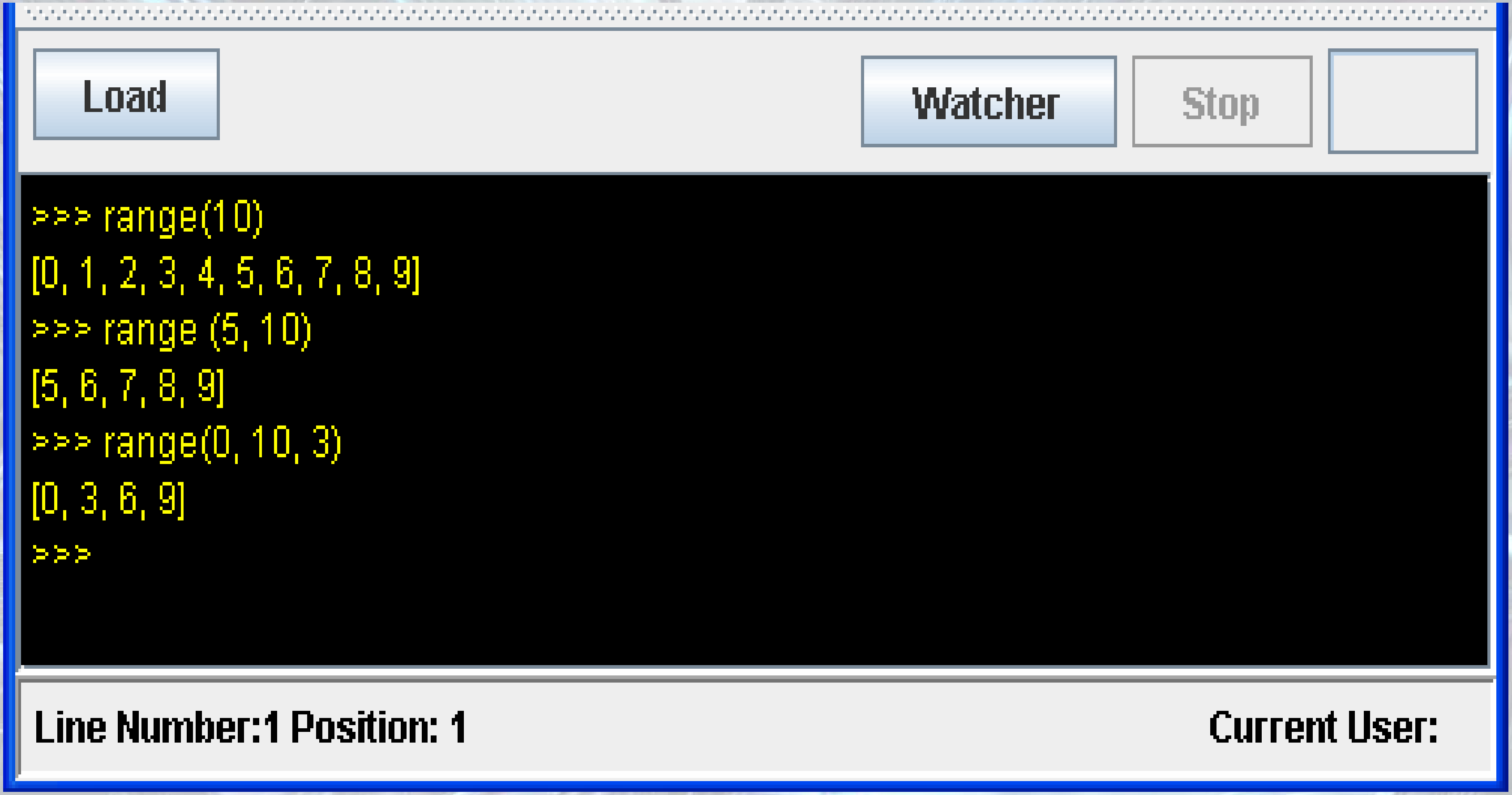
The Range Function, Conditionals, and Loops

