Subject Artificial Intelligence

This notebook will teach you about if elif else, Nested and while loop in the Python Programming Language. By the end of this lab, you'll know the basic concepts about conditional statements and how to use these functions.

if, elif, else Conditions

Python uses the if keyword to implement decision control. Python's syntax for executing a block conditionally is as below:

Syntax:
if [boolean expression]:
statement1
statement2
...
statementN

Any Boolean expression evaluating to True or False appears after the if keyword. Use the : symbol and press Enter after the expression to start a block with an increased indent. One or more statements written with the same level of indent will be executed if the Boolean expression evaluates to True.

To end the block, decrease the indentation. Subsequent statements after the block will be executed out of the if condition.

```
x = 12
if x >10:
    print("Hello")
```

output

Hello

```
price = 50

if price < 100:
    print("price is less than 100")
```

output

price is less than 100

Multiple Statements in the if Block

```
price = 50
quantity = 5
if price*quantity < 500:
    print("price*quantity is less than 500")
    print("price = ", price)
    print("quantity = ", quantity)
```

Output
price*quantity is less than 500
price = 50
quantity = 5

Out of Block Statements

```
price = 50
quantity = 5
if price*quantity < 100:
    print("price is less than 500")
    print("price = ", price)
    print("quantity = ", quantity)
print("No if block executed.")
```

Output
No if block executed.

Multiple if Conditions

```
price = 100

if price > 100:
 print("price is greater than 100")

if price == 100:
  print("price is 100")

if price < 100:
    print("price is less than 100")
```

Output
price is 100

else Condition:

Along with the if statement, the else condition can be optionally used to define an alternate block of statements to be executed if the boolean expression in the if condition evaluates to False.

Syntax:

```
if [boolean expression]:
   statement1
   statement2
   ...
   statementN
else:
   statement1
   statement2
   ...
   statementN
```

As mentioned before, the indented block starts after the : symbol, after the boolean expression. It will get executed when the condition is True. We have another block that should be executed when the if condition is False. First, complete the if block by a backspace and write else, put add the : symbol in front of the new block to begin it, and add the required statements in the block.

else Condition

```
price = 50

if price >= 100:
   print("price is greater than 100")
else:
   print("price is less than 100")
```

Output
price is less than 100

elif Condition:

Use the elif condition is used to include multiple conditional expressions after the if condition or between the if and else conditions.

Syntax:

```
if [boolean expression]:
   [statements]
elif [boolean expresion]:
   [statements]
elif [boolean expresion]:
   [statements]
else:
   [statements]
```

The elif block is executed if the specified condition evaluates to True.

if-elif Conditions

```
price = 100

if price > 100:
   print("price is greater than 100")
elif price == 100:
   print("price is 100")
elif price < 100:
   print("price is less than 100")
```

Output
price is 100

if-elif-else Conditions

```
price = 50

if price > 100:
   print("price is greater than 100")
elif price == 100:
   print("price is 100")
else price < 100:
   print("price is less than 100")
```

Output
price is less than 100

Invalid Indentation

```
price = 50

if price > 100:
   print("price is greater than 100")
 elif price == 100:
   print("price is 100")
  else price < 100:
   print("price is less than 100")
```

Output

```
  elif price == 100:
                ^
```

IdentationError: unindent does not match any outer indentation level

Nested if, elif, else Conditions:

Python supports nested if, elif, and else condition. The inner condition must be with increased indentation than the outer condition, and all the statements under the one block should be with the same indentation.

```
price = 50

quantity = 5
amount = price*quantity

if amount > 100:
   if amount > 500:
      print("Amount is greater than 500")
   else:
      if amount < 500 and amount > 400:
         print("Amount is")
      elif amount < 500 and amount > 300:
         print("Amount is between 300 and 500")
      else:
         print("Amount is between 200 and 500")
elif amount == 100:
   print("Amount is 100")
else:
   print("Amount is less than 100")
```

Output
Amount is between 200 and 500

While Loop:

Python uses the while and for keywords to constitute a conditional loop, by which repeated execution of a block of statements is done until the specified boolean expression is true.

The following is the while loop syntax.

Syntax:

```
while [boolean expression]:
   statement1
   statement2
   ...
   statement
```

Python keyword while has a conditional expression followed by the : symbol to start a block with an increased indent. This block has statements to be executed repeatedly. Such a block is usually referred to as the body of the loop. The body will keep executing till the condition evaluates to True. If and when it turns out to be False, the program will exit the loop. The following example demonstrates a while loop.

while loop

```
num =0

while num < 5:
   num = num + 1
   print('num = ', num)
```

Output
num = 1
num = 2
num = 3
num = 4
num = 5

```
i = 1
while i < 3:
   print(i ** 2)
   i = i + 1
print('Bye')
```

Output
1
4
Bye

Invalid Indentation

```
num =0
while num < 5:
   num = num + 1
     print('num = ', num)
```

Output

```
  print('num = ', num)
  ^
IndentationError: unexpected indent
```

Reference of website
https://www.tutorialsteacher.com/

Questions

- Print "Hello World" if `a` is greater than `b`?
- print "Yes" if `a` is equal to `b`, otherwise print "No"?
- Print `i` as long as `i` is less than 6?