

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

1. what is python and why it's is popular Ans-

python is high level,interpreted programming language known for its simplicity, versatility and readability
It's popular because of large framework support and library like matplotlib,numpy,pandas,seaborn 2)
community support 3) scalable -used in various domains like data science,web development ,games,etc 4)
cross platform -it works on various platforms like linux,macos, windows

2. what is interpreter in python

Ans: interpreter converts human readable code into machine code to display the results Interpreter
convert code line by line The error are generated in interpreter line by line And not generated once like
compiler

Example: print ("hello world")

It's will convert the code into Machine readable and display output as :

hello world

Double-click (or enter) to edit

3)what are pre-defined keywords in python

Ans: keywords are the reserved word in python that has their own specific meaning

Keywords cannot be used as identifier or variable

To generate keywords in python we use the following syntax `help(keywords)`

Or `Import keyword print (keyword.kwlist)`

4. Can keyword be used as variable name? No keywords can't be used as variable name because keywords are reserved word which have their own specific meaning

Example: Break - to break the block or loop

5. what is mutability in python ?

Mutability means can be change after creation(adding, removing, modifying) Like list

6. why are list mutable and tuples immutable Ans:- Lists are mutable because their elements can be changed, added, or removed after creation. Tuples are immutable because their elements cannot be changed once created — they're fixed to ensure data safety and consistency.

7.what is the difference between == and is operator in python

== Operator is comparison Operator used to check the equality of the value stored in variable

For example

```
a=5  
b=5  
print (a==b)
```

Output:

True

8) Is operator is identity operator used to check the variable memory location is same or not

```
A=(1,2,3)  
B=(1,2,3)  
print (A is B) #false
```

Value store in A and B is the same but they don't have same memory location

Double-click (or enter) to edit

9. what are logical operator in python

Ans: logical operator are operator use for checking the condition

Three type of operator And Or Not

1) And operator is true when the both the condition is true

2) or operator is true when anyone of the condition is true

3. Not operator - reverse of the condition

10) what is typecasting in python

Ans: typecasting means conversion of variable's datatype from one datatype to another datatype

For example A="5" print(type(A)) #output will be string

c=int(A) print (type(c)) #output will be int

11.

What is the difference between implicit and explicit typecasting? Ans: typecasting means changing variable data type from one datatype to another

Implicit typecasting is done by the python interpreter Example: a=2 c=a+0.5 print (c)

Explicit typecasting is done by the programmer to avoid data loss

Ex: a="2" c=int(a) print (type(c))

Double-click (or enter) to edit

- 11)1. Execute certain code only when a specific condition is true.
2. Control the flow of a program.
 3. Make decisions based on different inputs or situations.
 4. Perform different actions for different conditions.

How does elif statement work ?

Ans:elif allows you to check multiple conditions, one by one, without writing many separate if statements.

what is the difference between for loop and while loop in python ans:for loop is used for when number of iterations are known to user...and used for iterable while loop is used for when number of iterations are unknown

14. Describe a scenario where a while loop is more suitable than a for loop?

Ans: a for loop is used for when number of iterations are known whereas a while loop is used for when number of iterations are unknown While loop example

asking a user to enter a password until the password matches the given password

here user has to enter a password until it matches the password..here number of iterations are unknown

```
#write a python program to print hello world
```

```
print ("hello world!")
```

```
hello world!
```

```
#write a python program that displays your name and age
```

```
print ("My name is Kunal ")
```

```
print ("My age is 20.")
```

```
My name is Kunal
```

```
My age is 20.
```

```
#3Write a code to print all predefined keywords in python using the keyword library
```

```
import keyword
```

```
print(keyword.kwlist())
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
/tmp/ipython-input-2144603349.py in <cell line: 0>()
      1 #3Write a code to print all predefined keywords in python using the keyboard
library
      2
----> 3 import keyboard
      4 print(keyword.kwlist())
```

ModuleNotFoundError: No module named 'keyboard'

NOTE: If your import is failing due to a missing package, you can manually install dependencies using either `!pip` or `!apt`.

To view examples of installing some common dependencies, click the "Open Examples" button below.

OPEN EXAMPLES

Next steps: [Explain error](#)

```
import keyword

word = input("Enter a word: ")
if keyword.iskeyword(word):
    print(word, "is a Python keyword")
else:
    print(word, "is NOT a Python keyword")
```

```
-----
KeyboardInterrupt                                Traceback (most recent call last)
/tmp/ipython-input-1276089853.py in <cell line: 0>()
      1 import keyword
      2
----> 3 word = input("Enter a word: ")
      4 if keyword.iskeyword(word):
      5     print(word, "is a Python keyword")

-----
1 frames -----
/usr/local/lib/python3.12/dist-packages/ipykernel/kernelbase.py in _input_request(self,
prompt, ident, parent, password)
    1217         except KeyboardInterrupt:
    1218             # re-raise KeyboardInterrupt, to truncate traceback
-> 1219             raise KeyboardInterrupt("Interrupted by user") from None
    1220         except Exception:
    1221             self.log.warning("Invalid Message:", exc_info=True)
```

KeyboardInterrupt: Interrupted by user

```
#demonstrate list and tuple mutability

my_list = [10, 20, 30]
my_tuple = (10, 20, 30)

# Changing list value (Allowed)
my_list[1] = 50
print("Updated List:", my_list)
```

```
# Changing tuple value (Not Allowed → Error)h
try:
    my_tuple[1] = 50
except TypeError:
    print("Tuples are immutable, can't change values.")
```

```
#arithmetic operation
a=int(input("Enter a number "))
b=int(input("Enter a number "))
print ("addition is",a+b)
print ("substraction is ",a-b)
print ("multiplication and is ",a*b)
print ("division is ",a/b)
print ("exponential is ",a**b)
print ("floor division",a//b)
print ("modulus is ",a%b)
```

```
Enter a number 1
Enter a number 1
addition is 2
substraction is 0
multiplication and division is 1 1.0
```

```
#convert string into int flota and boolean
value = input("Enter a number: ")

int_value = int(value)
float_value = float(value)
bool_value = bool(value)

print(int_value, type(int_value))
print(float_value, type(float_value))
print(bool_value, type(bool_value))
```

```
#logical operator

x = True
y = False

print("x and y =", x and y)
print("x or y =", x or y)
print("not x =", not x)
```

```
#printing 1 to 10 number

for i in range(1, 11):
    print(i)
```

```
#factorial of given number
n = int(input("Enter a number: "))
fact = 1
i = 1

while i <= n:
    fact *= i
    i += 1
```

```
print("Factorial =" fact)
```

```
#sum of number from 1to 50 that is even
sum = 0
for i in range(1, 51):
    if i % 2 == 0:
        total += i

    print("Sum of even numbers:", sum)
```

```
#reverse of string

s = input("Enter a string: ")
i = len(s) - 1

rev = ""
while i >= 0:
    rev += s[i]
    i -= 1

print("Reversed String:", rev)
```

```
#positive, negative or zero

n= fint(input("Enter a number: "))

if n > 0:
    print("Positive")
elif n < 0:
    print("Negative")
else:
    print("Zero")
```

```
#typecasting of list
numbers = ["10", "20", "30"]
Type = [int(i) for i in numbers]
print(Type)
```