The Range Function: Numbers

- **range(x) – gives a list of the numbers from 0 to x-1: [0, …, x-1]**
- **range(x, y) – gives a list of the numbers from x to y-1: [x, …, y-1]**
- **range(x, y, z) – gives a list of the numbers that are multiples of z from x to y-1**

The Range Function: Numbers - Examples



```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> range (5, 10)
[5, 6, 7, 8, 9]
>>> range(0, 10, 3)
[0, 3, 6, 9]
>>>
```

Line Number:1 Position: 1 Current User:

The Range Function: Strings

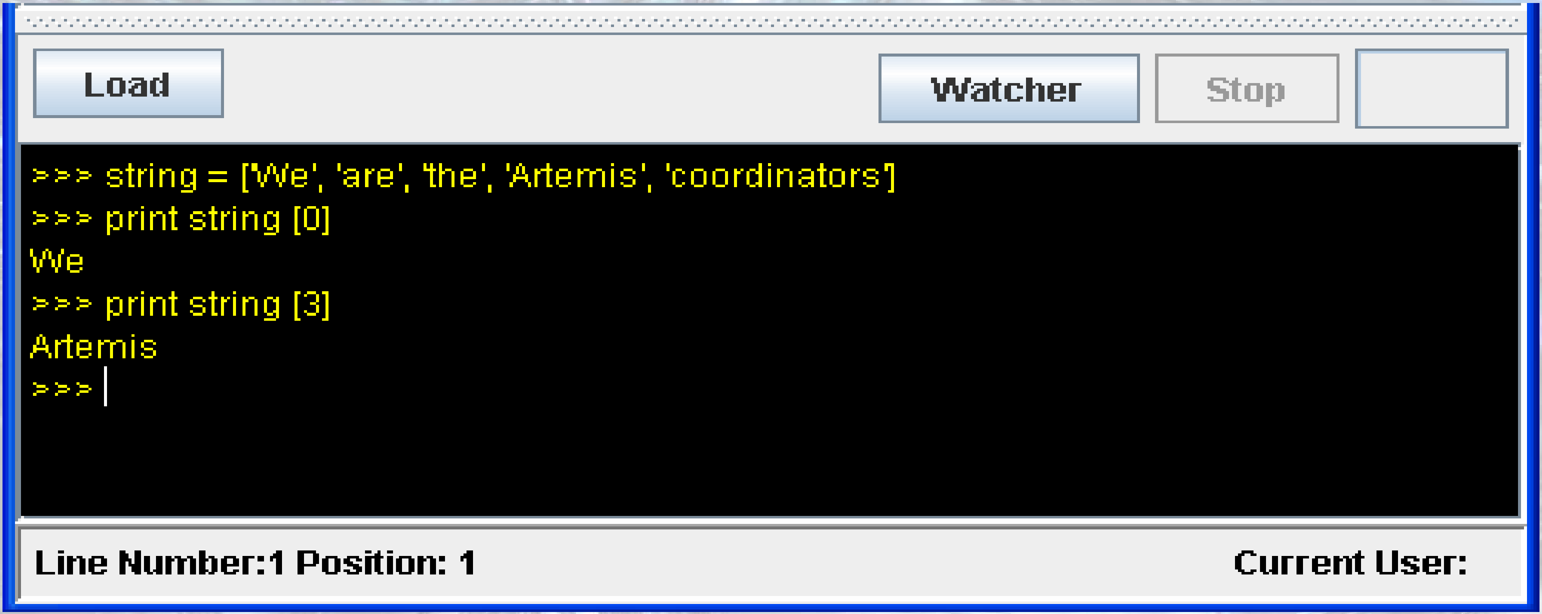
- You can store strings as a range as well, and use their indices to access them:

  string = [‘some, text’]

  string[0] = some

  string[1] = text

  Example:



```
>>> string = ['We', 'are', 'the', 'Artemis', 'coordinators']
>>> print string [0]
We
>>> print string [3]
Artemis
>>> 
```

