enable you to make decisions in your code, executing different blocks of code depending on whether a specified condition is true or false.

Here are the main types of conditional statements in Python Programming:

1. Simple IF (if statement):
2. if - else statement:
3. Ladder IF [if - elif - else statement]:
4. Nested IF Statement:

1 Simple if statement

→ The if statement is used to execute a block of code only if a specified condition is true.

An "if statement" is written by using the if keyboard provide by the python

Ex -

```
x = 10
if x > 5:
print ("x is greater than 5")
```

```
a = 33
b = 200
if b > a:
print ("b is Greater than a")
```

2.) And Operator

```
a = 200
b = 33
c = 500
if a > b and c > a:
print ("Both Conditions are true")
```

```
a = 200
b = 33
c = 500
if (a > b) & (c > a):
print ("Both Conditions are true")
```

Or

```
a = 200
b = 33
c = 500
if (a > b) or (c > a):
print ("Both Conditions are True")
```

Not

```
a = 33
b = 200
if not a > b:
```