Programming with Python Jan 24, 2025

Brief Summary of contents discussed.

if-else statement

```
if  <condition 1>:
    <statements>
elif <condition 2>:
    <statements>
elif <condition 3>:
    <statements>
…
…
else:
    <statements>
```

Exercise:

- Write a program to identify whether a number is odd or even.

- **Leap Year**

Build conditional trees as discussed in class:

a. Starting with ‘division by 4’ at root.
b. Starting with ‘division by 100’ at root.
c. Starting with ‘division by 400’ as root.

1. Write a program to find leap years based on all three conditional trees built above using `if-elif-else` ladder.
   Try counting the number of comparisons for each of the trees built above.

2. Write a program to find leap year with a single if condition:

   - Divisible by 4 and (not divisible by 300 or divisible by 400)

   Analyze the above conditional expression to see if there is a need to apply parenthesis to the above expression to overall default operator precedence.

One line expression for if-else

<expression 1> if <condition> else <expression 2>

Example:
num = eval(input("Enter any integer: "))
msg = 'Number is Even' if num % 2 == 0 else 'Number is Odd'
print(msg)

For loop
Loop: Repetitive execution of a statement or sequence of statements. **Iteration** is the execution of a such sequence of statements in the loop.

General form or `for`

```
for variable in sequence:
    Statements
```

A very useful built-in function used during `for` loop is **`range`**.
range: generates a sequence of integers

```
range(n)
range(start, end)
range(start, end, increment)
```

range(5) : 0, 1, 2, 3, 4
range(2, 7) : 2, 3, 4, 5, 6
range(2, 17, 3) : 2, 5, 8, 11, 14
range(6, 2, -1) : 6, 5, 4, 3

NOTE:
- Explore python.org website.
- Hopefully, by now everyone is using some IDE (Integrated Development Environment). It improves productivity and is very useful in the long run.