Line Number:1 Position: 1

Current User:

Conditionals: The "If" Statement

- If you want a certain block of code to only be executed if a certain condition is true, you can use an "if" statement!
- An example:

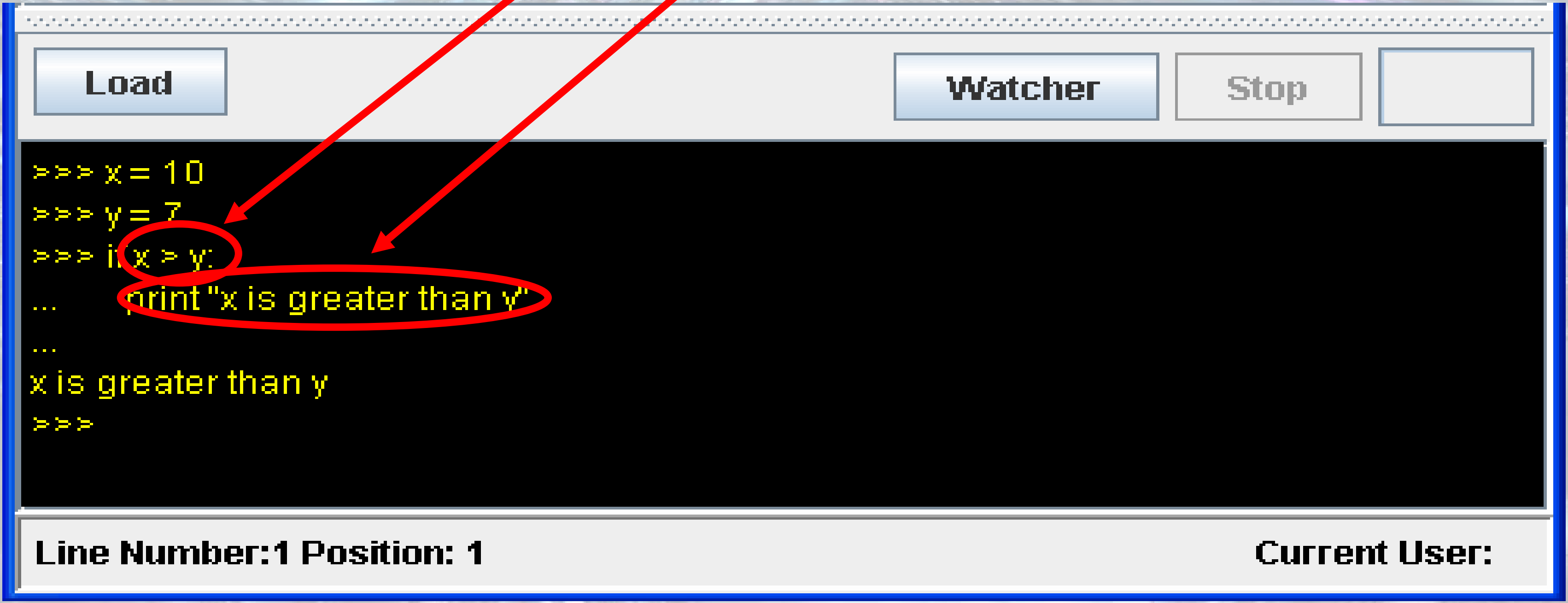
  if variable x is greater than variable y

  then print that variable x is greater

In code:

Condition: note that it is followed by a ":"

Consequence: note that it is in a "block". Press the tab button for each consequence for indentation.



