Python Programming Language

Lecture 10: For Loop, While Loop, List, Dictionary and Problem Solving

Hafsa Sultana

Computer Science and Engineering

Khulna University

Today's Learning Objectives

- For Loop
- While Loop
- List
- Dictionary (Dict)
- Practical Exercises

Basic Structure of a For Loop

- **Syntex:** for variable in sequence:

  # Code to execute
- Variable takes on each value in the sequence one at a time.
- The indented code block runs for each value.

- **Example:** Print 1 to 5 numbers.
- **Normal Code :**

```
print(1)
print(2)
print(3)
print(4)
print(5)
```

- **Using Loop:**

```
for num in range(1,6):
  print(num)
```

If condition in Loop

- Write a program that prints multiples of 3 between 1 and 20.

```
for i in range(1, 21):
    if i % 3 == 0:
     print(i)
```

Exercise

- Title: Find Maximum Element in a 2D List
- Input: matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
- Output: Maximum element: 9
- Solution:

```
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
max_element = matrix[0][0]  # Initialize with the first element of the matrix
for row in matrix:
    for element in row:
        if element > max_element:
            max_element = element

print("Maximum element:", max_element)
```

Basic Structure of a While Loop

- **Syntex:** while expression:

  statement(s)

- **Example:** Print 1 to 5 numbers.
- **Normal Code :**

```
print(1)
print(2)
print(3)
print(4)
print(5)
```

- **Using Loop:**

```
count=1
while count<=5:
    count+=1
    print (count)
print ("End of while loop")
```

Data Type: List

-
  ```
  list1 = ["apple", "banana", "cherry"]
  list2 = [1, 5, 7, 9, 3]
  list3 = [True, False, False]
  ```

-
  ```
  list4 = ["abc", 34, True, 40, "male"]
  ```

- List items are ordered, changeable, and allow duplicate values.
- The first item has index [0], and the second item has index [1].
- List Length :

```
thislist = ["apple", "banana", "cherry"]
print(len(thislist))
```

Data Type: List

-
  ```
  thislist =["apple", "banana", "cherry", "orange", "melon", "mango"]
  print(thislist[1])
  ```

```
print(thislist[:4])

print(thislist[-4:-1])
```

- Check if Item Exists:

-
  ```
  if "apple" in thislist:
    print("Yes, 'apple' is in the fruits list")
  ```

List Append

-
  ```
  thislist = ["apple", "banana", "cherry"]
  thislist.append("orange")
  print(thislist)
  ```
- Insert a Item :

-
  ```
  thislist = ["apple", "banana", "cherry"]
  thislist.insert(1, "orange")
  print(thislist)
  ```
- `Remove:`

-
  ```
  thislist.remove("banana")
  print(thislist)
  ```
- `Sort list:`
- `thislist.sort()`

List in Loop

```
thislist = ["apple", "banana", "cherry"]
for x in thislist:
  print(x)
```

- Index wise print :

```
thislist = ["apple", "banana", "cherry"]
for i in range(len(thislist)):
  print(thislist[i])
```

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = []

for x in fruits:
  if "a" in x:
    newlist.append(x)

print(newlist)
```

Data Type: Dictionary

- Dictionaries are used to store data values in key:value pairs.
- Dictionaries are changeable, meaning that we can change, add or remove items.

-
  ```
  thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
  }
  ```

-
  ```
  thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
  }
  print(thisdict["brand"])
  ```

-
  ```
  print(len(thisdict))
  ```

Data Type: Dictionary

- Accessing Items:
-
```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = thisdict["model"]
```
- `Get():`
- `x = thisdict.get("model")`
- `Kyes()`
- `x = thisdict.keys()`
- `Values():`
- `x = thisdict.values()`

Data Type: Dictionary

- Change Items:

-
  ```
  car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
  }

  x = car.values()

  print(x) #before the change

  car["year"] = 2020

  print(x) #after the change
  ```

Data Type: Dictionary

- Add Items:

-
  ```
  thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
  }
  thisdict["color"] = "red"
  print(thisdict)
  ```
- Using update:

-
  ```
  thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
  }
  thisdict.update({"color": "red"})
  ```
- Delete:
- `thisdict.pop("model")`

Data Type: Dictionary Loop

-
  ```
  for x in thisdict:
    print(x)
  ```

-
  ```
  for x in thisdict.values():
    print(x)
  ```

-
  ```
  for x in thisdict.keys():
    print(x)
  ```

Items:

```
for x, y in thisdict.items():
  print(x, y)
```

Practical Example 1

- **Sum of List:**
- Write a Python program to find the sum of all the numbers in a list.
- Solution:

```
numbers = [1, 2, 3, 4, 5]
total = 0
for number in numbers:
    total += number
print("Sum:", total)
```

Practical Example 2

- **Maximum Value in List**
- Write a Python program to find the maximum value in a list.
- Solution:

```
numbers = [1, 2, 3, 4, 5]
max_value = numbers[0]
for number in numbers:
    if number > max_value:
        max_value = number
print("Maximum value:", max_value)
```

Practical Exercise 3

- **Print List in Reverse**
- Write a Python program to print a list in reverse order.

```
numbers = [1, 2, 3, 4, 5]
 for i in range(len(numbers) - 1, -1, -1):
   print(numbers[i], end=" ")
```

Practical Exercise 4

- Title: Count Even Numbers

- Input: numbers = [1, 2, 3, 4, 5]
- Output: Number of even numbers: 2

Practical Exercise 5

- Title: Sum of Values

- data = {'a': 100, 'b': 200, 'c': 300}
- Sum of values: 600

Any Question?

References

- https://www.w3schools.com/python/
- https://www.tutorialspoint.com/python/

EXAM-2 (Solution)

Problem 1

7_Days_in_Week

Write a program that takes an integer input (1-7) and prints the corresponding day of the week. (Star from Sunday)

7_days(1)→ ‘Sunday’

7_days(4) → ‘Wednesday’

7_days(7)→ ‘Saturday’

Solution: 1

7_Days_in_Week

```
day_num = int(input("Enter a number (1-7): "))
if day_num == 1:
    print("Monday")
elif day_num == 2:
    print("Tuesday")
elif day_num == 3:
    print("Wednesday")
elif day_num == 4:
    print("Thursday")
elif day_num == 5:
    print("Friday")
elif day_num == 6:
    print("Saturday")
elif day_num == 7:
    print("Sunday")
else:
    print("Invalid number. Please enter a number between 1 and 7.")
```

Problem 2

Great_number_6

The number 6 is a truly great number. Given two int values, a and b, Print True if one of them is 6. Or if their sum or difference is 6. Note: the function abs(num) computes the absolute value of a number.

Great_number_6(6, 4) → True

Great_number_6(4, 5) → False

Great_number_6(1, 5) → True"

Problem 2

Great_number_6

```
sum_ab=a+b
sub_ab=abs(a-b)
if sum_ab==6 or sub_ab==6 or a==6 or b==6:
  print( True)
else:
  print( False)
```

Problem 3

Diff_21

Given an int n, return the absolute difference between n and 21, except return double the absolute difference if n is over 21.

[Absolute difference = big value – small value, the difference is always positive.]

diff21(19) → 2

diff21(10) → 11

diff21(21) → 0

Problem 3

Diff_21

```
if n<=21:
    difference = 21-n
  else:
    difference =2*(n-21)
  print( difference )
```