25-09-2024

# Training Day- 8

**CONDITIONAL STATEMENTS:** To Execute Set Of Statements On The Basis Of Some Conditions. Conditional Statements In Python Allow You To Control The Flow Of Your Program Based On Conditions. They Are Used To Execute Specific Code Blocks When Certain Conditions Are Met. Python Supports The Following Conditional Statements

### . `if` Statement

- Executes a block of code if the condition evaluates to `True`.
- Example:

```
python
Copy code
if x > 0:
    print("x is positive")
```

### 2. `if-else` Statement

- Provides an alternative block of code to execute when the condition evaluates to `False`.
- Example:

```
python
Copy code
if x > 0:
    print("x is positive")
else:
    print("x is non-positive")
```

### 3. `if-elif-else` Statement

- Allows checking multiple conditions in sequence.
- The first condition that evaluates to `True` is executed, and the rest are skipped.
- Example:

```
python
Copy code
if x > 0:
    print("x is positive")
elif x == 0:
    print("x is zero")
else:
    print("x is negative")
```

### 4. Nested `if` Statements

- Places one `if` statement inside another to handle more complex scenarios.

- Example:

```
python
Copy code
if x > 0:
    if x % 2 == 0:
        print("x is a positive even number")
```

**Conditional Statements:** ***3 Keywords:*** These keywords plays the role of a heading: Heading in python is a statement which is having some sub-statements inside it or a block of code inside it.

## Syntax To Create A Heading In Python:

HEADING_NAME:

All the statements inside a heading will be indented statements.

**Indentation:** Statements at a fixed gap (spacebars) w.r.to heading

If condition will be true then only statements will be executed.

if is a condition

elif is a condition

else executes if conditions above else are false

**"""Statements inside a heading ie the block of code inside heading in python is called a suite.**

**There should be al least one statement inside a heading.**

"""

```
# #New Program
# x=5
# if(x<=5):
#     print("CETPA")
#     print("Welcome")
# print("ABCD")
```

**Single Liner Headings:** Then we can mention the statement directly next to heading in same line

**……New Program……..**

```
# x=5
# if(x==5):print("CETPA")
```

```
# print("Welcome")
```
**……New Program……..**
```
# x=5
# if(x==5):pass
# print("Welcome")
```
**……New Program……..**
```
# x=5
# if(x==5):
#   pass
#   print("CETPA")
# print("Welcome")
```
**……New Program……..**
```
# x=5
# if(x==5):
#   pass
# print("CETPA")
# print("Welcome")
```
**……New Program……..**
```
#   print("CETPA")    #IndentationError: unexpected indent
#   print("Hello")
```
**……New Program……..**
```
# id=int(input("Enter the ID:"))
# if(id==1000):
#   print("Welcome to the System")
# else:
#   print("You are not allowed the entry")
```
**……New Program……..**
```
# id=input("Enter the ID:")
# if(id=="1000"):
#   print("Welcome to the System")
# else:
#   print("You are not allowed the entry")
```

**MULTIPLE CONDITIONS:** if, elif, else

elif and else can't work without if

elif: else if

"""

```
# day=input("Enter the day:")
# if(day=="Sunday"):
#   print("Take Rest")
# elif(day=="Saturday"):
#   print("Go to Movie")
# elif(day=="Friday"):
#   print("Go for Shopping")
# else:
#   print("Go for CETPA Class")
```

**If we want to check multiple conditions in single heading:**

**we can use logical operators 'and'  and 'or'**

**……New Program……..**

Better way

```
# day=input("Enter the day:")
# if(day=="Sunday"):
#   print("Take Rest")
# elif(day=="Saturday"):
#   print("Go to Movie")
# elif(day=="Friday"):
#   print("Go for Shopping")
# elif(day=="Monday" or day=="Tuesday" or day=="Wednesday" or day=="Thursday"):
#   print("Go for CETPA Class")
# else:
#   print("Incorrect input")
```

"""

**NESTED CONDITIONS:** Conditions inside conditions:

heading inside heading.

If there is a heading inside heading, then statements of the inner

heading will be indented with respect to inner heading

"""

**……New Program……..**

```
# a=5
# b=9
# if(a==5):
#   print("CETPA")
#   print("ABCD")
#   if(b==7):
#     print("Welcome")
#     print("PQRS")
#   print("UVWR")
# else:
#   if(a==6):
#     print("1234")
```

**……New Program……..**

```
# a=5
# b=7
# if(a==5):
#   print("CETPA")
#   print("ABCD")
#   if(b==7):
#     print("Welcome")
#     print("PQRS")
#   print("UVWR")
# else:
#   if(a==6):
#     print("1234")
```

**……New Program……..**

```
# a=6
# b=7
# if(a==5):
#   print("CETPA")
```

if, elif we get True inside condition then block will execute else won't

"""

**……New Program……..**

```
# a,b=5,7
# print(a==b)
# print(a<b)
# print(a!=b)
```

**……New Program……..**

```
# a,b=5,5
# if(a==b):
#   print("CETPA")
```

**False Values in Python:** In Python, a **false value** refers to any value that evaluates to `False` in a boolean context, such as when used in conditional statements. These are called **falsy values**. Here's a complete explanation:

## Falsy Values in Python

The following are the values that are treated as `False`:

0

False

None

All empty values

Rest all are True values

**……New Program……..**

```
# if(55):
#   print("CETPA")
```

**……New Program……..**

```
# if(""):
#   print("CETPA")
```