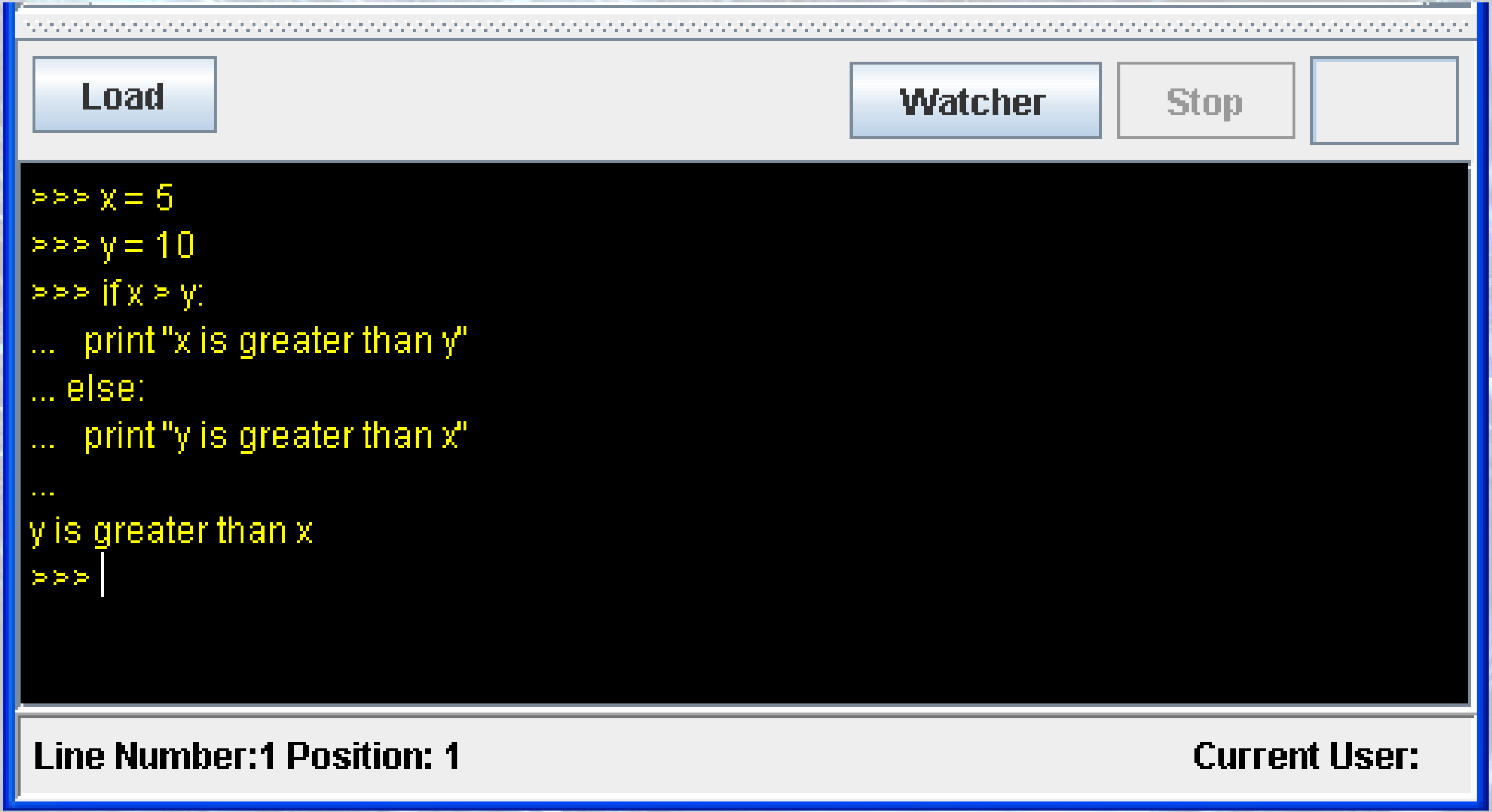
```
>>> x = 10
>>> y = 7
>>> if x > y:
...      print "x is greater than y"
...
x is greater than y
>>>
```

Line Number:1 Position: 1 Current User:

Conditionals: The "If…Else" Statement

- If a certain condition produces a certain result, what happens when that condition is not true?
- You can use an "if…else" statement!

