PROGRAMMING PYTHON

Lecture # 5 - Loops

Code Camp | Alpha

{"Dream" , "Code" , "Build"}

Loop

Used for iterating over a sequence

Some sequence types …

Strings

Strings are sequence of characters.

Lists

Lists are used to store multiple items in a single variable.

Tuple, Sets

Tuples, and sets are also used to store multiple values in a single variable but there are differences.

Dictionary

Dictionaries are used to store data values in **key:value** pairs.

Two Types

For Loop

While Loop

For Loop

- For is like an iterator method.
- A for loop is used for iterating over a sequence.
- With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.
- **Example:**
    - my_string = “Hello World”
    - for charact in my_string:
    -       print(charact)

- **Example:**
    - cars = [“alto", “civic", "cherry"]
    - for x in cars:
    -    print(x)

- Remember For Loop is mainly used for performing similar operations on a collection of items.
- Practice:
    - Verbs = [“eat”, “work”, “sleep”]
    - Use for loop to print eating, working, sleeping.

While Loop

- With the while loop we can execute a set of statements as long as a condition is true.
- While is good for performing mathematical operations
  - i = 1
  - while i < 6:
  -   print(i)
  -   i += 1
- Here: last line of the code is increamental operation and it is a must.
- Remember to increment i, or else the loop will continue **forever**.
- Also note that it is important to assign an initial value to the variable we are iterating
- **Practice:**
  - Write a program to take any number from as input from the user and print its table in the form 2x1 = 2

Break Statement

- With the With the break statement we can stop the loop even if the while condition is true. It is used in a scenario when we want the loop to stop when a certain scenario arrives within the loop
- Example:

```
i = 1
while i < 6:
  print(i)
  if i == 3:
    break
  i += 1
```

- Break statement also works with for loop
- Example:

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

- **Practice: Try breaking the loop before printing “banana”.**

Continue Statement

- With the continue statement we can stop the current iteration, and continue with the next.
- Unlike the break statement, it does not terminate the loop, it only skips the current iteration and move on to the next iteration.
- We use it to skip certain outcomes during the execution of our loops.
- **Example:**

```
i = 0
while i < 6:
  i += 1
  if i == 3:
    continue
  print(i)
```

- **Example:**

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

Else Statement

- Additionally we can also use Else in for and while loops to tell the program once the looping condition is no longer true
- Example:

```
for x in range(6):
  print(x)
else:
  print("Finally finished!")
```

- Example:

```
i = 1
while i < 6:
  print(i)
  i += 1
else:
  print("i is no longer less than 6")
```

Range Function

- Notice that For Loop is basically iterating over collectables.
- What if we want them to iterate over a range of numbers?
- For this purpose we use Range()
- A general structure of the range function is this:
  - Range(lower, upper)
- Sometimes an additional parameter is also used to tell about the increment
  - Range (lower, upper, increment)
  - By default the third parameter is 1
  - Also note that the range function by default starts with 0
- Example:
  - for x in range(6):
  -   print(x)
- Example:
  - for x in range(2, 6):
  -   print(x)
- Example:
  - for x in range(2, 30, 3):
  -   print(x)

Code Camp | Alpha
{"Dream", "Code", "Build"}

Nesting in Loops

- A nested loop is a loop inside a loop.
- The "inner loop" will be executed one time for each iteration of the "outer loop“
- Example:

```
Days = [“Mon”, “Tues”, “Wed”]
Things = [“Eat”, “Sleep”, “Repeat”]
For day in days:
   print(day+ ” “ + “Routine:”)
   print(“============“)
   for thing in things:
        print(thing)
```

- NOTE: There is another method called Pass:
- for loops cannot be empty, but if you for some reason have a for loop with no content, put in the pass statement to avoid getting an error.

```
for x in [0, 1, 2]:
  pass
```

Code Camp | Alpha

Practice

- Write a Python program to print those numbers which are divisible by 7 and multiples of 5, between 1500 and 2700 (both included).
- Write a Python program to construct the following pattern, using a nested for loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

- Note: to print in the same line, we use end property in print function.
  - for i in range(1, 6):
  -     print(i, end=' ')
  - In Python, you can print in the same line within a loop by using the end parameter of the print() function. The end parameter allows you to specify what should be printed at the end of each print statement, by default it's set to '\n' (newline), which causes a line break.

Practice

- Write a Python program to count the number of even and odd numbers in a series of numbers:
  - Sample numbers : numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)
  - Expected Output :
  - Number of even numbers : 5
  - Number of odd numbers : 4
- Write a python program that prints from 0 to 30 but excluding multiples of 3