COMP 141

For Loops

Rhodes College

Announcements

- Reminders
  - Program 4 due Sunday, Sept. 29th by 11:55pm
  - Midterm 1 is on Wednesday, Oct. 2nd
    - Review worksheet on course website

Practice From Last Time

1. Write a while loop that prints all divisors of 30.
   - Your code should print out the following:
   1, 2, 3, 5, 6, 10, 15, 30

2. Modify this loop to print out all common divisors of 30 AND 50

3. Now let the user select any 2 integers and print out the common divisors of these 2 integers

4. Challenge: Print out only the largest of the common divisors of these 2 numbers

Class Practice

Write a *while loop* that will compute the sum of the first *n* positive odd numbers. For example, if *n* is 5, you should compute 1 + 3 + 5 + 7 + 9.

The `for` Loop

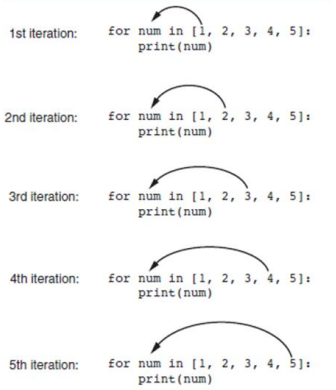
Count-Controlled loop: iterates a specific number of times

- Use a `for` statement to write count-controlled loop
  - Designed to work with sequence of data items
    - Iterates once for each item in the sequence
  - General format:

```
for variable in [val1, val2, etc]:
    statements
```

5

The `for` loop



6

Using the `range` function

The `range` function simplifies the process of writing a `for` loop

- **`range` returns an iterable object**
  - **Iterable**: contains a sequence of values that can be iterated over

`range` characteristics:

- **One argument:** used as ending limit
- **Two arguments:** starting value and ending limit
- **Three arguments:** third argument is step value

7

Using `range` Function

Which range gives us the output 1, 2, 3, 4, 5 ?

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

1
2
3
4
5

```
for num in range(5):
    print(num)
```

0
1
2
3
4

8

From Highest to Lowest

The `range` function can be used to generate a sequence with numbers in descending order

- Make sure starting number is larger than end limit, and step value is negative
- Example: `range (10, 0, -1)`

[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

9

For Loop Example 1

```
for num in range(1, 10, 1):
    square = num * num
    if square % 5 != 0:
        print("The square of", num, "is", square)
```

Output

The square of 1 is 1
The square of 2 is 4
The square of 3 is 9
The square of 4 is 16
The square of 6 is 36
The square of 7 is 49
The square of 8 is 64
The square of 9 is 81

10

For Loop Example 2

```
total = 0
for num in range(2, 11, 2):
    total += num
print(total)
```

Output
30

Note: total = 2 + 4 + 6 + 8 + 10

11

For Loop Example 3

```
def f_to_c(degrees_f):
    c = (degrees_f - 32) * 5/9
    return c

def main():
    fmin = int(input("Min temp: "))
    fmax = int(input("Max temp: " ))

    for fah_temp in range(fmin, fmax+1, 10):
        cel_temp = f_to_c(fah_temp)
        print(fah_temp, cel_temp)

main()
```

12

Rewrite GCD code to use a for loop

```
def main():
    num1 = int(input("Value 1: "))
    num2 = int(input("Value 2: "))

    cnt = 1
    gcd = 1

    #Code to determine which number is smaller
    minNum = num1
    if num2 < num1:
        minNum = num2


    while cnt <= minNum:
        if num1 % cnt == 0 and num2 % cnt == 0:
            gcd = cnt
        cnt += 1
    print(gcd)

main()
```

gcd.py in Box folder

Class Activity

Compute the sum of the first n odd positive integers using a `for` loop

Example:

- if n is 5, you should compute 1 + 3 + 5 + 7 + 9.