Note: the indentation – one tab for each consequence



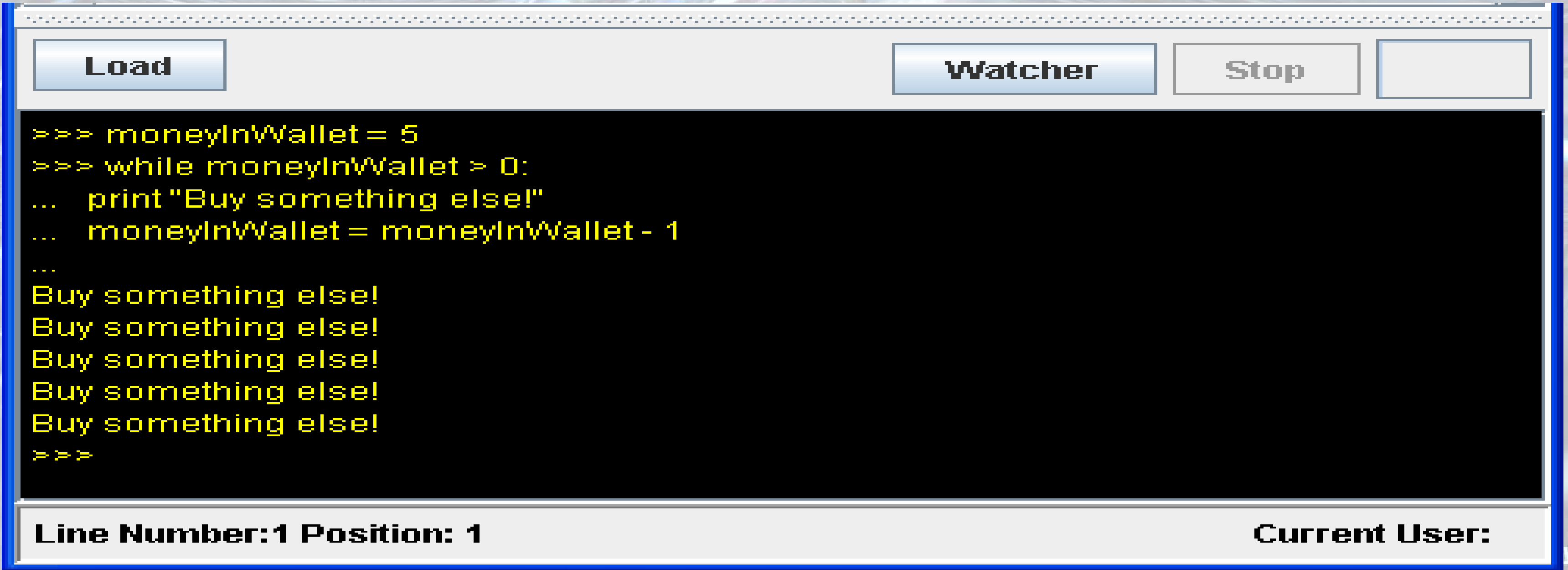
```
>>> x = 5
>>> y = 10
>>> if x > y:
...    print "x is greater than y"
... else:
...    print "y is greater than x"
...
y is greater than x
>>>
```

Line Number:1 Position: 1   Current User:

Loops: The "While" Loop

- If you want some code to execute only while a certain condition is true (and the condition is changing), use a "while" loop!
- Example: Let's say you're shopping at a dollar store and you have 5 dollars in your wallet and you want to spend it all:

