Chapter 4: Looping

CSCI-UA 0002 – Introduction to Computer Programming

Mr. Joel Kemp
Agenda

• while
• for
• range()
• for... in
Loop

• **Loop**: a coding construct that allows for some code to be run repeatedly.

• A loop, like a branch:
  – Has a condition (**loop condition**).
  – Executes a chunk of code (**loop body**) when the condition evaluates to **True**.

What does this look like in Python?
**while Loop**

• **One of the simplest forms of looping!**

• **Syntax:**
  ```python
  while (expression):
      # Loop body goes here
  ```

• **Example: The Simpsons**

  ```python
  answer = input("Homer, are we there yet? ")
  while (answer == "no"):
      answer = input("Homer, are we there yet? ")
  print("Finally! Ay Caramba!")
  ```

• **Alternative:**

  ```python
  answer = "no"  # Makes sure the loop runs
  while (answer == "no"):
      answer = input("Homer, are we there yet? ")
  print("Finally! Ay Caramba!")
  ```

Refer to: homerWhileLoop.py
Alternative Solutions

• **Multiple `input()` calls:**
  
  ```python
  answer = input("Homer, are we there yet? ")
  while (answer == "no"):
      answer = input("Homer, are we there yet? ")
  print("Finally! Ay Caramba!")
  ```

• **Preset answer to start the loop:**
  
  ```python
  answer = "no"  # Makes sure the loop runs
  while (answer == "no"):
      answer = input("Homer, are we there yet? ")
  print("Finally! Ay Caramba!")
  ```

• **Near-infinite loop:**
  
  ```python
  while (True):  # Say whaaaat?!
      answer = input("Homer, are we there yet? ")
      if (answer != "no"): 
          break  # Terminates the loop instantly
  print("Finally! Ay Caramba!")
  ```

Refer to: `whileLoopAlternatives.py`
Back to Guessing!

• Let’s fix the problem with our game!
  – While the guess is wrong, keep asking for a guess!
• Here’s where we left off!

```python
secret = 5

userGuess = input("Please guess a number: ")
userGuess = int(userGuess)

if(userGuess == secret):
    print("You Win!")
elif(userGuess > secret):
    print("Too High!")
else:
    print("Too Low!")
```
One Solution

• Multiple `input()` calls:

```python
secret = 5
userGuess = int(input("Please guess a number: "))

while(userGuess != secret):
    if(userGuess > secret):
        print("Too High!")
    else:
        print("Too Low!")
    userGuess = int(input("Please guess a number: "))

print("You Win!")
```

Let’s explore some alternate solutions!

Refer to: guessingv3.py
One Final Change

• Let’s make this game really worthwhile!

• Instead of the secret number always being 5, how can we make the number be a random number?
  – Let’s say between numbers 1 to 10.

• Hmm, random you say?
Python Random

• How can we generate a random number?

• At the top of your code, add the line:

```
from random import randint
```

– This tells the interpreter that we’re using `randint()` from the “random” Python library.

  • **Library** – a collection of related functions that can be referenced.

  – `randint()` takes two arguments (e.g., x and y) and generates a random number between x and y (inclusive).

```
from random import randint

count = 1
while(count <= 5):
    random_number = randint(1, 10)
    print ("Random number:", random_number)
    count += 1
```

Refer to:
random.py
The Final Code

```
secret = randint(1, 10)
userGuess = int(input("Please guess a number: "))

while(userGuess != secret):
    if(userGuess > secret):
        print("Too High!")
    else:
        print("Too Low!")

    userGuess = int(input("Please guess a number: "))

print("You Win!"))
```
**while Exercises**

1. Write a program that asks the user to enter a number that is **greater than** 0.
   - The program will keep on asking the user for the number until it is valid.
   - The program will then print the valid number.

2. Write a program that asks the user for the number \(N\) and range \(R\) of random numbers to generate.
   - Generate and print \(N\) random numbers between 0 and \(R\).
   - If \(N\) is 0, then output “No numbers will be generated!”

3. Write a program that asks the user for the number, \(N\), of even numbers to print.
   - Your program should compute and print the first \(N\) even numbers starting from 0.

Refer to: whileExercises.py
OTHER TYPES OF LOOPS
for loop

• A for loop is a very popular looping construct
• In python, a for loop iterates over:
  – Lists of numbers
  – Lists of objects
  – Text files
• How do we define a list?
  – We’ll see this more formally later.
  – For now, we’ll use `range()`!
    • The `range()` function is predominantly used with for loops!
• Example:
  ```python
  for num in range(5):
    print(num)
  ```
range()

• The range function has the following form:
  \[ \text{range}(\text{lower\_bound}, \text{upper\_bound}, \text{step}) \]
  – lower\_bound is optional and defaults to 0.
  – step is optional and defaults to 1.

• What does it do?
  – Generates a list of numbers from the lower bound up to but not including the upper bound.

• Examples:
  \[ \text{range}(1, 5) \]  # [1, 2, 3, 4]
  \[ \text{range}(1, 6, 2) \]  # [1, 3, 5]
  \[ \text{range}(4) \]  # [1, 2, 3]

Refer to: range.py
Exercises

1. Generate and print the even numbers up to and including the number 20.

2. Print the square roots for the numbers from 1 to 10.
   - You’ll need to use the \texttt{sqrt()} function of the math library.
   - \texttt{from math import sqrt}

3. Create a program that computes the factorial of a user supplied number.
   
   $0! = 1$
   $1! = 1$
   $2! = 2 \times 1$
   $3! = 3 \times 2 \times 1$
   $4! = 4 \times 3 \times 2 \times 1$
Nested Loops

- We can have loops within loops!
- Example:

```python
for i in range(5):
    for j in range(2):
        print("Hi")
```

- How does it work?
1. Perform the inner loop until it completes
2. Iterate the outer loop
3. Repeat until we exceed the range of the outer loop.

- Output:
  - The string "Hi" will be printed 8 times.

Refer to: nestedLoops.py