**KEYWORDS**

**Keywords:** It is predefined words given by the python interpreter it is also known as reserved words.

Very keywords has some special meaning (functionality), This will can understand by python interpreter.

We cannot use a keyword as a variable name, function name or other identifiers.

All keywords are small case letter except True, False and None.

Keywords are case sensitive.

## AND:

AND keyword is a logical operator and it is used to perform logical AND operations the AND keyword is used to check the two conditions if both the statements are true then the condition is true otherwise false

# Example using AND  
a = 50  
b = 40  
c = 10  
if a > b and b > c:  
 print("Both conditions are true")

Output: Both conditions are true

# OR:

OR Keyword is a logical operator it is used to perform logical OR operations. It evaluates two operations if any one statement is true then the condition is true if two statements are false then the condition is false

a = 50  
b = 40  
c = 10

if c < b or c > b:

print(" one condition is true")

Output: one condition is true

# NOT:

NOT is also a logical operator it is used to perform logical NOT operations. It returns true value if conditions are not true otherwise it will return false

a = 50  
b = 40

c = 10

if not a == c:

print("a is not equal to c")

Output: a is not equal to c

# IF:

IF keyword is used for conditional statement. It executes the code only if condition is true otherwise it blocks the code

fruit = "apple"

if fruit == "apple":

print("It's an apple")

else:

print("It's not an apple.")

Output: It's an apple

# ELSE:

ELSE keyword is a conditional statement, it decides what to do if condition is false and it is also used in try, except blocks

age = 14

if age >= 18:

print("you are eligible to use app")

else:

print("you are not eligible to use app")

Output: you are not eligible to use app

# ELIF:

ELIF is also a if-else statement here we have two conditions if condition is true then if statement will be execute otherwise else statement will be executed

temperature = 22

if temperature >= 30:

print("it's sunny day")

elif temperature <= 20:

print("It's normal day")

Output: It’s normal day

# WHILE:

In python while is used to create while loop it is a control flow statement it allows code number of times until the condition is true

count = 50

while count <= 50:

print(f" Count is: {count}")

count += 1

print("Loop finished")

Output: Count is: 50

Loop finished

# FOR:

For loop is a control flow statement and it is iterative sequences like list, tuple, It repeatedly execute the group of statements until the condition is true

numbers = [1, 2, 3, 4, 5]

for number in numbers:

print(f" Current number: {number}")

Output:

Current number: 1

Current number: 2

Current number: 3

Current number: 4

Current number: 5

# IN:

The IN keyword is used to check that the value is in sequence or not like lists, ranges

fruits = ['apple', 'grapes', 'banana', 'mango']

if 'kiwi' in fruits:

print("kiwi is present")

else:

print("not present")

Output: not present

TRY:

It is used for exception handling it allows us to test a block of code for errors and try is used to detect and monitor exception in program

a = 20  
b = 0

try:

c = a / b

print("div : ",c)

except Exception as e:

print("e")

print("Div by 0 NOT-OK in python")

Output: e

Div by 0 NOT-OK in python

# 11. EXCEPT:

The exception can be handled by using the try statement the try block raises error and the except block will be executed

a = 20  
b = 0  
try:

c = a / b

print("div : ",c)

except Exception as e:

print("e")

print("Div by 0 NOT-OK in python")

Output:

E

Div by 0 NOT-OK in python

# FINALLY:

The finally keyword is used in try, except blocks. It defines a block of code to run when the try, except, else block is final. The finally block will be executed no matter if the try block raises an error or not.

marks = 120  
marks = 90

try:

if(marks>100):

raise Exception("marks are greater than 100")

print("given marks:",marks)

except Exception as e:

print(e)

finally:

print("Finally block is executed")

Output:

given marks: 90

Finally block is executed

# 13. DEF:

It is used to define a function it is reusable block of code, functions takes input and perform actions and they return outputs

def sum (a, b):

return a + b

print(sum(1,6))

Output: 7

RETURN:

The **RETURN** keyword is used to exit a function and send a value back to the function’s call. To provide multiple outputs from a single function.

**Syntax: return [expression]**

def square(num):  
 return num \* num  
result = square(6)  
print(result)

Output: 36

IMPORT:

**Import** keywords is used to include modules in your python program.

Import pandas as pd and it is used numpy as np.

Import math as m

**Syntax: import module\_name as alias\_name**

import math as m  
print(m.sqrt(4))

Output: 2

CLASS:

The keyword class is used to define a class**.** It will give the blue print of an object.

Classes are fundamental to object-oriented programming (OOP) in Python.

**Syntax:**

**class Class name:**

**def \_init\_(self, parameters):**

class Car:  
 def start(self):  
 print("The car has start")  
my\_car = Car()  
my\_car.start()

Output: The car has start

FROM:

It is used to import the module or library. And It allows you to import only the functions, classes, or variables.

**Syntax:**

From module\_name import specific\_func

We can also import multiple items:

From modul\_name import func1, func2

Ex: from math import sqrt

from math import sqrt  
number = 121  
result = sqrt(number)  
print(f"The square root of {number} is {result}")

Output:

The square root of 121 is 11.0

AS:

It is used to create an alias for a module, function, or object. The **as** will give a module name as shorter name(alias) to make code easier to write.

* Aliasing Modules or Functions in Imports
* Handling Exceptions in try… except
* Context Management with the with Statements

**Syntax: import module\_name as alias\_name**

import numpy as np  
  
array = np.array([1, 2, 3])  
print(array)

Output: [1,2,3]

TRUE:

If the statement is true then it will print true in a Boolean.

x = 25  
y = 20  
if x>y and y<x:  
 print('true')  
else:  
 print('flase')

Output: true

FALSE

If the statement is wrong then it will print **False** is a Boolean.

x = 25  
y = 20  
if x<y and y>x:  
 print('true')  
else:  
 print('flase')

Output: flase

NONE:

The keyword **NONE** represents the absence of a value or a null value.

**Syntax: None**

x = None  
print(x)

Output: none

IS:

It is used to check if two objects refer to the same memory location.

**Is:** It checks whether two variables point to the same object.

**Syntax: object1 is object2**

a = [1, 2, 3]  
b = a  
  
print(a is b)

Output: True

a = [1, 2, 3]  
b = a  
c = [1, 2, 3]  
  
print(a is b)  
print(a is c)

Output:

True

False

LAMBDA:

It is used to create anonymous (nameless) functions. These functions are also called lambda functions.

**Syntax: lambda arguments: expression**

add = lambda x, y: x + y  
result = add(27, 31)  
print(result)

Output: 58

WITH:

It is used to simplify the management of resources, such as file handling.

This ensures that the file is properly closed after it is used, even if an error occurs during the read or write operation.

**Syntax: with expression [ as variable];**

with open("example.txt", "w") as file:  
 file.write("Hello, world!")

GLOBAL:

It allows us to modify the variable outside of the current scope.

It is used to create a global variable and make changes to the variable in a local context.

**Syntax: global variable\_name**

x = 45  
def modify\_global():  
 global x  
 x = 77  
print("Before modification:", x)  
modify\_global()  
print("After modification:", x)

Output:

**NON-LOCAL:**

It is used to access and modify a variable in an outer function from an inner function. It is used to work with nested functions.

**Syntax: nonlocal variable\_name**

def outer():  
 x = 30  
 def inner():  
 x = 25  
 print("Inner x:", x)  
 inner()  
 print("Outer x:", x)  
outer()

Output:

Inner x: 25

Outer x: 30