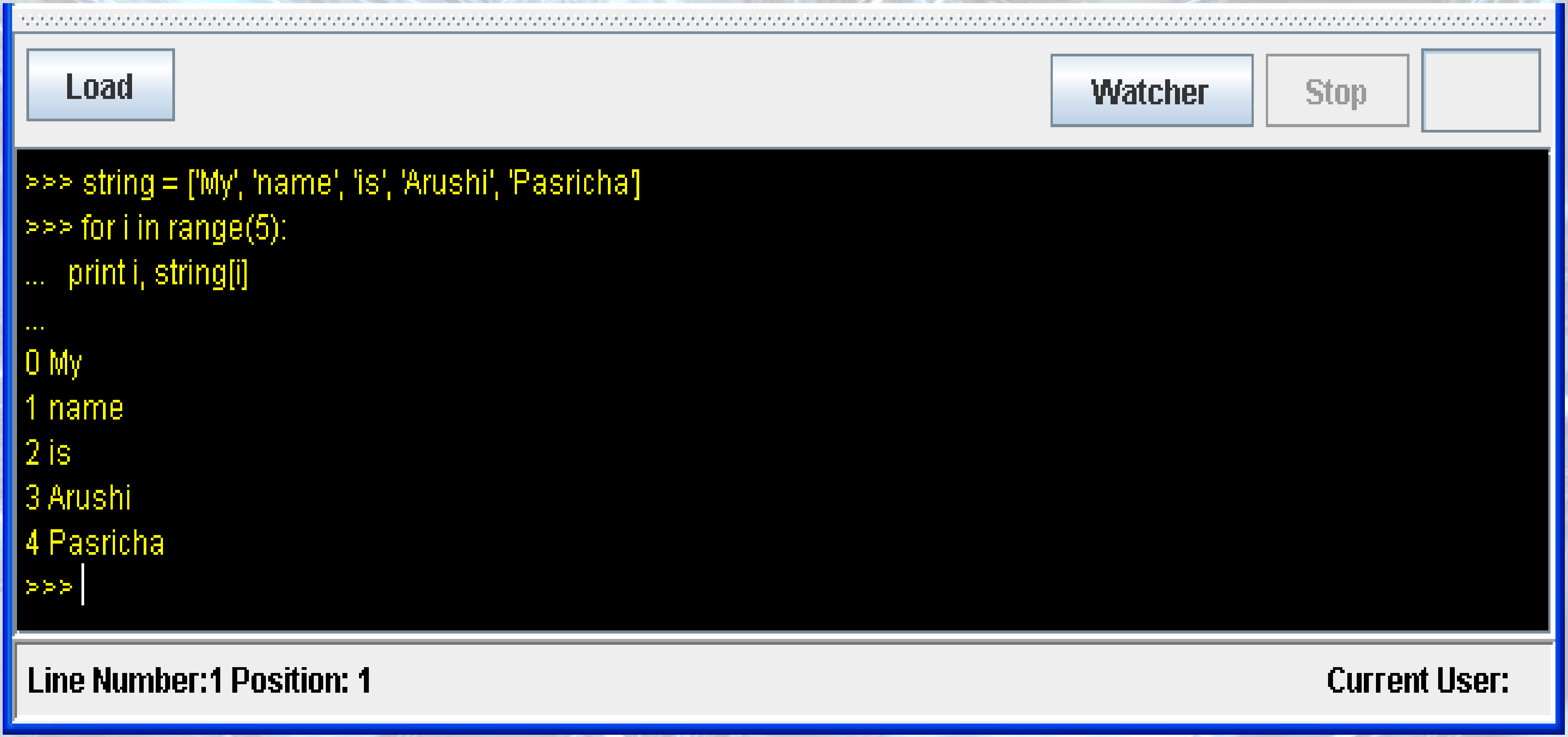
```
while I have money in my wallet
        print that I must buy something else
        decrease the amount of money in my wallet
```



```
>>> moneyInWallet = 5
>>> while moneyInWallet > 0:
...   print "Buy something else!"
...   moneyInWallet = moneyInWallet - 1
...
Buy something else!
Buy something else!
Buy something else!
Buy something else!
Buy something else!
>>>
```

Loops: The "For" Loop

- Another type of loop is a "for" loop.
- You can use this kind of loop with a range function:



```
>>> string = ['My', 'name', 'is', 'Arushi', 'Pasricha']
>>> for i in range(5):
...    print i, string[i]
...
0 My
1 name
2 is
3 Arushi
4 Pasricha
>>> 
```

Line Number:1 Position: 1 Current User:

JJMARX